Briefings in Bioinformatics, 00(0), 2020, 1–15

doi: 10.1093/bib/bbaa012
Advance Access Publication Date: 00 Month 20xx
Review Article

Biological applications of knowledge graph embedding models

Sameh K. Mohamed, Aayah Nounu and Vít Nováček

Corresponding author: Sameh K. Mohamed, Insight Centre for Data Analytics, IDA Business Park, Lower Dangen, Galway, Ireland. Tel.: +353 91 495730. Email: sameh.kamal@insight-centre.org

Abstract

Complex biological systems are traditionally modelled as graphs of interconnected biological entities. These graphs, i.e. biological knowledge graphs, are then processed using graph exploratory approaches to perform different types of analytical and predictive tasks. Despite the high predictive accuracy of these approaches, they have limited scalability due to their dependency on time-consuming path exploratory procedures. In recent years, owing to the rapid advances of computational technologies, new approaches for modelling graphs and mining them with high accuracy and scalability have emerged. These approaches, i.e. knowledge graph embedding (KGE) models, operate by learning low-rank vector representations of graph nodes and edges that preserve the graph's inherent structure. These approaches were used to analyse knowledge graphs from different domains where they showed superior performance and accuracy compared to previous graph exploratory approaches. In this work, we study this class of models in the context of biological knowledge graphs and their different applications. We then show how KGE models can be a natural fit for representing complex biological knowledge modelled as graphs. We also discuss their predictive and analytical capabilities in different biology applications. In this regard, we present two example case studies that demonstrate the capabilities of KGE models: prediction of drug–target interactions and polypharmacy side effects. Finally, we analyse different practical considerations for KGEs, and we discuss possible opportunities and challenges related to adopting them for modelling biological systems.

Key words: biomedical knowledge graphs; knowledge graph embeddings; tensor factorization; link prediction; drug–target interactions; polypharmacy side effects.

Sameh K. Mohamed is a PhD student in Computer Science at Insight Centre, National University of Ireland Galway, and a researcher at the Data Science Institute in Galway, Ireland. His main interests lie within the areas of machine learning and bioinformatics. He is currently focused on representational learning and knowledge graphs mining.

Vit Novacek holds a PhD from National University of Ireland Galway and currently leads the Biomedical Discovery Informatics Unit at Data Science Institute, National University of Ireland Galway where he also is a research fellow and adjunct lecturer. His personal research interests revolve around developing machine-aided discovery solutions by means of machine/representation learning, explainable AI and text mining, with a strong focus on biomedical use cases.

Aayah Nounu holds a PhD from The University of Bristol. Her background is in combining both laboratory-based methods and epidemiological methods to understand drug mechanisms associated with cancer prevention. She is currently working in a research post looking at the effect of aspirin for the prevention of colorectal cancer.

Data Science Institute is specialized in research technologies at the convergence of computer science, web science and artificial intelligence to build a fundamental understanding of how information and knowledge are increasingly driving society through digital processes and of the tools, techniques and principles supporting a data-enhanced world.

Submitted: 18 September 2019; **Received (in revised form):** 10 January 2020

Introduction

Biological systems consist of complex interconnected biological entities that work together to sustain life in living systems. This occurs through complex and systematic biological interactions of the different biological entities. Understanding these interactions is the key to elucidating the mechanism of action of the different biological functions (e.g. angiogenesis, metabolism, apoptosis, etc.) and thus understanding causes and activities of diseases and their possible therapies. This encouraged the development of multiple physical and computational methods to assess, verify and infer different types of these interactions. In this study, we focus on the use of computational methods for assessing and inferring interactions (associations) between different biological entities at the molecular level. We hereof study the use of knowledge graphs and their embedding models for modelling molecular biological systems and the interactions of their entities.

Initially, basic networks, i.e. uni-relational graphs, were adopted by early efforts for modelling complex interactions in biological systems [1–4]. Despite their initial success [5], these networks could not preserve the semantics of different types of associations between entities. For example, protein–protein interaction networks modelled with basic networks cannot differentiate between different types of interactions such as inhibition, activation, phosphorylation, etc. Therefore, more recent works modelled biological systems using heterogeneous multi-relational networks i.e. knowledge graphs, where they utilized different visual [6, 7] and latent representations [8, 9] of graph entities to infer associations between them.

In the context of biological applications, knowledge graphs were used to model biological data in different projects such as the UNIPROT [10], Gene Ontology [11] and Bio2RDF [12] knowledge bases. Moreover, they were the basis of multiple predictive models for drug adverse reactions [6, 8], drug repurposing [9, 13] and other predictions for different types of biological concepts associations [13, 14]. The task of learning biological associations in this context is modelled as link prediction in knowledge graphs [15]. Predictive models then try to infer a typed link between two nodes in the graph using two different types of features: graph features and latent-space vector representations.

Graph features models (i.e. visual feature models) are part of the network analysis methods, which learn their predictions using different feature types such as random walks [16, 17], network similarity [18], nodes connecting paths [19] and subgraph paths [19, 20]. They are used in multiple biological predictive applications such as predicting drug targets [21] and protein–protein interaction analysis [18]. Despite the expressiveness of graph feature models predictions, they suffer from two major drawbacks: limited scalability and low accuracy [22, 23]. They are also focused on graph local features compared to embedding models, which learn global latent features of the processed graph.

Latent feature models i.e. embedding models, on the other hand, express knowledge graphs' entities and relations using low-rank vector representations that preserve the graph's global structure. Knowledge graph embedding (KGE) models on the contrary are known to outperform other approaches in terms of both the accuracy and scalability of their predictions despite their lack of expressiveness [23–25].

In recent years, KGE models witnessed rapid developments that allowed them to excel in the task of link prediction [24–30]. They have then been widely used in various applications including computational biology in tasks like predicting drug–target interactions (DTIs) [9] and predicting drug polypharmacy side effects [8]. Despite their high-accuracy predictions in different biological inference tasks, KGEs are in their early adoption stages in computational biology. Moreover, many computational biology studies that have used KGE models adopted old versions of these models [31, 32]. These versions have then received significant modifications through recent computer science research advances [25].

In a previous study, Su *et al.* [14] have introduced the use of network embedding methods in biomedical data science. The study compiles a taxonomy of embedding methods for both basic and heterogeneous networks where it discusses a broad range of potential applications and limitation. The study's objective was to introduce the broad range of network embedding methods; however, it lacked deeper investigation into the technical capabilities of the models and how can they be integrated with a specific biological problem. The study also did not compare the investigated models in terms of their accuracy and scalability, which is essential to assist reader from the biological domain to understand the key differences between these methods as to their applicability.

In this study, we exclusively explore KGE models, focusing on the best performing models in terms of both scalability and accuracy across various biological tasks. We use these case studies to demonstrate the analytical capabilities of KGE models, e.g. learning clusters and similarity measures in different biological problems. We also explore the process of building biological knowledge graphs for generic and specific biological inference tasks. We then present computer-based experimental evaluation of KGE models on different tasks such as predicting DTIs, drug polypharmacy side effects and prediction of tissue-specific protein functions.

The rest of this study is organized as follows: Section 2.1 discusses knowledge graphs as a data modelling technique and their applications in the biological domain. Section 2.2 discusses KGE models, their design and how they operate on different types of data. Section 3 presents the example case studies that we will use throughout the study. Section 4 discusses the predictive and analytical capabilities of KGE models on the designated case studies discussed in Section 3. Section 5 discusses the performance of KGE models on biological data in terms of the predictive accuracy and scalability. Section 6 discusses the current challenges and possible opportunities of the use of KGE models to model the different types of biological systems. Finally, we discuss our conclusions in Section 7.

Background

In this section, we discuss both knowledge graphs and KGE models in the context of biological applications.

Knowledge graphs

A knowledge graph is a data modelling technique that models linked data as a graph, where the graph's nodes represent data entities and its edges represent the relations between these entities. In recent years, knowledge graphs became a popular means for modelling relational data where they were adopted in various industrial and academic applications such as semantic search engines [33], question answering systems [34] and general knowledge repositories [35]. They were also used to model data from different types of domains such as general human knowledge [35], lexical information [36] and biological systems [12].

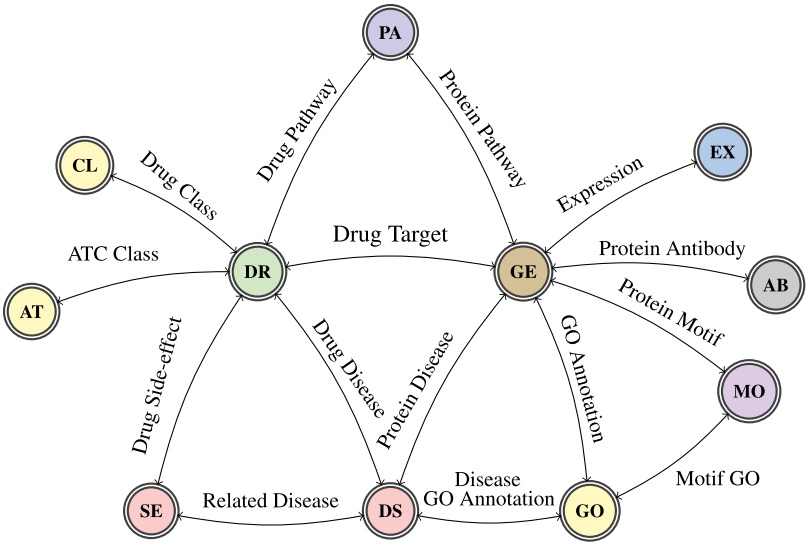

Figure 1. A schema of a knowledge graph that models a complex biological system of different types of entities and concepts. The abbreviation DR represents drugs, GE represents proteins (their genes), EX represents protein expressions (tissues and cell-lines), AB represents protein antibodies, MO represents protein motifs and other sequence annotations, GO represents gene ontology, DS represents diseases, SE represents drug side-effects, AT represents ATC classes, CL represents drug classes and PA represents pathways.

Knowledge graphs model facts as subject, predicate and object (SPO) triples, where subjects and objects are the knowledge entities and predicates are the knowledge relations. In this context, the subject entity is associated to the object entity with the predicate relation e.g. (*Aspirin*, *drug_target* and *COX1*). Figure 1 shows an illustration of a schema of a knowledge graph that models complex associations between different types of biological entities such as drugs, proteins, antibodies, etc. It also models different types of relations between these entities, where these relations carry different association semantics.

In our study, we use $\mathcal{G}$ to denote a knowledge graph, $\mathcal{E}$ to denote entities and $\mathcal{R}$ to denote relations i.e. predicates. We also use $\mathcal{N}_e$ and $\mathcal{N}_r$ to denote the total count of both entities and relations in a knowledge graph, respectively.

Popular Biological Sources. Online knowledge bases are a popular means for publishing large volumes of biological data [37]. In recent years, the number of these knowledge bases has grown, where they cover different types of data such as paper abstracts [38], raw experimental data [39], curated annotations [10, 40, 41], etc. Biological knowledge bases store data in different structured and unstructured (free text e.g. comments) forms. Although both data forms can be easily comprehended by humans, structured data are significantly easier for automated systems. In the following, we explore popular examples of these knowledge bases that offer structured data that can be easily and automatically consumed to generate knowledge graphs.

Table 1 summarizes the specializations and the different types of covered biological entities of a set of popular biological knowledge bases. The table also shows that most of the current knowledge bases are compiled around proteins (genes). However, it also shows their wide coverage of the different types of biological entities such as drugs, their indications, gene ontology annotations, etc.

Building Biological Knowledge Graphs. Knowledge graphs store information in a triplet form, where each triplet (i.e. triple) model a labelled association between two unique unambiguous entities. Data in biological knowledge bases, however, lack these association labels. Different knowledge bases also use different identifier systems for the same entity types, which results in the ambiguity of entities of merged databases. Building biological knowledge graph process therefore mainly deals with these two issues.

In the association labelling routine, one can use different techniques to provide meaningful labels for links between different biological entities. This, however, is commonly achieved by using entity types of both subject and object entities to denote the relation labels as shown in Figure 1 (e.g. 'Drug Side-effect' as a label for link between two entities that are known to be types of drug and side effect, respectively).

The ambiguity issue, i.e. merging entities of different identifier systems, is commonly resolved using identifier mapping resource files. Different systems study entities on different speciality levels. As a result, the links between their different identifier systems is not always in a form of one-to-one relationships. In such cases, a decision is made to apply a specific filtering strategy based on either expert's opinion or problem-specific properties (for instance, deciding on an authoritative resource such as UniProt for protein entities and resolving all conflicts by sticking to that resource's naming scheme and conventions).

To complement the basic principles introduced in the previous paragraphs, we refer the reader to the Bio2RDF initiative [55] that has extensively studied the general topic of building interlinked biological knowledge graphs [see also Bio2RDF scripts (https://github.com/bio2rdf/bio2rdf-scripts/wiki) for corresponding scripts and conversion convention details]. General principles as well as an example of actual implementation of conversion from (relational) databases into RDF (i.e. knowledge graphs) are discussed in the study of Bizer *et al.* [56]. Possible solutions to the problem of aligning and/or merging several such knowledge graphs are reviewed in the study of Amrouch *et al.* [57] that focuses on ontology matching. An example of a more data-oriented method is for instance LIMES [58]. All these approaches may provide a wealth of inspiration for building bespoke approaches to building knowledge graphs in specific biomedical use cases, should the information we provide in this section be insufficient.

Table 1. A comparison between popular biological knowledge graph in terms of the coverage of different types of biological entities. The abbreviation S represents structured data, U represents unstructured data, DR represents drugs, GE represents proteins, GO represents gene ontology,PA represents pathways and CH denotes chemicals

	Properties		Entity coverage								
Knowledge base	**Format**	**Speciality**	**Proteins**	**Drugs**	**Indications**	**Diseases**	**Gene ontology**	**Expressions**	**Antibodies**	**Phenotypes**	**Pathways**
UNIPROT [10]	S/U	GE	✓	✓		✓	✓	✓	✓		✓
REACTOME [42]	S	PA	✓				✓				✓
KEGG [40, 43]	S	PA	✓	✓		✓					✓
DrugBank [44]	S/U	DR	✓	✓							✓
Gene Ontology [11]	S	GO	✓				✓				✓
CTD [45]	S/U	CH	✓	✓			✓			✓	✓
ChEMBL [46]	S/U	CH	✓	✓	✓	✓		✓			
SIDER [47]	S	DR		✓	✓						
HPA [48]	S/U	GE	✓				✓	✓	✓		
STRING [49]	S	GE	✓								
BIOGRID [50]	S	GE	✓								
InAct [41]	S	GE	✓								
InterPro [51]	S	GE	✓								
PharmaGKB [52]	S	DR	✓	✓							
TTD [53]	S	DR	✓	✓							
Supertarget [54]	S	DR	✓	✓							

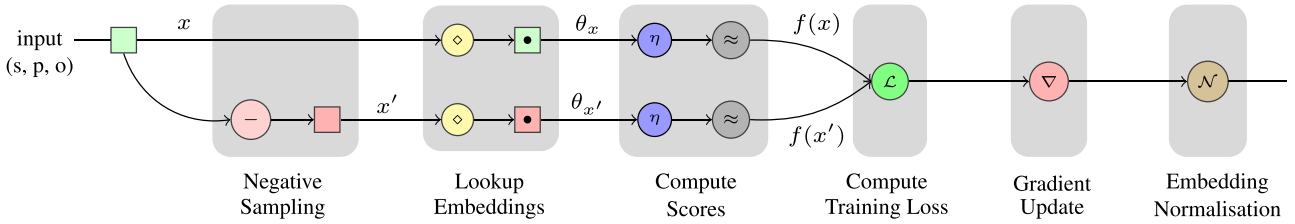

Figure 2. An illustration of the training network of one training instance of a KGE model.

Knowledge graph embeddings

In this section, we discuss KGE models where we briefly explore their learning procedure. We then explore different embedding representation types and their potential uses and application.

The learning procedure. Multiple studies have explored KGE models, their technical design, training objectives and predictive capabilities on general benchmarking settings [15, 24, 59]. Therefore, in the following, we only focus on providing a brief and concise description of how KGE models work.

KGE models operate by learning low-rank representations of knowledge graph entities and relations. The KGE learning step is a multi-phase procedure as shown in Figure 2, which is executed iteratively on knowledge graph data. Initially, all entities and relations are assigned random embeddings (noise). They are then updated using a multi-phase learning procedure.

KGE models consume knowledge graphs in the form of SPO triplets. They first generate negative samples from the input true triplets using uniform random corruptions of the subjects and objects [60]. KGE models then lookup corresponding embedding of both the true and corrupted triplets. The embeddings are then processed using model-dependent scoring functions (cf. mechanism of action in Table 2) to generate scores for all the triplets. The training loss is then computed using model-dependent loss functions where the objective is to maximize the scores of true triplets and minimize the scores of corrupted triplets. This objective can be formulated as follows:

$$\forall_{t\in\mathbb{T},t'\in\mathbb{T}'} f(\theta_t) > f(\theta_{t'}), \tag{1}$$

where $\mathbb{T}$ denotes the set of true triplets, $\mathbb{T}'$ denotes the set of corrupted triplets, f denotes the model-dependent scoring function and θ_t denotes the embeddings of the triplet t.

Traditionally, KGE models use a ranking loss, e.g. hinge loss or logistic loss, to model the objective training cost [26, 28, 29]. This strategy allows KGE models to efficiently train their embeddings in linear time, $\mathcal{O}(d)$, where K denotes the size of the embedding vectors. On the other hand, some KGE models such as the ConvE [30] and the ComplEx-N3 [25] models adopt multi-class based strategies to model their training loss. These approaches have shown superior predictive accuracy compared to traditional ranking-based loss strategies [25, 30]. However, they suffer from limited scalability as they operate on the full entity vocabulary.

The KGE models minimize their training loss using different variations of the gradient descent algorithm e.g. Adagrad, AMSGrad, etc. Finally, some KGE models normalize their embeddings as a regularization strategy to enhance their generalization. This strategy is often associated to models, which adopt ranking-based training loss strategies such as the TransE and DistMult models [26, 28].

The learning multi-phase procedure is executed iteratively to update the model's embeddings until they reach an optimal state that satisfies the condition in Equation 1. Table 2 also provides a summary of properties of popular KGE models, their mechanism of action i.e. scoring mechanism, output embeddings format, runtime complexity, release year and available code bases.

KGE models ingest graph data in triplets form where they learn global graph low-rank latent features, which preserve the

Table 2. A comparison between popular KGE models, their learning mechanism, published year and available code bases. Em. format column denotes the format of the model embeddings in the form $(g(d), h(d))$, where d denotes the embeddings size, $g(d)$ denotes the shape of the entities embeddings and $h(d)$ denotes the shape of the relations embeddings. n and m denote the number of entities and relations, respectively, in the space complexity column

Model	Scoring mechanism	Em. Format	Time complexity	Space complexity	Year	Repository (Python)
RESCAL [27]	Tensor factorization	(d, d^2)	$\mathcal{O}(d^2)$	$\mathcal{O}(nd + md^2)$	2011	mnick/rescal.py
TransE [26]	Linear translation	(d, d)	$\mathcal{O}(d)$	$\mathcal{O}(nd + md)$	2014	ttrouill/complex
DistMult [28]	Bilinear dot product	(d, d)	$\mathcal{O}(d)$	$\mathcal{O}(nd + md)$	2015	ttrouill/complex
HolE [62]	Fast Fourier transformation	(d, d)	$\mathcal{O}(d \log d)$	$\mathcal{O}(nd + md)$	2016	mnick/holographic-embeddings
ComplEx [29]	Complex product	$(2d, 2d)$	$\mathcal{O}(d)$	$\mathcal{O}(nd + md)$	2016	ttrouill/complex
ANALOGY [63]	Analogical structure	(d, d)	$\mathcal{O}(d)$	$\mathcal{O}(nd + md)$	2017	quark0/ANALOGY
ConvE [30]	Convolutional filters	(d, d)	$\mathcal{O}(d)$	$\mathcal{O}(nd + md)$	2018	TimDettmers/ConvE
TriModel [64]	Multi-part embeddings	$(3d, 3d)$	$\mathcal{O}(d)$	$\mathcal{O}(nd + md)$	2019	samehkamaleldin/libkge

graph's coherent structure. These features encode semantics such as node types and their neighbours by isolating nodes' embeddings on different embedding dimensions [23]. However, they have limited ability to encode indirect semantics such as logical rules and in-direct relations [61].

Embedding representation. KGEs have different formats e.g. vectors, matrices, etc., which serve as numerical feature representations of their respective objects. These representations can be used in both general tasks such as clustering and similarity analysis, as well as in specific inference tasks such as predicting different association types. Similarly, in computational biology, they can be used to cluster biological entities such as protein, drugs, etc., as well as to learn specific biological associations such as drug targets, gene-related diseases, etc. Embeddings of biological entities can also be used as representative features in traditional regression and classification models e.g. logistic regression or SVM classifiers.

Popular KGE models. Table 2 presents a comparison between a set of popular KGE models, their scoring mechanism, embeddings format, time complexity, space complexity, year of publication and corresponding source code repository. These models use different approaches to learn their embeddings where they can be categorized into three categories: distance-based models, factorization-based models and convolutional models. Distance-based models such as the TransE model use linear translations to model their embeddings interactions using a linear time and space complexity procedure. Convolution-based methods such as the ConvE use convolutional neural networks to model embedding interactions, which also have a linear time and space complexity. Factorization-based models, on the other hand, use dot product-based procedures to model embedding interactions, where they also have linear time and space complexity. However, tensor factorization-based models commonly use higher rank embeddings than convolution and distance-based models [29, 64].

In this study, we are focused on embedding methods, which operate on multi-relational graphs as we mentioned in the introduction of the paper. The DeepWalk [65], Node2Vec [66], etc. are uni-relational graphs embedding methods; thus, they we do not include them in this study.

Examples of biological case studies

In the following, we present two example biological case studies that we use through this study to demonstrate the capabilities of KGE models. Firstly, we discuss the task of predicting DTIs where we model biological information as a knowledge graph. We then evaluate the predictive accuracy of KGE models, and we compare them to other state-of-the-art approaches. Secondly, we discuss the task of predicting drug polypharmacy side effects, where we model the investigated drug polypharmacy data as a 3D tensor.

Predicting DTIs

The study of drug targets has become very popular with the objective of explaining mechanisms of actions of current drugs and their possible unknown off-target activities. Knowing targets of potential clinical significance also plays a crucial role in the process of rational drug development. With such knowledge, one can design candidate compounds targeting specific proteins to achieve intended therapeutic effects. Large-scale and reliable prediction of DTIs can substantially facilitate development of such new treatments. Various DTI prediction methods have been proposed to date. Examples include chemical genetic [67] and proteomic methods [68] such as affinity chromatography and expression cloning approaches. These, however, can only process a limited number of possible drugs and targets due to the dependency on laboratory experiments and available physical resources. Computational prediction approaches have therefore received a lot of attention lately as they can lead to much faster assessments of possible DTIs [69, 70].

Data. We consider the DrugBank_FDA [71] benchmarking data set as an example to evaluate the predictive accuracy of KGE models and to compare them to other approaches. We also utilize the UNIPROT [10] database to provide richer information about both drugs and their protein targets in the input knowledge graph. The data set contains 9881 known DTIs, which involve 1482 drugs and 1408 protein targets.

Related work. The work of Yamanishi *et al.* [69] was one of the 1st approaches to predict drug targets computationally. Their approach utilized a statistical model that infers drug targets based on a bipartite graph of both chemical and genomic information. The BLM-NII [70] model was developed to improve the previous approach by using neighbour-based interaction-profile inference for both drugs and targets. More recently, Cheng *et al.* [72, 73] proposed a new way for predicting DTIs, where they have used a combination of drug similarity, target similarity and network-based inference. The COSINE [74] and NRLMF [75] models introduced the exclusive use of drug–drug and target–target similarity measures to infer possible drug targets. This has an advantage of being able to compute predictions even for drugs and targets with limited information about their interaction data. However, these methods only utilized a single measure to model components similarity. Other approaches such as the KronRLS-MKL [76] model used a linear combination of multiple

similarity measures to model the overall similarity between drugs and targets. Non-linear combinations were also explored in an early study [70] and shown to provide better predictions. Recently, further predictive models were developed to utilize matrix factorization [77] and biological graph path features [7] to enable more accurate drug–target prediction.

Predicting polypharmacy side effects

Polypharmacy side effects are a specific case of adverse drug reactions that can cause significant clinical problems and represent a major challenge for public health and pharmaceutical industry [78]. Pharmacology profiling leads to identification of both intended (target) and unintended (off-target) drug-induced effects, i.e. biological system perturbations. While most of these effects are discovered during pre-clinical and clinical trials before a drug release on the market, some potentially serious adverse effects only become known when the drug is in use already.

When more drugs are used jointly (i.e. polypharmacy), the risk of adverse effects rises rather rapidly [79, 80]. Therefore, reliable automated predictions of such risks are highly desirable to mitigate their impact on patients.

Data. In this case study, we consider the data set compiled by Zitnik *et al.* [8] as an example benchmark. The data set includes information about multiple polypharmacy drug side effects (http://snap.stanford.edu/decagon/). The data set also contains facts about single drug side effects, protein–protein interactions and protein–drug targets. The drug side effects represented in the data set are collected from the SIDER (Side Effect Resource) database [47] and the OFFSIDES and TWOSIDES databases [80]. These side effects are categorized into two groups: mono-drug and polypharmacy drug–drug interaction side effects.

In our study, we only consider the polypharmacy side effects, and we filter out both the mono-side effects and drug targets data.

Related work. The research into predictive approaches for learning drug polypharmacy side effects is in its early stages [8]. The Decagon model [8] is one of the 1st introduced methods for predicting polypharmacy side effects, which models the polypharmacy side-effect data as a knowledge graph. It then solves the problem as a link prediction problem using a generative convolution-based strategy. Despite its effectiveness, this approach still suffers from a high rate of false positives. Furthermore, other approaches considered using a multi-source embedding model [81] to learn representations of drugs and polypharmacy side effects. These approaches achieved similar performance to the Decagon model with a more scalable training procedure [81].

Predicting tissue-specific protein functions

Proteins are usually expressed in specific tissues within the body where their precise interactions and biological functions are frequently dependent on their tissue context [82, 83]. The disorder of these interactions and functions results in diseases [84, 85]. Deep understanding of tissue-specific protein activities is therefore essential to elucidate the causes of diseases and possible treatments.

Data. We consider the tissue-specific data set compiled by Zitnik *et al.* [86] to study tissue-specific protein functions. The data set contains protein–protein interactions and protein functions of 144 tissue types (http://snap.stanford.edu/ohmnet/).

Related work. Recently, Zitnik *et al.* have developed the state-of-the-art model, the OhmNet model [86], a hierarchy-aware unsupervised learning method for multi-layer networks. It models each tissue information as a separate network and learns efficient representations for proteins and functions by generating their embeddings using the tissue-specific protein–protein interactome and protein functions. They have also examined other different approaches such as the LINE model [87], which uses a composite learning technique where it learns half of the embeddings' dimensions from the direct neighbour nodes and the other half from the 2nd hop connected neighbours. The GeneMania model [88] is another model that has suggested a propagation-based approach for predicting tissue-specific protein functions. In this method, the tissue-specific networks are firstly combined into one weighted network, and they are then propagated to allow predicting other unknown protein functions.

Capabilities of KGE models

KGE models can be used in different supervised and unsupervised applications where they provide efficient representations of biological concepts. They can be used in applications such as learning biological associations, concepts similarity and clustering biological entities. In this section, we discuss these applications in different computational biology tasks. We provide a set of example uses cases where we present the data integrated in each example, how the KGE models were utilized and we report the predictive accuracy of the KGE models and we compare it to other approaches when possible.

Learning biological associations

KGE models can process data in the form of a knowledge graph. They then try to learn low-rank representations of entities and relations in the graph, which preserve its coherent structure. They can also process data in a three-dimensional (3D) tensor form where they learn low-rank representations for the tensor entities that preserve true entity combination instances in the tensor.

In the following, we provide two examples for learning biological associations on a knowledge graph and a 3D tensor in a biological application. First, we discuss the task of predicting DTIs where we model biological information as a knowledge graph. We then evaluate the predictive accuracies of KGE models, and we compare them to other state-of-the-art approaches. Secondly, we discuss the task of predicting drug polypharmacy side effects, where we model the related data as a 3D tensor. We then apply KGE models to perform tensor factorization, and we evaluate their predictive accuracy in learning new polypharmacy side effects compared to other state-of-the-art approaches.

- Drug–target prediction benchmark. We present a comparison between state-of-the-art drug–target predictors and KGE models in predicting DTIs. The KGE models in this context utilize the fact that the current drug–target knowledge bases like DrugBank [71] and KEGG [40] are largely structured as networks representing information about drugs and their relationship with target proteins (or their genes), action pathways and targeted diseases. Such data can naturally be interpreted as a knowledge graph. The task of finding new associations between drugs and their targets can then be formulated as a link prediction problem on a biological knowledge graph.

  We use the standard evaluation protocol for the DTI task [7] on the DrugBank_FDA data set that we introduced in Section 3.1. We use a 5-fold cross-validation evaluation on the DTIs where they are divided into splits with uniform

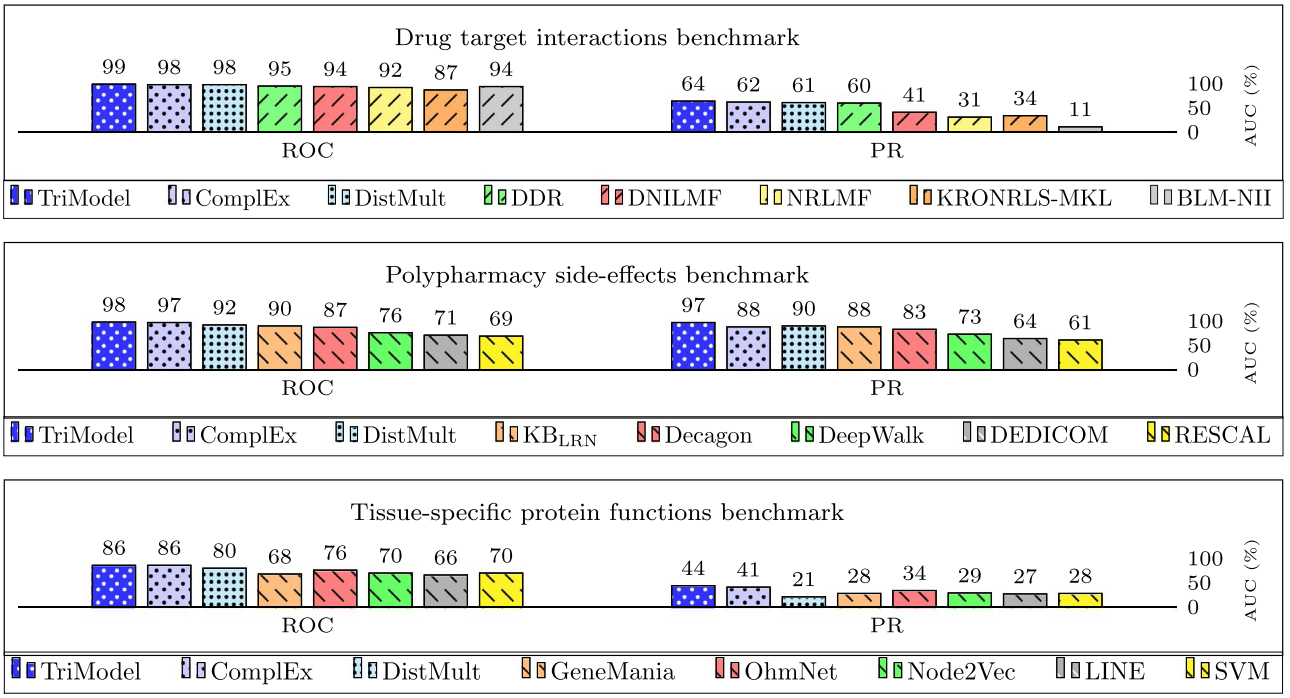

Figure 3. A summary of results of an evaluation of the predictive accuracy of knowledge graph embedding models compared to other models on two biological inference tasks: predicting drug targets and predicting polypharmacy side-effects. The reported results represent the score percentage of the area under the ROC and precision recall curves for the left and right side bars respectively.

random sampled negative instances with a 1:10 positive to negative ratio.

Figure 3 presents the outcome results of the KGE models (DistMult, ComplEx and TriModel) compared to other approaches (DDR [7], DNILMF [77], NRLMF [77], NRLMF [75], KRONRLS-MKL [76], COSINE [89] and BLM-NII [70]) on the DrugBank_FDA data set. The figure shows that the KGE models outperform all other approaches in terms of both the area under the ROC and precision recall curves.

- Polypharmacy side effects prediction benchmark. In Section 3.2, we discussed the problem of predicting polypharmacy side effects, the currently available data and related works. In the following, we present an evaluation benchmark for present polypharmacy side effects where we compare the KGE models with current state-of-the-art approaches. We first split the data into two sets, train and test splits, where the two splits represent 90% and 10% of the data, respectively. We then generate random negative polypharmacy side effects by randomly generating combinations of drugs for each polypharmacy side effect where the ratio between negative and positive instances is 1:1. We only consider drug combinations that did not appear in both training and test splits to enhance the quality of sampled negatives and decrease the ratio of false negatives.

  We use the holdout test defined by Zitnik *et al.* [8] where we train the predictive models on the training data and test their accuracy on the testing data split. We also run a 5-runs averaged 5-fold cross-validation evaluation to ensure the consistency of the model reported results over the different folds; however, we only report the holdout test results, which are comparable with state-of-the-art methods. Our k-fold cross validation experiments confirm that the model results are similar or insignificantly different across different random testing splits.

  We use the area under the ROC and precision recall metrics to assess the quality of the predicted scores. Figure 3 presents the results of our evaluation where we compare KGE models such as the DistMult, ComplEx and TriModel models to the current popular approaches (Decagon [8], KB_LRN [91], RESCAL [27], DEDICOM [92] and DeepWalk [65]). The results show that KGE models outperform other state-of-the-art approaches in terms of both the area under the ROC and precision recall curves.
- Tissue-specific protein function prediction benchmark. In Section 3.3, we have presented the problem of tissue-specific protein function prediction benchmark where we have discussed current predictive models and established benchmarking data sets. In the following, we present an evaluation benchmark between a set of traditional approaches such as the OhmNet [86], LINE [87], GeneMania [88] and SVM [86] models and other KGE models. We use the data set generated by Zitnik *et al.* [86], which provides training and testing data with both positive and negative instances where the negative to positive ratio is 1 to 10.

  We conduct a holdout test using the provided training and testing data set where we train our models on the training split and evaluate them on the testing using the area under the ROC and precision recall curves. Figure 3 presents the outcome of our experiments where it shows that KGE models such as the TriModel and ComplEx models achieve the best results in terms of both the area under the ROC and precision recall curves. Similar to the previous experiments, we also ran a 5-runs 5-fold cross-validation test to ensure the consistency of our results, and the results of our experiments confirm the results reported in the holdout test. However, we only report the holdout test results to be able to compare to other approaches.

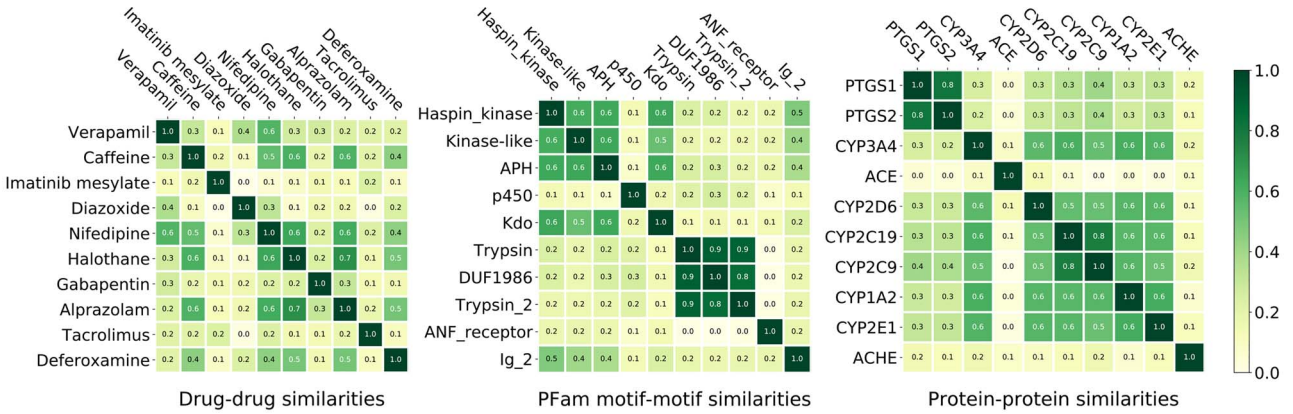

Figure 4. Three similarity matrices that denote the drug–drug similarities, motif–motif similarities and protein–protein similarities. The similarity values are generated by computing the cosine similarity between the embeddings of the pairs of compared entities. All the embeddings used to generate this figure are computed on the DrugBank_FDA data sets with the proteins associated to their PFam [90] motifs and protein families.

In all of our holdout test experiments, we learn the best hyperparameters using a grid search on the validation data split, where the training set is divided into two sets for training and validation (90% and 10%, respectively) in the absence of a validation set. On the other hand, in the cross validation experiments, we re-split each into training and validation splits (90% and 10%, respectively) in order to learn the model's best hyperparameters. We have found the embedding size is the most sensitive hyperparameters where it correlates with the graph size. The regulation weight and embedding dropout also are important hyperparameters, which affect the generality of the models from the validation to the testing split.

Example source code scripts and data sets of the experiments, which we executed in this study, are available at https://github.com/samehkamaleldin/bio-kge-apps.

Learning similarities between biological entities

The KGE models enable a new type of similarity that can be measured between any two biological entities using the similarity between their vector representation. The similarity between vectors can be computed using different techniques such as the cosine and p-norm similarities. Since the KGE representation is trained to preserve the knowledge graph structure, the similarity between two KGE representations reflects their similarity in the original knowledge. Therefore, the similarities between vector representations of KGE models, which are trained on a biological knowledge graphs, represent the similarities between corresponding entities in the original knowledge graph.

In the following, we explore a set of examples for using KGE similarities on biological knowledge graphs. We have used the drug–target knowledge graph created for the drug–target prediction task to learn embeddings of drugs, their target proteins and the entities of the motifs of these proteins according to the PFam database [90]. We have then computed the similarities between embeddings of entities of the same type such as drugs, proteins and motifs as shown in Figure 4. All the similarity scores in the illustration are computed using cosine similarity between the embeddings of the corresponding entity pair. The results show that the similarity scores are distributed from 0.0 to 1.0, where the 0.0 represents the least similar pairs and the 1.0 scores represent the similarity between the entity and itself. We then assess the validity of resulting scores by investigating the similarity of attributes of a set of the examined concepts with highest and lowest scores.

- Drug–drug embedding similarity. The left similarity matrix in Figure 4 illustrates the drug–drug similarity scores between the set of the most frequent drugs in the DrugBank_FDA data set. The scores are computed on the embeddings of drugs learnt in the DTI training pipeline. The figure shows that the majority of drug pairs have a low similarity (0.0 $\sim$ 0.2). For example, the similarity score between the drug pairs (diazoxideand caffeine) and (tacromlimus and diazoxide) is zero. We asses these results by assessing the commonalities between the investigated drugs in terms of indications, pharmacodynamics, mechanism of action, targets, enzymes, carriers and transporters. The caffeine and diazoxide in this context have no commonalities except for that they are both diuretics [93, 94]. On the other hand, halothane and alprazolam does not share any of the investigated commonalities.

  The results also shows a few drug–drug similarities with relatively higher scores (0.6 $\sim$ 0.7). For example, the similarity scores of the drug pairs (alprazolam and halothane), (alprazolam and caffeine) and (halothane and caffeine) are 0.7, 0.6 and 0.6, respectively. These findings can be supported by the fact that the two drug pairs share common attributes in terms of their targets, enzymes and carriers. For example, both alprazolam and halothane act on sedating individuals, and they target the GABRA1 protein [95, 96]. They are also broken by CYP3A4 and CYP2C9 enzymes and carried by albumin [97]. Similarly, the (alprazolamand caffeine) and (halothaneand caffeine) pairs have common associated enzymes.
- Motif–motif embedding similarity. The middle similarity matrix in Figure 4 illustrates the motif–motif similarity scores between the set of the most frequent PFam motifs associated with protein targets from the DTI benchmark. The lowest motif–motif KGE-based similarity scores correspond to the pairs (*ANF_receptor* and *Trypsin*), (*ANF_receptor* and *DUF1986*) and (*ANF_receptor* and *Trypsin_2*).
- On the other hand, the highest similarity scores (0.8, 0.9 and 0.9) exist between the pairs (*Trypsin*and *DUF1986*), (*Trypsin_2* and *DUF1986*) and (*Trypsin* and *Trypsin_2*), respectively.

  We assess the aforementioned findings by investigating the nature and activities of each of the discussed motifs. For example, *Trypsin* is a serine protease that breaks down proteins and cleaves peptide chains while *Trypsin_2* is

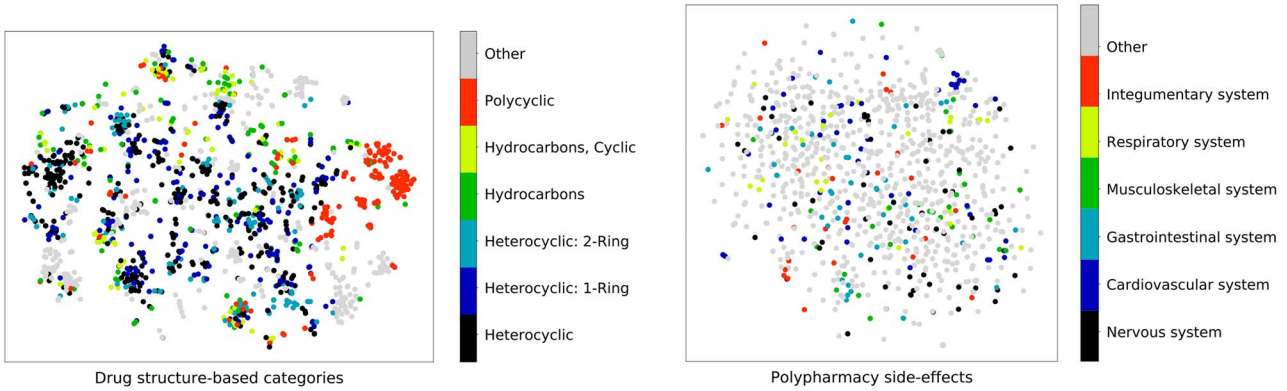

Figure 5. Three similarity matrices that denotes the drug–drug similarities, motif–motif similarities and protein–protein similarities. The similarity values are generated by computing the cosine similarity between the embeddings of the pairs of compared entities. All the embeddings used to generated this figure are computed on the DrugBank_FDA data sets with the proteins associated to their PFam [90] motifs and protein families.

an isozyme of *Trypsin*, which has a different amino acid sequence but catalyzes the same chemical reaction as *Trypsin* [98].

Moreover, the DUF1986 is a domain that is found in both of these motifs, which supports the high similarity scores. On the other hand, the *ANF receptor* is an atrial natriuretic factor receptor that binds to the receptor and causes the receptor to convert *GTP* to *cGMP*, and it plays a completely different role to trypsin, which supports its reported low similarity scores with trypsin.

- Protein–protein embedding similarity. The right similarity matrix in Figure 4 illustrates the protein–protein similarity scores between the set of the most frequent protein targets from the DTI benchmark. The highest-scored protein–protein pairs are (*PTGS1*, *PTGS2*) and (*CYP2C19*, *CYP2C9*) with the scores 0.8 and 0.8, respectively. This can be supported by the fact that the proteins CYP2C9, CYP1A2 and CYP2E1 belong to the same family of enzymes and thus they have similar roles.

  On the other hand, the *ACE* protein have the lowest similarity scores with the *CYP2C9*, *CYP1A2* and *CYP2E1* proteins with 0.0 similarity score. This can be supported by the fact that ACE is a hydrolase enzyme, which is completely different from *CYP2C9*, *CYP1A2* and *CYP2E1*, which are Oxidoreductases enzymes.

Clustering biological entities

In the following, we demonstrate the possible uses of embeddings based clustering in different biological tasks. We explore two cases where we use the embeddings of KGE models to generate clusters of biological entities such as drugs and polypharmacy side effects. We use visual clustering as an example to demonstrate cluster separation on a 2D space. However, in real scenarios, clustering algorithms utilize the full dimensionality of embedding vectors to build richer semantics of outcome clusters. Figure 5 shows two scatter plots of the embeddings of drugs from the DrugBank_FDA data set and the polypharmacy side effects reduced to a 2D space. We reduced the original embeddings using the T-SNE dimensionality reduction module [99] with the cosine distance configuration to reduce the embedding vectors to a 2D space.

The following examples examines two cases that differs in terms of the quality of generated clusters where we examine both drugs and polypharmacy side effects according to different properties. In the 1st example (drug clustering), the generated embeddings is able to provide efficient clustering. On the other hand, in the 2nd example, the polypharmacy side effects, the learnt embeddings could not be separated into visible clusters according to the investigated property.

- Clustering drugs. The left plot in Figure 5 shows a scatter plot of the reduced embedding vectors of drugs coloured according to their chemical structure properties. The drugs are annotated with seven different chemical structure annotations: *Polycyclic*, *Hydrocarbons Cyclic*, *Hydrocarbons*, *Heterocyclic*, *Heterocyclic 1-Ring*, *Heterocyclic 2-Ring* and other chemicals. These annotations represent the six most frequent drug chemical structure category annotation extracted from the DrugBank database.

  We can see in the plot that the *Polycyclic* chemicals are located within a distinguishable cluster in the right side of the plot. The plot also shows that other types of *Hydrocarbons* and *Heterocyclic* chemicals form different micro-clusters in different locations in the plot.

  These different clusters can be used to represent a form of similarity between the different drugs. It can also be used to examine the relation between the embeddings as a representation with the original attributes of the examined drugs.
- Clustering polypharmacy side effects. The right plot in Figure 5 shows a scatter plot of the reduced embedding vectors of polypharmacy side effects. The plot polypharmacy side-effect points are coloured according to the human body systems they affect. The plot includes a set of six categories of polypharmacy side effects that represent six different human body systems e.g. nervous system.

  Unlike the drug clusters illustrated in the left plot, the polypharmacy side-effect system-based categorization does not yield obvious clusters. They, however, form tiny and scattered groups across the plot. This shows that the KGE models are unable to learn representations that can easily separate polypharmacy side effects according to their associated body system.

Practical considerations for KGE models

In this section, we discuss different practical considerations related to the use of KGE models. We discuss their scalability

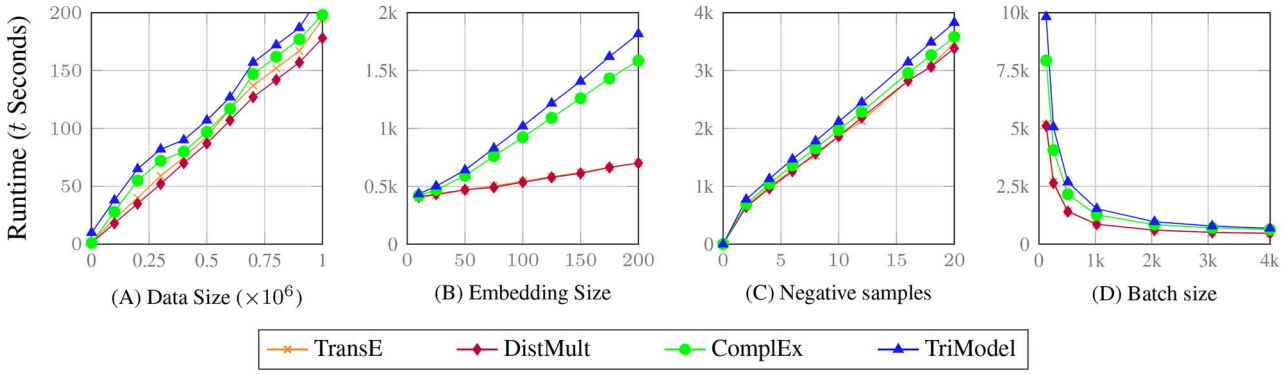

Figure 6. A set of line plots that describe the relation between the training runtime and the data size and configurable parameters of the TransE, DistMult, ComplEx and TriModel KGE models. The y-axis in all the plots represents the training time in seconds with different scales while the x-axis represents the data size and the models' parameters embedding size, negative samples and batch size respectively. The reported results are acquired by running the KGE models on the polypharmacy side-effects' full dataset (≈ 4.5M instances).

on different experimental configurations, and we explore their different training and implementation strategies.

Scalability

Not only KGE models outperform other approaches in biological knowledge graphs completion tasks but they also have better scalability compared to usual graph exploratory approaches. Often, complex biological systems are modelled as graphs where exploratory graph analytics methods are applied to perform different predictive tasks [5–7]. These models however suffer from limited scalability as they depend on graph traversal techniques that require complex training and predictions times [100, 101]. On the other hand, KGE models operate using linear time and space complexity [29, 59].

On the other hand, explanatory graph models use graph path searches, which require higher time and space complexity [22]. For example, the DDR model [21] is an exploratory graph drug–target predictor, which uses graph random walks as features. A recent study [102] has shown that KGE models can outperform such models with higher scalability and better predictive accuracy. This is due to their linear time and space complexity procedures [29] compared to other exploratory models, which use polynomial and exponential time and space procedures [23, 103].

In the following, we provide an empirical study of the scalability of KGE models in terms of different experimental configuration. We have studied the relation between the training runtime of KGE models and several training configuration parameters to examine their scalability capabilities. We have investigated the relation between the training runtime and the data size, embedding size, training negative samples and the training data batch size. We have performed our study on the polypharmacy side-effect data where the objective was to learn embeddings of drugs and polypharmacy side effects.

Figure 6 shows the outcome results of our study across the different investigated attributes. Plot 'A' shows the relation between the training runtime and the size of the processed data. The plot shows that all the four investigated have a linear relation between their training runtime and the investigated data size. The plot also shows that the investigated models have a consistent growth in terms of their runtime across all the data sizes. The DistMult model consistency achieves the smallest runtime followed by the TransE, ComplEx and TriModel models, respectively.

Plot 'B' shows the relationship between the training runtime and the model embedding size. The plot shows that all the investigated models have a linear growth of their training runtime corresponding to the growth of the embeddings size. However, the growth rate of the TransE and DistMult models is considerably smaller than the growth of both the ComplEx and TriModel models. This occurs as both the TransE and DistMult models use a single vector to represent each of their embeddings while the ComplEx and TriModel models use two and three vectors, respectively. Despite the better scalability of both the TransE and DistMult models, the ComplEx and TriModel models generally achieve better predictive accuracy than the TransE and DistMult models [64].

Plot 'C' shows the relation between the runtime of KGE models and the number of negative samples they use during training. The plot shows that there is a positive linear correlation between training runtime and the number of negative samples—where all the KGE models have similar results across all the investigated sampling sizes. The TriModel, however, consistently have the highest runtime compared to other models.

Plot 'D' shows the effects of the size of the batch on the training runtime. The plot shows an exponential decay of the training runtime with the linear growth of the data batch size. The KGE models process all the training data for each training iteration i.e. epoch, where the data are divided into batches for scalability and generalization purposes. Therefore, the increase of the training data batch sizes leads to a decrease of the number of model executions for each training iteration. Despite the high scalability that can be achieved with large batch sizes, the best predictive accuracy is often achieved using small data batch sizes. Usually, the most efficient training data batch size is chosen during a hyperparameter grid search along with other parameters such as the embedding size and the number of negative samples.

Implementation and training strategies

Different implementations of KGE models are available online in different repositories as shown in Table 2. The high scalability of KGE models allows them to be ported to both CPUs and GPUs where they can benefit from the high-performance capabilities of GPU cores. They can also be implemented to operate in a multi-machine design, where they perform embedding training in a distributed fashion [104]. This configuration is better suited

for processing knowledge graph of massive volumes that is hard to fit into one machine.

In this study, all our experiments are implemented in Python 3.5 using the Tensorflow library where we train our models on a single GPU card on one machine. We run our experiments on a Linux machine with an Intel(R) Core(TM) i7 processor, 32 GB RAM and an nVidia Titan Xp GPU.

Opportunities and challenges

In this section, we discuss the challenges and opportunities related to the general and biological applications of KGE models. We begin by discussing the scope of input data for these models. We then discuss possible applications of KGE models in the biological domain. We conclude by discussing the limited interpretability of KGE models and other general limitations related to their biological applications.

Potential applications

KGE models can build efficient representations of biological data, which are modelled as 3D tensors or knowledge graphs. This includes multiple types of biological data such as protein interactome and DTIs. In the following, we discuss examples of biological tasks and applications that can be performed using KGE models.

- Modelling proteomics data. KGE models can be used to model the different types of protein–protein interactions such as binding, phosphorylation, etc. [105, 106]. This can be achieved by modelling these interactions as a knowledge graphs and applying the KGE models to learn the embeddings of the different proteins and interaction types. They can also be used to model the tissue context of interactions where different body tissues have different expression profiles of proteins, and these differences in expression affect the proteins' interaction network. KGE can be used to model these interactions with their associated contexts as tensors [6].

  The biological activities of proteins also differ depending on their tissue context [86]. This type of information can easily be modelled using tensors where KGE models can be used to analyse the different functions of proteins depending on their tissue context [107].
- Modelling genomics data. Genomics data have been widely used to predict multiple gene associated biological entities such as gene–disease and gene–function associations [108, 109]. These approaches model the gene association in different ways including tensors and graph-based representations [110]. KGE models can be easily utilized to process such data and provide efficient representations of genes and their associated biological objects. They can be further used to analyse and predict new disease–gene and gene–function associations.
- Modelling pharmacological systems. Information on pharmaceutical chemical substances is becoming widely available on different knowledge bases [46, 71]. This information includes the drug–drug and drug–protein interactome. In this context, KGE models can be a natural fit, where they can be used to model and extend the current pharmacological knowledge. They can also be used to model and predict both traditional and polypharmacy side effects of drugs as shown in recent works [8, 111].

More details and discussion of the possible uses of KGE models and other general network embedding methods can be found in the study of Su *et al.* [14], which discusses further potential uses of these methods in the biological domain.

Limitations of the KGE models

In the following, we discuss the limitations of the KGE models in both general and biological applications.

- Lack of interpretability. In KGE models, the learning objective is to model nodes and edges of the graph using low-rank vector embeddings that preserve the graph's coherent structure. The embedding learning procedure operates mainly by transforming noise vectors to useful embeddings using gradient decent optimization on a specific objective loss. Despite the high accuracy and scalability of this procedure, these models work as a black box and they are hard to interpret. Some approaches have suggested enhancing the interpretability of KGE models by using constraining training with a set of predefined rules such as type constraints [112], basic relation axioms [113], etc. These approaches thus enforce the KGE models to learn embeddings that can be partially interpretable by their employed constraints.

  In recent studies, researchers have also explored the interpretability of KGE models through new predictive approaches on top of the KGE models. For example, Gusmão *et al.* [114] suggested the use of pedagogical approaches where they have used an alternative graphical predictive model, the SFE model [19], to link the learnt graph embeddings to the original knowledge graph. This approach was able to provide a new way for finding links between the embeddings and the original knowledge; however, the outcomes of these methods are still limited by the expressibility and feature coverage of the newly employed predictive models. The interpreting method in this context also depends on graph traversal methods, which have limited scalability on large knowledge graphs [20].
- Data quality. KGE models generate vector representations of biological entities according to their prior knowledge. Therefore, the quality of this knowledge affects the quality of the generated embeddings. For example, there is a high variance in the available prior knowledge on proteins where well-studied proteins have significantly higher coverage in most databases [115]. This has a significant impact on quality of the less represented proteins as KGE models will be biassed towards more studied proteins (i.e. highly covered proteins).

  In recent years, multiple works have explored the quality of currently available knowledge graphs [116] and the effect of low quality graphs on embedding models [117]. These works have shown that the accuracy KGE predictions degrade as sparsity and unreliability increase [117].

  This issue can be addressed by extending the available knowledge graph facts through merging knowledge bases of similar content. For example, drug–target prediction using KGE models can be enhanced by extending the knowledge of protein–drug interactions by extra information such as protein–protein interactions and drug properties [102].
- Knowledge evolution. Biological knowledge evolves everyday, where new chemicals and drugs are introduced and different associations between biological entities are discovered. However, KGE models in this context are unable to encode the newly introduced entities. This results from their

dependence on prior knowledge instead of the structural informations of proteins and chemical substances.

This issue can be addressed by combining KGE scoring procedure with other sequence- and structure-based scoring mechanisms. This can allow informed prediction on new unknown objects. However, such a strategy will affect the scalability of predictions due to the newly introduced sequence- and structure-based features.

- Hyperparameter sensitivity. The outcome predictive accuracy of KGE embeddings is sensitive to their hyperparameters [118]. Therefore, minor changes in these parameters can have significant effects on the outcome predictive accuracy of KGE models. The process of finding the optimal parameters of KGE models is traditionally achieved through an exhausting brute-force parameter search. As a result, their training may require rather time-consuming grid search procedure to find the right parameters for each new dataset.

  In this regard, new strategies for hyperparameter tuning such as differential evolution [119], random searches [120] and Bayesian hyperparameter optimization [121]. These strategies can yield a more informed parameter search results with less running time.
- Reflecting complex semantics of biological data in models based on knowledge graphs. KGE methods are powerful in encoding direct links between entities; however, they have limited ability in encoding simple indirect semantics such as types at different abstraction levels (i.e. taxonomies). For example, a KGE model can be very useful in encoding networks of interconnecting proteins, which are modelled using direct relations. However, it has limited ability in encoding compound, multi-level relationships such as protein involvement in diseases due to their involvement in pathways that cause this disease. Such compound relationships that could be used for modelling complex biological knowledge are notoriously hard to reflect in KGE models [122]. However, the KGE models do have some limited ability to encode for instance type constraints [123], basic triangular rules [122] or cardinality constraints [124]. This could be used for modelling complex semantic features reflecting biological knowledge in future works. One has to bear in mind, though, that the designs of these semantics-enhanced KGE models typically depends on an extra computational routines to regularize the learning process, which affects their scalability.

In their study, Su *et al.* [14] have also discussed further general limitations of network embedding methods and the effects and consequences of such limitations on the use of network embedding methods in the biological domain.

Conclusions

In this study, we discussed KGE models and their biological applications. We presented two biological case studies, predicting drug targets and predicting polypharmacy side-effects, to demonstrate the predictive and analytical capabilities of KGE models. We demonstrated by computational experimental evaluation that KGE models outperform state-of-the-art approaches in solving the two studied problems on standard benchmarks. We also demonstrated the analytical capabilities of KGE such as clustering and measuring concept similarities. In this regard, we demonstrated KGE models' abilities to learn efficient similarities between different biological entities such as drugs and proteins. We also showed that the KGE models can efficiently be used as clustering methods for biological entities.

Furthermore, we discussed different practical considerations regarding the scalability and training strategies of KGE models. We also discussed the potential applications of KGE models in the biological domain. We finally discussed the challenges and limitations that face KGE models where we explored both their general limitations and the challenges that face them in the biological domain. In conclusion, we believe that the presented study can be a solid stepping stone towards many promising applications of the emergent KGE technology in the field of computational biology.

Key Points

- Knowledge graphs allow easy, automated integration of multiple diverse biological data sets in order to model complex biological systems.
- KGE models enable scalable and efficient predictions on biological knowledge graphs.
- KGE models provide state-of-the-art predictive accuracy in learning biological associations with high scalability.
- KGE models provide high-quality analytics, e.g. clustering and concept similarities, of complex biological systems that can be modelled as graphs or 3D tensors.
- KGE models can be utilized to model and analyse different types of biological data including genomics, proteomics and pharmacological data.
- Despite their accurate and scalable predictive capabilities, however, KGE models have limited interpretability. They are also sensitive to data quality, knowledge evolution and training configurations.

Funding

The work presented in this paper was supported by the CLARIFY project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 875160, and by Insight research centre supported by the Science Foundation Ireland (SFI) grant (12/RC/2289_2).

Conflict of interest

None.

References

1. Cohen JD, Servan-Schreiber D. Context, cortex, and dopamine: a connectionist approach to behavior and biology in schizophrenia. *Psychol Rev* 1992;**99**(1):45–77.
2. Gibrat J-F, Madej T, Bryant SH. Surprising similarities in structure comparison. *Curr Opin Struct Biol* 1996;**6**(3):377–85.
3. Barabási A-L, Oltvai ZN. Network biology: understanding the cell's functional organization. *Nat Rev Genet* 2004;**5**:101–13.
4. Albert R. Scale-free networks in cell biology. *J Cell Sci* 2005;**118**(Pt 21):4947–57.
5. Janjic V, Przulj N. Biological function through network topology: a survey of the human diseasome. *Brief Funct Genomics* 2012;**11**(6):522–32.
6. Muñoz E, Novácek V, Vandenbussche P-Y. Facilitating prediction of adverse drug reactions by using knowledge graphs and multi-label learning models. *Brief Bioinform* 2019;**20**(1): 190–202.

7. Olayan RS, Ashoor H, Bajic VB. Ddr: efficient computational method to predict drug–target interactions using graph mining and machine learning approaches. *Bioinformatics* 2017;**34**(7):1164–73.
8. Zitnik M, Agrawal M, Leskovec J. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics* 2018;**34**(13): i457–66 .
9. Mohamed SK, Nováček V, Nounu A. Drug target discovery using knowledge graph embeddings. In: *Proceedings of the 34th Annual ACM Symposium on Applied Computing, SAC '19*, pp. 11–18. Limassol, Cyprus: ACM, 2019.
10. The UniProt Consortium. Uniprot: the universal protein knowledgebase. *Nucleic Acids Res* 2017;**45**(D1): D158–69.
11. The Gene Ontology Consortium. The gene ontology resource: 20 years and still GOing strong. *Nucleic Acids Res* 2019;**47**(D1): D330–8.
12. Dumontier M, Callahan A, Cruz-Toledo J, *et al*. Bio2rdf release 3: a larger, more connected network of linked data for the life sciences. In: *Proceedings of the ISWC 2014 Posters & Demonstrations*, pp. 401–4. Riva del Garda, Italy: CEUR-WS.org, 2014.
13. Alshahrani M, Khan MA, Maddouri O, *et al*. Neuro-symbolic representation learning on biological knowledge graphs. *Bioinformatics* 2017;**33**(17): 2723–30.
14. Su C, Tong J, Zhu Y, *et al*. Network embedding in biomedical data science. *Brief Bioinform* 2018. doi: 10.1093/bib/bby117
15. Nickel M, Murphy K, Tresp V, *et al*. A review of relational machine learning for knowledge graphs. *Proc IEEE* 2016;**104**(1):11–33.
16. Lao N, Mitchell TM, Cohen WW. Random walk inference and learning in a large scale knowledge base. In: *EMNLP*, Edinburgh, UK: ACL, 2011.
17. Xu B, Guan J, Wang Y, *et al*. Essential protein detection by random walk on weighted protein-protein interaction networks. *IEEE/ACM Trans Comput Biol Bioinform* 2017;**16**: 377–87.
18. Raman K. Construction and analysis of protein-protein interaction networks. *Autom Exp* 2010;**2**;12.
19. Gardner M, Mitchell TM. Efficient and expressive knowledge base completion using subgraph feature extraction. In: *EMNLP*, pp. 1488–98. Lisbon, Portugal: The Association for Computational Linguistics, 2015.
20. Mohamed SK, Nováček V, Vandenbussche P-Y. Knowledge base completion using distinct subgraph paths. In: *Proceedings of the 33rd Annual ACM Symposium on Applied Computing, SAC '18*, pp. 1992–9. Pau, France: ACM, 2018.
21. Olayan RS, Ashoor H, Bajic VB. DDR: efficient computational method to predict drug-target interactions using graph mining and machine learning approaches. *Bioinformatics* 2018;**34**(7):1164–73.
22. Toutanova K, Chen D. Observed versus latent features for knowledge base and text inference. In: *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pp. 57–66. Beijing, China: ACL, 2015.
23. Nickel M, Murphy K, Tresp V, *et al*. A review of relational machine learning for knowledge graphs. *Proc IEEE* 2016;**104**(1):11–33.
24. Wang Q, Mao Z, Wang B, *et al*. Knowledge graph embedding: a survey of approaches and applications. *IEEE Trans Knowl Data Eng* 2017;**29**(12):2724–43.
25. Lacroix T, Usunier N, Obozinski G. Canonical tensor decomposition for knowledge base completion. In: *ICML*, pp. 2869–78. JMLR Workshop and Conference Proceedings, Vol. 80. Stockholm, Sweden: JMLR.org, 2018.
26. Bordes A, Usunier N, García-Durán A, *et al*. Translating embeddings for modeling multi-relational data. In: *NIPS*, 2013, pp. 2787–95. Lake Tahoe, Nevada, United States: NIPS.
27. Nickel M, Tresp V, Kriegel H-P. A three-way model for collective learning on multi-relational data. In: *ICML*, pp. 809–16. Bellevue, Washington, USA: Omnipress, 2011.
28. Yang B, Yih W-T, He X, *et al*. Embedding entities and relations for learning and inference in knowledge bases. In: *ICLR*, San Diego, CA, USA: ICLR, 2015.
29. Trouillon T, Welbl J, Riedel S, *et al*. Complex embeddings for simple link prediction. In: *ICML*, pp. 2071–80. JMLR Workshop and Conference Proceedings, Vol. 48. New York City, NY, USA: JMLR.org, 2016.
30. Dettmers T, Pasquale M, Pontus S, *et al*. Convolutional 2d knowledge graph embeddings. In: *Proceedings of the 32th AAAI Conference on Artificial Intelligence*, New Orleans, Louisiana, USA: AAAI Press, 2018.
31. Zitnik M, Zupan B. Collective pairwise classification for multi-way analysis of disease and drug data. *Pac Symp Biocomput* 2016;**21**:81–92.
32. Abdelaziz I, Fokoue A, Hassanzadeh O, *et al*. Large-scale structural and textual similarity-based mining of knowledge graph to predict drug-drug interactions. *J Web Semant* 2017;**44**:104–17.
33. Qian R. Understand your world with bing, 2013. Bing Blogs.
34. Ferrucci DA, Brown EW, Chu-Carroll J, *et al*. Building Watson: an overview of the deepqa project. *AI Magazine* 2010;**31**(3):59–79.
35. Mitchell TM, Cohen WW, Hruschka ER, Jr, *et al*. Never-ending learning. In: *AAAI*, pp. 2302–10. New Orleans, Louisiana, USA: AAAI Press, 2015.
36. Miller GA. Wordnet: a lexical database for english. *Commun ACM* 1995;**38**(11):39–41.
37. Zhu Y, Elemento O, Pathak J, *et al*. Drug knowledge bases and their applications in biomedical informatics research. *Brief Bioinform* 2019;**20**(4): 1308–21.
38. Aronson AR, Mork JG, Gay CW, *et al*. The nlm indexing initiative's medical text indexer. *Stud Health Technol Informatics* 2004;**107**(Pt. 1):268–72.
39. Landrum MJ, Lee JM, Riley GR, *et al*. Clinvar: public archive of relationships among sequence variation and human phenotype. *Nucleic Acids Res* 2014;**42**(D1): D980–5.
40. Kanehisa M, Furumichi M, Tanabe M, *et al*. Kegg: new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Res* 2017;**45**(D1):D353–61.
41. Orchard SE, Ammari MG, Aranda B, *et al*. The mintact project intact as a common curation platform for 11 molecular interaction databases. *Nucleic Acids Res* 2014;**42**(D1): 358–63.
42. Fabregat A, Jupe S, Matthews L, *et al*. The reactome pathway knowledgebase. *Nucleic Acids Res* 2018;**46**(D1): D649–55.
43. Kanehisa M, Sato Y, Kawashima M, *et al*. Kegg as a reference resource for gene and protein annotation. *Nucleic Acids Res* 2016;**44**(D1):D457–62.
44. Wishart DS, Knox C, Guo AC, *et al*. Drugbank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic Acids Res* 2008;**36**:D901–6.
45. Mattingly CJ, Colby GT, Forrest JN, *et al*. The comparative toxicogenomics database (CTD). *Environ Health Perspect* 2003;**111**:793–5.
46. Gaulton A, Hersey A, Nowotka M, *et al*. The chembl database in 2017. *Nucleic Acids Res* 2017;**45**(D1):D945–54.

47. Kuhn M, Letunic I, Jensen LJ, *et al*. The sider database of drugs and side effects. *Nucleic Acids Res* 2016;**44**(D1): D1075–9.
48. Uhlén M, Fagerberg L, Hallstrom BM, *et al*. Tissue-based map of the human proteome. *Science* 2015;**347**(6220):1260419.
49. Szklarczyk D, Morris JH, Cook HV, *et al*. The string database in 2017: quality-controlled protein-protein association networks, made broadly accessible. *Nucleic Acids Res* 2017;**45**(D1): D362–8.
50. Stark C, Breitkreutz B-J, Chatr aryamontri A, *et al*. The BioGRID interaction database: 2011 update. *Nucleic Acids Res* 2007;**39**:D698–704.
51. Mitchell AL, Attwood TK, Babbitt PC, *et al*. Interpro in 2019: improving coverage, classification and access to protein sequence annotations. *Nucleic Acids Res* 2019;**47**(D1): D351–60.
52. Hewett M, Oliver DE, Rubin DL, *et al*. Pharmgkb: the pharmacogenetics knowledge base. *Nucleic Acids Res* 2002;**30**(1):163–5.
53. Chen X, Ji ZL, Chen YZ. TTD: therapeutic target database. *Nucleic Acids Res* 2002;**30**(1):412–5.
54. Hecker N, Ahmed J, von Eichborn J, *et al*. Supertarget goes quantitative: update on drug-target interactions. *Nucleic Acids Res* 2012;**40**(D1): 1113–7.
55. Belleau F, Nolin M-A, Tourigny N, *et al*. Bio2RDF: towards a mashup to build bioinformatics knowledge systems. *J Biomed Inform* 2008;**41**(5):706–16.
56. Bizer C, Cyganiak R. D2R server-publishing relational databases on the semantic web. In: *Poster at the 5th International Semantic Web Conference*, Vol. 175, 2006.
57. Amrouch S, Mostefai S. Survey on the literature of ontology mapping, alignment and merging. In: *2012 International Conference on Information Technology and e-Services*, pp. 1–5. Sousse, Tunisia: IEEE, 2012.
58. Ngomo A-CN, Auer S. Limes—a time-efficient approach for large-scale link discovery on the web of data. In: *Twenty-Second International Joint Conference on Artificial Intelligence*, Barcelona, Catalonia, Spain: IJCAI, 2011.
59. Mohamed SK, Muñoz E, Nováček V, *et al*. Loss functions in knowledge graph embedding models. In: *DL4KGS@ESWC*. CEUR Workshop Proceedings, Vol. 2106. Portoroz, Slovenia: CEUR-WS.org, 2019.
60. Bordes A, Glorot X, Weston J, *et al*. A semantic matching energy function for learning with multi-relational data—application to word-sense disambiguation. *Mach Learn* 2014;**94**(2):233–59.
61. Guo S, Wang Q, Wang L, *et al*. Jointly embedding knowledge graphs and logical rules. In: *EMNLP*, Austin, Texas, USA: ACL, 2016.
62. Nickel M, Rosasco L, Poggio TA. Holographic embeddings of knowledge graphs. In: *AAAI*, pp. 1955–61. Phoenix, Arizona USA: AAAI Press, 2016.
63. Liu H, Wu Y, Yang Y. Analogical inference for multi-relational embeddings. In: *ICML*, Sydney, Australia: ICML, 2017.
64. Mohamed SK, Novácek V. Link prediction using multi part embeddings. In: *ESWC*, pp. 240–54. Lecture Notes in Computer Science, Vol. 11503. Springer, 2019.
65. Perozzi B, Al-Rfou' R, Skiena S. Deepwalk: online learning of social representations. In: *SIGKDD*. 701–10. New York, USA: ACM, New York, 2014.
66. Grover A, Leskovec J. node2vec: scalable feature learning for networks. *KDD: Proceedings International Conference on Knowledge Discovery & Data Mining* 2016;**2016**: 855–64.
67. Terstappen GC, Schlüpen C, Raggiaschi R, *et al*. Target deconvolution strategies in drug discovery. *Nat Rev Drug Discov* 2007;**6**(11):891.
68. Sleno L, Emili A. Proteomic methods for drug target discovery. *Curr Opin Chem Biol* 2008;**12**(1):46–54.
69. Yamanishi Y, Araki M, Gutteridge A, *et al*. Prediction of drug–target interaction networks from the integration of chemical and genomic spaces. *Bioinformatics* 2008;**24**(13):i232–40.
70. Mei J-P, Kwoh C-K, Yang P, *et al*. Drug–target interaction prediction by learning from local information and neighbors. *Bioinformatics* 2012;**29**(2):238–45.
71. Wishart DS, Knox C, Guo AC, *et al*. Drugbank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic Acids Res* 2006;**34**:D668–72.
72. Cheng F, Zhou Y, Li W, *et al*. Prediction of chemical-protein interactions network with weighted network-based inference method. *PLoS One* 2012;**7**(7): e41064. https://doi.org/10.1371/journal.pone.0041064.
73. Cheng F, Liu C, Jiang J, *et al*. Prediction of drug-target interactions and drug repositioning via network-based inference. *PLoS Comput Biol* 2012;**8**(5): e1002503. https://doi.org/10.1371/journal.pcbi.1002503.
74. Rosdah AA, Holien JK, Delbridge LMD, *et al*. Mitochondrial fission–a drug target for cytoprotection or cytodestruction? *Pharmacol Res Perspect* 2016;**4**(3):e00235.
75. Liu H, Sun J, Guan J, *et al*. Improving compound–protein interaction prediction by building up highly credible negative samples. *Bioinformatics* 2015;**31**(12):i221–9.
76. Nascimento ACA, Prudêncio RBC, Costa IG. A multiple kernel learning algorithm for drug-target interaction prediction. *BMC Bioinform* 2016;**17**(1):46.
77. Hao M, Bryant SH, Wang Y. Predicting drug-target interactions by dual-network integrated logistic matrix factorization. *Sci Rep* 2017;**7**:40376.
78. Bowes J, Brown AJ, Hamon J, *et al*. Reducing safety-related drug attrition: the use of in vitro pharmacological profiling. *Nat Rev Drug Discov* 2012;**11**(12):909–22.
79. Kantor ED, Rehm CD, Haas JS, *et al*. Trends in prescription drug use among adults in the United States from 1999-2012. *JAMA* 2015;**314**(17):1818–31.
80. Tatonetti NP, Ye P, Daneshjou R, *et al*. Data-driven prediction of drug effects and interactions. *Sci Transl Med* 2012;**4**(125):125ra31.
81. García-Durán A, Niepert M. Kblrn: End-to-end learning of knowledge base representations with latent, relational, and numerical features. In: *UAI*, Monterey, California, USA: AUAI Press, 2018.
82. Fagerberg L, Hallström BM, Oksvold P, *et al*. Analysis of the human tissue-specific expression by genome-wide integration of transcriptomics and antibody-based proteomics. *Mol Cell Proteomics* 2014;**13**(2):397–406.
83. Greene CS, Krishnan A, Wong AK, *et al*. Understanding multicellular function and disease with human tissue-specific networks. *Nat Genet* 2015;**47**(6):569.
84. D'Agati VD. The spectrum of focal segmental glomerulosclerosis: new insights. *Curr Opin Nephrol Hypertens* 2008;**17**(3):271–81.
85. Cai JJ, Petrov DA. Relaxed purifying selection and possibly high rate of adaptation in primate lineage-specific genes. *Genome Biol Evol* 2010;**2**:393–409.

86. Zitnik M, Leskovec J. Predicting multicellular function through multi-layer tissue networks. *Bioinformatics* 2017;**33**(14): i190–8.
87. Tang J, Qu M, Wang M, *et al*. Line: large-scale information network embedding. In WWW, Florence, Italy: ACM, 2015.
88. Warde-Farley D, Donaldson SL, Comes O, *et al*. The genemania prediction server: biological network integration for gene prioritization and predicting gene function. *Nucleic Acids Res* 2010;**38**(Web-Server-Issue): 214–20.
89. Lim H, Gray P, Xie L, *et al*. Improved genome-scale multi-target virtual screening via a novel collaborative filtering approach to cold-start problem. *Sci Rep* 2016;**6**:38860.
90. Bateman A, Coin LJM, Durbin R, *et al*. The pfam protein families database. *Nucleic Acids Res* 2000;**28**(1):263–6.
91. Malone B, García-Durán A, Niepert M. Knowledge graph completion to predict polypharmacy side effects. In: *DILS*, Hannover, Germany: Springer, 2018.
92. Papalexakis EE, Faloutsos C, Sidiropoulos ND. Tensors for data mining and data fusion: models, applications, and scalable algorithms. *ACM Trans Intell Syst Technol* 2016;**8**:16:1–16:44.
93. Lipschitz WL, Hadidian Z, Kerpcsar A. Bioassay of diuretics. *Pharmacol Exp Ther* 1943, **79**(2):97–110.
94. Pohl JE, Thurston HF, Swales JD. The antidiuretic action of diazoxide. *Clinical Science* 1972, **42**(2):145–52.
95. Verster JC, Volkerts ER. Clinical pharmacology, clinical efficacy, and behavioral toxicity of alprazolam: a review of the literature. *CNS Drug Rev* 2004;**10**(1):45–76.
96. Overington JP, Al-Lazikani B, Hopkins AL. How many drug targets are there? *Nat Rev Drug Discov* 2006;**5**:993–6.
97. Minoda Y, Kharasch ED. Halothane-dependent lipid peroxidation in human liver microsomes is catalyzed by cytochrome P4502A6 (CYP2A6). *Anesthesiology* 2001;**95**(2): 509–14.
98. Rungruangsak-Torrissen K, Carter CG, Sundby A, *et al*. Maintenance ration, protein synthesis capacity, plasma insulin and growth of Atlantic salmon (salmo Salar L.) with genetically different trypsin isozymes. *Fish Physiol Biochem* 1999;**21**:223–33.
99. van der Maaten L. Accelerating t-sne using tree-based algorithms. *J Mach Learn Res* 2014;**15**:3221–45.
100. Cheung T-Y. Graph traversal techniques and the maximum flow problem in distributed computation. *IEEE Trans Softw Eng* 1983;**4**:504–12.
101. Fraigniaud P, Gasieniec L, Kowalski DR, *et al*. Collective tree exploration. *Network* 2006;**48**(3):166–77.
102. Mohamed SK, Novácek V, Nounu A. Discovering protein drug targets using knowledge graph embeddings. *Bioinformatics* 2019;**36**(2): 603–10.
103. Mohamed SK, Muñoz E, Novácek V, *et al*. Identifying equivalent relation paths in knowledge graphs. In: *LDK*, Galway, Ireland: Springer, 2017.
104. Lerer A, Ledell W, JS, *et al*. Pytorch-biggraph: a large-scale graph embedding system. In: *The 2nd SysML Conference*, Palo Alto, CA, USA: ACM, 2019.
105. Tuncbag N, Kar G, Keskin O, *et al*. A survey of available tools and web servers for analysis of protein-protein interactions and interfaces. *Brief Bioinform* 2008;**10**(3):217–32.
106. Zhang J, Kurgan LA. Review and comparative assessment of sequence-based predictors of protein-binding residues. *Brief Bioinform* 2018;**19**:821–37.
107. Mohamed SK. Predicting tissue-specific protein functions using multi-part tensor decomposition. *Inform Sci* 2020;**508**:343–57.
108. Bamshad MJ, Ng SB, Bigham AW, *et al*. Exome sequencing as a tool for Mendelian disease gene discovery. *Nat Rev Genet* 2011;**12**:745–55.
109. Zeng X, Ding N, Rodríguez-Patón A, *et al*. Probability-based collaborative filtering model for predicting gene-disease associations. *BMC Med Genomics* 2017;**10**:76. https://doi.org/10.1186/s12920-017-0313-y.
110. Bauer-Mehren A, Bundschus M, Rautschka M, *et al*. Gene-disease network analysis reveals functional modules in mendelian, complex and environmental diseases. *PLoS One* 2011;**6**(6):e20284. doi: 10.1371/journal.pone.0020284. Epub 2011 Jun 14.
111. Muñoz E, Novácek V, Vandenbussche P-Y. Using drug similarities for discovery of possible adverse reactions. In: *AMIA 2016*, American Medical Informatics Association Annual Symposium, Chicago, IL, USA, November 12–16, 2016. Chicago, IL, USA: AMIA, 2016.
112. Krompass D, Baier S, Tresp V. *Type-constrained representation learning in knowledge graphs*. In: *International Semantic Web Conference*, Bethlehem, PA, USA: Springer, 2015.
113. Minervini P, Costabello L, Muñoz E, *et al*. Regularizing knowledge graph embeddings via equivalence and inversion axioms. In: *ECML/PKDD*, Skopje, Macedonia: Springer, 2017.
114. Gusmão AC, Correia AHC, De Bona G, *et al*. Interpreting embedding models of knowledge bases: a pedagogical approach. In: *Proceedings of WHI*, Stockholm, Sweden: CoRR, 2018.
115. The Uniprot Consortium. Uniprot: a hub for protein information. *Nucleic Acids Res* 2015;**43**(D1): 204–12.
116. Färber M, Bartscherer F, Menne C, *et al*. Linked data quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO. *Semantic Web* 2017;**9**:77–129.
117. Pujara J, Augustine E, Getoor L. Sparsity and noise: where knowledge graph embeddings fall short. In: *EMNLP*, Copenhagen, Denmark: ACL, 2017.
118. Kadlec R, Bajgar O, Kleindienst J. Knowledge base completion: Baselines strike back. In: *Rep4NLP@ACL*, pp. 69–74. Vancouver, Canada: Association for Computational Linguistics, 2017.
119. Wei F, Nair V, Menzies T. Why is differential evolution better than grid search for tuning defect predictors? arXiv, abs/1609.02613. 2016.
120. Solis FJ, Wets RJ-B. Minimization by random search techniques. *Math Oper Res* 1981;**6**:19–30.
121. Snoek J, Larochelle H, Adams RP. Practical bayesian optimization of machine learning algorithms. In: *NIPS*, Lake Tahoe, Nevada, United States: NIPS, 2012.
122. Weber L, Minervini P, Münchmeyer J, *et al*. Nlprolog: reasoning with weak unification for question answering in natural language. In: *ACL (1)*, pp. 6151–61. Florence, Italy: Association for Computational Linguistics, 2019.
123. Minervini P, Costabello L, Muñoz E, *et al*. Regularizing knowledge graph embeddings via equivalence and inversion axioms. In: *ECML/PKDD (1)*. Lecture Notes in Computer Science, Vol. 10534. Springer, 2017, 668–83.
124. Muñoz E, Minervini P, Nickles M. Embedding cardinality constraints in neural link predictors. In: *SAC*, pp. 2243–50. Limassol, Cyprus: ACM, 2019.