BMC Bioinformatics

REVIEW Open Access

Graph embedding on mass spectrometry- and sequencing-based biomedical data

Edwin Alvarez-Mamani[1,2], Reinhard Dechant[2,3], César A. Beltran-Castañón[1] and Alfredo J. Ibáñez[2,4*]

*Correspondence:
aibanez@pucp.edu.pe

[1] Engineering Department, Pontificia Universidad Católica del Perú, San Miguel, Lima, Peru
[2] Institute for Omics Sciences and Applied Biotechnology (ICOBA PUCP), Pontificia Universidad Católica del Perú, San Miguel, Lima, Peru
[3] Present Address: Calico Life Sciences, 1170 Veterans Blvd, San Francisco, CA 94080, USA
[4] Science Department, Pontificia Universidad Católica del Perú, San Miguel, Lima, Peru

Abstract

Graph embedding techniques are using deep learning algorithms in data analysis to solve problems of such as node classification, link prediction, community detection, and visualization. Although typically used in the context of guessing friendships in social media, several applications for graph embedding techniques in biomedical data analysis have emerged. While these approaches remain computationally demanding, several developments over the last years facilitate their application to study biomedical data and thus may help advance biological discoveries. Therefore, in this review, we discuss the principles of graph embedding techniques and explore the usefulness for understanding biological network data derived from mass spectrometry and sequencing experiments, the current workhorses of systems biology studies. In particular, we focus on recent examples for characterizing protein–protein interaction networks and predicting novel drug functions.

Keywords: Graph embedding, Biomedical data, Biological network

Introduction

In the literature, several reviews present graph embedding models used to solve multiple tasks such as pathogen-host protein interactions, predicting drug efficiency, linking a metabolite with a metabolic network, etc [1–3]. However, wide spread application of graph embedding techniques in the life-science community has been scarce, which may be in part because the complex mathematical framework underlying graph embedding requires considerable bioinformatical expertise. To make graph embedding known to a wider research community we have focused our review to be accessible for wet-lab biologists as well as bioinformaticians, mainly using more accessible wording for life scientists and focussing on potential future applications.

Biological data is usually presented as graphs; some of the most famous ones are represented in the book Cellular Biochemical Networks (Editor: Gerhard Michal), which describes the known metabolomic network of eukaryotic cells and comprises most of the cellular metabolites and their interactions (i.e., possible conversions and connections between metabolic pathways such as sugar and amino acid metabolism). Although traditional biology tools have been extremely successful in identifying most components and

some of the major linear interactions contained in the Cellular Biochemical Networks graphs, one of the significant challenges in biology is comprehending the nonlinear or dynamic interactions among the cellular constituents to unravel the organization and interactions within cellular networks. For example, understanding which metabolic subnetworks are active in a particular cell type under specific conditions is critical to decipher the influence of the metabolic network on cellular function.

Mass spectrometry (MS) is an excellent example of a tool for understanding the underlying interactions among large numbers of cellular constituents. MS-based metabolomic and proteomics studies can follow various linear and nonlinear interactions (based on signal abundances) and dynamic interactions from time series measurements. The interactions are visualized via correlation plots of the MS signals [4, 5]. In a correlation plot, metabolites, proteins, etc., are represented as dots (or nodes), and a line illustrates their correlations with other network elements. Using carefully designed experiments and bioinformatic tools makes it possible to model and quantify the different types of interactions between the nodes. Hence, a traditional approach in molecular biology is to compare two or more graphs to identify which metabolites or proteins in the biological network are associated with a particular physiology (i.e., disease) or phenotype of interest [4, 5].

Unfortunately, clear insight into biological information via visual inspection of the correlation plots is challenging due to the large number of biological species present in cells that MS can detect. Furthermore, artifacts such as the presence of ghost peaks or batch effects can futher obscure the information within these graphs [4, 5]. Graph embedding techniques have been developed to analyze complex graphs of diverse origins. A graph embedding technique takes graphs as input and converts the graphs into a matrix of vectors (i.e., a lower-dimensional latent space), thus allowing researchers to better identify the interactions between their different elements. Although graph embedding techniques have been applied to various fields of study, e.g., to analyze relationships between client and providers in financial transactions [6], to recommend locations using recommender systems [7], or to detect malware [8]; they have not been routinely applied to biological systems and are not well-known to life-scientists.

This review discusses the suitability of graph embedding techniques for analyzing masss spectrometry- and sequencing-based biomedical data and explains the theoretical background to understand graph embedding. We classify graph embedding techniques from the perspective of biomedical data, considering the canonical classification, thereby subdividing graph embedding techniques into random walk-based, matrix factorization-based, and deep learning-based algorithms. Additionally, we review articles that applied graph embedding for link prediction, node classification, and node clustering tasks on biomedical data and highlight novel biological insights obtained by graph embedding. In particular, we will focus on protein–protein and drug–protein interactions. Our review will help future readers to identify, which graph embedding models can be applied to solve a given task on biomedical datasets, which datasets can be used, and which metrics are available to evaluate the results.

The paper is structured as follows: section "Theory of graphs embeddings" contains the necessary definitions and summarizes the theoretical background of graph embedding. Then, section "Applications of graph embeddings in mass spectrometry- and

sequencing-based biomedical data" describes the existing applications of graph embedding techniques on biomedical data. Finally, section "Conclusion" discusses conclusions and future applications.

Theory of graphs embeddings

Background techniques

To be able to understand graph embedding, we first must introduce the term word embedding, which transforms a group of words (i.e., text) into a matrix of vectors and is frequently used in natural language processing (NLP) [9]. In more detail, word embedding technique results in the (n-dimensional) vector representation of a word (token) within a text [10]. Since words often occur in the same semantic or syntactic context, a cosine similarity measure among the vectors in the matrix can be used to identify the relationship between words. Hence, the semantic and syntactic similarity between words can be mathematically identified [11, 12]. For example, word embedding is used when a word processing program suggests a phrase after the computer user types just a few words. Two different strategies were proposed for word embedding (i.e., architectures): `Continuous bag-of-words` (CBOW) [13] predicts a word w_i in one particular position in the sentence based on the context of words surrounding that position $w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2}$, while `continuous skip-gram model` [13] predicts the context (surrounding words) with respect to a particular word in the sentence. The first formulation of `skip-gram` model defines the conditional likelihood $P(w_{context} \mid w_{center}) \approx P(w_o \mid w_c)$ utilizing the function `softmax` [13, 14],

$$P(w_o \mid w_c) = \frac{exp(u_o^\top v_c)}{\sum_{i=1}^{|W|} exp(u_i^\top v_c)} \tag{1}$$

where o is the index of the context word (output) in the dictionary, c is the index of the central word (input) in the dictionary, and W is the vocabulary.

Similarly, *continuous bag-of-words* defines conditional likelihood $P(w_c \mid w_{o_1}, \ldots, w_{o_{2m}})$ [13, 14], where $o_1, \ldots, o_{2m}$ are the indexes of the context words in the dictionary.

$$P(w_c \mid w_{o_1}, \ldots, w_{o_{2m}}) = \frac{exp\left(\frac{1}{2m} u_o^\top (v_{o_1} + \cdots + v_{o_{2m}})\right)}{\sum_{i=1}^{|W|} exp\left(\frac{1}{2m} u_i^\top (v_{o_1} + \cdots + v_{o_{2m}})\right)} \tag{2}$$

Although the `skip-gram` architecture performs slightly worse on syntactic tasks than the `CBOW` model, it does much better on semantic tasks [13]. Executing the definition (Equ. 1) has a very high computational cost [13, 14]. Therefore, [15] optimized the training process of the `skip-gram` model by adding the `hierarchical softmax` and `negative sampling` techniques.

Graph embedding is applied to dot (scatter) graphs. In analogy to *word embedding*, in *graph embedding* a point (i.e., node) in a graph is considered as a word, which is surrounded by other points (i.e., words). Furthermore, the graph contains information about the relationship between any two given points (words); this relationship is defined as an edge between two nodes. Hence, graph embedding can be used to create a matrix of vectors for all the nodes in a graph based on their edges by using the following

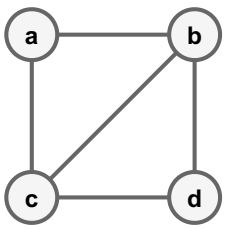
analogy [16–19]: given a sequence of words, $S_1^n = (w_1, w_2, \ldots, w_n)$ where $w_i \in W$, it can be inferred $P(w_n \mid w_1, w_2, \ldots, w_{n-1}) \approx P(v_i \mid v_1, v_2, \ldots, v_{i-1})$ and v_i represents a node in a graph G.

Graph embedding

The following definitions are useful to better understand and develop graph embedding and its applications.

Definition 1 (Graph) In mathematics and computer science, a graph is a scatter plot with a defined data structure. Let G be a graph, defined as $G = (V, E)$ where $V = \{v_1, v_2, \ldots, v_n\}$ is a set of nodes (vertices), and E represents the connection (edge) between 2 nodes $(v_i, v_j) \in V$ [1, 20–23]. Given a graph (Fig. 1a), this graph can be represented by an adjacency matrix: is 1 when there is an edge from node v_i to node v_j, and is 0 when there is no edge (Fig. 1b). The adjacency list groups the neighboring nodes of each node v_i (Fig. 1c), while the edge list consists of ordered pairs (v_i, v_j) when there is an edge from node v_i to node v_j (Fig. 1d) [20, 21, 24, 25].

Definition 2 (Homogeneous and heterogeneous graphs) In a homogeneous graph, all nodes and/or edges are of the same type. For example, in the friends' network, each node represents a person, and an edge represents friendship between two people. In contrast, in heterogeneous graphs, nodes and edges can be of different types. Heterogeneous graphs are exemplified by an education network, in which there may be nodes representing teachers and students, and it is possible to have the relationships (edges) between teachers (colleagues), between teachers and students, and between students (classmates) [1, 20, 21, 24]. By their nature, biochemical networks can be defined as homogeneous or heterogeneous graphs. For example, protein–protein interaction studies are represented



(a) Input graph

	a	b	c	d
a	0	1	1	0
b	1	0	1	1
c	1	1	0	1
d	0	1	1	0

(b) Adjacency matrix

a	b	c	
b	a	c	d
c	a	b	d
d	b	c	

(c) Adjacency list

a	b
a	c
b	c
b	d
c	d

(d) Edge list

Fig. 1 Graph representation

in homogeneous graphs [26–29], while miRNA-disease/gene interaction studies are represented by heterogeneous graphs [30–32].

Definition 3 (Directed and undirected graphs) In directed graphs (digraph), the list of nodes (i.e., vertices) that generates the graph is ordered, and each interaction (i.e., edge) has a direction. Traversal in this type of graph is done according to the direction of the interactions among nodes, while in undirected graphs traversal can be done in both directions [1, 20, 21, 24]. In metabolic pathways, both types of graphs are present. Metabolic pathways, in which each product (i.e., node) is solely dependent on its precursor (i.e., a previous node in the pathway), can be defined as directed. However, most metabolic pathways are represented as undirected graphs, since their chemical reactions are reversible and regulated by feedback loops, where downstream products influence the formation of their upstream precursors (e.g., in glycolysis) [33, 34].

Definition 4 (First-order and second-order proximity) The first-order proximity measures the proximity between a pair of nodes v_i, v_j, and represents the weight w of the edge e_{ij} ($w \geq 0$). If the edge does not have a weight, then the default value is 0. Then, first-order proximity is defined as the neighborhood of the node v_i containing a set of adjacent nodes $N_{v_i} = \{v_k \mid e_{ik} > 0, k \neq i\}$. The second-order proximity measures the number of 2-hop paths between a pair of nodes v_i, v_j [2, 24].

Definition 5 (Graph embedding) Given a graph as input $G = (V, E)$, graph embedding (see Fig. 2) is defined as a mapping function $f : v_i \rightarrow Z_i \in \mathbb{R}^d$ (latent space) with $i \in \{1, 2, \ldots, n\}$ where $d \ll |V|$ and Z_i is a vector of dimension d known as an *embedding* [2, 22, 24].

Classification of graph embedding techniques

Most commonly, graph embedding techniques are classified as either matrix factorization-based, random walk based, or deep learning-based [1, 2, 22–24, 35].

However, in the literature, an alternative classification has been introduced based on the point of view of the mathematical problems, which can be *vector point-based*, *gaussian distribution-based*, or based on *dynamic graph embedding* [1]. Vector point-based approaches aim to project the nodes of a high-dimensional graph onto low-dimensional vectors within a vector space [1]. Gaussian distribution-based methods allow the vector representation (embedding) of a node as *potential functions* of *continuous densities* in a

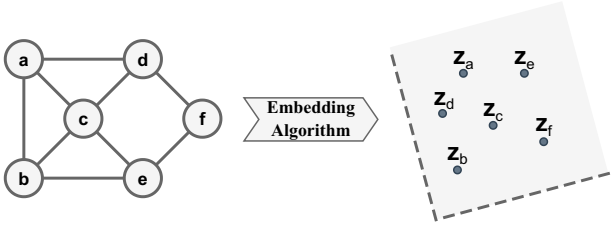


Fig. 2 Graph embedding scheme

Table 1 Network embedding models

Category	Publications
Non-attributed network	
Shallow embeddings	[16, 17, 19, 37–53]
Graph neural networks	[54–62]
Attribute network	
Semantic matching models	[63–78]
Translational distance models	[79–85]
Meta-path-based methods	[18, 86–90]

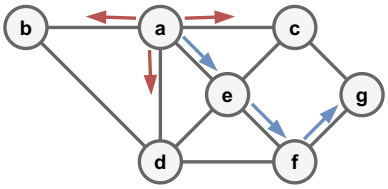


Fig. 3 BFS (red arrows) and DFS (blue arrows) traversals, from node *A* with a path length of 3

vector space. [1]. Dynamic graph embedding is often the method of choice for practical applications, as many networks are dynamic and evolve, leading to the addition of removal of nodes or edges [1].

Alternatively, it was proposed that embedding techniques can be grouped from the perspective of biomedical networks, including biomedical relation data, biomedical knowledge graphs, biomedical ontology, or clinical data, in *non-attributed network embedding* and *attributed network embedding* [36]. Below is the classification of non-attributed network embedding [1, 2, 22, 24, 35, 36] and attributed network embedding [2, 36]. Table 1 shows the graph embedding models published by category.

Mathematical concepts behind graph embeddings

Shallow embeddings are the earliest graph embedding technique applied to life-science data based on homogenous networks (i.e., networks based on only one biological entity, such as proteins). Shallow graph embeddings are subdivided into random-walk and matrix-factorization algorithms. Examples of random-walk algorithms are (`DeepWalk` [16] and `Node2vec` [17]; while matrix-factorization examples are graph factorization [43] and GraphRep [44].

DeepWalk [16] was the first graph embedding technique used to represent the vertices (nodes) of a homogeneous graph in vectors [91]. The process begins when the random walk algorithm generates a sequence of vertices. The model is then trained using the `skip-graph` algorithm [13]. Finally, the result is the vector representation for each vertex, also called embedding.

Node2vec [17] is a generalization of `DeepWalk` [16]. The authors added two parameters, p, and q, which drive the generation of paths (see Fig. 3) by using the idea of breadth-first traversal (BFS) and depth-first traversal (DFS). When $q > 1$, the traversal approaches BFS, and the random walks lead to a *micro-view* of node

neighborhoods. In contrast, $q < 1$ is an exploration *macro-view* that approximates a DFS traversal for node neighborhoods [1]. The authors of the base article used the values of $p = 1$, $q = 2$ for a micro-view and the values of $p = 1$, $q = 0.5$ for a macro-view. The parameters p and q also control how fast a path is explored, and the neighborhood of an initial node v_i is left. The authors performed multi-label classification and link prediction experiments to verify their proposal. Results were evaluated using the F1-score metric.

While Deepwalk and Node2vec provided a solution to tasks such as link prediction, node classification, node clustering (community detection), and visualization, two random-walk algorithms, `Netpro2vec` [92] and `Pathway2vec` [33], were proposed to better analyze biomedical datasets.

Netpro2vec [92]: In the techniques described above, nodes of a network were transformed into tokens. Instead, the main concept of `Netpro2vec` is to transform networks into documents. The process is carried out in 3 steps: 1) building the probability distributions representing each graph, 2) extracting tokens from probability distributions, and 3) building the graph embedding using token extraction. The graph is then represented as a word document (a set tokens), and the `Doc2vec` (document embedding) technique is applied to obtain the graph embedding [93]. The proposal was compared with other techniques of whole-graph embeddings to solve classification tasks in gene networks. The results were evaluated based on accuracy, precision, recall, F-measure, and Matthews correlation coefficient (MCC) metrics.

Pathway2vec [33] incorporates multiple random walk-based techniques, `Node2vec` [17], `Metapath2vec`, `Metapath2vec++` [18], `JUST` [94], and `RUST` [33], to represent learning by automatically generating features of metabolic pathways. It consists of three layers that interact: compounds, enzymes, and pathways. This interaction between layers results in a heterogeneous network of multi-layer information, and each layer has associated nodes. The layered architecture captures meaningful relationships to learn a low-dimensional space based on neural embeddings of metabolic features. Finally, applying the `skip-gram` [13] model, the embeddings for each node are extracted. `Pathway2vec` was applied for node clustering, embedding visualization, and pathway prediction tasks. Evaluation of the results was performed using `MetaCyc` software and F1-micro metric.

Graph Factorization (GF) [43]: `GF` is a factorization technique based on partitioning a graph to minimize the number of neighboring vertices instead of edges between partitions. `GF` begins from the assumption that the information regarding the presence of an edge (i, j) with a weight Y_{ij} can be captured by the inner product between vertices with attributes $\langle Z_i, Z_j \rangle$. Finally, the value of the vector Z is determined by the following objective function:

$$f(Y, Z, \lambda) = \frac{1}{2} \sum_{(i,j) \in E} (Y_{ij} - \langle Z_i, Z_j \rangle)^2 + \frac{\lambda}{2} \sum_i \|Z_i\|^2 \tag{3}$$

where λ is the regularization parameters, E is the list of edges. To validate their proposal, the authors applied `GF` on a graph of 200 million vertices and 10 billion edges. In order to evaluate convergence and execution time, they used 3 architectures: a single machine,

a synchronous parallel implementation and an asynchronous parallel implementation. The results showed that asynchronous parallel implementation is very beneficial for scalability.

GraRep [44]: GrapRep is a model for learning node representation. This model captures the relational information of different k-steps with different values of k between vertices of the graph, directly manipulating different global transition matrices defined on the graph without slow and complex sampling processes. GraRep defines different loss functions and optimizes each model with matrix factorization techniques, constructing global representations of each node by combining the different model representations. Experiments were run to solve the node clustering and node classification tasks on linguistic networks and social networks, respectively. In both tasks, GraRep showed an empirical efficiency of the learned representations compared to the LINE and DeepWalk models.

While shallow-embedding algorithm applications focus on solving link prediction, node classification, and community detection tasks, more complex problems such as graph matching, subgraph matching, and calculating the maximum common subgraphs require more complex models. Graph-neural network (GNN) algorithms can address these problems by combinatorial optimization using graph theory. Furthermore, these problems are solved through representation learning (deep learning); for example, in [95] a GNN model is proposed that addresses the subgraph matching problem for molecular fingerprint detection.

Graph Convolutional Network (GCN): Kipf et al. [56] present GCN for semi-supervised learning that works directly on graphs. GCN is a variation of convolutional neural networks. It scales linearly with the number of edges and encodes the local structure of the graph and features of nodes. The task of node classification is approached on a graph with partially labeled nodes, using a neural network $f(X, A)$ trained in a supervised environment with node feature matrix X and adjacency matrix $A \in \mathbb{R}^{N\times N}$. For this purpose, a multilayer GCN is considered with the following layer-wise propagation rule.

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}) \tag{4}$$

where $\tilde{A} = A + I_N$ is the adjacency matrix of undirected graph with added loops, I_N is the identity matrix, $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ and $W^{(l)}$ is a layer-specific trainable weight matrix, $\sigma(.)$ is an activation function, $H^{(l)} \in \mathbb{R}^{N\times D}$ is the matrix of activations in the l^{th} layer $H^{(0)} = X$. The experiments were run on 4 datasets (Citeseer, Cora, Pubmed, and NELL), and the results showed that CGN significantly outperforms DeepWalk.

In the case of attribute data—biomedical data based on heterogeneous networks (i.e., networks based on more than one biological entity, such as drug–protein target interactions)—the graph embedding algorithms must consider both the node distribution and the edge information of the graph. Embeddings are generated that encode the proximity between nodes based on their attributes and connectivity patterns. Graph embeddings algorithms for attribute data can be divided into semantic matching models (e.g., DDKG, DistMult, etc.), translational distance models (e.g., TransE, TransR), and metapath-based methods (e.g., Metapath2vec).

DDKG: Xiaori et al. [96] used an approach denominated "attention-based knowledge graph representation learning framework" or DDKG to simultaneously consider drug

attributes and triple facts in knowledge graphs (KG). A triple fact is the link between one entity (e.g., metabolite, protein, etc.), usually referred as subject or head, and another entity referred as object or tail. The relationship between this two entities is referred as relationship or label. Xiaori et al.'s work aimed to use all the information available in biomedical KGs and improve the results in the link prediction task in drug–drug interaction (DDI) networks. The proposal was developed in 4 steps: 1) Building the KG, 2) Generating the initial embeddings for each drug according to its KG, 3) Generating the global embeddings of the drugs considering the node-embeddings of their neighbors, 4) finally, DDKG determines the probability of interaction of drugs in pairs with their respective embeddings through a binary classification. The experiments were conducted on two biomedical KGs and compared with ten state-of-the-art models, including `LINE` and `SDNE`. Results obtained from DDKGs were evaluated by metrics of accuracy, sensitivity, specificity, AUC, and AUPR, demonstrating that DDKGs outperformed the state-of-the-art models.

DistMult [67] considered learning entity and relationship representations in knowledge bases (KBs) using the neural-embedding approach. The learning process seeks to learn entity and relationship representations such that valid triple facts (i.e., known facts) receive high scores. The triple facts are denoted by (e_1, r, e_2), where e_1 is the subject, e_2 is the object, and r is the relationship between the two. The first layer of the model projects a pair of entities from the input into low-dimensional vectors, and the second layer combines these two vectors into a scalar to be compared by a scoring function. Entity representation learning can be defined as:

$$y_{e_1} = f(WX_{e_1}), \quad y_{e_2} = f(WX_{e_2}) \tag{5}$$

where f can be a linear/nonlinear function, W is a parameter matrix, W can be initialized randomly/pre-trained, and X is a one-hot/n-hot vector representing the input entities e_1 and e_2. `DistMult` was empirically evaluated for link prediction tasks on the Freebase dataset. The results showed that a bilinear model successfully captures the compositional semantics of the relationships. It is also reported that `DistMult` outperforms `TransE` with a top-10 accuracy of 73.2% versus 54.7%.

TransE: Antoine et al. [80] addressed the problem of embedding different class entities (e.g., metabolites, proteins, etc.) and relationships of multi-relational data in low-dimensional latent spaces. The primary condition is that all the different entities (e.g., protein, metabolite, gene, etc.) must be present in a directed graph. In this directed graph, a triple fact consists of one entity (designated head), which is related to another entity (designated tail) by an edge (designated label). `TransE` is an energy-based model that learns embeddings of low-dimensional entities. For `TransE` the relationships are represented as translations in latent space; if a strong relation (edge) exists among two nodes (i.e., head and tail), then the embedding of the tail entity must be similar to the embedding of the head entity plus some vector that satisfies the relationship. For its simplicity, TransE has a small number of parameters and is scalable. Experiments showed that `TransE` performs well and significantly outperforms the `RESCAL` method in the link prediction task on two large knowledge base, Firebase and Wordnet.

TransR [82]: In contrast to the `TransE` model, where entities and relations (edges) are embedded in the same latent space, in `TransR` it was proposed to build the embeddings

of the entities and the edges in separate latent spaces linked by specific relation matrices, yielding one entity space and multiple relation spaces. `TransR` was based on the idea that entities that have a relationship of the form (head, label, tail) are first projected from the entity space into the r-relation space as h_r and t_r with M_r operation, and then $h_r + r \approx t_r$. The relation-specific projection can make the head/tail entities that actually hold a strong relation (edge) close to each other and also move away those that do not. In the experiments, Lin et al. [82] evaluated the model with three tasks: link prediction, triple classification, and relational fact extraction using the WordNet and Freebase datasets. The results showed that `TransR` obtains significant improvements compared to `TransE`. Additionally, they proposed `CTransR`, a combination of `TransR` and Clustering.

Metapath2vec [18]: Unlike `DeepWalk` [16] and `Node2vec` [17], `Metapath2vec`, guides and generates paths using random walks through meta-path schemes. It captures the structural and semantic relationships between different types of nodes in heterogeneous networks. Formally, a meta-path is defined as a path $\mathcal{P}$ represented by, $\mathcal{P} : V_1 \xrightarrow{R_1} V_2 \xrightarrow{R_2} \dots V_t \xrightarrow{R_t} V_{t+1} \dots \xrightarrow{R_{l-1}} V_l$, where $R = R_1 \circ R_2 \circ \cdots \circ R_{l-1}$ defines complex relationships between node types V_1 and V_l. The `skip-gram` architecture is also used by `Metapath2vec` to determine embeddings. Dong et, al. [18] evaluated their proposal on heterogeneous graphs for solving multi-classification nodes, node clustering, and similarity search problems. The results were evaluated using the F1-score metric.

Applications of graph embeddings in mass spectrometry- and sequencing-based biomedical data

Applications of graph embedding techniques for mass spectrometry- and sequencing-based data covered in this review are summarized in Table 2 [26, 31, 33, 92, 97, 98]. By their nature, certain—OMICs data can be stored in a graph data structure. For example, gene–gene, protein–protein, and metabolite-metabolite interactions can be stored in homogeneous graphs. In contrast, heterogeneous graphs can contain multiple species, e.g., drug–protein, gene–protein interactions, etc. and analyzing these graphs can contribute to biological knowledge. However, computational tools to study graph data structures in biological graphs can suffer from high computational and space costs, especially in large-scale information containing graphs [28]. Graph embedding algorithms can then be used to identify interactions between heterogeneous nodes such as: drug–target [26, 99–101], miRNA-disease [30, 31], miRNA-target [32], miRNA-gene [32], microbe-drug [102], gene–disease [31, 103], gene–pathway [31], cell–gene [104], chemical–disease [31]. On the other hand, the interaction between homogeneous nodes may be protein–protein [26–29], drug–drug [34, 100, 102], microbe-microbe [102], gene–gene [104].

As an example, Su et al. [28] applied graph embedding to improve the identification of protein–protein interactions. To avoid the high computational cost of identifying the possible protein–protein interactions based on previous graph embedding techniques, the authors studied different approaches (algorithms) to accelerate graph embedding and improve its accuracy. The authors' contribution was 2-fold. Firstly, their approach denominated LPPI integrated protein attributes into the graph embedding task. This way, multi-view information was used, improving the accuracy of the

Table 2 Summary of graph embedding on biomedical data

Techniques	Dataset	Applications	Evaluation Metrics
Combined DeepWalk, LINE, Node2vec, and SDNE [26]	MATADOR, PubTator, and BioGRID	Link prediction	AUC, AUPR, MAP, Avg. R-precision, and Precision@k
HeteWalk [115]	HPRD, MISIM, MimMiner, DisGeNET, and miRTarBase	Link prediction	AUC
Cascade model [97]	BioChem, Drug Bank, and PubChem	Link prediction	Accuracy, hits@10, and AUC
[29]	Krogan, Dip, and BioGRID	Node clustering	Precision, recall, F-score, fraction, geometry accuracy, and MMR
HNERMDA [102]	MDAD and aBiofilm	Link prediction	Accuracy, AUC, and AUPR
PmDNE [30]	HMDD3.0	Link prediction	AUC, AUPR, precision, accuracy, recall, and F1-score
HO-VGAE [27]	HI-II-14, HI-III, Lit-BM-13, BioGRID, and Bioplex	Link prediction	AUPR, Precision@k
HMNE [105]	Lazega, CKM, DBLP, C.elegans, H.genetic, PPI, and Twitter	Link prediction and node classification	F1-micro, F1-macro, and AUC
TriModel [99]	DrugBank_FDA, KEGG_MED, and Yamanishi_08	Link prediction	AUC and AUPR
FactorHNE [103]	DisGeNet, HPO and Orphanet, STRING 10	Link prediction	AUPR, AUC, Precision@K, and Recall@K
[100]	DrugBank_FDA, UNIPROT	Link prediction, node clustering	Accuracy
Hybrid model GVS [31]	GO, HPRD, CTD, HMDD and MATADOR	Link prediction	Accuracy and F1-score
DeepWalk and Node2vec [98]	DrugBank, Bio2RDF, human disease network, SIDER, KEGG, and PharmGKB	Link prediction	AUC and AUPR
Netpro2vec [92]	LFR, MREG, Kidney RNASeq, Brain fMRI COBRE, Breast RNAseq, Breast Microarray, MUTAG	Node classification	Accuracy, precision, recall, F-score, and MCC
BiLSTM [101]	Human, DUD-E, and ChEMBL	Node classification	AUC, precision, and recall
scLINE [104]	Usoskin, Li, Pollen, Patel, Darmanis, Camp, Muraro, and Petropoulos	Node clustering	DBI, NMI, ARI, Jaccard and Purity
PRD [34]	Bio2RDF, and DDI Corpus	Link prediction	AUC, AUPR
ACNE and ACNE-ST [116]	Cora, Citeseer, Wiki, and DBLP_C4	Node classification and node clustering	F1-micro and F1-macro
Pathway2vec [33]	EcoCyc, HumanCyc, AraCyc, YeastCyc, LeishCyc, and TrypanoCyc	Link prediction and node clustering	F1-micro
LPPI [28]	PPI network and GraphSAGE-PPI	Link prediction and Node Classification	Accuracy, sensitivity, precision, MCC, and AUC
MRMTI [32]	miRTarBase, miRBase, HumanNet, and biomaRt	Link prediction	AUC, AUPR, precision, recall, F1-score, and balanced accuracy
CANE [117]	Disease Encylopedia Section of XYWY.com.	Link prediction	Precision@k and recall@k

AUPR area under precision-recall curve, *ROC* receiver operating characteristics, *AUC* area under the curve ROC, *MCC* Matthews correlation coefficient, *DBI* Davies–Bouldin index, *NMI* normalized mutual information, *ARI* adjusted rand index, *MMR* maximum matching ratio, *MAP* mean average precision

graph embedding process. Secondly, the graph was reconstructed using the `GraphZoom` algorithm to reduce the graph's size. Therefore, the authors could accelerate the efficiency of the embedding algorithms. Combining the above two aspects, the authors' algorithm, LPPI, saves execution time without losing accuracy (AUC 0.99996) in identifying protein–protein interactions in a large dataset.

Despite representing a major advance in the use of graph embedding, Su et al. [28] only used a homogeneous dataset from protein data. However, biological information from systems biology studies is typically derived from multi-omics datasets and contain heterogeneous information (DNA, RNA, protein, and metabolite information). Furthermore, as the network of interactions is time or condition-sensitive, multilayer networks must be considered [4].

Gong et al. [105] proposed the use of a multilayer network embedding to handle data sets with multiple types of nodes and edges found in heterogeneous graphs. This approach becomes extremely useful for evaluating the performance of node embedding in link prediction, which tries to predict edges that most likely will appear in theoretical networks (not experimentally measured data); this is similar to the approach performed by bioinformatics in in-silico studies. As some tested datasets are very large and complex, it is hard to predict links on the whole node sets. Hence, Gong et al. [105] suggested first extracting a core set of nodes of each dataset and conducting link prediction in these core sets. Hence, many authors similar to Gong et al. are encouraging the use of more complex graph-embedding algorithms that are based on combinations of the above-mentioned ones. These combinations of graph-embedding algorithms are known as encores or graph neural networks.

For example, Ray et al. [106] used a combination of graph-embedding algorithms as proposed by Gong et al. to generate a graph embedding encore algorithm approach to identify potential drugs that could affect the protein–protein interaction (PPI) between the SARS-CoV-2 virus and its human target proteins. The SARS-CoV-2 viral protein and human interaction datasets (i.e., protein interaction graph) were based on the experimental data obtained by Gordon et al. [107] by means of affinity-purification mass spectrometry (AP-MS) screening and on the theoretical data by Dick et al. [108].

The graph embedding-based algorithm proposed by Ray et al. [106] to repurpose drugs against COVID-19 considered that the available data was heterogeneous. They suggested to combine three different data sets: (i) SARS-CoV-2—host protein interactions, (ii) human protein–protein interactions, and (iii) drug–human protein interactions to predict possible novel treatments to interfere with infection. As described by Gong et al. [105], these three datasets were very large and complex; hence, Ray et al. [106] had to reduce the dataset complexity by performing the data reduction step, i.e., a first graph embedding based on the `Nod2vec` algorithm to obtain the feature matrix (`X`). In the second step, the novel graph embedding algorithm denominated variational graph autoencoder (VGAE) was used for link prediction tasks. As input, VGAE receives the adjacency matrix (`A`) and the feature matrix (`X`) from the original graph (`X` replaces the one-hot matrix that the VGAE model uses by default and also helps improve prediction precision). The encoder of VGAE converts the input data to lower-dimensional representation (`Z`) and the decoder takes `Z` to reconstruct the

original input in ($\hat{\mathbb{A}}$), where $\hat{\mathbb{A}}$ is similar to $\mathbb{A}$, and in $\hat{\mathbb{A}}$ new connections between the different types of nodes can be discovered.

The results of Ray et al. [106] were compatible with those observed by other authors. For example, Ray et al. [106] identified the angiotensin-converting enzyme-2 (ACE-2) as a potential drug target against SARS-CoV-2 [109, 110]. Interestingly, the authors also found that drugs used to prevent Malaria and pneumocystis pneumonia (PCP) relapses, such as Primaquine, have therapeutic potential against SARS-CoV-2 based on the interaction of Primaquine with the TIM complex, consisting of TIMM29 and ALG11.

Similarly, Zitnik et al. [111] used a graph convolutional network, a combination of graph-embedding algorithms with a convolutional neural network that can work directly on graphs, to predict clinical side effects in patients taking multiple drugs simultaneously.

As in the case of Ray et al. [106], Zitnik et al. [111] combined multimodal graphs of protein–protein interactions, drug–protein target interactions, and known clinical drug side effects. Their new graph embedding algorithm, named Decagon, could accurately predict drug side effects in patients with complex diseases or co-existing conditions necessitating simultaneous medication for their treatment.

The use of shallow embeddings, such as (Nod2vec) is limited as shallow embeddings do not share information between the nodes and do not take advantage of the characteristics of the nodes in the coding process. To mitigate these limitations, graph neural networks (GNN) have more sophisticated encoders that take advantage of the structure, features, and attributes of graphs [112].

Su et al. [113] proposed constrained multi-view nonnegative matrix factorization (CMNMF), a model based on GNN, to determine the similarity between drugs and viruses within their space of characteristics (latent space). Therefore, CMNMF is oriented towards preserving drug and virus similarity information as much as possible. Then, they apply a graph convolutional network (GCN) with attention-based neighbor sampling to optimize the vectorial representation of drugs and viruses in virus-drug associations (VDA) networks, whereas VDA networks are considered heterogeneous graphs. The experiments were executed on three VDA datasets to identify possible drugs against SARS-CoV-2. The embedding algorithm from Su et al. outperformed other models and was evaluated with the accuracy, F1, AUC, and AUPR metrics.

Decagon, a DeepWalk neural graph embedding, outperformed baseline algorithms by up to 69% (accuracy). Specifically, Decagon could automatically predict side effects with a known strong molecular basis with high precision, but still performed well on predicting side effects with a non-molecular basis due to its effective sharing of model parameters across edge types.

Finally, Nelson et al. [114] mentioned the advantages of graph embedding techniques compared to other techniques that operate directly on biological/biomedical networks. One advantage is a more rapid analysis of the learnt embedding. Unlike the tasks mentioned in the other works (link prediction, node classification, and node clustering), Nelson et al. [114] demonstrated the usefulness of graph embeddings for more specific tasks in biology, such as protein network alignment, protein module detection, and protein function prediction. Taken together, these examples establish the high value of graph embedding techniques for the analysis of mass spectrometry—and

sequencing-based—OMICs datasets. Several other applications have been published, which could not be discussed in greater detail, but have been showcased in Table 2 and classified for the use for (i) link prediction, (ii) node classification, and (iii) node clustering tasks.

Although Table 2 shows how graph embedding algorithms have become popular for representing biomedical data, several major limitations are apparent that limit the general applicability of graph embedding to life sciences:

- Most graph embedding algorithms have been developed to accomplish a specific task on a specific dataset, with no standards or even flexibility for incorporating other datasets. For using the same graph embedding algorithm to solve a different task, the new data set must be rewritten, thus limiting the application for other researchers.
- Shallow-embedding algorithm applications are limited in their applications, such as link prediction, node classification, and community detection tasks. More complex problems such as graph matching, subgraph matching, and calculating the maximum common subgraphs require more complex models requiring combinatorial optimization (graph theory). Furthermore, these problems are solved through representation learning (deep learning). However, most deep-learning graph embedding techniques are not deterministic because they use probabilities to perform their tasks, yielding similar, but not identical results for different runs.
- Loss of structural information: graph embedding methods typically aim to preserve the proximity of nodes based on their graph structure. However, they may lose certain structural information during the embedding process. For instance, (i) higher-order relationships within the graph may not be accurately captured. Furthermore, (ii) graph embeddings may not effectively leverage node attributes or features. Node attributes (metadata) can provide valuable information in life sciences, such as measurement conditions. It may be computationally expensive to maintain graph embeddings for (iii) dynamic data sets where nodes and edges are frequently added, removed, or modified (due to experimental conditions).
- Interpretability: The interpretability of graph embeddings can be more challenging compared to other clustering techniques, as it is often difficult to interpret the specific features or relationships each dimension captures.

Addressing these limitations is an active area of research, and researchers continue to develop new techniques and algorithms to enhance the performance and versatility of graph embedding methods to make them more applicable to life-science research questions.

Conclusion

As can be easily appreciated from the by far not exhaustive list of discussed algorithms for graph embedding in this review, there is currently not yet a gold standard for graph embedding for biological data emerging that can provide reliable data for biologists and serve as a reference point for future developments of in the field. So far, the presented applications for graph embedding on biological data have all been developed for the specific data sets at hand. All these studies have thus mainly remained theoretical, focusing

on the development of computational techniques rather than taking the interpretation of the data to the identification of novel biology or drug developments. Yet, with the ever-growing datasets available to life science researchers, the community needs novel tools to understand better the underlying biological processes. Given their nature of reducing the dimensionality of complex data, graph embedding algorithms are an exciting and novel tool for extracting novel insight from large biological datasets (Table 2). We envision that graph embedding will become an essential tool aiding hypothesis generation leading to novel biological discoveries.

Specifically, graph embedding techniques hold significant potential in various biological and biomedical research fields. In the context of the drug–disease association (DDA), disease-gene association (DGA), drug–target interaction (DTI), protein–protein interaction (PPI), and drug–drug interaction (DDI) (Table 2), graph embedding methods can provide valuable insights and aid in understanding complex relationships. By representing drugs, diseases, genes, targets, and proteins as nodes in a graph and capturing their interactions as edges, graph embedding algorithms can (i) infer novel insight into a biological system based on information about its elements (i.e., link prediction), (ii) classify the relevance of biological elements (e.g., proteins, metabolites, etc.) and their interactions within a system (i.e., node classification), and (iii) identify a phenotype or physiology of interest based on the networks formed by their elements (i.e., node clustering).

Furthermore, with the help of low-dimensional representations obtained using graph-neural networks (GNN) algorithms, it is possible to encode the underlying relationships and functional associations to find similarities between individuals sharing the same condition (e.g., graph matching or subgraph matching). These low-dimensional embeddings can then be leveraged to gain an understanding of the underlying molecular events occurring within the biological system (i.e., molecular phenotype characterization).

Hence, the ability to integrate multiple data sources, such as genomic, transcriptomic, proteomic, metabolomic, and clinical data, further enhances the predictive power and potential impact of graph embedding techniques, mainly in the field of personalized medicine, paving the way for improved disease management, identifying potential therapeutic targets, elucidating underlying molecular mechanisms, and exploring drug synergy or adverse interactions.

In conclusion, this increased predictive power gained by using graph embedding techniques on biological data will allow life-science researchers to conduct more targeted experiments by extracting novel unseen links. Developing applications will require substantial further research on the bioinformatic side to identify the most promising approaches to be applied to specific types of datasets, as well as thorough experimental validation of the generated outputs. Despite posing a challenging problem to either field, the rapid rise of AI tools in our everyday life as a researcher will certainly fuel interest in incorporating novel AI-based analysis methods on high dimensional biological data. Therefore, we anticipate that graph embedding applications will soon be invaluable in the broader life science community.

Acknowledgements
EA doctoral studies are funded by *Consejo Nacional de Ciencia, Tecnología e Innovación Tecnológica* (CONCYTEC), and *Fondo Nacional de Desarrollo Científico, Tecnológico y de Innovación Tecnológica* (FONDECYT), under contract No. 174-2020-FONDECYT "Doctoral Programs in Peruvian Universities". AI thank to "The Max Planck Partner Group" (Max Planck Institute for Chemical Ecology-Jena) for their financial support.

Author contributions
E.A.M.,R.D., C.A.B.C., and A.J.I. wrote the main manuscript text, and E.A.M. prepared figures and tables. All authors reviewed the manuscript. All authors read and approved the final manuscript.

Funding
This work was supported by*Consejo Nacional de Ciencia, Tecnología e Innovación Tecnológica(CONCYTEC),Fondo Nacional de Desarrollo Científico, Tecnológico y de Innovación Tecnológica(FONDECYT), under contract No. 174-2020-FONDECYT "Doctoral Programs in Peruvian Universities", and the Max-Planck-Gesellschaft "The Max Planck Partner Group" (Max Planck Institute for Chemical Ecology-Jena and the Pontificia Universidad Católica del Perú).*

Availability of data and materials
Not Applicable.

Declarations

Ethics approval and consent to participate
Not Applicable.

Consent for publication
Not Applicable.

Competing interests
We, the authors, declare not to have competing interests as defined by BMC or other interests that might be perceived to influence the results and/or discussion reported in this paper.

Received: 16 January 2023 Accepted: 11 December 2023
Published online: 02 January 2024

References

1. Xu M. Understanding graph embedding methods and their applications. SIAM Rev. 2021;63(4):825–53.
2. Makarov I, Kiselev D, Nikitinsky N, Subelj L. Survey on graph embeddings and their applications to machine learning problems on graphs. PeerJ Comput Sci. 2021;7:357.
3. Park J, Jo J, Yoon S. Mass spectra prediction with structural motif-based graph neural networks. arXiv preprint arXiv:2306.16085 2023.
4. Schmidt AM, Fagerer SR, Jefimovs K, Buettner F, Marro C, Siringil EC, Boehlen KL, Pabst M, Ibánez AJ. Molecular phenotypic profiling of a saccharomyces cerevisiae strain at the single-cell level. Analyst. 2014;139(22):5709–17.
5. Buettner F, Jay K, Wischnewski H, Stadelmann T, Saad S, Jefimovs K, Mansurova M, Gerez J, Azzalin CM, Dechant R, et al. Non-targeted metabolomic approach reveals two distinct types of metabolic responses to telomerase dysfunction in *S. cerevisiae*. Metabolomics. 2017;13(5):1–10.
6. Khazane A, Rider J, Serpe M, Gogoglou A, Hines K, Bruss CB, Serpe R. Deeptrax: embedding graphs of financial transactions. In: 2019 18th IEEE international conference on machine learning and applications (ICMLA). IEEE; 2019. p. 126–33.
7. Xie M, Yin H, Wang H, Xu F, Chen W, Wang S. Learning graph-based poi embedding for location-based recommendation. In: Proceedings of the 25th ACM international on conference on information and knowledge management; 2016. p. 15–24.
8. Ye Y, Hou S, Chen L, Lei J, Wan W, Wang J, Xiong Q, Shao F. Out-of-sample node representation learning for heterogeneous graph in real-time android malware detection. In: 28th International joint conference on artificial intelligence (IJCAI); 2019.
9. Li Y, Yang T. Word embedding for understanding natural language: a survey. In: Guide to big data applications. Berlin: Springer; 2018. p. 83–104.
10. Liu Y, Liu Z, Chua T-S, Sun M. Topical word embeddings. In: Twenty-ninth AAAI conference on artificial intelligence; 2015.
11. Drozd A, Gladkova A, Matsuoka S. Word embeddings, analogies, and machine learning: beyond king–man + woman = queen. In: Proceedings of Coling 2016, the 26th international conference on computational linguistics: technical papers; 2016. p. 3519–30.
12. Orkphol K, Yang W. Word sense disambiguation using cosine similarity collaborates with word2vec and wordnet. Future Internet. 2019;11(5):114.
13. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781; 2013.
14. Zhang A, Lipton ZC, Li M, Smola AJ. Dive into deep learning. arXiv preprint arXiv:2106.11342; 2021.
15. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. Adv Neural Inf Process Syst. 2013;26:66.
16. Perozzi B, Al-Rfou R, Skiena S. Deepwalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining; 2014. p. 701–10.
17. Grover A, Leskovec J. node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining; 2016. p. 855–64.

18. Dong Y, Chawla NV, Swami A. metapath2vec: scalable representation learning for heterogeneous networks. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining; 2017. p. 135–44.
19. Ribeiro LF, Saverese PH, Figueiredo DR. struc2vec: learning node representations from structural identity. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining; 2017. p. 385–94.
20. Goodrich MT, Tamassia R, Goldwasser MH. Data structures and algorithms in python. New York: Wiley; 2013.
21. Lee KD, Lee KD, Steve Hubbard SH. Data structures and algorithms with python. Berlin: Springer; 2015.
22. Goyal P, Ferrara E. Graph embedding techniques, applications, and performance: a survey. Knowl Based Syst. 2018;151:78–94.
23. Cai H, Zheng VW, Chang KC-C. A comprehensive survey of graph embedding: problems, techniques, and applications. IEEE Trans Knowl Data Eng. 2018;30(9):1616–37.
24. Aggarwal M, Murty MN. Machine learning in social networks: embedding nodes, edges, communities, and graphs. Berlin: Springer; 2020.
25. Stamile C, Aldo Marzullo ED. Graph machine learning: take graph data to the next level by applying machine learning techniques and algorithms. Packt Publishing; 2021.
26. Crichton G, Guo Y, Pyysalo S, Korhonen A. Neural networks for link prediction in realistic biomedical graphs: a multi-dimensional evaluation of graph embedding-based approaches. BMC Bioinform. 2018;19(1):1–11.
27. Xiao Z, Deng Y. Graph embedding-based novel protein interaction prediction via higher-order graph convolutional network. PLoS ONE. 2020;15(9):0238915.
28. Su X-R, You Z-H, Hu L, Huang Y-A, Wang Y, Yi H-C. An efficient computational model for large-scale prediction of protein–protein interactions based on accurate and scalable graph embedding. Front Genet. 2021;12: 635451.
29. Zhu J, Zheng Z, Yang M, Fung GPC, Huang C. Protein complexes detection based on semi-supervised network embedding model. IEEE/ACM Trans Comput Biol Bioinf. 2019;18(2):797–803.
30. Li J, Liu Y, Zhang Z, Liu B, Wang Y. Pmdne: prediction of mirna-disease association based on network embedding and network similarity analysis. BioMed Res Int. 2020;2020:66.
31. Bai T, Li Y, Wang Y, Huang L. A hybrid vae based network embedding method for biomedical relation mining. Neural Process Lett. 2021;66:1–12.
32. Luo J, Ouyang W, Shen C, Cai J. Multi-relation graph embedding for predicting mirna-target gene interactions by integrating gene sequence information. IEEE J Biomed Health Inform. 2022;6:66.
33. Basher MA, Rahman A, Hallam SJ. Leveraging heterogeneous network embedding for metabolic pathway prediction. Bioinformatics. 2021;37(6):822–9.
34. Wang M, Wang H, Liu X, Ma X, Wang B, et al. Drug-drug interaction predictions via knowledge graph and text embedding: instrument validation study. JMIR Med Inform. 2021;9(6):28277.
35. Yue X, Wang Z, Huang J, Parthasarathy S, Moosavinasab S, Huang Y, Lin SM, Zhang W, Zhang P, Sun H. Graph embedding on biomedical networks: methods, applications and evaluations. Bioinformatics. 2020;36(4):1241–51.
36. Su C, Tong J, Zhu Y, Cui P, Wang F. Network embedding in biomedical data science. Brief Bioinform. 2020;21(1):182–97.
37. Kruskal JB, Wish M. Multidimensional scaling, vol. 11. London: Sage; 1978.
38. Tenenbaum JB, Silva Vd, Langford JC. A global geometric framework for nonlinear dimensionality reduction. Science. 2000;290(5500):2319–23.
39. Roweis ST, Saul LK. Nonlinear dimensionality reduction by locally linear embedding. Science. 2000;290(5500):2323–6.
40. Belkin M, Niyogi P. Laplacian eigenmaps and spectral techniques for embedding and clustering. Adv Neural Inf Process Syst. 2001;66:14.
41. Shaw B, Jebara T. Structure preserving embedding. In: Proceedings of the 26th annual international conference on machine learning; 2009. p. 937–44.
42. Luo D, Ding CH, Nie F, Huang H. Cauchy graph embedding. In: ICML; 2011.
43. Ahmed A, Shervashidze N, Narayanamurthy S, Josifovski V, Smola AJ. Distributed large-scale natural graph factorization. In: Proceedings of the 22nd international conference on World Wide Web; 2013. p. 37–48.
44. Cao S, Lu W, Xu Q. Grarep: learning graph representations with global structural information. In: Proceedings of the 24th ACM international on conference on information and knowledge management; 2015. p. 891–900.
45. Yang C, Liu Z, Zhao D, Sun M, Chang E. Network representation learning with rich text information. In: Twenty-fourth international joint conference on artificial intelligence; 2015.
46. Ou M, Cui P, Pei J, Zhang Z, Zhu W. Asymmetric transitivity preserving graph embedding. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining; 2016. p. 1105–14.
47. Tang J, Qu M, Wang M, Zhang M, Yan J, Mei Q. Line: Large-scale information network embedding. In: Proceedings of the 24th international conference on World Wide Web; 2015. p. 1067–77.
48. Cho H, Berger B, Peng J. Diffusion component analysis: unraveling functional topology in biological networks. In: International conference on research in computational molecular biology. Berlin: Springer; 2015. p. 62–4.
49. Perozzi B, Kulkarni V, Skiena S. Walklets: multiscale graph embeddings for interpretable network classification. arXiv preprint arXiv:1605.02115:043238-23.
50. Li J, Zhu J, Zhang B. Discriminative deep random walk for network classification. In: Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers); 2016. p. 1004–13.
51. Chen H, Perozzi B, Hu Y, Skiena S. Harp: hierarchical representation learning for networks. In: Proceedings of the AAAI conference on artificial intelligence; 2018. p. 32.
52. Rozemberczki B, Sarkar R. Fast sequence-based embedding with diffusion graphs. In: International workshop on complex networks. Berlin: Springer; 2018. p. 99–107.
53. Rozemberczki B, Davies R, Sarkar R, Sutton C. Gemsec: graph embedding with self clustering. In: Proceedings of the 2019 IEEE/ACM international conference on advances in social networks analysis and mining; 2019. p. 65–72.

54. Wang D, Cui P, Zhu W. Structural deep network embedding. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining; 2016. p. 1225–34.
55. Cao S, Lu W, Xu Q. Deep neural networks for learning graph representations. In: Proceedings of the AAAI conference on artificial intelligence; 2016. p. 30.
56. Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907; 2016.
57. Kipf TN, Welling M. Variational graph auto-encoders. arXiv preprint arXiv:1611.07308; 2016.
58. Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs. Adv Neural Inf Process Syst. 2017;66:30.
59. Veličković P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y. Graph attention networks. arXiv preprint arXiv:1710.10903; 2017.
60. Wang H, Wang J, Wang J, Zhao M, Zhang W, Zhang F, Xie X, Guo M. Graphgan: graph representation learning with generative adversarial nets). arXiv; 2017;30(22):11–9.
61. Veličković P, Fedus W, Hamilton WL, Liò P, Bengio Y, Hjelm RD. Deep graph infomax. arXiv preprint arXiv:1809.10341; 2018.
62. Zeng H, Zhou H, Srivastava A, Kannan R, Prasanna V. Graphsaint: graph sampling based inductive learning method. arXiv preprint arXiv:1907.04931; 2019.
63. Lin Y-Y, Liu T-L, Chen H-T. Semantic manifold learning for image retrieval. In: Proceedings of the 13th annual ACM international conference on multimedia; 2005. p. 249–58.
64. Nickel M, Tresp V, Kriegel H-P. A three-way model for collective learning on multi-relational data. In: ICML; 2011.
65. Jenatton R, Roux N, Bordes A, Obozinski GR. A latent factor model for highly multi-relational data. Adv Neural Inf Process Syst. 2012;25:66.
66. Socher R, Chen D, Manning CD, Ng A. Reasoning with neural tensor networks for knowledge base completion. Adv Neural Inf Process Syst. 2013;26:66.
67. Yang B, Yih W.-t, He X, Gao J, Deng L. Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint arXiv:1412.6575 2014.
68. Bordes A, Glorot X, Weston J, Bengio Y. A semantic matching energy function for learning with multi-relational data. Mach Learn. 2014;94(2):233–59.
69. Dong X, Gabrilovich E, Heitz G, Horn W, Lao N, Murphy K, Strohmann T, Sun S, Zhang W. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining; 2014. p. 601–10.
70. Yang Z, Tang J, Cohen W. Multi-modal Bayesian embeddings for learning social knowledge graphs. arXiv preprint arXiv:1508.00715; 2015.
71. Nickel M, Rosasco L, Poggio T. Holographic embeddings of knowledge graphs. In: Proceedings of the AAAI conference on artificial intelligence; 2016. p. 30.
72. Ren X, He W, Qu M, Voss CR, Ji H, Han J. Label noise reduction in entity typing by heterogeneous partial-label embedding. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining; 2016. p. 1825–34.
73. Zhang D, Yin J, Zhu X, Zhang C. Homophily, structure, and content augmented network representation learning. In: 2016 IEEE 16th international conference on data mining (ICDM). IEEE; 2016. p. 609–18.
74. Pan S, Wu J, Zhu X, Zhang C, Wang Y. Tri-party deep network representation. Network. 2016;11(9):12.
75. Chen J, Zhang Q, Huang X. Incorporate group information to enhance network embedding. In: Proceedings of the 25th ACM international on conference on information and knowledge management; 2016. p. 1901–4.
76. Tu C, Zhang W, Liu Z, Sun M, et al. Max-margin deepwalk: discriminative learning of network representation. In: IJCAI, vol. 2016; 2016. p. 3889–95.
77. Yang Z, Cohen W, Salakhudinov R. Revisiting semi-supervised learning with graph embeddings. In: International conference on machine learning. PMLR; 2016. p. 40–8.
78. Chen H, Anantharam AR, Skiena S. Deepbrowse: similarity-based browsing through large lists. In: International conference on similarity search and applications. Springer; 2017. p. 300–14.
79. Bordes A, Weston J, Collobert R, Bengio Y. Learning structured embeddings of knowledge bases. In: Twenty-fifth AAAI conference on artificial intelligence; 2011.
80. Bordes A, Usunier N, Garcia-Duran A, Weston J, Yakhnenko O. Translating embeddings for modeling multi-relational data. Adv Neural Inf Process Syst. 2013;26:66.
81. Wang Z, Zhang J, Feng J, Chen Z. Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the AAAI conference on artificial intelligence; 2014. p. 28.
82. Lin Y, Liu Z, Sun M, Liu Y, Zhu X. Learning entity and relation embeddings for knowledge graph completion. In: Twenty-ninth AAAI conference on artificial intelligence; 2015.
83. Ji G, He S, Xu L, Liu K, Zhao J. Knowledge graph embedding via dynamic mapping matrix. In: Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: long papers); 2015. p. 687–96.
84. Feng J, Huang M, Wang M, Zhou M, Hao Y, Zhu X. Knowledge graph embedding by flexible translation. In: Fifteenth international conference on the principles of knowledge representation and reasoning; 2016.
85. Ji G, Liu K, He S, Zhao J. Knowledge graph completion with adaptive sparse transfer matrix. In: Thirtieth AAAI conference on artificial intelligence; 2016.
86. Chang S, Han W, Tang J, Qi G-J, Aggarwal CC, Huang TS. Heterogeneous network embedding via deep architectures. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining; 2015. p. 119–28.D
87. Chen T, Sun Y. Task-guided and path-augmented heterogeneous network embedding for author identification. In: Proceedings of the tenth ACM international conference on web search and data mining; 2017. p. 295–304.
88. Huang Z, Mamoulis N. Heterogeneous information network embedding for meta path based proximity. arXiv preprint arXiv:1701.05291; 2017.

89. Fu X, Zhang J, Meng Z, King I. Magnn: metapath aggregated graph neural network for heterogeneous graph embedding. In: Proceedings of the web conference 2020; 2020. p. 2331–41.
90. Yang L, Xiao Z, Jiang W, Wei Y, Hu Y, Wang H. Dynamic heterogeneous graph embedding using hierarchical attentions. In: European conference on information retrieval. Berlin: Springer; 2020. p. 425–32.
91. Zhou J, Liu L, Wei W, Fan J. Network representation learning: from preprocessing, feature extraction to node embedding. ACM Comput Surv. 2022;55(2):1–35.
92. Manipur I, Manzo M, Granata I, Giordano M, Maddalena L, Guarracino MR. Netpro2vec: a graph embedding framework for biomedical applications. IEEE/ACM Trans Comput Biol Bioinf. 2021;19(2):729–40.
93. Le Q, Mikolov T. Distributed representations of sentences and documents. In: International conference on machine learning. PMLR; 2014. p. 1188–96.
94. Hussein R, Yang D, Cudré-Mauroux P. Are meta-paths necessary? Revisiting heterogeneous graph embeddings. In: Proceedings of the 27th ACM international conference on information and knowledge management; 2018. p. 437–46.
95. Roy I, Velugoti VSBR, Chakrabarti S, De, A. Interpretable neural subgraph matching for graph retrieval. In: Proceedings of the AAAI conference on artificial intelligence, vol. 36; 2022. p. 8115–23.
96. Su X, Hu L, You Z, Hu P, Zhao B. Attention-based knowledge graph representation learning for predicting drug–drug interactions. Brief Bioinform. 2022;23(3):140.
97. Liang X, Li D, Song M, Madden A, Ding Y, Bu Y. Predicting biomedical relationships using the knowledge and graph embedding cascade model. PLoS ONE. 2019;14(6):0218264.
98. Zong N, Wong RSN, Yu Y, Wen A, Huang M, Li N. Drug–target prediction utilizing heterogeneous bio-linked network embeddings. Brief Bioinform. 2021;22(1):568–80.
99. Mohamed SK, Nováček V, Nounu A. Discovering protein drug targets using knowledge graph embeddings. Bioinformatics. 2020;36(2):603–10.
100. Mohamed SK, Nounu A, Nováček V. Biological applications of knowledge graph embedding models. Brief Bioinform. 2021;22(2):1679–93.
101. Chen W, Chen G, Zhao L, Chen CY-C. Predicting drug–target interactions with deep-embedding learning of graphs and sequences. J Phys Chem A. 2021;125(25):5633–42.
102. Long Y, Luo J. Association mining to identify microbe drug interactions based on heterogeneous network embedding representation. IEEE J Biomed Health Inform. 2020;25(1):266–75.
103. He M, Huang C, Liu B, Wang Y, Li J. Factor graph-aggregated heterogeneous network embedding for disease–gene association prediction. BMC Bioinform. 2021;22(1):1–15.
104. Li H, Xiao X, Wu X, Ye L, Ji G. scline: a multi-network integration framework based on network embedding for representation of single-cell rna-seq data. J Biomed Inform. 2021;122: 103899.
105. Gong M, Liu W, Xie Y, Tang Z, Xu M. Heuristic 3d interactive walk for multilayer network embedding. IEEE Trans Knowl Data Eng. 2020;6:66.
106. Ray S, Lall S, Bandyopadhyay S. A deep integrated framework for predicting sars-cov2-human protein–protein interaction. IEEE Trans Emerg Top Comput Intell. 2022;6:66.
107. Gordon DE, Jang GM, Bouhaddou M, Xu J, Obernier K, White KM, O'Meara MJ, Rezelj VV, Guo JZ, Swaney DL, et al. A sars-cov-2 protein interaction map reveals targets for drug repurposing. Nature. 2020;583(7816):459–68.
108. Dick K, Chopra A, Biggar KK, Green JR. Multi-schema computational prediction of the comprehensive sars-cov-2 vs. human interactome. PeerJ. 2021;9:11117.
109. Bakowski MA, Beutler N, Wolff KC, Kirkpatrick MG, Chen E, Nguyen T-TH, Riva L, Shaabani N, Parren M, Ricketts J, et al. Drug repurposing screens identify chemical entities for the development of Covid-19 interventions. Nat Commun. 2021;12(1):1–14.
110. Riva L, Yuan S, Yin X, Martin-Sancho L, Matsunaga N, Pache L, Burgstaller-Muehlbacher S, De Jesus PD, Teriete P, Hull MV, et al. Discovery of sars-cov-2 antiviral drugs through large-scale compound repurposing. Nature. 2020;586(7827):113–9.
111. Zitnik M, Agrawal M, Leskovec J. Modeling polypharmacy side effects with graph convolutional networks. Bioinformatics. 2018;34(13):457–66.
112. Hamilton WL. Graph representation learning. Synth Lectu Artif Intell Mach Learn. 2020;14(3):1–159.
113. Su X, Hu L, You Z, Hu P, Wang L, Zhao B. A deep learning method for repurposing antiviral drugs against new viruses via multi-view nonnegative matrix factorization and its application to sars-cov-2. Brief Bioinform. 2022;23(1):526.
114. Nelson W, Zitnik M, Wang B, Leskovec J, Goldenberg A, Sharan R. To embed or not: network embedding as a paradigm in computational biology. Front Genet. 2019;10:381.
115. Xiong Y, Guo M, Ruan L, Kong X, Tang C, Zhu Y, Wang W. Heterogeneous network embedding enabling accurate disease association predictions. BMC Med Genom. 2019;12(10):1–17.
116. Chen J, Gong Z, Mo J, Wang W, Wang C, Dong X, Liu W, Wu K. Self-training enhanced: network embedding and overlapping community detection with adversarial learning. IEEE Trans Neural Netw Learn Syst. 2021;6:66.
117. Zhang Z, Xiong H, Xu T, Qin C, Zhang L, Chen E. Complex attributed network embedding for medical complication prediction. Knowl Inf Syst. 2022;64(9):2435–56.

Publisher's Note