Knowledge Graphs and Drug Repurposing

A haphazard summary of the state-of-the-art: existing papers, methods, applications and challenges

Siddharth Sahay

March 8, 2025

Contents

[1]In general, not biology related

Disclaimer: This document is mostly written up by me (based on ***my understanding***) and made up of excerpts (as well as tables and figures) ***copied and pasted*** from original research papers and other ***freely available sources*** on the internet, ***collected for personal reference, and not really for anything else***! I have also used NotebookLM by Google, which is a RAG system atop an LLM (Gemini) to extract and summarise certain sections for clarity and brevity. This is also the reason for the ***unconventional use of citations***, which have been used at the section/subsection level rather than sentence level. For the most accurate details, ***please refer to the sources cited***!

1 Knowledge Graphs vs. Knowledge Graph Embeddings [1][2][3][4][5]

A Knowledge Graph (KG) is a structured representation of knowledge where entities (nodes) are connected through relations (edges). Formally, a KG can be represented as a set of triples (h, r, t), where h (head) and t (tail) are entities, and r is the relation linking them. KGs are used in various applications, including search engines, recommender systems, and question-answering.

However, KGs are often incomplete, which motivates the use of Knowledge Graph Embeddings (KGEs). KGEs are learned **low-dimensional vector representations of entities and relations that capture semantic relationships in a continuous space**. Instead of operating directly on discrete symbols, KGEs encode entities and relations in a way that preserves structural and semantic properties, making it easier to predict missing links and perform reasoning tasks.

The effectiveness of a KGE model depends on how well it preserves the structure of the original knowledge graph while enabling efficient computation for downstream tasks. Various embedding models, such as translational models (e.g., TransE) and semantic matching models (e.g., ComplEx, DistMult), have been developed to optimize this representation.

1.1 Efficient Computation

Operations on embeddings in a Euclidean space, such as vector addition or finding the distance/similarity between vectors, are computationally efficient. This efficiency is beneficial for tasks like entity resolution, relationship prediction, and graph completion.

1.2 Dimensionality Reduction

Knowledge graphs can be highly complex and large, consisting of numerous nodes (entities) and edges (relationships). Embeddings help reduce this complexity by mapping high-dimensional graph structures into a lower-dimensional vector space while preserving important information.

1.3 Semantic Similarity

By embedding entities and relationships into a continuous space, similar entities and relationships can be positioned closer to each other in the vector space. This makes it easier to perform tasks like clustering, classification, and link prediction.

2 Knowledge Graph Embeddings [1][6]

Knowledge graph (KG) embedding is to embed components of a KG including entities and relations into **continuous vector spaces**, so as to **simplify the manipulation while preserving the inherent structure of the KG**. This representation makes KGs easier to manipulate and apply to downstream tasks like KG completion, relation extraction, and question answering.

Motivation for KG Embeddings

1. KGs are symbolic and hard to manipulate.
2. Embedding them in a vector space allows for easier computation while preserving their structure.

2.1 A typical KG embedding technique

A typical KG embedding technique generally consists of three steps: **(i) representing entities and relations, (ii) defining a scoring function, and (iii) learning entity and relation representations.**

The first step specifies the form in which entities and relations are represented in a continuous vector space. Entities are usually represented as vectors, i.e., deterministic points in the vector space. Relations are typically considered operations in vector space. They can be represented as vectors, matrices, tensors, multivariate Gaussian distributions, or even mixtures of Gaussian distributions.

In the second step, a scoring function $f_r(h, t)$ is defined for each fact (h, r, t) to measure its plausibility. Facts observed in the knowledge graph (KG) tend to have higher scores than those that have not been observed.

Finally, to learn entity and relation representations (i.e., embeddings), the third step involves solving an optimization problem that maximizes the total plausibility of the observed facts. This usually means **minimising a loss function** (such as a margin-based ranking loss function that considers true and false triples).

Once we've "solved for total plausibility", we have:

1. Maximized the scores of observed (true) facts.
2. Minimized the scores of unobserved or false facts.
3. Used a loss function to enforce this separation.
4. Updated embeddings via backpropagation.

We roughly categorize such embedding techniques into two groups: **translational distance models and semantic matching models**. The former uses distance-based scoring functions, and the latter similarity-based ones.

2.2 Translational Distance Models

Translational distance models exploit distance-based scoring functions. They measure the plausibility of a fact as the distance between the two entities, usually after a translation carried out by the relation.

2.2.1 TransE

TransE is a translational distance model that represents entities and relations in the same vector space. It assumes that a relation acts as a translation from the head entity to the tail entity. TransE represents entities and relations as vectors $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^d$ and models the relation as a translation:

$$\mathbf{h} + \mathbf{r} \approx \mathbf{t}. \tag{1}$$

The plausibility of a triple is scored using a distance function:

$$f_r(h, t) = -\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_p, \tag{2}$$

where $p = 1$ (Manhattan distance) or $p = 2$ (Euclidean distance).

Advantages of TransE

1. Simple and computationally efficient *O(d)*.
2. Performs well for simple one-to-one relations.

Limitations of TransE

- Fails for 1-to-N, N-to-1, and N-to-N relations.

Example: If "**City** $\rightarrow$ **LocatedIn** $\rightarrow$ **Country**" has multiple correct answers (e.g., **Paris** $\rightarrow$ **LocatedIn** $\rightarrow$ **France** and **Lyon** $\rightarrow$ **LocatedIn** $\rightarrow$ **France**), TransE forces all "France" embeddings to collapse into one vector. Cannot model relation-specific semantics (e.g., hierarchical or complex relations).

2.2.2 TransR

TransR extends TransE by introducing a **relation-specific projection matrix $\mathbf{M}_r$** to map entities into a relation-specific space:

$$\mathbf{h}_\perp = \mathbf{M}_r \mathbf{h}, \quad \mathbf{t}_\perp = \mathbf{M}_r \mathbf{t}. \tag{3}$$

The relation then operates in this transformed space:

$$\mathbf{h}_\perp + \mathbf{r} \approx \mathbf{t}_\perp. \tag{4}$$

The scoring function is:

$$f_r(h, t) = -\|\mathbf{M}_r \mathbf{h} + \mathbf{r} - \mathbf{M}_r \mathbf{t}\|_p. \tag{5}$$

Why TransR?

1. In **TransE**, both entities and relations are in the same space.
2. However, different relations may have different semantic meanings.
3. TransR **projects** entities into a space specific to each relation before applying translation.

Advantages of TransR

1. Better at modelling multi-relational knowledge graphs.
2. Handles 1-to-N, N-to-1, and N-to-N relations **more effectively**.
3. Allows different relations to operate in different vector spaces.

Limitations of TransR

- Increased computational complexity:

Projection requires *O(dk)* operations per entity per relation.

2.3 Semantic Matching Models

2.3.1 RESCAL

RESCAL (Nickel et al., 2011) is an early and expressive knowledge graph embedding model based on **bilinear factorization**. Unlike translational models, RESCAL explicitly models **higher-order dependencies** between entities.

Many simple KGE models (like TransE, DistMult, ComplEx) assume that **each triple (h,r,t) is independent** of other triples in the knowledge graph. This means:

- If **(Paris, CapitalOf, France)** is a valid fact, its score is computed **independently** of other related facts.
- There is **no mechanism** in these models to recognize that:
  - (France, LocatedIn, Europe)
    (Paris, LocatedIn, Europe)
    are **logically connected** (through **transitivity**).

This independence assumption **simplifies learning but ignores dependencies** between multiple relations.

Higher-order dependencies refer to **complex, multi-step relationships** between entities. Examples:

- **Transitivity**: If A $\rightarrow$ B and B $\rightarrow$ C, then A $\rightarrow$ C. Example: (Paris, LocatedIn, France) + (France, LocatedIn, Europe) then (Paris, LocatedIn, Europe)
- **Symmetry**: If A is related to B, then B is related to A. Example: (John, MarriedTo, Alice) then (Alice, MarriedTo, John)
- **Hierarchies**: Capturing multi-level relationships in taxonomies. Example: (Dog, SubclassOf, Mammal) + (Mammal, SubclassOf, Animal) then (Dog, SubclassOf, Animal)

Many KGE models **fail to capture these higher-order patterns** because they **score each triple separately**.

RESCAL explicitly models **pairwise interactions** between **all entities** in the KG using a **bilinear interaction matrix**. This allows it to:

- Recognize **multi-step relationships** between entities.
- Capture **dependencies between different relations** (e.g., "CapitalOf" implies "LocatedIn").
- Better handle **complex structures like taxonomies, social networks, and molecular graphs**.

Key Advantages

- Captures **higher-order dependencies** and complex relational patterns.
- More **expressive** than translational models (e.g., TransE, TransR).
- Effectively models **hierarchical, symmetric, and asymmetric relations**.

Limitations

- Computationally expensive: requires $O(nd + md^2)$ storage.
- Overfitting risk: large $\mathbf{M}_r$ matrices require regularization.

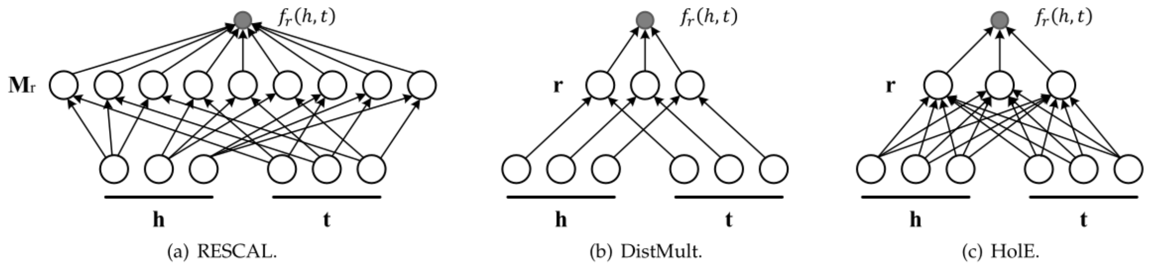

Figure 1: RESCAL, DistMulti, and HolE.

2.3.2 DistMulti

DistMult is like a **weighted dot product** between entity embeddings. It assumes that every relation has a **separate set of weights** that interact with entity embeddings, but the interaction happens **independently across each dimension**.

Intuition Imagine we have a **spreadsheet** where:

- Each **row** represents an entity (e.g., “Paris”, “France”, “Eiffel Tower”).
- Each **column** represents a feature (e.g., “is a capital”, “is a country”, “is a landmark”).

In DistMult, relations like **“CapitalOf”** assign **weights** to these features and decide how the entities interact. Since DistMult only uses **diagonal weights**, it assumes that each feature of the head entity and tail entity interact **independently**.

Key Advantages

- Efficient – Only learns a single weight per feature.
- Works well for symmetric relations (e.g., “is similar to”).

Limitations

- Fails for asymmetric relations (e.g., “BornIn” should behave differently from “ParentOf”, but DistMult treats them the same).

2.3.3 HolE

HolE combines the expressive power of RESCAL with the efficiency and simplicity of DistMult. HolE fixes DistMult’s problem by introducing a **compression trick** that lets it capture more complex interactions **without extra parameters**. Instead of just doing a **dot product**, it **mixes features across dimensions** using a mathematical operation called **circular correlation**.

Intuition Think of an image **hologram** – it stores 3D information using 2D projections. HolE does something similar by **encoding interactions across features without explicitly storing them all**.

- Instead of treating each feature separately (like DistMult), HolE **blends them** so that different parts of the entity embeddings can interact.
- This allows it to model **asymmetric relations** while keeping the model **compact and efficient**.

Key Advantages

- Captures asymmetric relations (e.g., “BornIn”, “ParentOf”)
- More expressive than DistMult, but still lightweight.

Limitations

- **Less interpretable** – Circular correlation is harder to understand than simple dot products.

Ultimately... DistMult = "Feature-by-feature interaction with weights" (good for symmetric relations, simple but limited). HolE = "Compressed but expressive interactions using a clever mathematical trick" (can handle asymmetric relations without bloating the model).

2.4 Matching with Neural Networks

Neural network-based methods for knowledge graph embeddings model relationships between entities through semantic matching. These methods define a scoring function to evaluate the plausibility of a fact (h,r,t) by matching the latent semantics of entities and relations in a continuous vector space. The latent semantics just mean the "hidden meaning" or knowledge that is brought out once we capture the hidden structure in data learning distributed representations.

2.4.1 ConvE

ConvE is a Knowledge Graph Embedding (KGE) model that uses a convolutional neural network (CNN) to capture interactions between head entities and relations.

- **Interaction Model:** For each triple (h, r, t), the input to ConvE is a matrix $A \in \mathbb{R}^{2\times d}$ where the first row of A represents $h \in \mathbb{R}^d$ and the second row represents $r \in \mathbb{R}^d$. A is reshaped to a matrix $B \in \mathbb{R}^{m\times n}$ where the first $m/2$ half rows represent h and the remaining $m/2$ half rows represent r.

- **Convolutional Layer:** In the convolution layer, a set of 2-dimensional convolutional filters $\Omega = \{\omega_i \mid \omega_i \in \mathbb{R}^{r\times c}\}$ are applied on B that capture interactions between h and r. The resulting feature maps are reshaped and concatenated in order to create a feature vector $v \in \mathbb{R}^{|\Omega|rc}$.

- **Linear Transformation:** In the next step, v is mapped into the entity space using a linear transformation $W \in \mathbb{R}^{|\Omega|rc\times d}$, that is $e_{h,r} = v^T W$. The score for the triple $(h, r, t) \in K$ is then given by: $f(h, r, t) = e_{h,r}t$.

- **1-N Scoring:** Since the interaction model can be decomposed into $f(h, r, t) = \langle f'(h, r), t\rangle$, the model is particularly designed for 1-N scoring, i.e., efficient computation of scores for (h, r, t) for fixed h, r and many different t.

- **Training:** ConvE and TuckER were originally trained in a multi-class setting using the binary cross-entropy loss where each (h, r)-pair has been classified against $e \in E$ simultaneously. If $|E| = n$, the label vector for each (h, r)-pair has n entries indicating whether the triple (h, r, e_i) is (not) part of the KG, and along each dimension of the label vector a binary classification is performed. It should be noted that there exist different implementation variants of the binary cross-entropy loss that address numerical stability. ConvE and TuckER employed a numerically unstable variant, and in the context of this work, we refer to this variant when referring to the binary cross-entropy loss.

- **Performance:** In a benchmarking study, ConvE achieved 56.33% Hits@10 on the WN18RR dataset, compared to 52.00% in the original paper. On the Kinships dataset, ConvE is among the top-ten-performing interaction models. On FB15K-237, ConvE also performs well among the interaction models.

- **Explicitly Modeling Inverse Relations:** Explicitly modeling inverse relations can improve the computational efficiency of ConvE.

2.4.2 Semantic Matching Energy (SME)

SME models interactions between entities and relations by transforming their vector embeddings through a hidden layer and computing a matching score.

SME is like a **translator** between entities and relations. Instead of using direct dot products (like DistMult), it first **transforms** entity and relation embeddings **into a common space** and then measures how well they match.

2.4.3 Neural Tensor Network (NTN)

NTN extends SME by introducing a relation-specific tensor to model complex interactions between entities. NTN is like **a neural network that dynamically decides how entities interact**. Instead of just adding or multiplying embeddings, it **uses a learned tensor (multi-dimensional matrix) to model interactions between every feature of both entities**.

2.4.4 Multi-Layer Perceptron (MLP)

MLP is a simpler architecture that uses a fully connected neural network to learn non-linear interactions. MLP treats knowledge graph reasoning like a black-box decision process. It takes in entity and relation embeddings, processes them through multiple non-linear layers, and outputs a score for how likely a fact is true.

MLP offers flexibility and can learn complex interactions, but requires significantly more training data compared to simpler models. Needs a lot of data to generalize well, but it's highly flexible – can learn very complex patterns.

Quick summary of latent semantics in neural networks

1. **In Semantic Matching Energy (SME):** The neural network transforms entity and relation embeddings to **discover** latent similarities between concepts.
2. **In Neural Tensor Networks (NTN):** A relation-specific tensor models interactions between entities to **learn** latent semantic patterns.
3. **In Multi-Layer Perceptron (MLP):** The network's hidden layers learn **non-linear latent features** that capture implicit relationships.

Some other general intuitions

- **Latent semantics** capture implicit meaning beyond explicit textual data.
- **Vector embeddings** of entities encode these hidden relationships.
- **Neural network-based models** use latent semantics to improve reasoning in knowledge graphs.

2.5 Model Training

2.5.1 Open World Assumption

Here, we assume that knowledge graphs contain only true facts, but missing facts are not necessarily false—they might just be unknown. Since missing facts could be either true or false, we can't assume all unobserved facts are negative. Instead, we generate negative examples carefully.

Steps in training with OWA

1. **Initialize embeddings** for entities and relations (usually random or pretrained from word vectors).
2. **Sample positive facts** from the knowledge graph (e.g., (Einstein, BornIn, Germany)).
3. **Generate negative samples**:
   - Replace the **head** or **tail** with a random entity (e.g., (Tesla, BornIn, Germany)).
   - Avoid generating obviously false facts.
4. **Train using a loss function**:
   - **Logistic loss** (pushes true facts to higher scores).
   - **Ranking loss** (ensures true facts have higher scores than negatives).
5. **Optimize using Stochastic Gradient Descent (SGD)** with adaptive learning rates.

2.5.2 Closed World Assumption

In the closed world assumption (CWA), we assume that any fact not explicitly in the knowledge graph is false.

2.6 Model Comparison

Table 1: Comparison of Models in Space and Time Complexity

Method	**Space complexity**	**Time complexity**
TransE	$\mathcal{O}(nd + md)$	$\mathcal{O}(d)$
TransH	$\mathcal{O}(nd + md)$	$\mathcal{O}(d)$
TransR	$\mathcal{O}(nd + mdk)$	$\mathcal{O}(dk)$
TransD	$\mathcal{O}(nd + mk)$	$\mathcal{O}(\max(d, k))$
TranSparse	$\mathcal{O}(nd + (1 - \theta)mdk)$	$\mathcal{O}(dk)$
TransM	$\mathcal{O}(nd + md)$	$\mathcal{O}(d)$
ManifoldE	$\mathcal{O}(nd + md)$	$\mathcal{O}(d)$
TransF	$\mathcal{O}(nd + md)$	$\mathcal{O}(d)$
TransA	$\mathcal{O}(nd + md^2)$	$\mathcal{O}(d^2)$
KG2E	$\mathcal{O}(nd + md)$	$\mathcal{O}(d)$
TransG	$\mathcal{O}(nd + mdc)$	$\mathcal{O}(dc)$
UM	$\mathcal{O}(nd)$	$\mathcal{O}(d)$
SE	$\mathcal{O}(nd + md^2)$	$\mathcal{O}(d^2)$
RESCAL	$\mathcal{O}(nd + md^2)$	$\mathcal{O}(d^2)$
TATEC	$\mathcal{O}(nd + md^2)$	$\mathcal{O}(d^2)$
DistMult	$\mathcal{O}(nd + md)$	$\mathcal{O}(d)$
HolE	$\mathcal{O}(nd + md)$	$\mathcal{O}(d \log d)$
ComplEx	$\mathcal{O}(nd + md)$	$\mathcal{O}(d)$
ANALOGY	$\mathcal{O}(nd + md)$	$\mathcal{O}(d)$
SME (linear)	$\mathcal{O}(nd + md)$	$\mathcal{O}(d^2)$
SME (bilinear)	$\mathcal{O}(nd + md)$	$\mathcal{O}(d^3)$
NTN	$\mathcal{O}(nd + md^2k)$	$\mathcal{O}(d^2k)$
SLM	$\mathcal{O}(nd + mdk)$	$\mathcal{O}(dk)$
MLP	$\mathcal{O}(nd + md)$	$\mathcal{O}(d^2)$
NAM	$\mathcal{O}(nd + md)$	$\mathcal{O}(Ld^2)$

2.7 Applications of KGEs

KG embeddings have a bunch of applications, categorized into **in-KG applications** and **out-of-KG applications**.

2.7.1 In-KG Applications (within the Knowledge Graph)

- **Link Prediction**: Predicting missing links in the KG, e.g., inferring missing relations like (Tesla, BornIn, ?).
- **Triple Classification**: Determining whether a given fact (h, r, t) is true or false based on learned embeddings.
- **Entity Classification**: Assigning semantic categories to entities using KG embeddings.
- **Entity Resolution**: Identifying and merging duplicate entities across different KGs.

Link Prediction Link prediction aims to infer missing entities or relations in a knowledge graph by **predicting the most plausible missing entity in a triple** (h, r, ?) or (?, r, t). Different approaches model these relationships in distinct ways. Translational distance models, such as TransE, represent relations as vector translations in the embedding space, enforcing the constraint $h + r \approx t$. Given a query like (Einstein, BornIn, ?), the model identifies the entity t that is closest to Einstein + BornIn in vector space. **Semantic matching models**, including DistMult, HolE, and ComplEx, **use bilinear similarity functions to assess entity interactions**. If a triple (h, r, t) has a high score, it is considered a likely missing link. In contrast, **neural network-based models**, such as Neural Tensor Networks (NTN) and Multi-Layer Perceptron (MLP), **employ deep learning architectures to model non-linear transformations**, capturing more expressive and complex relationships between entities and relations. These models are particularly useful in scenarios where entity interactions are too intricate for simple algebraic formulations.

Triple Classification Triple classification determines whether a fact **(h, r, t)** is true or false. Embedding models like **TransE** or **DistMult** compute a **scoring function**, and a **threshold** is applied to classify triples. The challenge lies in **false negatives**, where missing but valid facts may be misclassified due to knowledge graph incompleteness.

Entity Classification Entity classification assigns entities to **semantic categories** using embeddings from models like **TransE** or **ComplEx**. A **classifier** (e.g., MLP or softmax) is trained on labeled embeddings. For unseen entities, classification relies on **nearest neighbors** in embedding space. The key challenge is handling **multi-label** cases where entities belong to overlapping categories.

Entity Resolution (Deduplication & Linking) Entity resolution identifies when different entity representations refer to the **same real-world entity**. Embeddings (e.g., **TransE, BERT**) encode entity descriptions, and **cosine similarity** measures how closely two entities align. If the similarity surpasses a **threshold**, the entities are merged. The challenge is handling **inconsistent or incomplete data** across sources.

2.7.2 Out-of-KG Applications (beyond the Knowledge Graph)

- **Relation Extraction**: Extracting structured facts from unstructured text using KG embeddings to improve NLP models.
- **Question Answering (QA)**: Using KG embeddings to enhance knowledge-based QA systems.
- **Recommender Systems**: Using KG embeddings to improve collaborative filtering by capturing item relationships.

Relation Extraction Relation extraction identifies and classifies relationships between entities in unstructured text, often using **knowledge graph embeddings** to enhance accuracy. The process typically involves **pre-trained language models (e.g., BERT, RoBERTa)** or **neural networks** that encode textual context and extract relational patterns. Entities are mapped to embeddings, and a classifier predicts the relation type based on learned interactions. Knowledge graph embeddings like **TransE** or **ComplEx** can be integrated to improve relation prediction, ensuring extracted facts align with existing structured knowledge.

Question Answering (QA) QA systems use **knowledge graphs** to provide structured answers to natural language queries. Given a question, entity linking maps keywords to KG entities, while relation matching identifies relevant facts. Models like **Graph Neural Networks (GNNs)** or **Transformer-based retrievers** (e.g., **BERT for KGQA**) retrieve and rank possible answers. Embedding-based approaches score candidate triples using techniques like **TransE or DistMult**, ensuring the most relevant fact is returned. Hybrid models also integrate **retrieval-based search** with **neural networks** for deeper reasoning.

Recommender Systems KG embeddings improve recommendations by modeling user-item interactions as **entity relationships**. In **knowledge-aware recommenders**, users and items are represented as nodes, with relations encoding interactions (e.g., "likes," "purchased"). Graph-based models like **RGCN (Relational Graph Convolutional Networks)** or **TransE-based link prediction** infer missing connections, predicting potential user preferences. Personalized recommendations emerge from **embedding similarity** or **graph traversal**, allowing systems to make context-aware suggestions beyond traditional collaborative filtering.

2.8 Benchmarking and Reproducibility

Many newly published KG embedding approaches are hard to compare consistently because of differences in codebases, hyperparameters, training regimes, and how evaluation metrics are computed. Many published results were only replicable after tuning different hyperparameters than those originally reported—sometimes because important details were missing from the papers or official code was unavailable. In some cases, results could not be matched at all[1], suggesting that minor implementation differences or unreported steps (e.g., non-default initializations, gradient clipping, partial early stopping routines) can significantly affect outcomes.

2.8.1 Key observations

1. Baseline models (like DistMult or TransE) can sometimes match or surpass "fancier" ones if given the right hyperparameters or well-chosen loss configurations.
2. Some high-parameter models (e.g., RESCAL, TuckER, or QuatE) can perform extremely well, but simpler ones (DistMult, ComplEx, RotatE) often come surprisingly close—especially if optimized carefully.
3. Many incremental "state-of-the-art" claims can shrink or disappear when older baselines are pushed to their optimal configurations! (Of course, "optimal" being the operating keyword here.)

2.9 Evaluation Metrics

Given a test triple (h,r,t), the model is asked to predict either the correct tail entity t (the "tail prediction" task) or the correct head entity h (the "head prediction" task). In each case, the model's score for the ground-truth triple to the scores it assigns to "corrupted" triples (replacing h or t with other entities) is compared.

Different models are evaluated using various **benchmark tasks**, with common evaluation metrics including:

- **Mean Rank (MR)**: The average rank of correct answers in a ranked list.
- **Mean Reciprocal Rank (MRR)**: The average of the reciprocal ranks of correct answers.
- **Hits@N**: The proportion of correct answers ranked in the top-N predictions.
- **AUC-PR (Area Under Precision-Recall Curve)**: Measures classification performance in tasks like triple classification.

Evaluation strategies depend on whether the KG follows the **Open World Assumption (OWA)** (missing data is unknown) or the **Closed World Assumption (CWA)** (missing data is considered false).

2.9.1 Mean Rank (MR)

The *Mean Rank* (MR) computes the average rank of the correct entity across all test triples. Given a set of test triples $\mathcal{T}_{test}$, the MR is defined as:

$$\text{MR} = \frac{1}{|\mathcal{T}_{test}|} \sum_{(h,r,t)\in\mathcal{T}_{test}} \text{rank}(h,r,t), \tag{6}$$

where $\text{rank}(h,r,t)$ is the rank assigned to the correct entity in a ranked list of possible entities. A lower MR indicates better performance.

2.9.2 Mean Reciprocal Rank (MRR)

The *Mean Reciprocal Rank* (MRR) is a widely used metric that computes the average reciprocal of the rank assigned to the correct entity:

$$\text{MRR} = \frac{1}{|\mathcal{T}_{test}|} \sum_{(h,r,t)\in\mathcal{T}_{test}} \frac{1}{\text{rank}(h,r,t)}. \tag{7}$$

Unlike MR, MRR is less sensitive to extreme rank values and provides a smooth evaluation score in the range $[0,1]$.

2.9.3 Hits@K and Hits@N

The *Hits@K* metric measures the proportion of test triples for which the correct entity appears in the top-K predictions:

$$\text{Hits@K} = \frac{1}{|\mathcal{T}_{test}|} \sum_{(h,r,t)\in\mathcal{T}_{test}} \mathbb{I}(\text{rank}(h,r,t) \leq K), \tag{8}$$

where $\mathbb{I}(\cdot)$ is an indicator function that returns 1 if the correct entity is ranked within the top K, and 0 otherwise. Higher Hits@K values indicate better performance.

The *Hits@N* metric is simply a generalization of Hits@K, where N represents a specific value of K. In practice, Hits@10 is sometimes referred to as Hits@N when $N = 10$. Both metrics provide insight into how frequently the correct entity is ranked within a given threshold.

2.9.4 Adjusted Mean Rank (AMR)

The *Adjusted Mean Rank* (AMR) normalizes the MR by the expected rank under random scoring:

$$\text{AMR} = \frac{\text{MR}}{\frac{1}{2}\sum_{(h,r,t)\in\mathcal{T}_{test}}(\xi(h,r,t)+1)}, \tag{9}$$

where $\xi(h,r,t)$ denotes the number of candidate triples against which the true triple is ranked. The AMR helps compare different datasets by adjusting for dataset size.

2.9.5 Filtered vs. Unfiltered Evaluation

In knowledge graph completion, often *filtered* evaluation is employed, where known true triples (from the training, validation, and test sets) are removed from the candidate set before ranking. This prevents unfair penalization when multiple correct answers exist for a given query.

Filtered metrics typically yield more reliable insights into model performance and are now standard practice in KGE evaluation.

2.9.6 Two Training Philosophies (LCWA vs. sLCWA)

Local Closed World Assumption (LCWA) Under LCWA, for each true triple (h,r,t) all triples (h,r,t') $\notin$ KG are considered negative. In practice, you do a **1-to-N scoring pass**: for each (h,r), the model outputs scores over all candidate tails in one go, computing a cross-entropy or other multi-class style loss. This can be memory-heavy on larger knowledge graphs—because you have to consider "all possible tails" in each step—but it ensures the model compares the golden triple to every negative candidate tail in a single batch.

Stochastic Local Closed World Assumption (sLCWA) Under sLCWA, negative samples are generated by *corrupting* a single triple, either replacing h or t. You do a *1-to-k* scoring pass (e.g., for each positive triple, you generate k negative samples). This approach is more common in earlier translational models (TransE, RotatE, etc.) because it's easy to batch, uses less memory, and can be combined with advanced negative-sampling heuristics (like Bernoulli sampling or self-adversarial sampling).

2.10 Future Research Directions for KGEs[2]

- **Incorporating External Knowledge**: Integrating textual descriptions, ontologies, or logical rules into KG embeddings.
- **Scalability**: Developing more efficient models that can handle very large knowledge graphs.
- **Handling Dynamic KGs**: Many KGs evolve over time, requiring embeddings to be updated without retraining from scratch.
- **Explainability**: Improving interpretability of KG embeddings to better understand learned relationships.
- **Multi-modal Learning**: Using images, text, and structured data together to enhance entity representation.

[2]In general, not biology related

3 Biological Applications of KGE models [7]

Knowledge graph embeddings provide *low-dimensional vector representations* for both entities (e.g. proteins, drugs) and relations (e.g. "targets," "inhibits") in a multi-relational graph. The paper highlights that these embeddings are learned by **scoring factual triplets**—subject, relation, object (S, R, O)—and **contrasting** them with negative (corrupted) triplets. After training, the embeddings encode many global aspects of the data, enabling tasks such as *link prediction* (finding missing edges in the graph) and *clustering* (grouping entities by similarity).

In **biology**, KGE models thus turn large multi-relational databases—like drug–target relationships, protein–protein interactions, gene ontology information—into *dense numeric representations.* They can then infer new links (e.g. *predicting unknown drug–target interactions*) or detect emergent patterns (e.g. *grouping drugs by chemical scaffolds* or *learning "similar" proteins* based on functional closeness).

Multiple studies have explored KGE models, their technical design, training objectives and predic- tive capabilities on general benchmarking settings. 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, 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. KGE models then lookup corresponding embed- ding of both the true and corrupted triplets. The embeddings are then processed using model-dependent scoring functions 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.

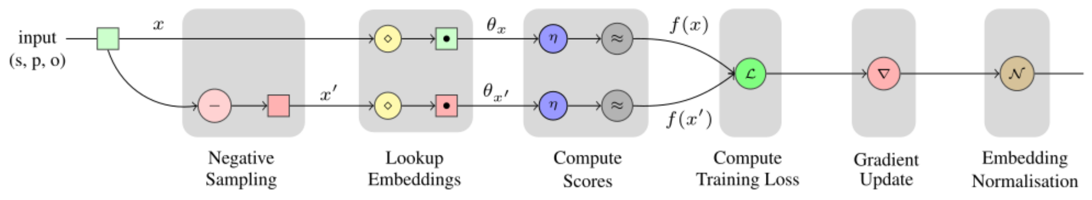

Figure 2: An illustration of the training network of one training instance of a KGE model.

Traditionally, KGE models use a ranking loss, e.g. hinge loss or logistic loss, to model the objective training cost. This strategy allows KGE models to efficiently train their embeddings in linear time, O(d), where K denotes the size of the embedding vectors. On the other hand, some KGE models such as the ConvE and the ComplEx-N3 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. 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, AMS- Grad, 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. [7]

3.1 Drug–Target Interaction (DTI) Prediction

1. **Setup**

   - DTIs are (Drug, targets, Protein) triplets within a larger knowledge graph that may also include other edges such as (Drug, side_effect, SE), (Protein, involved_in_pathway, Pathway), etc.
   - KGE models learn embeddings for every drug and every protein, as well as each relation type ("targets," "involved_in," "side_effect," etc.). During training, negative samples are created by randomly replacing the protein or drug with another entity.

2. **Scoring Function** For each (Drug, targets, Protein) triplet, the model computes a scalar *score* using its chosen scoring mechanism (e.g. a translational distance in TransE, a dot

product in DistMult, etc.). True triplets are pushed to have higher scores, while corrupted triplets are forced lower.

3. **Use Case** After training, the model can rank new candidate (Drug?, targets, Protein?) triplets by computing the embedding-based score. If the score is high, it suggests a potential novel DTI. On standard benchmarks (e.g. the "DrugBank_FDA" subset), KGE-based solutions (DistMult, ComplEx, TriModel) outperformed prior matrix-factorization or random-walk-based methods in ROC and PR metrics.

4. **Practical Benefits**

   - **Scalability**: KGEs handle large volumes of drugs and proteins with near-linear complexity in the number of edges and embedding dimensions.
   - **Ease of Integration**: Additional relations (such as gene ontology annotations or drug–drug interactions) can be trivially added to the same graph.

3.2 Polypharmacy Side Effects

1. **Problem Setting** Polypharmacy side effects occur when taking multiple drugs together. The authors treat this scenario as a link-prediction task in a 3D tensor or knowledge graph: (Drug 1, side_effect_type, Drug 2). If the side effect is known, that edge is positive; unknown or random drug pairs are negative.

2. **Tensor Factorization Perspective** One can interpret polypharmacy events as a 3D adjacency matrix, where axis 1 is "drug 1," axis 2 is "drug 2," and axis 3 is "side_effect_type." A "1" means that side effect type is observed when drug1 is co-administered with drug2. KGE-based factorization then learns 3D embeddings that can fill missing entries.

3. **Decagon vs. KGE** The well-known "Decagon" model uses graph neural networks. The paper's experiments show that classical KGE methods (DistMult, ComplEx, TriModel) can achieve even stronger or comparable performance with simpler, more scalable training. They directly do link prediction on the (Drug, side_effect, Drug) edges.

4. **Outcome** The embeddings are used to identify potential adverse events from drug combinations not previously recorded. This is critical for clinical risk management.

3.3 Tissue-Specific Protein Functions

1. **Motivation** A single protein can have different roles in different tissues (e.g. it may be expressed in brain tissue vs. heart tissue). The relevant data can be modeled as a multi-relational graph or as multiple "layers," each representing one tissue's interactions.

2. **KGE Workflow**

   - Each protein–function or protein–protein interaction within a tissue context is a triple (Protein, interacts_in_tissueA, Protein) or (Protein, has_function_in_tissueB, Function).
   - KGEs learn embeddings that capture these tissue-specific relationships.

3. **Comparison** Past approaches (OhmNet, GeneMania) employed specialized multi-layer or propagation-based models. KGE methods, by contrast, can directly integrate the extra "tissue" dimension as an additional relation type, often matching or exceeding performance.

3.4 Capabilities of KGE models

KGE models can be used in different supervised and unsuper- vised applications where they provide efficient representations of biological concepts.

3.4.1 Learning biological associations

... by processing data in knowledge graphs or 3D tensors.

In a knowledge graph, entities and relations are represented as nodes and edges, respectively. **KGE models learn low-rank representations that preserve the graph's inherent structure**. When processing data in a 3D tensor, KGE models learn low-rank representations that maintain the integrity of entity combinations within the tensor. For example, in **drug–target**

interaction (DTI) prediction, biological information can be modelled as a **knowledge graph**. By evaluating the predictive accuracies of KGE models, their performance can be compared against other state-of-the-art approaches. Similarly, in predicting **drug polypharmacy side effects**, data can be modelled as a **3D tensor**. KGE models can then be applied to **perform tensor factorization**, and their predictive accuracy in learning new polypharmacy side effects can be evaluated against other methods. A comparison of state-of-the-art drug–target predictors and KGE models demonstrates the effectiveness of KGE models in predicting DTIs. Drug-target knowledge bases, such as **DrugBank** and **KEGG**, are structured as networks that represent information about drugs and their relationships with target proteins, action pathways, and targeted diseases. This data can be interpreted as a knowledge graph, where finding new associations between drugs and their targets can be framed as a link prediction problem. The standard evaluation protocol for DTI prediction on the DrugBank_FDA dataset involves a 5-fold cross-validation. The KGE models (**DistMult, ComplEx, and TriModel**) **outperform** other approaches (DDR, DNILMF, NRLMF, KRONRLS-MKL, COSINE, and BLM-NII) **in terms of both the area under the ROC and precision-recall curves**.

In predicting polypharmacy side effects, KGE models are compared with state-of-the-art approaches using a holdout test. The data is split into training and testing sets, and random negative polypharmacy side effects are generated. **KGE models** such as **DistMult, ComplEx, and TriModel outperform** approaches like Decagon, KB_LRN, RESCAL, DEDICOM, and DeepWalk8. Similarly, in predicting tissue-specific protein functions, KGE models are evaluated against traditional approaches such as OhmNet, LINE, GeneMania, and SVM8. The KGE models, such as **TriModel and ComplEx**, achieve the best results **in terms of both the area under the ROC and precision-recall curves**.

3.4.2 Measure similarities between biological entities

...by evaluating the similarity between their vector representations.

These vector similarities can be computed using techniques such as **cosine and p-norm similarities**. Since KGE representations are trained to preserve the knowledge graph structure, the similarity between two KGE representations reflects their similarity in the original knowledge graph. For instance, in a drug–target knowledge graph, embeddings of drugs, their target proteins, and the motifs of these proteins can be learned. The similarities between embeddings of entities of the same type, such as drugs, proteins, and motifs, can then be computed. These similarity scores range from 0.0 to 1.0, where 0.0 represents the least similar pairs and 1.0 represents the similarity between an entity and itself. The validity of these scores can be assessed by investigating the similarity of attributes of concepts with the highest and lowest scores. For example, drug-drug similarity scores can be computed on the embeddings of drugs learned in the DTI training pipeline. These scores reflect the commonalities between the investigated drugs in terms of indications, pharmacodynamics, mechanism of action, targets, enzymes, carriers, and transporters. Similarly, motif-motif similarity scores can be computed between the most frequent PFam motifs associated with protein targets from the DTI benchmark. These scores reflect the nature and activities of each of the discussed motifs. Protein–protein similarity scores can also be computed between the most frequent protein targets from the DTI benchmark. These scores reflect the family and roles of the proteins.

3.4.3 Clustering biological entities

... providing efficient representations of biological concepts.

Visual clustering can demonstrate cluster separation on a 2D space, but clustering algorithms typically use embedding vectors' full dimensionality to build richer outcomes clusters' semantics. For example, the embeddings of drugs from the DrugBank_FDA dataset and polypharmacy side effects can be reduced to a 2D space using the T-SNE dimensionality reduction module. This allows for the examination of drug and polypharmacy side effect clustering based on different properties. In **drug clustering**, the generated embeddings **can provide efficient clustering based on chemical structure properties**. Polycyclic chemicals, hydrocarbons, and heterocyclic chemicals form distinguishable clusters, representing a form of similarity between the different drugs. These clusters can be used to examine the relation between the embeddings as a representation with the original attributes of the examined drugs. However, in clustering polypharmacy side effects, the system-based categorization may not yield obvious clusters. **This indicates that KGE models may not always be able to learn representations that can easily separate polypharmacy side effects according to their associated body system**. 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, **KGE models have limited interpretability and are sensitive to data quality, knowledge evolution and training configurations**.

3.5 Scalability

KGE models show better scalability compared to traditional graph exploratory approaches. Complex biological systems are often modelled as graphs, and exploratory graph analytics methods are applied to perform different predictive tasks. However, these models suffer from limited scalability because they depend on graph traversal techniques that require complex training and prediction times. KGE models, on the other hand, operate using linear time and space complexity. Explanatory graph models use graph path searches, which require higher time and space complexity. For example, the DDR model is an exploratory graph drug-target predictor, which uses graph random walks as features. A recent study 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 compared to other exploratory models, which use polynomial and exponential time and space procedures.

3.6 Training & Inference

1. **Negative Sampling** For each true biological edge (e.g. "Aspirin targets PTGS1"), the KGE pipeline creates random negative examples by corrupting the subject or object (e.g. "Aspirin targets PTGS2" if that is known to be false). The model is trained to give higher scores to real edges vs. corrupted edges.
2. **Scoring & Loss Function** Different KGE methods define different "scoring" functions (translational, bilinear, convolutional). They typically use margin-based or logistic ranking loss. The goal is: *true triplets > negative triplets.*
3. **Optimization** Gradients are computed via mini-batch stochastic gradient descent or variants (Adam, Adagrad), ensuring the model remains scalable.
4. **Predicting New Links** Once trained, to check if "Drug X targets Protein Y," the model calculates the *score* for (X, targets, Y). A high score suggests a likely new link.
5. **Integration of Additional Info** Biological knowledge often includes data like expression levels, gene family membership, or disease associations. In a KGE pipeline, these are simply more edges. The same model updates embeddings in a multi-task manner.

3.7 Challenges, Limitations & Practical Notes

1. **Interpretability** Embeddings are powerful but are "black box." This makes it tough to provide explicit mechanistic explanations.
2. **Data Quality & Coverage** KGE models rely on *existing knowledge graph completeness.* If a protein is understudied (few known edges), its embedding quality is limited. Also, standard KGE approaches cannot predict embeddings for truly new entities not appearing in the training graph, without retraining or hybridizing with other (e.g. sequence-based) models.
3. **Hyperparameter Sensitivity** Different KGE approaches (e.g. DistMult, ComplEx) and embedding sizes can drastically change results. The embedding dimension, negative sampling ratio, and learning rate are especially important; performance can vary significantly with different settings. Grid or Bayesian searches are typically employed.
4. **Knowledge Evolution** If new drugs or proteins appear, typical KGE methods have no embeddings for them unless retrained (or augmented with additional sub-models that incorporate sequence data, etc.).
5. **Complex Semantics** Real biology often has multi-level or conditional logic that is hard to capture with standard "triplet" embeddings, though some specialized research tries to integrate ontological constraints.

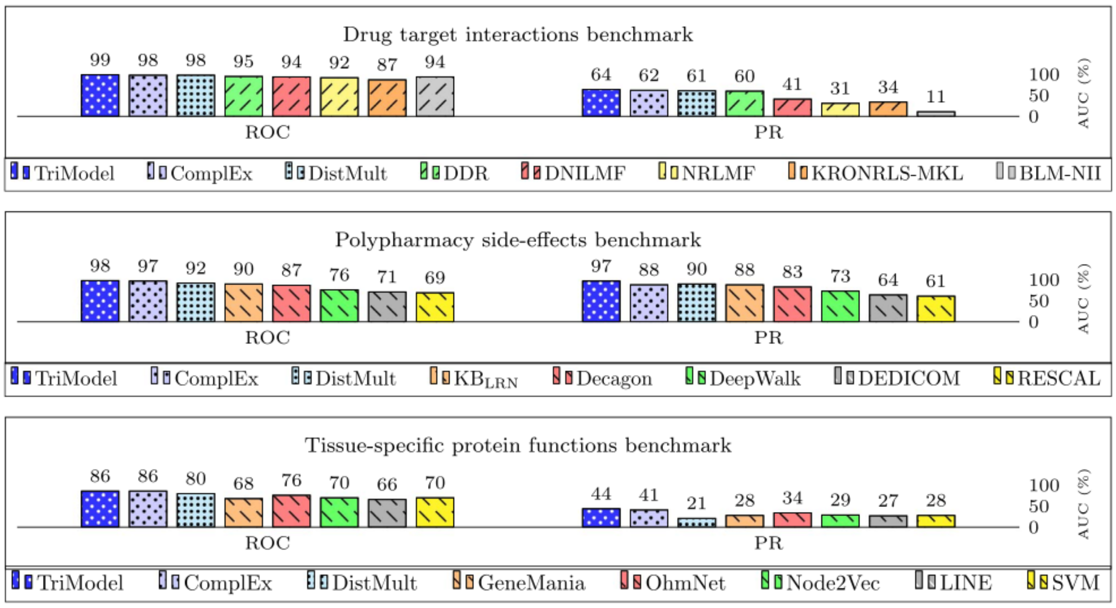

Figure 3: A summary of results of an evaluation of the predictive accuracy of knowledge graph embedding models compared to other models on two biological inference tasks: predicting drug targets and predicting polypharmacy side-effects. The reported results represent the score percentage of the area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

3.8 Key Points about KGEs in biological applications

1. Knowledge graphs allow easy, automated integration of multiple diverse biological data sets in order to model complex biological systems.
2. KGE models enable scalable and efficient predictions on biological knowledge graphs.
3. KGE models provide state-of-the-art predictive accuracy in learning biological associations with high scalability.
4. 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.
5. KGE models can be utilized to model and analyse different types of biological data including genomics, proteomics and pharmacological data.
6. 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.

4 Drug Repurposing [8][9][10][11][5][12]

Currently (March 2025), there are about 7000 identified rare diseases, together affecting 10the population. However, fewer than 6% of all rare diseases have an approved treatment option, highlighting their tremendous unmet needs in drug development. The process of repurposing drugs for new indications, compared with the development of novel orphan drugs, is a time-saving and cost-efficient method resulting in higher success rates, which can therefore drastically reduce the risk of drug development for rare diseases. Although drug repurposing is not novel, new strategies have been developed in recent years to do it in a systematic and rational way.

4.1 Rare Diseases [9][11]

A rare disease (RD) can be any heterogeneous condition affecting a small percentage of the population (Europe: 1 person per 2000; USA: ¡200 000 individuals). RDs are often chronic, resulting in lifelong disability or early death; many RDs have a pediatric onset and about 30% of children with

RD die be- fore the age of 5 years. Seventy percent of all RDs are genetic, caused by both germline and somatic gene mutation. Of the RDs with a genetic origin, many have a monogenic origin; they are caused by a single gene defect and follow a Mendelian inheritance pattern (dominant, re- cessive, X-linked). Additionally, RDs also show non-Mendelian inheritance, which includes epigenetic changes (e.g., Beckwith-Wiedemann syndrome) and mitochondrial disorders (e.g., Rett syndrome) resulting from maternal transmission of variants in mitochondrial DNA.

Around 7000 RDs have been identified to date. While individually rare, they globally affect 300 million people (10% of the population) worldwide. Traditional drug development is time-consuming (10-15 years) and costly ($2.5 billion per drug). Repurposing offers a cheaper, faster alternative!

4.1.1 Some rare diseases from the literature

1. **Hutchinson-Gilford Progeria Syndrome (HGPS)** – A genetic disorder that results in premature aging, with an incidence of about 1 in 8 million live births.
2. **Huntington's Disease** – A neurodegenerative genetic disorder that affects movement, cognition, and psychiatric health, with a prevalence of 3 to 7 per 100,000 people of European ancestry.
3. **Rett Syndrome** – A rare neurological disorder that primarily affects females and results in severe cognitive and motor impairments.
4. **Beckwith-Wiedemann Syndrome** – A genetic condition involving epigenetic changes that lead to overgrowth and an increased risk of childhood tumors.
5. **African Trypanosomiasis (Sleeping Sickness)** – A parasitic disease caused by the *Trypanosoma* species, leading to neurological and systemic complications.
6. **Fabry Disease** – A lysosomal storage disorder caused by a deficiency of the enzyme α-galactosidase A, leading to a buildup of fats in the body.
7. **Muckle-Wells Syndrome** – A rare autoinflammatory disorder caused by excessive interleukin-1 activity, leading to symptoms like fever, joint pain, and hearing loss.
8. **Niemann-Pick Disease Type 1C** – A lysosomal storage disorder characterized by cholesterol accumulation and neurological symptoms.
9. **Spinal Muscular Atrophy (SMA)** – A genetic disorder that causes progressive muscle wasting due to motor neuron degeneration.
10. **Cystic Fibrosis (CF)** – A genetic disorder that affects the respiratory and digestive systems due to thick mucus buildup.

4.1.2 Key Challenges in Rare Disease Treatment

1. **Diagnosis Issues**
   - Many RDs have **overlapping symptoms**, making diagnosis difficult.
   - On average, it takes **6 years** for a patient to receive a correct diagnosis.
   - DNA sequencing technologies like **Whole-Exome Sequencing (WES)** and **Whole-Genome Sequencing (WGS)** have improved detection.
2. **Lack of Market Incentives**
   - Developing drugs for **small patient populations** is **not commercially viable** for pharmaceutical companies.
   - Orphan drugs are often **very expensive**, e.g., *Spinraza* for **Spinal Muscular Atrophy (SMA)** costs **$750,000 for the first year**.

4.1.3 Challenges in Drug Development for RDs

The **identification of disease-causing gene mutations, however, is merely the first step** in confirming a clinical diagnosis. Understanding underlying affected genetic and molecular mechanisms and pathways is essential for disease comprehension and selection of a target for therapy. This can be achieved by engineering and studying model organisms, applying multi-omics sequencing approaches from RNAseq to epigenetic information in tissue types of interest, microarray technology, or in silico techniques. However, despite the significant increase in the speed of RD gene discovery, **the gap in our understanding of the molecular and cellular mechanisms of RDs still remains**.

Currently, most patients **merely receive symptomatic or comfort treatment**, which address second-order complications instead of the underlying disease cause. Current treatment is therefore aimed at the improvement of life quality of those affected, but does not prevent the inevitable decline in function.

One way to tackle the obvious gap in clinical management of RDs is the development of novel orphan drugs. The development of such new drugs, however, is a major challenge, **with often only limited knowledge available regarding disease epidemiology, manifestations, heterogeneity, natural course, and progression.**

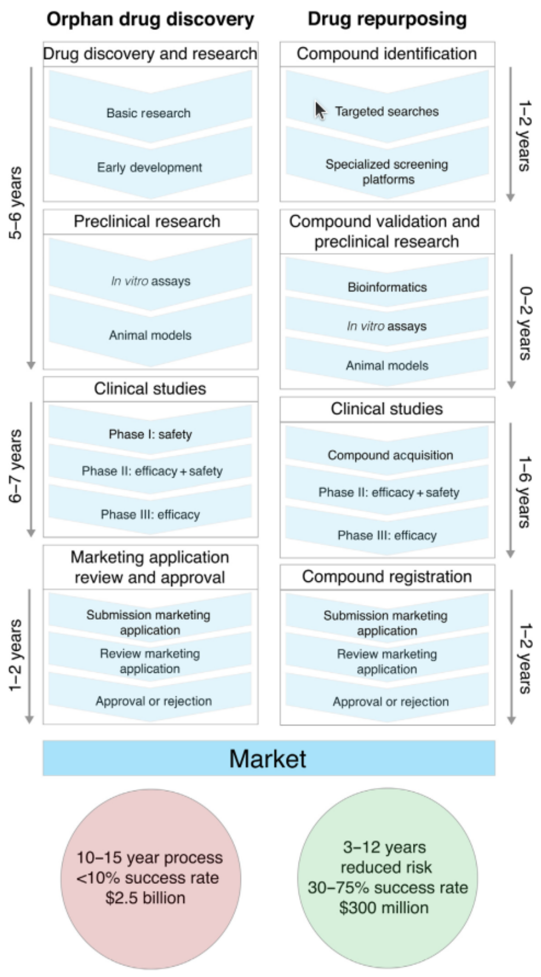

Figure 4: Advantages of Drug Repurposing over the Traditional Way of Orphan Drug Development

4.2 Scientific Basis of Drug Repurposing [10][11][5]

The scientific foundation of drug repurposing is based on **polypharmacology, molecular networks, shared pathways, and bioinformatics-driven approaches**. Below is a deeper dive into the **biological, chemical, and computational principles** that enable the discovery of new therapeutic uses for existing drugs.

4.2.1 Polypharmacology: Multi-Target Nature of Drugs

Polypharmacology refers to the ability of a single drug to interact with multiple biological targets rather than a single receptor or enzyme.

Key Principles

- **Off-Target Effects**: Drugs may have unintended interactions that prove beneficial for other diseases.
- **Target Similarity**: Structurally or functionally similar proteins may share drug interactions.
- **Pathway Crosstalk**: Drugs influencing a pathway in one disease may have effects in another.

Drug	Original Indication	Repurposed Use
Thalidomide	Sedative	Multiple Myeloma, Leprosy
Minoxidil	Hypertension	Hair Loss Treatment
Sildenafil (Viagra)	Angina	Erectile Dysfunction, Pulmonary Hypertension

Table 2: Examples of repurposed drugs

Examples

4.2.2 Shared Molecular Pathways Across Diseases

Many diseases share common genetic mutations, signaling pathways, and metabolic routes, enabling drug repurposing.

Key Concepts

- **Gene-Disease Associations**: Genes involved in one disease may be implicated in another.
- **Biological Pathway Overlaps**: Conditions such as immune response disorders and neurodegenerative diseases often share molecular mechanisms.
- **Protein-Protein Interactions (PPI)**: A drug modulating one protein may affect an entire pathway.

Examples

- Cancer and autoimmune diseases share pathways targeted by kinase inhibitors.
- NSAIDs (e.g., ibuprofen) show potential in reducing Alzheimer's disease inflammation.
- Statins, used for cholesterol reduction, may help in treating muscular dystrophies.

4.2.3 Computational Approaches in Drug Repurposing

Modern bioinformatics techniques provide data-driven predictions for drug repurposing.

Molecular Docking and Structural Biology

- Predicts drug-target interactions based on 3D structures.
- Identifies new binding sites on existing drugs.

Network Pharmacology and Systems Biology

- Maps gene interactions, metabolic pathways, and protein networks.
- Uses **gene expression datasets** to predict drug effects.

Artificial Intelligence and Machine Learning

- Deep learning models analyze biomedical datasets.
- AI-driven predictions suggest repurposing candidates based on mechanism of action.

4.2.4 Omics-Based Drug Repurposing

Genomics and Transcriptomics

- Whole-genome sequencing (WGS) and whole-exome sequencing (WES) help identify rare disease mutations.
- Gene expression analysis can indicate potential new drug applications.

Metabolomics

- Identifies metabolic pathways affected by diseases.
- Example: Metformin (diabetes drug) is being explored for aging-related diseases.

4.2.5 Experimental Approaches for Drug Repurposing

High-Throughput Screening (HTS)

- Tests thousands of FDA-approved drugs on disease models.
- Example: Chloroquine (anti-malarial) shows anti-cancer properties.

iPSC Disease Models

- Patient-derived induced pluripotent stem cells (iPSCs) allow for drug testing on patient-specific models.
- Example: iPSCs were used to test drugs for Niemann-Pick disease.

4.2.6 Side Effect Signatures

The analysis of side effect signatures can help in drug repurposing. This approach is based on the idea that drugs with similar side effects may act on the same target or pathway, leading to a similar therapeutic effect.

4.3 Some Examples of Repurposed Drugs [9][11]

4.3.1 Propranolol for infantile hemangiomas

Propranolol, a beta-blocking medicine **traditionally used for cardiovascular conditions**, **was found to effectively treat hemangiomas in infants**. This discovery stemmed from a clinical observation and led to a public/private partnership between Bordeaux University and Pierre Fabre laboratories. A randomised trial confirmed propranolol's effectiveness, resulting in its approval for infantile hemangioma treatment via a Paediatric Use Marketing Authorisation (PUMA) in Europe Further research is exploring its potential in other rare conditions.

4.3.2 Fenfluramine for Dravet syndrome and Lennox–Gastaut syndrome:

Fenfluramine, **initially an appetite suppressor withdrawn due to cardiovascular risks, was reevaluated for epilepsy treatment**. Clinical trials demonstrated its antiseizure properties in Dravet syndrome. Despite previous safety concerns, the benefits outweighed the risks for this patient group, leading to the approval of fenfluramine solution (Fintepla) as an add-on treatment for seizures associated with Dravet syndrome. In 2023, it was also approved for seizure treatment in Lennox–Gastaut syndrome.

4.3.3 Alpelisib for PIK3CA-related overgrowth spectrum (PROS):

Alpelisib, a PIK3CA inhibitor **initially studied for breast cancer**, **was explored for PROS, a group of genetic disorders with similar genetic mutations**. Collaboration between Canaud's research group and Novartis led to positive outcomes in PROS mouse models and patients. This resulted in FDA approval of alpelisib (Vijoice) for severe manifestations of PROS. This case highlights the benefits of early collaboration between academia and industry in drug repurposing. Although the marketing authorisation application of Alpelisib was withdrawn in the European Union due to the need for more data to support the benefit-risk assessment, Novartis intends to submit a new application when prospective data is available.

4.3.4 Farnesyltransferase for Hutchinson-Gilford progeria syndrome (HGPS)

The rare and fatal **premature aging disease HGPS**, caused by variants in the Lamin A/C gene (LMNA), has seen therapeutic potential through the repurposing of Farnesyltransferase inhibitor (FTI) drugs. These drugs reduce the amount of permanently farnesylated progerin. Based on clinical trial observations, lonafarnib (Sarasar), **originally used for cancer** treatment, is under consideration for approval by the FDA as the first treatment for HGPS.

4.3.5 Canakinumab for Muckle-Wells syndrome (MWS)

For MWS, **an autoinflammatory disorder resulting from increased interleukin-1** (**IL-1**), the drug canakinumab (Ilaris), **initially approved for rheumatoid arthritis, was repurposed**. Canakinumab, a human IgG1 anti-IL-1β monoclonal antibody, selectively blocks IL-1β, effectively neutralising the excess of IL-1β. Clinical studies demonstrated sustained control of disease activity and remission of associated symptoms in MWS patients. The FDA and the European Commission approved it in 2009 for treating MWS patients.

Disease	**Repurposed Drug**	**Mechanism**
Alzheimer's Disease	Rifampin	Reduces amyloid accumulation
Parkinson's Disease	Exenatide	Improves neuronal survival
Ebola Virus	Favipiravir	Inhibits viral replication
Osteoporosis	Raloxifene	Selective estrogen receptor modulator (SERM)

Table 3: Other case studies in drug repurposing

4.4 OA vs. CA in Biomedical Research [12]

- **Open Access (OA):** Scientific literature that is freely available to the public without paywalls. It includes publications in OA journals, preprints, and self-archived versions of research articles.
- **Controlled Access (CA):** Research that is behind paywalls, typically available only through institutional or personal subscriptions to journals and databases.

Key Findings:

- **More than 50% of drug-disease relationships relevant to rare diseases come from OA sources.**
- **45% of relationships exist only in CA sources**, meaning they cannot be accessed by researchers who rely solely on OA literature.
- Elsevier's AI NLP extracted **1.5 times more statements from CA sources** than from OA sources.
- However, **the number of relations extracted from OA and CA sources is roughly equal**, indicating that knowledge graphs built from OA literature can still provide comprehensive information for drug repurposing.

5 Key Knowledge Graphs [1][8][12][3][13]

5.1 BOCK (Biological networks and Oligogenic Combinations as a KG) [3]

- **Construction and Purpose**:
  - BOCK is a knowledge graph constructed to explore disease-causing genetic interactions, integrating curated information on oligogenic diseases from clinical cases with relevant biomedical networks and ontologies.
  - It puts oligogenic combinations into a biological context, specifically focusing on networks relevant to understanding the molecular mechanisms of epistasis.
  - The corresponding source databases were selected based on their quality, accessibility, and interoperability.
- **Nodes and Edges**:
  - BOCK comprises **158,964 nodes of 10 different types**.
  - It contains **2,659,064 edges of 17 different types**.
  - Genes are the central entities.
- **Accessibility**:
  - The complete KG is open-access in semantically rich formats that facilitate its exchange and use.
  - It is available as an RDF graph (Resource Description Framework) along with its data model as an OWL file.
  - BOCK is also provided in the Graph Markup Language (GraphML) and in tab-separated files compatible for direct import into the Neo4J Graph Database.
- **Edge Filtering**:
- Some edge types were filtered before integration, allowing only the connections with a minimal level of confidence provided in the original network resource.
- **Applications**: BOCK allows for the extraction of predictive rules based on path information between known pathogenic gene pairs in the KG.
- **Explanation**: All the rules from the model can be translated into a KG query, retrieving a manageable subset of paths from BOCK to assist in generating hypotheses about the potential molecular mechanisms underlying the disease of interest.
- **Performance Evaluation**:
  - Two decision set models were analysed: one including Phenotype-traversing paths (DS incl.Pheno) and one excluding them (DS excl.Pheno).
  - The phenotype-inclusive model (DS incl. Phenotype) can capture indirect Phenotype relationships due to the metapath-based design of its rules.
- **Limitations**:
  - When excluding phenotype information, the gene pair pathogenicity predictor exhibits a relatively high false positive rate.
  - The large number of rules potentially generated requires stringent constraints on the search space, which limit exploration of patterns.

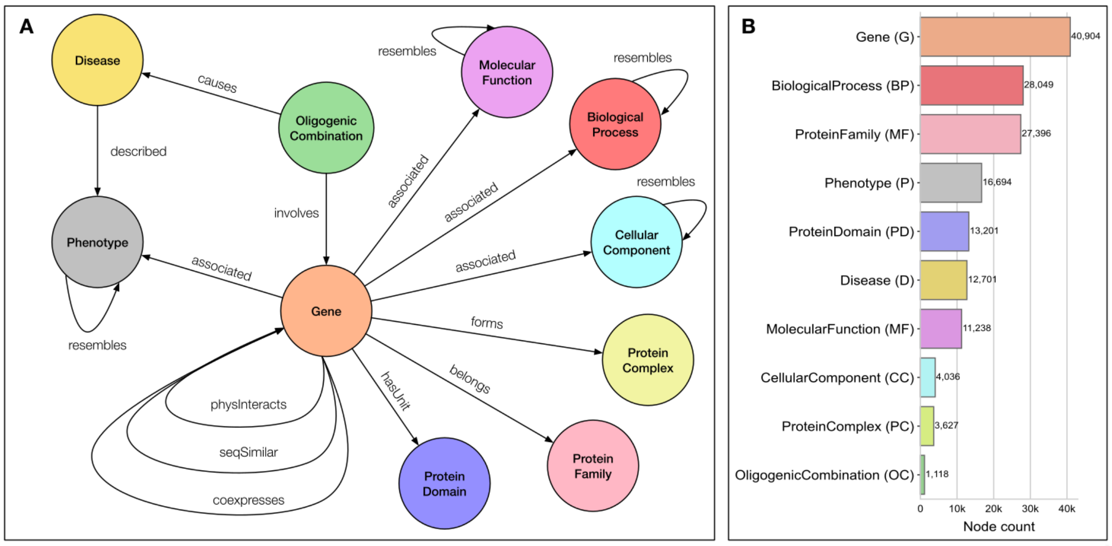

Figure 5: Schema and node statistics of the KG (BOCK). A KG schema representing the different node types (i.e. metanodes) as circles and their relationships as arrows (bidirectional arrows indicate undirected associations). B Number of nodes in the KG per metanode.

5.2 Hetionet [8][3][4][14]

Hetionet is accessible via a Neo4j server, offering a user-friendly interface to explore its interconnected information. Meta-path features derived from Hetionet have been applied in gene-disease prioritization and drug repurposing. Among the 2,250,197 triplets that make up the knowledge graph, only 755 correspond to "compound treats disease".

- **Construction and Content**:
  - Hetionet v1.0 was created by integrating data from **29 publicly available resources** into a single data structure.
  - It accommodates the breadth and depth of information, implementing network pharmacology that natively incorporates polypharmacology and high-throughput phenotypic screening.
  - Hetionet includes nodes from **five ontologies**, selected for their conformity to current best practices.
  - The goal was to integrate a broad diversity of information types of medical relevance, ranging in scale from molecular to organismal.
- **Accessibility**:
  - Hetionet v1.0 is accessible via a Neo4j Browser at https://neo4j.het.io.
  - It is publicly available.
- **Nodes and Edges**:
  - Hetionet comprises **11 metanodes** (types of nodes).
  - It contains **24 edge types (metaedges)**.
  - The 11 metanodes have 66 possible source–target pairs, all connected at a path length of 3.
  - **Table of Metaedges**: Hetionet v1.0 contains 24 edge types (metaedges).
    * Examples include Anatomy–downregulates–Gene (AdG), Anatomy–expresses–Gene (AeG), and Anatomy–upregulates–Gene (AuG).
    * All metaedges besides `GenefiregulatesfiGene` are undirected.
  - **Nodes**: Nodes encode entities extracted from standard terminologies.
- **Queries and Versatility**:

  - Hetionet differentiates itself in its ability to flexibly query across multiple domains of information.
  - It allows identifying drugs that target a specific pathway, biological processes involved in a specific disease, drug targets causing a specific side effect, and anatomies with transcriptional relevance for a specific disease.
  - These queries are simple to write and fast to run using the Hetionet Browser.

- **Applications**:
  - Ideally suited for drug repurposing, the network has broader biological applicability.
  - It can identify drugs that target a specific pathway.

- **Project Rephetio**:
  - Project Rephetio uses the hetnet data structure to unite diverse information to predict the probability that a compound treats a disease.
  - The predictions successfully prioritised novel indications.

- **Limitations**: Hetionet contains a smaller number of disease nodes compared to MSI and KEGG because the disease nodes defined are at a higher or broader level.

5.3 GNBR (Global Network of Biomedical Relationships) [12][13]

- GNBR is a large, heterogeneous knowledge graph comprising **drug, disease, and gene (or protein) entities linked by a small set of semantic themes derived from the abstracts of biomedical literature**.

- It consists of over **130,000 entities and over two million edges**, with each edge represented by a set of several important, semantic themes, and the confidence associated with each of these themes is quantified as a continuous value.

- The Dr–disease (Dz) themes, particularly “Treatment” (T) and “Inhibits cell growth (esp. cancer)” (C), are relevant for drug repurposing.

- A GNBR-like knowledge graph derived from full texts as well as abstracts could provide even more power to discover novel relationships.

5.4 Bioteque [5]

- Bioteque integrates and formats biomedical data as pre-calculated knowledge graph embeddings.

- It contains biological entities, descriptors, data sources, and algorithmic interpretations.

5.5 Clinical Knowledge Graph (CKG) [5]

- CKG is one of the most expansive KGs within the drug repurposing domain, including data from **26 biomedical databases and 9 ontologies**, culminating in a rich repository of knowledge.

- It includes a staggering **16 million nodes and over 220 million relationships**. Its nodes are classified into 19 distinct types, including publications, drugs, diseases, proteins, and clinical variables, while its edges are categorized into 57 types, facilitating intricate relationship representation.

- CKG offers various algorithms and machine learning techniques that can be used for data analysis.

5.6 PharMeBINet [5]

- PharMeBINet represents a significant evolution from HetioNet, integrating data from 19 additional public resources in addition to Hetionet's original 29 sources.
- This expansion results in a noteworthy augmentation of nodes, increasing from 47,031 to 2,869,407—an impressive 61-fold increase.
- The relationships within PharMeBINet experience a notable growth, escalating from 2,250,197 to 15,883,653—a sevenfold amplification.
- Moreover, PharMeBINet's node diversity expands exponentially from 11 to 66 types, while the variety of edge types increases from 24 to 208.

5.7 Multiscale Interactome (MSI) [1]

- The MSI knowledge graph contains 29,959 nodes and 478,728 edges.
- It has 4 node types and 5 edge types.

5.8 DRKG [1]

DRKG was constructed by integrating seven publicly available databases (namely, DrugBank, ChEMBL, CTD, DisGeNET, SIDER, REACTOME, and KEGG).

5.9 BioKG [1]

- BioKG was built using 18 different data sources, including DrugBank, UniProt, and KEGG.
- It contains a total of 7 node types (drugs, proteins, indications, diseases, gene ontology, expression, and pathways) and 10 edge types.
- One of the main advantages of this KG is that it contains a mapping module that allows it to be interoperable with other KGs.

5.10 Elsevier Biology Knowledge Graph (EBKG) [12]

- Elsevier builds its biology knowledge graph from both controlled and open access publications using its proprietary AI NLP technology.
- The node types, edge types and data sources in EBKG facilitate comparison of contributions to biomedical knowledge.

6 Explainable Artificial Intelligence (XAI) in Drug Repurposing and Link Prediction [5][14][15][16]

Explainable Artificial Intelligence (XAI) is a rapidly advancing field aimed at enhancing the **trustworthiness and transparency** of AI model predictions, particularly in domains like healthcare where decisions have significant real-world impacts. By providing clear explanations for AI-driven predictions, XAI not only fosters **trust** but also facilitates **knowledge discovery**, enabling researchers to uncover complex patterns, generate testable hypotheses, and address biases or errors in AI models. Regulatory frameworks like the **EU's General Data Protection Regulation (GDPR)** further emphasize the necessity of a **"right to explanation"** when AI is involved in critical decisions.

6.1 Some XAI Techniques for Drug Repurposing

- **Attribution Maps:** Assign importance scores to graph features, including nodes and edges, highlighting significant subgraphs or node sets for improved interpretability.
- **Counterfactual Explanations:** Offer alternative scenarios to assess how changes in graph structure could alter predictions, enhancing model transparency.
- **GNNExplainer:**

  - One of the pioneering XAI methods applied to graph-based models.
  - Utilizes **Mutual Information (MI)** to maximize the shared information between predictions from the complete graph and a subgraph.
  - Provides explanations in the form of a **subgraph** by training edge and node masks.
  - While effective, retraining the mask for each explanation may lead to inconsistent results.

- **SubgraphX:**
  - Uses **Monte Carlo Tree Search (MCTS)** and **Shapley values** to generate human-readable explanations.
  - Iteratively removes nodes to compute the importance of each element statistically.

- **PaGE-Link:**
  - Generates explanations as paths using a **k-core pruning module** and a **path-enforcing mask**.
  - Balances two loss terms: **Lpred** (to maximize MI) and **Lpath** (to select path-forming edges).
  - Produces a path as an explanation, offering a structured and human-understandable output.

- **(Graph)LIME:**
  - Performs small perturbations to node features to observe prediction variability, providing insights into feature importance.

- **DrugChat:**
  - Leverages a **Large Language Model (LLM)** in a chatbot format to provide human-like explanations for drug compounds.
  - Takes a molecule represented as a **SMILE graph**, processes it through a **pre-trained GNN**, and uses an adapter to transform the embedding for the LLM.
  - Though not initially designed for **link prediction** or **drug repurposing**, its approach could be adapted for these tasks, offering **ChatGPT-like explanations**.
  - Shares common LLM limitations, including the risk of **hallucinations** (i.e., generating plausible but incorrect answers).

- **rd-explainer:**
  - Specifically developed for **drug repurposing for rare diseases**, combining a **graph learning model** with an explainer module to tackle the **"black-box" problem**.
  - Provides explanations as **semantic graphs**, which mirror **human reasoning** and offer valuable insights for further research.
  - Comprises three modules: the **Knowledge Graph Construction**, **Prediction**, and **Explainer** modules.
  - The **Explainer module** generates **semantic subgraphs** to elucidate the connection between predicted drugs and symptoms.

- **XG4Repo:**
  - A framework that uses **knowledge graphs** to predict diseases treatable by a given compound.
  - Offers rule-based explanations, highlighting the **importance of supporting evidence** for predictions.

6.2 Explanation as Semantic Graphs

- **Semantic graphs**, used by methods like **rd-explainer**, provide explanations that align with human logic.
- These graphs can guide further, more precise investigations, even when complete supporting evidence is lacking.

6.3 Challenges and Considerations in XAI for Drug Repurposing

- **Benchmarking and Metrics:** The absence of standardized benchmarks and metrics for evaluating explainers can hinder consistent assessment of explanations.
- **Reproducibility:** Some XAI techniques exhibit reproducibility issues, affecting the reliability of explanations.
- **Data Topology Impact:** The representation of graphical knowledge plays a critical role in model performance and explanation accuracy.
- **Method Selection:** It is essential to choose the appropriate **Knowledge Graph**, **AI method**, and **XAI technique** to align with specific research goals and datasets.

7 Specific Techniques and Algorithms [5][3][4][14][16][17][2]

7.1 Machine Learning

7.1.1 DT2Vec+ [17]

DT2Vec+ is a machine learning pipeline designed for drug repurposing. It predicts **drug-target interactions (DTIs)** and determines the specific type of interaction involved. It integrates the triplet associations of drug–target–disease data to a heterogeneous graph by incorporating drug–drug and protein–protein similarity net- works, together with verified drug-disease and protein-disease associations. This is achieved through a multi-stage process that integrates heterogeneous data, applies network embedding techniques, and utilises gradient boosted trees for classification.

- **Data Integration and Heterogeneous Graph Construction**
  - DT2Vec+ begins by integrating data from diverse sources to construct a **heterogeneous graph**. This graph is a crucial component as it represents the complex relationships between drugs, targets (proteins), and diseases.
  - The graph consists of three types of nodes:
    * **Drugs**: Representing chemical compounds with potential therapeutic effects.
    * **Proteins (Targets)**: Representing the biological molecules with which drugs interact.
    * **Diseases**: Representing the conditions that the drugs are intended to treat.
  - These nodes are connected through four types of edges:
    * **Drug-Drug Similarity (DDS)**: Edges connecting drugs that share similar characteristics.
    * **Protein-Protein Similarity (PPS)**: Edges connecting proteins that have similar sequences or functions.
    * **Drug-Disease Associations (DDis)**: Edges connecting drugs known to be associated with specific diseases.
    * **Disease-Protein Associations (DisP)**: Edges connecting diseases to the proteins that are implicated in their pathology.
  - By integrating these diverse data types into a unified graph, DT2Vec+ captures a rich set of relationships that are essential for predicting DTIs.
- **Node Embedding with Node2Vec**
  - To extract meaningful features from the heterogeneous graph, DT2Vec+ employs **node2vec**, a network embedding technique. Node2Vec maps the nodes (drugs, proteins, and diseases) into a **low-dimensional vector space**, capturing the structural properties of the graph.
  - **Node2vec** is used to map nodes to vectors. It is a scalable feature learning method for networks.
  - The algorithm performs **random walks** starting from each node in the graph. These random walks generate sequences of nodes, which are then used to learn vector representations (embeddings) for each node.

  - Node2vec uses parameters to regulate the preference of performing a breadth-first search (BFS) or a depth-first search (DFS) walk. BFS tries to stay as close as possible to the previous node, while DFS tries to move away from the previous node.
  - The resulting embeddings encode the node's position and context within the network, allowing machine learning models to leverage network information.

- **Multi-Label, Multi-Class Classification**
  - DT2Vec+ formulates the prediction of drug-target interactions (DTIs) as a **multi-label, multi-class classification** problem.
  - This means the model aims to predict not only whether a drug interacts with a target but also the **type and degree of the interaction**.
  - The interaction types include:
    * Increases expression
    * Decreases expression
    * Decreases reaction
    * Increases reaction
    * Increases activity
    * Decreases activity
  - This detailed classification is crucial for understanding drug action and guiding targeted treatment strategies.

- **Model Training with XGBoost**
  - The prediction of DTIs is achieved using **gradient boosted trees (XGBoost)**.
  - **Drug-target pairs** are represented by concatenating the **drug vector and the target vector** extracted from the graph embedding. These concatenated vectors serve as input features for the XGBoost model.
  - The **XGBoost model** is trained to predict the interaction type. A one-vs-rest strategy is employed to train separate models for each interaction type, and cross-validation is used to evaluate the performance.

- **Novel DTI Extraction and Drug Repurposing**
  - DT2Vec+ uses a two-step prediction process to identify novel DTIs.
  - First, experimentally validated negative interactions are used alongside DT2Vec to predict highly positive interactions.
  - Then, DT2Vec+ is applied to these positive interactions to identify the six types of DTIs.

- **Performance Evaluation**
  - The performance of DT2Vec+ is evaluated using several metrics.
  - These metrics include:
    * Accuracy
    * F1-score
    * Precision
  - In experiments, DT2Vec+ has demonstrated high performance, achieving an average accuracy of 77.09%, F1-score of 74.39%, and precision of 84.58% on external test sets.

- **Implementation**
  - Categorical labels for degree and type of interaction were converted to binary vectors through one-hot encoding, and one-vs-rest strategies were used to train the model against each label.
  - Cross-validation is used to evaluate performance.
  - The average performance of all models was measured on external test-sets.

DT2Vec+ offers a comprehensive and effective approach to drug repurposing by integrating heterogeneous data, leveraging network embedding techniques, and employing a powerful machine-learning model for prediction. By predicting potential drugs and their specific action modes, DT2Vec+ refines the search space for experimental validation and reduces wet-laboratory work and experimental costs. The code for DT2Vec+ is available on GitHub, facilitating its use and adaptation by other researchers.

7.1.2 ARBOCK [3]

- Association Rule learning Based on Overlapping Connections in Knowledge Graphs!
- **Knowledge Graph Foundation**: ARBOCK hinges on the **Biomedical Oligogenic Connectivity Knowledge Graph (BOCK)**. This KG serves as the foundation for identifying connections between genes. The quality and structure of BOCK directly influence ARBOCK's ability to discover meaningful associations.
  - BOCK is not just a database but a structured representation of biomedical knowledge, where nodes represent biological entities (genes, proteins, pathways, etc.) and edges represent relationships between them.
  - Unlike methods relying solely on direct phenotypic associations, ARBOCK, through its metapath-based rules, captures indirect phenotype relationships, covering a wider pool of genes.
- **Path Traversal and Feature Extraction**: ARBOCK's capacity to traverse all possible paths within BOCK between identified gene pairs is central to its methodology. It uses a path length cut-off (a maximum number of edges in a path), set to 3 in the study, to ensure connectivity among known pathogenic gene pairs.
  - Each path begins with the gene of lowest Residual Variant Intolerance Score (RVIS) and ends at the gene with the highest score.
  - The approach retains edge directionality and calculates a path reliability score based on the geometric mean of edge scores.
- **Association Rule Mining with Apriori**: The Apriori algorithm is a crucial step, identifying sets of metapaths that frequently co-occur in positive examples (disease-causing gene pairs).
  - This step uncovers combinations of relationships that are statistically enriched in pathogenic gene interactions.
  - The mined rules undergo a refinement process that optimises path reliability thresholds to improve interpretability and filter out potentially spurious paths. This involves a differential evolution algorithm to determine optimal threshold values for a rule.
  - The mining of simple metapath associations is extended by searching for unification constraints between metapaths, looking for nodes at the intersection of paths associated with at least two different metapaths. Only closed itemsets are selected to minimise redundancy of mined patterns.
- **Decision Set (DS) Classifier**: This is the final predictive model. The decision set algorithm selects a subset of the generated rules that collectively provide good coverage of the positive examples (known pathogenic gene pairs) while minimising the inclusion of negative examples (non-pathogenic pairs). The model translates rules into KG queries, retrieving a manageable subset of paths from BOCK. The subgraph contains concrete relationships and entities to assist in forming hypotheses about the molecular mechanisms underlying the disease.
- **Explainability**: A core strength of ARBOCK is its focus on explainability. By translating the rules into KG queries, the model retrieves a specific subgraph from BOCK that connects the gene pair in question. This subgraph serves as an explanation for the model's prediction, highlighting the key relationships and biological entities involved. All the rules from the model can be translated into a KG query, which retrieves a manageable subset of paths from BOCK.
- **Performance**: Studies have shown that ARBOCK outperforms models based on graph topology alone, such as random walk with restart (RWR), by harnessing the semantics of heterogeneous paths.

In essence, ARBOCK functions by:

1. Exploring the complex relationships between genes within a knowledge graph.
2. Identifying statistically significant patterns of these relationships that are indicative of disease-causing interactions.
3. Combining these patterns into a predictive model.
4. Providing a clear, interpretable explanation for each prediction in the form of a subgraph extracted from the knowledge graph.

Metapaths are a key component of the **ARBOCK** framework presented in the paper, serving as a way to capture and analyse relationships between genes in the knowledge graph. They are used to mine association rules and provide interpretable explanations for predictions of pathogenic gene interactions.

- **Definition**: Metapaths are defined as **sequences of node and edge types** within a knowledge graph. They record the semantic pattern of a relationship between entities. In simpler terms, a metapath describes a specific type of connection between two nodes in the graph, specifying the types of nodes and relationships involved in that connection.
- **Usage in ARBOCK**:
  - **Mining Association Rules**: The ARBOCK framework leverages metapaths to mine association rules from frequently observed sets of metapaths in pathogenic gene pairs. This involves identifying patterns of metapaths that commonly occur together in gene pairs known to cause disease.
  - **Predicting Genetic Interactions**: The mined association rules are then used to identify novel pathogenic genetic interactions. By analysing the metapaths associated with gene pairs, the model can predict whether a particular gene interaction is likely to be disease-causing.
  - **Providing Interpretability**: ARBOCK provides a fully interpretable model by using metapaths. The model offers graphical explanations for each prediction, showing the specific entities and relationships (i.e., the metapaths) that led to the prediction.
- **Example**:
  - The paper mentions **GaPaG** and **GaPrPaG** (reflecting common and related phenotypes between gene pairs) as examples of metapaths that hold the most influence among high-confidence rules when considering all valid paths.
  - **GpGaBPaG** and its reverse, which capture shared processes between a gene pair and an interacting gene, stand out in the highest confidence rules when excluding paths traversing Phenotype nodes.
- **Technical Details**:
  - Although the traversal of paths in BOCK disregards edge directionality, the original direction of the edges is encoded in the recorded paths.
  - For each path, a path reliability score is attributed based on the geometric mean of its edge scores.
  - Paths are then grouped into corresponding path types, or metapaths.
- **Benefits**:
  - **Transparency**: Path-based approaches using metapaths offer a transparent way of inferring new relationships.
  - **Interpretability**: They provide meaningful explanations for predictions by highlighting specific connections and relationships within the knowledge graph.
  - **Feature Generation**: Metapaths can be employed to generate features from KGs for various classification tasks.

In short, metapaths are a way of **abstracting and generalising relationships** in a knowledge graph, allowing the ARBOCK framework to identify meaningful patterns and make interpretable predictions about disease-causing gene interactions.

7.1.3 Gradient Boosted Trees (XGBoost) [17]

Gradient Boosted Trees (GBT) are a machine learning technique used for both classification and regression tasks. They are **ensemble models** that combine the predictions of multiple decision trees to create a more accurate and robust prediction.

How they work:

- **Sequential Tree Building**: GBTs build trees sequentially, with each tree attempting to correct the errors of the previous trees.
- **Loss Function**: A loss function is used to measure the difference between the predicted values and the actual values. The goal of each tree is to minimise this loss.
- **Gradient Descent**: GBTs use gradient descent to find the best way to reduce the loss. The gradient indicates the direction of the steepest increase in the loss function, so the algorithm moves in the opposite direction to minimise it.
- **Weak Learners**: Each tree is a "weak learner," meaning it only makes a small improvement to the overall prediction. However, by combining many weak learners, the model can achieve high accuracy.
- **Boosting**: The process of sequentially adding trees and weighting their predictions is called boosting. Each tree is given a weight that reflects its contribution to the final prediction.

Key aspects and advantages of GBTs:

- **High Accuracy**: GBTs are known for their high predictive accuracy, often outperforming other machine learning algorithms.
- **Feature Importance**: GBTs can provide estimates of feature importance, indicating which features are most relevant to the prediction.
- **Handling Imbalanced Data**: GBTs can effectively handle imbalanced datasets, where one class is much more frequent than the other.
- **Regularisation**: Regularisation techniques are often used to prevent overfitting, where the model learns the training data too well and performs poorly on new data.
- **Flexibility**: GBTs can be used for a variety of prediction tasks, including predicting drug-target interactions and their types.

One implementation of gradient boosted trees is **XGBoost (Extreme Gradient Boosting)**. It is known for its speed, accuracy, and ability to handle imbalanced datasets. XGBoost is an ensemble model that combines multiple weak learners, in this case decision trees, through a boosting algorithm to create a stronger learner. A boosting algorithm works by sequentially training weak learners based on the residual error of the previous learner. To prevent over-fitting during the training phase, number of boosting rounds can be limited, and maximum depth of the trees can be constrained.

In the context of drug repurposing, GBTs, such as XGBoost, can be used to:

- Predict novel drug-target interactions (DTIs), as well as to predict the type of interaction induced.
- Integrate various data sources, such as drug-drug and protein-protein similarity networks, drug-disease and protein-disease associations.
- Rank DTIs based on the confidence of the model.

7.2 Random Walk Algorithms

Drug repurposing can be achieved through the use of **knowledge graphs (KGs)**, which are data structures representing knowledge of a specific domain. These graphs consist of **nodes representing entities** like drugs, diseases, and proteins, and **edges representing the relationships** between these entities, such as "treats", "interacts with", or "causes". KGs are directed heterogeneous graphs and can use semantics or ontologies to define their nodes and edges. Many methods are applied to analyse KG and make link predictions, including Deep Learning models (autoencoders and Graph Neural Networks, GNNs), random walks, translational embeddings, matrix factorisation and metapath-based methods.

Random walk algorithms are a way of analysing a graph by creating several paths by **randomly traversing the graph** and then using these paths to **create different embeddings for each node**. A random walk algorithm moves from node to node along the edges of the graph in a random manner, creating a sequence of nodes (a path). The generated paths are used to create embeddings, which are vector representations of nodes that capture information about their relationships and context within the graph. These algorithms are useful for exploring relationships within knowledge graphs. They can be biased, such as with DREAMwalk, which has a teleportation function that utilises semantic similarity to move to other nodes. Other algorithms for this task include Node2Vec, which uses parameters to regulate the preference for breadth-first search (BFS) or depth-first search (DFS) walks, and WalkPool which combines GNNs and random walks.

7.2.1 DREAMwalk [5][3][4]

DREAMwalk is an AI method that uses random walks to generate embeddings for drugs and diseases, which are then used as input for an XGBoost classifier. DREAMwalk incorporates **semantic information-guided teleportation** to populate drugs and disease entities on a random walk-based path generation process. It operates on the principle of **semantic multi-layer guilt-by-association (GBA)**, assuming that semantically similar drugs share biologically relevant targets with diseases.

How DREAMwalk works:

- **Teleport-guided random walk**: DREAMwalk performs a teleport operation during random walking, using semantic similarity as a guide. The widely used Anatomical Therapeutic Chemical (ATC) classification and medical subject headings (MeSH) describe the semantic hierarchy of drugs and diseases, respectively. When the random walker arrives at a drug or disease node, it chooses between network traversing and teleport operation. If teleport is selected, the walker randomly samples the next node from a similarity matrix using similarity values as the sampling distribution. The probability of choosing teleportation is determined by a user-defined teleport factor.
- **Heterogeneous Skip-gram model**: The semantic information-integrated random walk sequences are then passed to a heterogeneous Skip-gram model-based node representation learning, which generates an embedding space for computing relationships between entities.
- **XGBoost classifier**: Using the generated node representations, an XGBoost classifier is trained to output the drug-disease treatment probability, given the subtracted vector of drug and disease nodes. The trained XGBoost model is then utilised for drug repurposing by prioritising highly probable treatment drug-disease relationships.
- **Semantic Similarity**: It utilises semantic similarity measure as teleport probability between drug-drug or disease-disease nodes.

Why DREAMwalk is useful:

- **Improved drug-disease association prediction**: DREAMwalk incorporates semantic information to generate random walk paths that provide both biological and semantic perspectives, leading to accurate representation learning that reflects the molecular and semantic contexts of entities. It generates a more effective embedding space that allows for mapping drugs and diseases in the same space.
- **Interpretability**: DREAMwalk's node sequences can be analysed to identify neighbouring genes for a given entity, offering a way to infer the mechanism of action (MoA) of a biomedical entity based on these neighbouring genes.
- **Addresses PPI network bias**: DREAMwalk uses biased random walks that allows for teleportation to semantically similar nodes, which helps to address the issue that regular random walk approaches are not very effective in solving the drug repurposing problem because the protein-protein interaction (PPI) network is much larger than the drug–disease, drug–gene, and gene–disease networks.

What DREAMwalk has discovered:

- **Breast Carcinoma and Alzheimer's Disease**: DREAMwalk was tested on two different case studies, Alzheimer Disease and breast cancer, showing promising results. DREAMwalk can identify a related yet diverse range of repurposable drugs when compared to baseline models.
- **Parkinson's Disease**: The path-based interpretation of gabapentin and PD show the semantic information integrated neighbourhoods' potential ability to explain biological mechanisms of drugs and diseases, which are difficult to identify solely via molecular-level neighborhoods.
- DREAMwalk's semantic information-guided path enables interpretation of drug/disease mechanisms.
- That integrating semantic information, such as semantic hierarchies of drugs (ATC classification) and diseases (MeSH term, DiseaseOntology, ICD-11) produces a more accurate prediction.
- It can predict the DDAs with the highest median probability and rank compared to the seven baseline models.

7.2.2 Node2Vec [5][17]

Node2Vec is a method for **scalable feature learning for networks**. It is a **random walk-based** approach that learns low-dimensional representations for nodes in a graph, designed to preserve the network's structural properties and node neighbourhoods.

How Node2vec works:

- **Biased Random Walks**: Node2vec uses **biased random walks** to explore the graph. It employs two parameters, **p (return parameter) and q (in-out parameter)**, to control the exploration strategy.
  - The **return parameter (p)** influences the likelihood of revisiting a previously visited node. A low p-value encourages the walk to return to the previous node, approximating BFS.
  - The **in-out parameter (q)** governs whether the walk explores nodes further away from the starting node (DFS-like behaviour) or stays closer to the initial node (BFS-like behaviour). A low q-value encourages DFS walks.
- **Skip-Gram Model**: Once the random walks are generated, a **skip-gram model** is trained to learn embeddings for each node. The skip-gram model is a neural network with a single hidden layer. It takes a one-hot encoded feature vector of a target node as input and is trained to predict the probability of finding each node in the neighbourhood of the target node. The hidden layer of the trained model is then used as the feature vector for each node.

Why Node2vec is useful:

- **Captures Network Structure**: Node2vec is designed to preserve the network's structural properties and node neighbourhoods.
- **Scalability**: The method is scalable, making it suitable for large networks.
- **Feature Learning**: Node2vec automatically learns features that can be used for various downstream tasks.
- **Node embeddings**: The algorithm outputs node embeddings, or vector representations of nodes that capture information about their relationships and context within the graph.

It has been shown that Node2vec is a superior method for node embedding compared to other existing state-of-the-art methods. It is applied in the DT2Vec+ pipeline to map nodes in the drug–target–disease association graph into a 100-dimensional vector.

7.3 Graph Neural Networks (GNNs)

7.3.1 GNNExplainer [5][15][16]

- GNNExplainer is an XAI method used to provide explanations for predictions made by GNNs.
- It provides explanations in the form of a subgraph that can be easily understood.
- GNNExplainer leverages **Mutual Information (MI)** from information theory to identify the most important features for a given prediction. MI quantifies the shared information between two random variables, measuring their dependence:
  - $MI(Y, (Gs, Xs)) = H(Y) - H(Y|(G = Gs, X = Xs))$
  - Where *H(Y)* is the entropy of the original predictions, and *H(Y—(G = Gs, X = Xs))* is the entropy of the predictions using the subgraph.
- **The goal of GNNExplainer is to maximise the MI between predictions from the complete graph and a subgraph**. Since *H(Y)* is fixed, this is equivalent to minimising $H(Y|(G = Gs, X = Xs))$.
- The optimisation is achieved by training a mask on the adjacency matrix, with values between 0 and 1. This mask, when applied, weakens certain edges, effectively excluding them from predictions. The result is a subgraph that serves as the explanation.
- A key issue is that retraining the mask for each explanation can lead to different results for the same instance upon repeated runs.

7.3.2 GraphSAGE [5][15][16]

- GraphSAGE (Graph SAmple and aggreGatE) is a framework for **inductive graph representation learning**.
- GraphSAGE performs **inductive graph representation learning by leveraging rich node attribute information**.
- It extends Graph Convolutional Networks (GCNs) by allowing neighbour sampling. Instead of applying the message passing process through every node, it only makes use of a subgroup of neighbours.
- It incorporates the possibility to work in mini-batches. By applying mini-batches, the space and time complexity of the mode can be fixed to: *O(Ki=1 Si)*, where *K* is the number of layers of the GNN and *Si* corresponds to the number of neighbours sampled on that layer.
- Each mini-batch is a subset of computational graphs (a computational graph is the individual GNN that is built for each node) of N nodes.
- By applying this technique, the GNN can better manage larger graphs.

7.3.3 Graph Attention Networks (GAT) [2][16]

- The idea of GAT is that, **instead of applying a normalisation that only relies on the degree of the nodes like GCNs, it makes use of an attention mechanism** that assigns different weights to the neighbouring nodes based on their relevance to the target node.
- This attention mechanism allows GATs to dynamically adapt the importance of each neighbour during the aggregation process, enabling more flexible and context-aware information propagation within the graph.

Key concepts common to GNNs:

- GNNs are an adaptation of regular neural networks but applied to graph structures.
- The idea is to create a neural network structure for each node, where in each layer, the node gathers information from its neighbours. A 1-layer GNN will gather information from its direct neighbours, while a 2-layer GNN will look at the 2-hop neighbourhood.

- In each layer, there are two processes taking place, a message passing process and an aggregation process.
  - During the message passing process, information from each neighboring node is used as input in a regular NN.
  - Then, during the aggregation process, the resulting vectors from the message passing operation are combined.
  - There are multiple aggregation operations that can be used, including summation, pooling and average aggregators.

7.4 Graph Autoencoders [5][4][16][2]

Graph Autoencoders (GAEs) are a type of neural network architecture used for knowledge graph representation learning, which can be applied to drug repurposing. GAEs are unsupervised learning methods that aim to reconstruct the input graph from a lower-dimensional latent representation, capturing essential structural information.

How can they be used for drug repurposing:

Architecture

- GAEs consist of two main components: an encoder and a decoder.
  - The encoder maps the input graph into a latent space, generating node embeddings that capture the graph's structural features.
  - The decoder then reconstructs the original graph from these embeddings.

Learning Process

- The learning process involves optimising the GAE to minimise the reconstruction loss, ensuring that the reconstructed graph closely resembles the original graph.
  - The reconstruction loss measures how well the new graph resembles the original graph.
  - A regularization loss is often included to ensure that the latent space distribution of encoded data approximates a Gaussian Distribution.

7.4.1 Variational Graph Autoencoders

- A common approach involves using Variational Graph Autoencoders (VGAEs), which make use of Graph Convolutional Networks (GCNs) to create a latent feature matrix used to reconstruct the adjacency matrix. The learning approach in VGAEs is similar to that in regular autoencoders.

7.4.2 Node Embeddings from Static Subgraphs

- Node Embeddings from Static Subgraphs (NESS) is another approach that uses graph autoencoders.
  - NESS divides the graph into smaller, non-overlapping subgraphs that pass through the same encoder and decoder, which can increase the model's performance relative to regular GAEs.
  - One limitation of NESS is that it only works in a transductive setting, which can be relevant in a drug repurposing setting where new knowledge is constantly being introduced.

Application to Drug Repurposing

- GAEs can be used to predict potential connections between drug nodes and disease nodes in a knowledge graph, framing drug repurposing as a link prediction task.
  - Graph embeddings transform the graph's nodes, edges, and features into a low-dimensional space while preserving its structure.

  – This facilitates the application of machine learning algorithms by simplifying the graph's complex structure, making it easier to analyse and derive predictions from.

By learning these embeddings, GAEs can identify potential new uses for existing drugs by finding drugs with similar embedding vectors to diseases of interest.

7.5 Large Langauge Models (LLMs)

7.5.1 LMKE - Language Model for Knowledge Graph Embedding [5]

LMKE approaches knowledge graph completion by treating it as a **language modelling task**. Instead of directly learning embeddings for entities and relations, it leverages the power of pre-trained **Masked Language Models (MLMs)**. Specifically, LMKE takes a triple (head entity, relation, tail entity) and their descriptions as input. The MLM is then tasked with predicting the missing tail entity, given the head entity, the relation, and their respective textual descriptions. The description is taken from available text and integrated with the knowledge of the LLM. By framing KG completion as a cloze task (which is a fill-in-the-blank task), LMKE can effectively utilise the pre-existing knowledge and reasoning capabilities embedded within LLMs. LMKE has demonstrated strong performance, particularly on datasets like WN18RR, showcasing its ability to capture complex relationships between entities.

7.5.2 MoCoSA [5]

MoCoSA adopts a **hybrid approach**, combining the strengths of both **structural and descriptive information** present in knowledge graphs. It employs a dual-encoder architecture, featuring a **structural encoder** and a **description encoder**. The structural encoder is responsible for capturing the graph's inherent structure, often utilising traditional knowledge graph embedding techniques such as translational models (e.g., TransE). Simultaneously, the description encoder leverages the power of LLMs to encode the textual descriptions associated with entities and relations. By fusing these structural and descriptive representations, MoCoSA can capture more nuanced and comprehensive knowledge about the entities and their relationships. This enables superior link prediction performance, as evidenced by its state-of-the-art results on challenging datasets like OpenBG500 and WN18RR.

7.6 Rule Mining

7.6.1 AnyBURL [5][3][14]

AnyBURL, is a technique used in knowledge graphs for drug repurposing that focuses on **learning logical rules from the graph's structure**.

AnyBURL operates as a **bottom-up approach**, efficiently identifying logical rules within large knowledge graphs by **sampling random paths** over a set time. These paths are then used to extract rules. A confidence score is assigned to each rule, and those exceeding a predetermined threshold are retained. These rules can then be used to perform link prediction. AnyBURL selects the candidate node that satisfies the highest confidence rule, and in the event of a tie, the second-highest confidence rule is considered. This approach prioritises single, high-confidence rules, as AnyBURL posits that combining scores from multiple rules may not be as efficient due to the potential lack of rule independence.

Within a general architecture for path-based drug repurposing models, AnyBURL functions as a path generator. It samples paths using a random walk approach to generate rules and then applies these rules to the graph to produce predictions, which are subsequently ranked based on the rule's confidence.

There are some techniques that improve on AnyBURL. SAFRAN, for example, differs from AnyBURL by basing its predictions on a combination of scores from different rules rather than relying solely on the rule with the highest score. Also, **the rule miner in AnyBURL can be used to initialise the generator in XG4Repo**, as the generated rules are good for prediction tasks.

7.6.2 Other mining techniques

- **ARBOCK** is both a machine learning and rule mining technique! ARBOCK approach constructs a **rule-based classification model** using the characteristics of paths linking potential pathogenic gene pairs within the knowledge graph. It generates a rule set from local patterns of the pathogenic gene pairs and combines these rules into a decision set classifier to enable identification of potential pathogenic gene pairs.

- **AMIE** A fast and exact rule mining algorithm.
- **RUDIK** A rule discovery method in knowledge bases.
- **SAFRAN** An interpretable, rule-based link prediction method that improves upon Any-BURL by making predictions based on a **combination of scores of different rules**, rather than relying solely on the highest-scoring rule. SAFRAN groups similar rules together and then obtains a score, considering the score of different rules.
- **RNNLogic** A model that uses rules as latent variables to make predictions. It includes a rule generator and a reasoning predictor that apply the rules to propose candidate answers for the query.
- **RuDiK** A rule mining method.

Rule-based methods are more easier to interpret in the context of explainable AI (XAI), because the rules provide a **human-understandable explanation** for the model's predictions.

7.7 Statistical Methods

7.7.1 Fischer's Exact Test [8]

Fisher's exact test is used to find significant associations between diseases, anatomies, and symptoms. This test helps determine if the co-occurrence of these topics is statistically significant rather than due to random chance.

- The primary goal is to identify if there's a notable relationship between two categorical variables. In this case, the variables are diseases, anatomies, and symptoms found in articles. "Articles" here refers to scientific publications indexed in databases like MEDLINE, which is a subset of PubMed
- We use this test to find:
  - Articles featuring two or more disease topics.
  - Articles containing both a major disease topic and any anatomy topic.
  - Articles containing both a major disease topic and any symptom topic.
- **Search Options**:
  - `majr`: This option limits the search to articles where the topic is a **major focus**, ensuring relevance.
  - `noexp`: This option prevents the search from "exploding," meaning it **suppresses the inclusion of articles linked to MeSH (Medical Subject Headings) subterms**, thus focusing on the specific term itself.
- **Why Fisher's Exact Test?** Outside of the sources, it is particularly useful when sample sizes are small, where other tests like the chi-squared test might not be appropriate because their assumptions are not met. Fisher's exact test assesses whether observed proportions in a contingency table differ from what would be expected by chance alone.

8 Limitations

8.1 Scalability of GNNs [5][15][16]

The scalability of Graph Neural Networks (GNNs) is a significant concern, especially when dealing with large-scale, high-dimensional biomedical datasets. Several factors contribute to these scalability challenges, and various strategies are being explored to address them.

8.1.1 Challenges Contributing to Scalability Issues

- **Computational Demands** Training complex graph models on extensive biomedical datasets requires substantial computational resources.
- **Long-Range Connections** Modelling long-range connections in a large graph necessitates numerous layers in the GNN. However, most GNNs are only a few layers deep due to computational constraints.
- **Oversmoothing** Increasing the number of layers to model long-range connections can lead to oversmoothing, where node representations of different classes become indistinguishable.
- **Heterogeneous Minibatch Sampling** Creating random graph partitions for minibatching can result in batches with imbalanced node and edge types, especially in heterogeneous graphs.
- **Transductive Setting** Some GNNs work in a transductive setting, using a graph of a fixed size, which can be a major inconvenience if the knowledge graph is constantly being updated.

8.1.2 Strategies to Address Scalability Concerns

- **Distributed Computing, Parallel Processing, and Hardware Acceleration** Exploring these techniques can help handle the computational demands of training complex graph models on extensive biomedical datasets.
- **Algorithmic Techniques** Developing algorithmic techniques that can efficiently learn representations from massive graphs is essential for applying graph-based methods to high-dimensional medical datasets.
- **Neighbour Sampling** GraphSAGE extends Graph Convolutional Networks (GCNs) by allowing neighbour sampling, where the message-passing process is applied through a subgroup of neighbours instead of every node.
- **Minibatches** GraphSAGE incorporates the possibility to work in minibatches, fixing the space and time complexity of the model to O(Ki=1 Si), where K is the number of layers of the GNN and Si corresponds to the number of neighbours sampled on that layer.

8.1.3 GNN Variants for Complex Graphs

- **Temporal GNNs** These learn on dynamic graphs where nodes and edges change over time, providing a formal mechanism for temporal predictions and handling time-varying factors.
- **Hypergraph GNNs** These learn on hypergraphs, enabling n-level connections between nodes through hyperedges, and have been used to learn drug-disease relationships and EHR data.
- **Subgraph GNNs** These learn representations for subgraphs, which are subsets of nodes and edges in a larger graph, aiding in the classification of diseases represented as subgraphs of many phenotype nodes and edges.

8.1.4 Additional Considerations

- **Scalability vs. Expressivity** Graph transformers generally offer greater expressivity than message-passing neural networks (MPNNs), and recent research has sought to combine MPNNs and graph transformers to increase model expressivity.
- **Trade-offs in Model Selection** Choosing the appropriate knowledge graph depends significantly on the intended predictive algorithm and the requirements for explainable AI (XAI). Computational resources are a critical consideration, as larger graphs demand substantial RAM, CPU, and GPU capacities.
- **Limitations of GNNExplainer** Retraining the mask for each explanation can lead to different results for the same instance upon repeated runs.

Addressing these scalability issues and carefully selecting appropriate techniques and architectures are crucial for effectively applying GNNs in drug repurposing and other biomedical applications.

8.2 XAI [15]

Explainable AI (XAI) is critical for drug repurposing because it enhances the **trustworthiness and interpretability** of AI-driven predictions, particularly in healthcare, where decisions have substantial impacts. XAI methods can uncover potential errors or biases in the model and improve AI methods by identifying the biological mechanisms (genes, pathways, side effects, or anatomies) that link compounds to diseases. A key aspect of XAI involves generating explanations as paths within knowledge graphs, transforming abstract feature spaces into concrete entities and relationships. These paths provide insights into how drugs interact with targets to treat diseases, making predictions interpretable for experts.

Several techniques facilitate the generation of explanations in drug repurposing:

- **Graph-based queries** Translating rules into knowledge graph queries to retrieve relevant paths.
- **Semantic graphs** Some methods generate explanations as semantic graphs, providing a researcher-centric interpretable machine learning approach for hypothesis generation. These explanations, presented as semantic graphs, resemble human reasoning and support researchers in formulating evidence-based hypotheses.
- **Subgraphs** Explanations can be produced as subgraphs in which an AI model has predicted that a drug could potentially be used to treat a disease. A possible subgraph explanation could be that Drug A targets Gene X, which is one of the genes that causes Disease B.
- **Metapaths** Methods can prioritise providing just connecting paths such as metapath based ones and on improving path visualisation for user interpretation.

However, there are some limitations including:

- **Reproducibility Issues** The known reproducibility issues of explainers may reduce the confidence and reliance on the explanations.
- **Data Topology Affects Explanations** Data topology affects explanations, highlighting the importance of investigating further how best to represent graphical knowledge for model performance and explanation accuracy. For example, different knowledge graphs can have different clustering coefficients which lead to more or less edges being present in the subgraphs.
- **Incomplete Knowledge** The knowledge depicted in knowledge graphs is often incomplete, as collecting all the knowledge of a given area is often an expensive task. Additionally, long-term management of a knowledge graph is often costly, and therefore many knowledge graphs don't contain updated information.
- **Lack of standard benchmarking** There is a lack of standard benchmarking and metrics to systematically evaluate explainers and explanations which makes it difficult to do comparisons.

8.3 GNNExplainer Limitations [15]

GNNExplainer is a method applied to graphs that leverages Mutual Information (MI) to provide explanations for predictions obtained with AI models. However, GNNExplainer has several limitations:

- **Reproducibility Issues** GNNExplainer lacks consistency when obtaining explanations, meaning explanations on the same prediction can significantly change if running GNNExplainer several times. The known reproducibility issues of explainers may reduce the confidence and reliance on the explanations.
- **Computational inefficiency** Directly maximising MI can be very inefficient as it suggests iterating through every possible sub-graph, with a complexity of 2*M.
- **Disconnectedness** The explanation would consist in a subgraph where the two targeted nodes would be disconnected from each other, which might bring confusion and could be seen as a 'bad' explanation. Work towards methods that prioritise or focus on providing just connecting paths such as metapath based ones is arguably recommended.

- **Data Topology Affects Explanations** Data topology affects explanations, highlighting the importance of investigating further how best to represent graphical knowledge for model performance and explanation accuracy. For example, different knowledge graphs can have different clustering coefficients which lead to more or less edges being present in the subgraphs.
- **Limited scope** GNNExplainer can only be used in node classification tasks.
- **Lack of standard benchmarking** There is a lack of standard benchmarking and metrics to systematically evaluate explainers and explanations which makes it difficult to do comparisons.

8.4 Other limitations [1][5][12][3][4][15]

- **Bias towards PPI (protein-protein interaction) Network:** Empirical analysis of drug-gene-disease knowledge graphs showed that random walk and network propagation algorithms were biased towards the PPI network. PPI network is much larger and denser than drug-gene and disease-gene networks. **Connecting two sparse networks (drug-gene and disease-gene) through a large and dense PPI network is difficult, leading to representation learning frameworks being biased towards the PPI network**.
- **Data Incompleteness:** Knowledge graphs (KGs) often suffer from incomplete information, as collecting all relevant knowledge is challenging. In KGs, nodes represent biological entities and edges represent relationships. If information on either nodes or edges is missing, the KG will be incomplete. Around half the relations in some knowledge graphs are supported by only one reference. This necessitates the development of additional criteria for assessing the confidence and reliability of KG relationships.
- **Outdated Information:** The long-term management and updating of KGs can be costly, leading to outdated information that may not provide new insights.
- **Standardisation**: There is a lack of standard benchmarking and metrics to systematically evaluate explainers and explanations, which complicates comparisons between different models. The lack of a common standard for drug repurposing makes it challenging to directly compare different methods. Evaluating AI models is challenging due to the diversity of metrics, such as Area Under the Receiver Operating Characteristic Curve (AUROC), Area Under the Precision-Recall curve (AUPRC), Hits@k, and Mean Reciprocal Rank (MRR). Different researchers use different KGs to assess their methods, complicating comparative analysis.
- **Lack of Personalisation:** Current drug repurposing approaches using KGs typically identify general drug candidates rather than offering personalised medicine solutions.
- **Algorithmic Bias**: The disproportionate gene-phenotype annotation coverage in disease-associated genes could introduce bias in machine learning models trained on this dataset. For example, only 23.5% of all human genes are linked to a phenotype term, but the proportion rises to 82.6% when considering genes involved in known oligogenic diseases.
- **Explainability and Trust:** It can be challenging to fully trust predictions made by AI models that lack explanatory support. Providing model transparency and explainability is crucial for obtaining domain experts' trust and confidence in these models for high-stakes decision-making.
- **Reproducibility of Explanations:** The reproducibility/inconsistency of explanations could be affected by the size and complexity of data, making users skeptical about their reliability.
- **Single Reference Relations:** Approximately half the relations in some KGs are supported by only one reference, necessitating the development of additional criteria for relation confidence. Approaches for a new confidence measure include epistemic discourse analysis, adding an article citation index to the relation confidence score, developing a statistical score that measures overall alignment of a new reported relation with the entire KG, and distinguishing self-citation versus independent observations.
- **Knowledge Graph and Ontology Changes:** Knowledge graph and ontology changes pose a great interoperability challenge to the community to keep up downstream bioinformatics and data science workflows and analyses.

- **Semantic Hierarchy**: Semantic hierarchy information may act as a guide if appropriately utilized; otherwise, it may introduce noise into the network, leading to performance degradation.
- **Data-related challenges**: Fusing complex datasets into a cohesive graph representation requires data harmonization, feature engineering, and graph construction techniques.
- **Single XAI Method**: Using only one explainable AI (XAI) method is an important limitation of some studies.
- **Redundant triples:** Knowledge graphs (KG) are a particular type of multirelational graph where the information is defined by a set of existing triples, including a head node (h), a tail node (t*) and a relation (r) that links them.
- **Edge directionality:** High-level semantic knowledge graphs have inherent limitations, and incorporating more granular phenotypic effects (e.g., Drug A increases blood pressure) and directionality in edges could lead to a greater ability.
- **Computational Demands:** The size and complexity of many KGs necessitate high computational power, posing a practical challenge. Larger graphs demand substantial RAM, CPU, and GPU capacities.

9 Future work

TODO

References

[1] Mehdi Ali, Max Berrendorf, Charles Tapley Hoyt, Laurent Vermue, Mikhail Galkin, Sahand Sharifzadeh, Asja Fischer, Volker Tresp, and Jens Lehmann. Bringing light into the dark: A large-scale evaluation of knowledge graph embedding models under a unified framework. 2021.

[2] Kairong Hu, Hai Liu, Choujun Zhan, Yong Tang, and Tianyong Hao. Learning knowledge graph embedding with a bi-directional relation encoding network and a convolutional autoencoder decoding network, 2021.

[3] Alexandre Renaux, Chloé Terwagne, Michael Cochez, Ilaria Tiddi, Ann Nowé, and Tom Lenaerts. A knowledge graph approach to predict and interpret disease-causing gene interactions, 8 2023.

[4] Dongmin Bang, Sangsoo Lim, Sangseon Lee, and Sun Kim. Biomedical knowledge graph learning for drug repurposing by extending guilt-by-association to multiple layers, 6 2023.

[5] Pablo Perdomo-Quinteiro and Alberto Belmonte-Hernández. Knowledge graphs for drug repurposing: a review of databases and methods, 1 2024.

[6] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. 2017.

[7] Sameh K Mohamed, Aayah Nounu, and Vít Nováček. Biological applications of knowledge graph embedding models. 2020.

[8] Daniel Scott Himmelstein, Antoine Lizee, Christine Hessler, Leo Brueggeman, Sabrina L Chen, Dexter Hadley, Ari Greena, Pouya Khankhanian, and Sergio E Baranzini. Systematic integration of biomedical knowledge prioritizes drugs for repurposing. 2017.

[9] AH Jonker, D O'Connor, M Cavaller-Bellaubi, C Fetro, M Gogou, PACT Hoen, M de Kort, H Stone, N Valentine, and AMG Pasmooij. Drug repurposing for rare: progress and opportunities for the rare disease community. 2024.

[10] Shabunina EA, Malyshev I Yu, Kuznetsova LV, and Lobanov EV. Evolution of the drug repurposing paradigm. 2021.

[11] Helen I Roessler, Nine V A M Knoers, Mieke M van Haelst, and Gijs van Haaften. Drug repurposing for rare diseases, 2021.

[12] Anton Yuryev, Maria Shkrob, Alex Tropsha, and Grant Mitchell. Exploring drug repurposing for rare diseases: Leveraging biomedical knowledge graphs and access to scientific literature, 2025.

[13] Daniel N Sosa, Alexander Derry, Margaret Guo, Eric Wei, Connor Brinton, and Russ B Altman. A literature-based knowledge graph embedding method for identifying drug repurposing opportunities in rare diseases, 2020.

[14] Ana Jiménez, María José Merino, Juan Parras, and Santiago Zazo. Explainable drug repurposing via path based knowledge graph completion, 1 2024.

[15] P. Perdomo-Quinteiro, K. Wolstencroft, M. Roos, and Queralt-Rosinach. Knowledge graphs and explainable ai for drug repurposing on rare diseases.

[16] Ruth Johnson, Michelle M. Li, Ayush Noori, Owen Queen, and Marinka Zitnik. Graph artificial intelligence in medicine, 2024.

[17] E. Amiri Souri, A. Chenoweth, S. N. Karagiannis, and S. Tsoka. Drug repurposing and prediction of multiple interaction types via graph embedding, 5 2023.