Bioinformatics, 37(6), 2021, 822–829
doi: 10.1093/bioinformatics/btaa906
Advance Access Publication Date: 20 October 2020
Original Paper

Systems biology

Leveraging heterogeneous network embedding for metabolic pathway prediction

Abdur Rahman M. A. Basher[1] **and Steven J. Hallam**[1,2,3,4,5,*]

[1]Graduate Program in Bioinformatics, [2]Department of Microbiology & Immunology, [3]Genome Science and Technology Program, [4]Life Sciences Institute and [5]ECOSCOPE Training Program, University of British Columbia, Vancouver, BC V6T 1Z3, Canada

*To whom correspondence should be addressed.

Associate Editor: Cowen Lenore

Received on March 24, 2020; revised on October 3, 2020; editorial decision on October 7, 2020; accepted on October 8, 2020

Abstract

Motivation: Metabolic pathway reconstruction from genomic sequence information is a key step in predicting regulatory and functional potential of cells at the individual, population and community levels of organization. Although the most common methods for metabolic pathway reconstruction are gene-centric e.g. mapping annotated proteins onto known pathways using a reference database, pathway-centric methods based on heuristics or machine learning to infer pathway presence provide a powerful engine for hypothesis generation in biological systems. Such methods rely on rule sets or rich feature information that may not be known or readily accessible.

Results: Here, we present pathway2vec, a software package consisting of six representational learning modules used to automatically generate features for pathway inference. Specifically, we build a three-layered network composed of compounds, enzymes and pathways, where nodes within a layer manifest inter-interactions and nodes between layers manifest betweenness interactions. This layered architecture captures relevant relationships used to learn a neural embedding-based low-dimensional space of metabolic features. We benchmark pathway2vec performance based on node-clustering, embedding visualization and pathway prediction using MetaCyc as a trusted source. In the pathway prediction task, results indicate that it is possible to leverage embeddings to improve prediction outcomes.

Availability and implementation: The software package and installation instructions are published on http://github.com/pathway2vec.

Contact: shallam@mail.ubc.ca

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Metabolic pathway reconstruction from genomic sequence information is a key step in predicting regulatory and functional potential of cells at the individual, population and community levels of organization (Abubucker *et al.*, 2012). Exponential advances in sequencing throughput continue to lower the cost of data generation with concomitant increases in data volume and complexity (Ansorge, 2009). Resulting datasets create new opportunities for metabolic reconstruction within biological systems that require the development of new computational tools and approaches that scale with data volume and complexity. Although the most common methods for metabolic pathway reconstruction are gene-centric e.g. mapping annotated proteins onto known pathways using a reference database based on sequence homology, heuristic or rule-based methods for pathway-centric inference including PathoLogic (Karp *et al.*, 2016) and MinPath (Ye and Doak, 2009) have become increasingly used to generate hypotheses and build quantitative models. For example, Pathologic generates pathway genome databases (PGDBs) that can be refined based on experimental validation e.g. EcoCyc (Karp *et al.*, 2018) and stored in repositories e.g. BioCyc (Caspi *et al.*, 2016a) for community access and use in flux balance analysis.

The development of accurate and flexible rule sets for pathway prediction remains a challenging enterprise informed by expert curators incorporating thermodynamic, kinetic and structural information for validation (Toubiana *et al.*, 2019). Updating these rule sets as new organisms or pathways are described and validated can be cumbersome and out of phase with current user needs. This has led to the consideration of machine-learning (ML) approaches for pathway prediction based on rich feature information. Dale *et al.* (2010) conducted a seminal study comparing the performance of Pathologic to different types of supervised ML algorithms (naive Bayes, *k*-nearest neighbors, decision trees and logistic regression), converting rules into features, defining new features and evaluating on experimentally validated pathways from six highly curated organisms in the BioCyc collection randomly divided into training and test sets. Resulting performance metrics indicated that generic ML methods equaled and, in some cases, exceeded performance of Pathologic with the benefit of probability estimation for pathway presence and increased flexibility and transparency of use.

Despite the potential benefits of adopting ML methods for pathway prediction from genomic sequence information, Pathologic remains the primary inference engine of Pathway Tools (Karp *et al.*, 2016), and alternative methods for pathway-centric inference expanding on the generic methods described above remain nascent. Several of these methods incorporate metabolite information to improve pathway inference and reaction rules to infer metabolic pathways (Carbonell *et al.*, 2018; Tabei *et al.*, 2016; Toubiana *et al.*, 2019). Other methods including BiomeNet (Shafiei *et al.*, 2014) and MetaNetSim (Jiao *et al.*, 2013) dispense with pathways all together and model reaction networks based on enzyme abundance information. Recently, M.A. Basher *et al.* (2020) implemented a multi-label classification approach to predict metabolic pathways for organismal and multi-organismal genomes e.g. microbiomes. One of the primary challenges encountered in developing mlLGPR related to engineering reliable features representing heterogeneous and degenerate functions within multi-organismal datasets (Lawson *et al.*, 2019).

Advances in representational learning have led to the development of scalable methods for engineering features from graphical networks e.g. networks composed of multiple nodes including information systems or social networks (Dong *et al.*, 2017; Grover and Leskovec, 2016; Perozzi *et al.*, 2014). These approaches learn feature vectors for nodes in a network by solving an optimization problem in an unsupervised manner, using random walks followed by Skip-Gram extraction of low-dimensional latent continuous features, known as *embeddings* (Mikolov *et al.*, 2013). Here, we present pathway2vec, a software package incorporating multiple *random walks-based* algorithms for representational learning used to automatically generate feature representations of metabolic pathways, which are decomposed into three interacting layers: compounds, enzymes and pathways, where each layer consists of associated nodes. A Skip-Gram model is applied to extract embeddings for each node, encoding smooth decision boundaries between groups of nodes in that graph. Nodes within a layer manifest inter-interactions and nodes between layers manifest betweenness interactions resulting in a multi-layer heterogeneous information network (Shi *et al.*, 2017). This layered architecture captures relevant relationships used to learn a neural embedding-based low-dimensional space of metabolic features (Fig. 1).

In addition to implementing several published random walk methods, we developed unit-cir̲cle based ju̲mp and s̲tay random walk (RUST), adopting a unit-circle equation to sample node pairs that generalize previous random walk methods (Dong *et al.*, 2017; Grover and Leskovec, 2016; Hussein *et al.*, 2018). The modules in pathway2vec were benchmarked based on node-clustering, embedding visualization and pathway prediction. In the case of pathway prediction, pathway2vec modules provided a viable adjunct or alternative to manually curated feature sets used in ML-based metabolic pathway reconstruction from genomic sequence information at different levels of complexity and completion. The distinctness of this work lies in decomposing pathway into components, so various graph-learning methods can be applied to automatically extract semantic features of metabolic pathways, and to incorporate the learned embeddings for pathway inference.

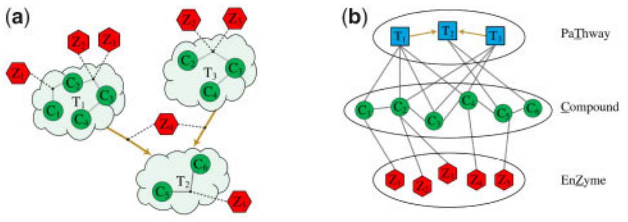

Fig. 1. Three interacting metabolic pathways (**a**), depicted as a cloud glyph, where each pathway is comprised of compounds (green) and enzymes (red). Interacting compound, enzyme and pathway components are transformed into a multi-layer heterogeneous information network (**b**)

2 Definitions and problem statement

In this section, we formulate the problem of metabolic features engineering using a heterogeneous information network. Throughout the article, all vectors are column vectors denoted by boldface lowercase letters (e.g. $\mathbf{x}$) while matrices are represented by boldface uppercase letters (e.g. $\mathbf{X}$). The $\mathbf{X}_i$ matrix denotes the i-th row of $\mathbf{X}$ and $\mathbf{X}_{i,j}$ denotes the (i,j)-th element of $\mathbf{X}$. A subscript character to a vector, $\mathbf{x}_i$, denotes an i-th cell of $\mathbf{x}$. Occasional superscript, $\mathbf{x}^{(i)}$, suggests an index to a sample, position, or current epoch during learning period. We use calligraphic letters to represent sets (e.g. $\mathcal{E}$) while we use the notation $|.|$ to denote the cardinality of a given set.

Definition 2.1.**Multi-label Pathway Dataset** (M.A.Basher *et al.*, 2020). A pathway dataset is characterized by $\mathcal{S} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) : 1 < i \leqslant n\}$ consisting of n examples, where $\mathbf{x}^{(i)}$ is a vector indicating abundance information for each enzymatic reaction denoted by z, which is an element of a set $\mathcal{Z} = \{z_1, z_2, \ldots, z_r\}$, having r possible reactions. The abundance of an enzymatic reaction for a given example i, say $z_l^{(i)}$, is defined as $a_l^{(i)} (\in \mathbb{R}_{\geq 0})$. The class label $\mathbf{y}^{(i)} = [y_1^{(i)}, \ldots, y_t^{(i)}] \in \{-1, +1\}^t$ is a pathway label vector of size t representing the total number of pathways obtained from a trusted source of experimentally validated metabolic pathways $\mathcal{Y}$. The matrix form of $\mathbf{x}^{(i)}$ and $\mathbf{y}^{(i)}$ are symbolized as $\mathbf{X}$ and $\mathbf{Y}$, respectively.

Both $\mathcal{Z}$ and $\mathcal{Y}$ are derived from trusted sources, such as KEGG (Kanehisa *et al.*, 2017) or MetaCyc (Caspi *et al.*, 2016b). We assume that there is a numerical representation behind every instance and label.

The pathway inference task can be formulated as retrieving a set of pathway labels for an example i given features learned according to a *heterogeneous information network* defined as:

Definition 2.2.**Heterogeneous Information Network.** A heterogeneous information network is defined as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ denote to the set of nodes and edges (either directed or undirected), respectively (Sun *et al.*, 2011). Each $v \in \mathcal{V}$ is associated with an object type mapping function $\phi(v) : \mathcal{V} \rightarrow \mathcal{O}$, where $\mathcal{O}$ represents a set of object types. Each edge $e \in \mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ includes multiple types of links, and is associated with a link type mapping function $\phi(e) : \mathcal{E} \rightarrow \mathcal{R}$, where $\mathcal{R}$ represents a set of relation types. In particular, when $|\mathcal{O}| + |\mathcal{R}| > 2$, the graph is referred to as a heterogeneous information network.

In heterogeneous information networks, both object types and relationship types are explicitly segregated. For the undirected edges, notice that if a relation exists from a type $O_i (\in \mathcal{O})$ to a type $O_j (\in \mathcal{O})$, denoted as $O_i R O_j$ and $R \in \mathcal{R}$, the inverse relation R^{-1} holds naturally for $O_j R^{-1} O_i$. However, in many circumstances, R and its inverse R^{-1} are not equal, unless the two objects are in the same domain, and R is symmetric. In addition, the network may be weighted where each edge $e_{i,j}$, of nodes i and j, is associated with a weight of type R. The linkage type of an edge automatically defines the node types of its end points. The graph articulated in this article is considered directed and weighted (in some cases), but for simplification is converted to an undirected network by simply treating edges as symmetric links.

Example 2.1.MetaCyc can be abstracted as a heterogeneous information network, in Figure 1b, which contains three types of objects, namely compounds (C), enzymes (Z) and pathways (T). There exist different types of links between objects representing semantic relationships e.g. '*composed* of' and '*involved in*', relationships between pathways and compounds or relations between enzymes and compounds e.g. '*transform*' and '*transformed by*'. An enzyme may be mapped to a numerical category, known as an enzyme commission number (EC) based on the chemical reaction it catalyzes.

Two objects within heterogeneous information networks describe meta-level relationships referred to as *meta-paths* (Sun *et al.*, 2011).

Definition 2.3.**Meta-Path.** A meta-path $P \in \mathcal{P}$ is a path over $\mathcal{G}$ in the form of $O_1 R_1 O_2 R_2 O_i R_k \ldots R_j O_{j+1}$, which defines an aggregation of relationships $U = R_1 {}^\circ R_2 {}^\circ \ldots {}^\circ R_j$ between type O_1 and O_{j+1}, where $^\circ$ denotes the composition operator on relationships and $O_i \in \mathcal{O}$ and $R_k \in \mathcal{R}$ are object and relation type, respectively.

Example 2.2.MetaCyc contains multiple meta-paths conveying different semantics. For example, a meta-path 'ZCZ' represents the co-catalyst relationships on a compound (C) between two enzymatic reactions (Z), and 'ZCTCZ' may indicate a meta-path that requires two enzymatic reactions (Z) transforming two compounds (C) within a pathway (T). Another important meta-path to consider is 'CZC', which implies '$C + Z \Rightarrow C$' transformation relationship.

Problem Statement. **Metabolic Pathway Prediction.** Given three inputs: (i) a heterogeneous information network $\mathcal{G}$, (ii) a dataset $\mathcal{S}$ and (iii) an optional set of meta-paths $\mathcal{P}$, the goal is to automatically resolve node embeddings such that leveraging the features will effectively improve pathway prediction for a hitherto unseen instance $\mathbf{x}^*$.

3 The pathway2vec framework

The pathway2vec framework is composed of five modules: (i) node2vec (Grover and Leskovec, 2016), (ii) metapath2vec (Dong *et al.*, 2017), (iii) metapath2vec++ (Dong *et al.*, 2017), (iv) JUST (Hussein *et al.*, 2018) and (v) RUST (this work), where each module contains a random walk modeling and node representation step. A graphical representation of the pathway2vec framework is depicted in Figure 2.

C1. Random Walks. In this step, a sequence of random walks over an input graph (whether heterogeneous or homogeneous) is generated based on the selected model (see Section 3.1).

C2. Learning Node Representation. Resulting walks are fed into the Skip-Gram model to learn node embeddings (Dong *et al.*, 2017; Fu *et al.*, 2017; Grover and Leskovec, 2016; Mikolov *et al.*, 2013). An embedding is a low-dimensional latent continuous feature for each node in $\mathcal{G}$, which encodes smooth decision boundaries between groups or communities within a graph. Details are provided in Section 3.2.

3.1 Random walks

To capture meaningful graph relationships, existing techniques, such as DeepWalk (Perozzi *et al.*, 2014), design simple but effective algorithms based on random walks for representational learning of features. However, DeepWalk does not address in-depth and in-breadth graph exploration. Therefore, node2vec (Grover and Leskovec, 2016) was developed to traverse local and global graph structures based on the principles of: (i) homophily (Fortunato, 2010; Newman, 2006) where interconnected nodes form a community of correlated attributes and (ii) structural equivalence (Henderson *et al.*, 2012), where nodes having similar structural roles in a graph should be close to one another. node2vec simulates a second-order random walk, where the next node is sampled conditioned on the previous and the current node in a walk. For this, two hyperparameters are adjusted, $s \in \mathbb{R}_{>0}$ that extracts local information of a graph, and $h \in \mathbb{R}_{>0}$ that enables local and global traversals by moving deep in a graph or walking within the vicinity of the current node. This method is illustrated in Figure 2b top.

First-order and second-order random walks were initially proposed for homogeneous graphs, but can be readily extended to heterogeneous information networks. Sun *et al.* (2011) have observed that random walks can suffer from implicit bias due to initial node selection or the presence of a small set of dominant node types skewing results toward a subset of interconnected nodes. metapath2vec was developed (Dong *et al.*, 2017) to resolve implicit bias in graph traversal to characterize semantic associations embodied between different types of nodes according to a certain path definition. This method is illustrated in Figure 2b bottom.

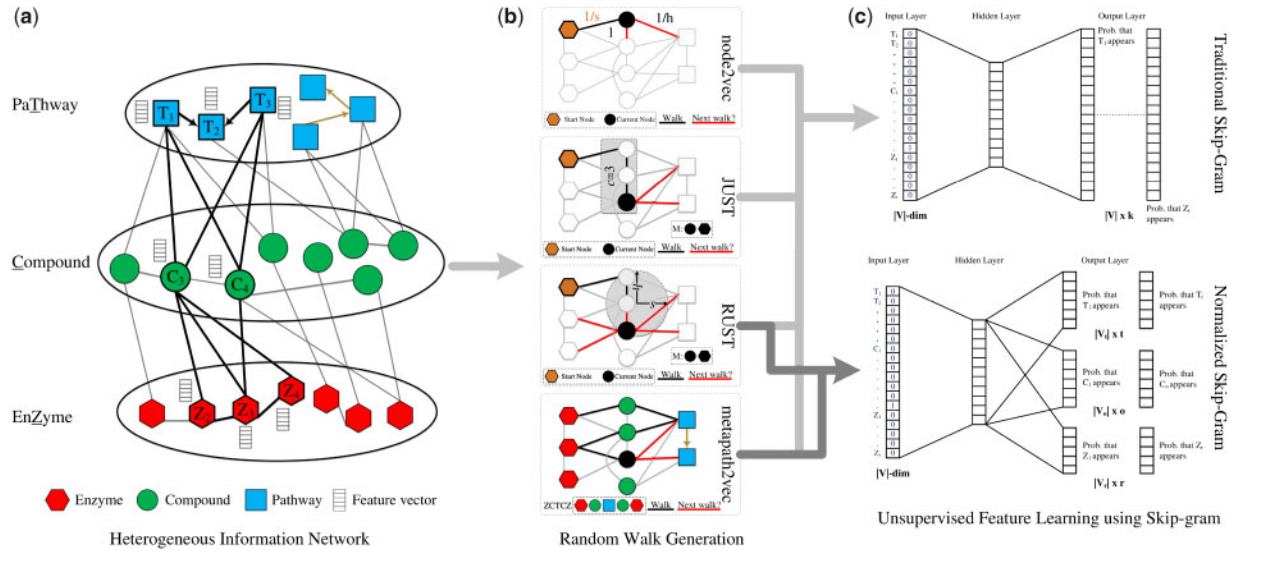

Fig. 2. Graphical representation of pathway2vec framework. Main components: (**a**) a multi-layer heterogeneous information network composed from MetaCyc, showing meta-level interactions among compounds, enzymes and pathways, (**b**) four random walks and (**c**) two representational learning models: traditional Skip-Gram (top) and Skip-Gram by normalizing domain types (bottom). In the subfigure (a), the highlighted network neighbors of T_1 (nitrifier denitrification) indicate this pathway interacts directly with T_2 [nitrogen fixation I (ferredoxin)] and indirectly to T_3 [nitrate reduction I (denitrification)] by second-order with relationships to several compounds, including nitric oxide (C_3) and nitrite (C_4) converted by enzymes represented by the EC numbers (Z_2: EC 1.7.2.6, Z_3: EC 1.7.2.1 and Z_4: EC 1.7.2.5). The black colored nodes in subfigure (b) indicate the current position of the walkers and red links suggest the next possible nodes to sample while black links indicate route taken by a walker to reach the current node. node2vec is parameterized by local search s and in–out h hyperparameters. These two hyperparameters constitute a unit circle, i.e. $h^2 + s^2 = 1$, for RUST. M stores previously visited node types, which is 2 and only applied for JUST and RUST. c is number of nodes of the same domain type as the current node, which is 3 and is associated with JUST. For metapath2vec, a walker requires a prespecified scheme, which is set to 'ZCTCZ'. The normalized Skip-Gram in the subfigure (c) bottom is simply trained based on the domain type, in contrast to the traditional Skip-Gram model. More information related to both learning strategies is provided in Section 3.2. Zoom for readability

metapath2vec overcomes the limitation of node2vec by enabling to extract semantical representations over heterogeneous graph. However, the use of meta-paths requires either prior domain-specific knowledge to recover semantic associations of HIN according to a certain path definition. As a result, groups of vertices with the heterogeneous information network may not be visited or revisited multiple times. This limitation was partially addressed by leveraging multiple path schemes (Fu *et al.*, 2017) to guide random walks based on a meta-path length parameter. Hussein *et al.* (2018) developed the Jump and Stay (JUST) heterogeneous graph embedding method using random walks as an alternative to meta-paths. JUST randomly selects the next node in a walk from either the same node type or from different node types using an exponential decay function and a tuning parameter based on two history records: (i) c corresponding the number of nodes consecutively visited in the same domain as the current node and (ii) a queue M of size m storing the previously visited node types. This method is illustrated in Figure 2b second from top.

However, in order to balance the node distribution over multiple node types, JUST constrains the number of memorized domains m to be within the range of $[1, |\mathcal{O}| - 1] \in \mathbb{Z}_{>1}$. This can misrepresent graph structure in two ways: (i) explorations within domain because the last visited consecutive c nodes may enforce sampling from another domain, or (ii) jumping deep toward nodes from other domains because M is constrained. To alleviate these problems, we develop a novel random walk algorithm, RUST, adopting a unit-circle equation to sample node pairs that generalize previous representational learning methods, as illustrated in Figure 2b second from bottom. The two hyperparameters s and h constitute a unit circle, i.e. $h^2 + s^2 = 1$, where $h \in [0, 1]$ indicates how much exploration is needed within a domain while $s \in [0, 1]$ defines the in-depth search toward other domains such that $s > h$ encourages the walk to explore more domains and vice versa. Consequently, RUST blends both semantic associations and local/global structural information for generating walks without restricting domain size m in M.

To better illustrate the effect of s and h on RUST, consider an example in Figure 3, where the walkers in JUST and RUST are currently stationed at C_3 of compound type. While JUST enforces its walker to jump toward pathway domain, because of the combined effect of c that holds three consecutive nodes of compound type and M that is currently storing EC and compound types, RUST may prefer returning to C_2 (no links exist to C_4) than jumping to T_1 or T_2. This is because $s < h$ promotes exploration within the same domain as C_3. If, however, $s > h$ then RUST will perform in-depth search by selecting a node of type pathway. For formal definitions about the discussed random walks, see Supplementary Section S1.

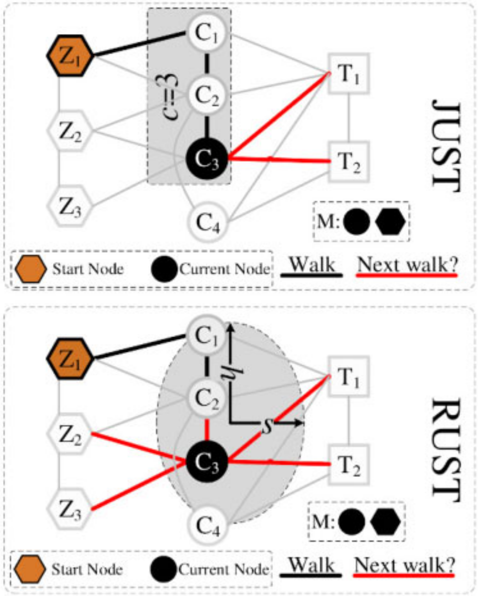

Fig. 3. An illustrative example showing the selection of the next node for both JUST and RUST on HIN extracted from MetaCyc. The walker is currently stationed at C_3 arriving from node C_2 (indicated by black colored link), where M stores two previously visited node types and c (for JUST) holds three consecutive nodes that are of the same domain as C_3. As can be seen JUST would prefer selecting the next node of type pathway while RUST may prefer returning to C_2 than jumping to T_1 or T_2, as indicated by red edges, because $s < h$ represented by an ellipsis glyph

3.2 Learning latent embedding in graph

Random walks $\mathcal{W}$ generated using node2vec, metapath2vec, JUST and RUST are fed into the Skip-Gram model to learn node embeddings (Mikolov *et al.*, 2013). The Skip-Gram model exploits context information defined as a fixed number of nodes surrounding a target node. The model attempts to maximize co-occurrence probability among a pair of nodes identified within a given window of size q in $\mathcal{W}$ based on log-likelihood:

$$\sum_{l \in \mathcal{W}} \sum_{j \in l} \sum_{-q \leq k \leq q, j \neq 0} \log p(v^{j+k} | v^j), \tag{1}$$

where $v^{j-c}, \ldots, v^{j+c}$ are the context neighbor nodes of node v^j and $p(v^{j+i}|v^j)$ defines the conditional probability of having context nodes given the node v^j. The $p(v^{j+k}|v^j)$ is the commonly used softmax function, i.e $= \frac{e^{\mathbf{D}_{v^{j+k}} . \mathbf{D}_{v^j}}}{\sum_{i \in \mathcal{V}} e^{\mathbf{D}_{v^i} . \mathbf{D}_{v^j}}}$, where $\mathbf{D} \in \mathbb{R}^{|V| \times d}$ stores the embeddings of all nodes and $\mathbf{D}_v$ is the v-th row corresponding to the embedding vector for node v. In practice, the vocabulary of nodes may be very large, which intensifies the computation of $p(v^{j+k}|v^j)$. The Skip-Gram model uses negative sampling, which randomly selects a small set of nodes N that are not in the context to reduce computational complexity. This idea represented in updated Equation (1) is implemented in node2vec, metapath2vec, JUST and RUST according to:

$$\sum_{l \in \mathcal{W}} \sum_{j \in l} \sum_{-q \leq k \leq q, j \neq 0} \left(\log \sigma(\mathbf{D}_{v^{j+k}} . \mathbf{D}_{v^j}) + \sum_{u \in N \wedge u \notin \mathcal{N}(j)} \mathbb{E}_{v^u} [\log p(v^u | v^j)] \right), \tag{2}$$

where $\sigma(v) = \frac{1}{1+e^{-v}}$ is the sigmoid function.

In addition to the equation above, Dong and colleagues proposed a normalized version of metapath2vec, called metapath2vec++, where the domain type of the context node is considered in calculating the probability $p(v^{j+k}|v^j)$, resulting in the following objective formula:

$$\begin{aligned} \sum_{l \in \mathcal{W}} \sum_{j \in l} \sum_{-q \leq k \leq q, j \neq 0} & \left(\log \sigma(\mathbf{D}_{v^{j+k}} . \mathbf{D}_{v^j}) \right. \\ & + \left. \sum_{u \in N \wedge u \notin \mathcal{N}(j) \wedge \phi(v^u) = \phi(v^{j+k})} \mathbb{E}_{v^u} [\log p(v^u | v^j)] \right), \end{aligned} \tag{3}$$

where $\phi(v^u) = \phi(v^{j+k})$ suggests that the negative nodes are of the same type as the context node $\phi(v^{j+k})$. The above formula is also applied for RUST, and we refer it to RUST-norm. Through iterative update over all the context nodes, whether using Equation (2) or (3), for each walk in $\mathcal{W}$, the learned features are expected to capture semantic and structural contents of a graph useful for pathway inference.

4 Predicting pathways

For pathway inference, the learned EC embedding vectors are concatenated into each example i according to:

$$\tilde{\mathbf{x}}^{(i)} = \mathbf{x}^{(i)} \oplus \frac{1}{r} \mathbf{x}^{(i)} \mathbf{D}_{v:v \in \mathcal{Z}}, \tag{4}$$

where $\oplus$ denotes the vector concatenation operation, $\mathbf{D} \in \mathbb{R}^{|V| \times d}$ stores the embeddings of all nodes and $\mathbf{D}_{v:v \in \mathcal{Z}}$ indicates feature vectors for r enzymatic reactions. By incorporating enzymatic reaction features into $\mathbf{x}^{(i)}$, the dimension size is extended to $r + d$, where r is the enzyme vector size while d corresponds to embeddings size. This modified version of $\mathbf{x}^{(i)}$ is denoted by $\tilde{\mathbf{x}}^{(i)}$, which then can be used by an appropriate ML algorithm, such as mlLGPR (M.A.Basher *et al.*, 2020), to train and infer a set of metabolic pathways from enzymatic reactions.

5 Experimental setup

In this section, we explain the experimental settings and outline materials used to evaluate the performance of pathway2vec modules that were written in Python v3 and trained using tensorflow v1.10 (Abadi *et al.*, 2016). Unless otherwise specified all tests were conducted on a Linux server using 10 cores of Intel Xeon CPU E5-2650.

5.1 Preprocessing MetaCyc

We constructed three hierarchical layers of HIN using MetaCyc v21 (Caspi *et al.*, 2016b), according to: EC (bottom-layer), compound (mid-layer) and pathway (top-layer) as in Figure 2a. Relationships among these layers establish inter-interactions and betweenness interactions. Three inter-interactions were built: (i) ECs interactions that were collected-based shared metabolites, e.g. if a compound is engaged in two ECs then the two ECs were considered connected; (ii) compounds interactions that were processed based on shared reactions, e.g. if any two compounds constituting substrate and product of an engaged enzymatic reaction they would be linked; and (iii) pathways interactions that were constructed based on shared metabolites, e.g. if any product in one pathway is being consumed by another then these two pathways were linked. With regard to betweenness interactions, we considered two forms: (i) EC-compound interaction if any enzyme (represented by an EC number) engages in any compound then nodes of both types were linked and (ii) compound-pathway interaction if any compound involves in any pathway then those nodes were considered related. After building multi-layer HIN, we applied different configurations, as summarized in Table 1, to explore relationship between different graph types and the quality of generated walks and embeddings.

5.2 Parameter settings

Parameterization for the other random walk methods can be found in Dong *et al.* (2017), Grover and Leskovec (2016) and Hussein *et al.* (2018). For training, we randomly initialized model parameters with a truncated Gaussian distribution, and set the learning rate to 0.01, the batch size to 100 and the number of epochs to 10. Unless otherwise indicated, for each module, the number of sampled path instances is $K = 100$, the walk length is $l = 100$, the embedding dimension size is $d = 128$, the neighborhood size is 5, the size of negative samples is 5 and the number of memorized domain m for JUST and RUST are 2 and 3, respectively. The explore and the in–out hyperparameters for node2vec and RUST are $h = 0.7$ (or $h = 0.55$) and $s = 0.7$ (or $s = 0.84$), respectively, using the uec configuration. For metapath2vec and metapath2vec++, we applied the meta-path scheme 'ZCTCZ' to guide random walks. For brevity, we denote node2vec, metapath2vec, metapath2vec++, JUST, RUST and RUST-norm as n2v, m2v, cm2v, jt, rt and crt, respectively.

Table 1. Different configurations of compound, enzyme (EC) and pathway objects extracted from the MetaCyc database: (i) full content (MetaCyc), (ii) reduced content based on trimming nodes below two links (MetaCyc r), (iii) links among enzymatic reactions are removed, following graph independence assumption (MetaCyc uec) and (iv) combination of unconnected enzymatic reactions and trimmed nodes (MetaCyc uec + r)

Database	# EC	# Compound	# Pathway	$\|\mathcal{V}\|$	$\|\mathcal{E}\|$
MetaCyc	6378	13 689	2526	22 593	37 631
MetaCyc (r)	3606	6469	2467	12 542	37 631
MetaCyc (uec)	6378	13 689	2526	22 593	33 353
MetaCyc (uec + r)	3229	6469	2467	12 165	33 353

6 Experimental results and discussion

In this section, we first evaluate parameter sensitivity of RUST prior to benchmarking the four random walk algorithms, jointly with the two learning methods, based on node-clustering, embedding visualization and pathway prediction.

6.1 Parameter sensitivity of RUST

6.1.1 Experimental setup

In this section, the effect of different hyperparameter settings in RUST on the quality of learned nodes embeddings is described. Since the hyperparameter space involved in RUST, is infinite, exhaustive searches for optimal settings are prohibitive. Therefore, settings were sub-selected to determine RUST performance. Specifically, the effects of the dimensions $d \in \{30, 50, 80, 100, 128, 150\}$, the neighborhood size $q \in \{3, 5, 7, 9\}$, the memorized domains $m \in \{3, 5, 7\}$ and the two hyperparameters s and h ($\in \{0.55, 0.71, 0.84\}$) were evaluated based on Normalized Mutual Information (NMI) scores, after 10 trials. The NMI produces scores between 0, indicating no mutual information exists, and 1, indicating node clusters (feature groups) are perfectly correlated based on class information: enzyme, compound and pathway. Clustering was performed using the k-means algorithm (Arthur and Vassilvitskii, 2007) to group data based on the learned representations from RUST as described in Dong *et al.* (2017) and Hussein *et al.* (2018). Random walks $\mathcal{W}$ were generated using MetaCyc with uec option for RUST test parameters.

6.1.2 Experimental results

Supplementary Figure S1a indicates that RUST performance tends to saturate when the memorized domains are concentrated around $m = 5$ and $h = 0.55$, indicating a preference to explore more domain types. By fixing $m = 3$ and $h = 0.55$, the optimal results of NMI score w.r.t. the number of embedding dimensionality was found to be at 80 and 128 (Supplementary Fig. S1b). Beyond this value RUST performance deteriorated. A similar trend was also observed when the context neighborhood size was increased beyond $q > 5$ (Supplementary Fig. S1c). Based on these observations, the following settings $m = 3$, $h = 0.55$, $d = 80$ or $d = 128$ and $q = 5$ provide the most efficient and accurate clustering outcomes using MetaCyc with uec option. For comparative purposes, we set $d = 128$.

6.2 Node clustering

6.2.1 Experimental setup

The performance of different random walk methods was tested in relation to node-clustering using NMI after 10 trials and the hyperparameters described above on all MetaCyc graph types depicted in Table 1. Clustering was performed using the k-means algorithm to group homogeneous nodes based on the embeddings learned by each method.

6.2.2 Experimental results

Supplementary Figure S2 indicates node-clustering results for node2vec, metapath2vec, JUST and RUST. node2vec, JUST and RUST exhibited similar performance across all configurations, indicating that these methods are less likely to extract semantic knowledge, characterizing node domains, from MetaCyc. However, RUST performed optimally better than node2vec and JUST in learning representations. In the case of metapath2vec, the random walk follows a predefined meta-path scheme, capturing the necessary relational knowledge for defining node types. For example, *nitrogenase* (EC-1.18.6.1), which reduces *nitrogen* gas into *ammonium*, is exclusively linked to the *nitrogen fixation I (ferredoxin)* pathway (Eady, 1996).

Without a predefined relation, a walker may explore more local/global structure of $\mathcal{G}$, hence, become less efficient in exploiting relations between these two nodes. Among the four walks, only metapath2vec is able to accurately group those nodes, according to their classes. Despite the advantages of metapath2vec, it is biased to a scheme, as described in Hussein *et al.* (2018), which is explicitly observed for the case of 'uec+r' (Supplementary Fig. S2d). Under these conditions, both isolated nodes and links among ECs are discarded, resulting in a reduced number of nodes that are more easily traversed by a meta-path walker. metapath2vec++ exhibited trends similar to metapath2vec because they share the same walks. However, metapath2vec++ is trained using normalized Skip-Gram. Therefore, it is expected to achieve good NMI scores, yielding over 0.41 on uec+full content (in Supplementary Fig. S3), which is also similar to RUST-norm NMI score (~0.38). This is interesting because RUST-norm employs RUST-based walks, but the embeddings are learned using normalized Skip-Gram.

Taken together, these results indicate that node2vec, JUST and RUST-based walks are effective for analyzing graph structure while metapath2vec can learn good embeddings. However, RUST strikes a balance between the two proprieties through proper adjustments of m and the two unit-circle hyperparameters. Regarding the MetaCyc type, we recommend 'uec' because the associations among ECs are captured at the pathway level. The trimmed graph is contraindicated, because it eliminates many isolated, but important pathways and ECs.

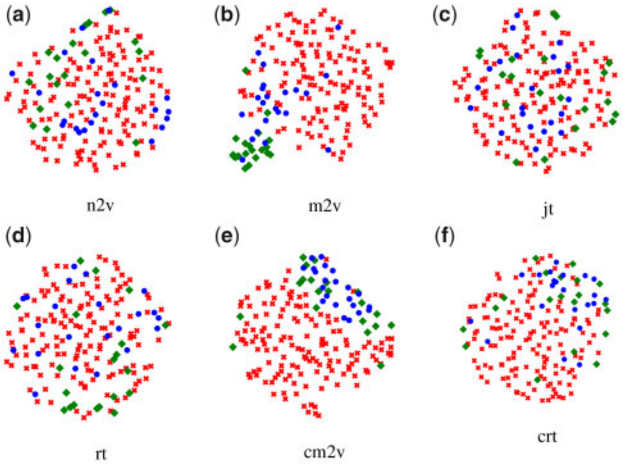

Fig. 4. The 2D UMAP projections of the 128-dimension embeddings, trained under uec+full setting depicting 185 nodes related to nitrogen metabolism. Node color indicates the category of the node type, where red indicates enzymatic reactions, green indicates compounds and blue is reserved for metabolic pathways

6.3 Manifold visualization

6.3.1 Experimental setup

In this section, learned high dimensional embeddings were visualized by projecting them onto a 2D space using two case studies. The first case examines the quality of learned nodes embeddings according to the generated random walks an approach commonly sought in most graph-learning embedding techniques (Grover and Leskovec, 2016; Wang *et al.*, 2016). We posit that a good representational learning method defines clear boundaries for nodes of the same type. For illustrative purposes, nodes corresponding to nitrogen metabolism were selected. The second case examines the limitations of meta-path-based random walks, extending our discussions in Section 6.2. For illustrative purposes, we focus on the pathway layer in Figure 2a and consider representation of pathways having no enzymatic reactions. For visualization, we use UMAP, a.k.a. uniform manifold approximation and projection (McInnes *et al.*, 2018) using 1000 epochs with the remaining settings set to default values.

6.3.2 Experimental results

Figure 4 visualizes 2D UMAP projections of the 128-dimension embeddings, trained under uec+full setting depicting 185 nodes related to nitrogen metabolism in MetaCyc. Each point denotes a node in HIN and each color indicates the node type. node2vec (Fig. 4a), JUST (Fig. 4c) and RUST (Fig. 4d) appear to be less than optimal in extracting walks that preserve three layer relational knowledge e.g. nodes belonging to different types form unclear boundaries and diffuse clusters. In the cases of metapath2vec (Fig. 4b), metapath2vec++ (Fig. 4f) and RUST-norm (Fig. 4f), nodes of the same color are more optimally portrayed. In the second use case, 80 pathways were identified, having no enzymatic reactions, with their 109 pathway neighbors, as shown in Supplementary Figure S4a. From Supplementary Figure S4, we observe that, in contrast to node2vec, JUST, RUST and RUST-norm, pathway nodes are skewed incorrectly in both metapath2vec and metapath2vec++ and with lesser degree. This demonstrates the rigidness of meta-path-based methods that follow a defined scheme that limits their capacity to exploit local structure in learning embeddings. Interestingly, RUST-norm, based on RUST walks, is the only method that combines structural and semantic information as indicated in Supplementary Figure S4g and f, respectively. Taken together, these results indicate that RUST-based walks with training using Equation (3) provide efficient embeddings, consistent with node-clustering observations.

6.4 Metabolic pathway prediction

6.4.1 Experimental setup

In this section, the effectiveness of the learned embeddings from pathway2vec modules is determined across different pathway inference methods including MinPath v1.2 (Ye and Doak, 2009), PathoLogic v21 (Karp *et al.*, 2016) and mlLGPR-elastic net (EN) (M.A.Basher *et al.*, 2020). In contrast to previous multi-label classification methods (Grover and Leskovec, 2016; Hussein *et al.*, 2018;

Table 2. Micro *F*1 scores of each comparing algorithm on six benchmark datasets

Methods	Micro *F*1 score ↑					
	EcoCyc	HumanCyc	AraCyc	YeastCyc	LeishCyc	TrypanoCyc
PathoLogic	0.7631	0.7460	0.7093	**0.7890**	0.6109	0.6447
MinPath	0.5161	0.4589	0.5489	0.4221	0.2990	0.3511
mlLGPR	0.7275	**0.7468**	**0.7343**	0.7392	**0.6220**	**0.6768**
mlLGPR+n2v	0.7614	0.3857	0.3938	0.4457	0.4780	0.4548
mlLGPR+m2v	0.7638	0.3883	0.3768	0.4642	0.4851	0.4293
mlLGPR+cm2v	0.7508	0.3783	0.3939	0.4598	0.4700	0.4697
mlLGPR+jt	0.7640	0.3783	0.3860	0.4726	0.4528	0.4515
mlLGPR+rt	0.7651	0.4076	0.3883	0.4633	0.4857	0.4680
mlLGPR+crt	0.7682	0.3654	0.4052	0.4451	0.4585	0.4653

Perozzi *et al.*, 2014), where the goal is to predict the most probable label set for nodes, we leverage the learned vectors and the multi-label dataset, according to Equation (4). Pathway prediction with mlLGPR-EN used the default hyperparameter settings, after concatenating features from each learning method, to train on BioCyc [v20.5 tier (T) T2 & T3] (Caspi *et al.*, 2016a) consisting of 9255 PGDBs with 1463 distinct pathway labels (see Supplementary Section S5). Results are reported on T1 golden datasets including EcoCyc, HumanCyc, AraCyc, YeastCyc, LeishCyc and TrypanoCyc. Four evaluation metrics are used to report performance scores after three repeated trials: *Hamming loss*, *micro precision*, *micro recall* and *micro F1 score*.

6.4.2 Experimental results

Table 2 shows micro $F1$ scores for each pathway predictor. Numbers in boldface represent the best performance score in each column while the underlined text indicates the best performance among the embedding methods. From the results, all variations of embedding methods tended to perform better than MinPath across the four T1 golden datasets (EcoCyc, YeastCyc, LeishCyc and TrypanoCyc). With the exception of EcoCyc, the performance of embeddings resulted in less optimal micro $F1$ scores than PathoLogic or mlLGPR. In the case of mlLGPR, embeddings were trained on <1470 pathways, potentially obscuring the actual benefits of the learned features. Taken together, different pathway2vec modules performed similar to one another indicating that embeddings are potential alternatives to the pathway and reaction evidence features used in M.A.Basher *et al.* (2020). Since RUST-norm is based on RUST walks that perform local and global graph structure exploration (Section 6.2) while generating meaningful semantic representation (Section 6.3), we suggest that users adopt RUST-norm. Full results are provided in Supplementary Section S6.

7 Conclusion

We have developed the pathway2vec package for learning features relevant to metabolic pathway prediction from genomic sequence information. The software package consists of six representational learning modules used to automatically generate features for pathway inference. Metabolic feature representations were decomposed into three interacting layers: compounds, enzymes and pathways, where each layer consists of associated nodes. A Skip-Gram model was applied to extract embeddings for each node encoding smooth decision boundaries between groups of nodes in a graph resulting in a multi-layer heterogeneous information network for metabolic interactions within and between layers. Three extensive empirical studies were conducted to benchmark pathway2vec, indicating that the representational learning approach is a promising adjunct or alternative to features engineering based on manual curation. At the same time, we introduced RUST, a novel and flexible random walk method that uses unit-circle and domain size hyperparameters to exploit local/global structure while absorbing semantic information from both homogeneous and heterogeneous graphs. Looking forward, we intend to leverage embeddings and graph structure on more complex community-level metabolic pathway prediction problems. Because random walk-based methods depend on many hyperparameters (e.g. the length of a random walk) that must be tuned, and many walks that must be generated, we are exploring alternative graph convolutional neural networks to reduce computational complexity. Such methods aggregate feature information based on node co-occurrences patterns automatically without dependence on hyperparameter settings (Abu-El-Haija *et al.*, 2018; Cohen *et al.*, 2019; Pei *et al.*, 2020).

Acknowledgements

We would like to thank Connor Morgan-Lang, Julia Glinos, Kishori Konwar and Aria Hahn for lucid discussions on the function of the pathway2vec framework, Ryan MacLaughlin for his participation in preliminary performance evaluations and all members of the Hallam Lab for helpful comments along the way.

Funding

This work was performed under the auspices of Genome Canada, Genome British Columbia, the Natural Sciences and Engineering Research Council (NSERC) of Canada, and Compute/Calcul Canada. A.R.M.A.B was supported by a UBC four-year doctoral fellowship (4YF) administered through the UBC Graduate Program in Bioinformatics.

Conflict of Interest: S.J.H. is a co-founder of Koonkie Inc., a bioinformatics consulting company that designs and provides scalable algorithmic and data analytics solutions in the cloud.

References

Abadi,M. *et al.* (2016) Tensorflow: a system for large-scale machine learning. In: *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*. pp. 265–283. Savannah, GA, USA.

Abubucker,S. *et al.* (2012) Metabolic reconstruction for metagenomic data and its application to the human microbiome. *PLoS Comput. Biol.*, **8**, e1002358.

Abu-El-Haija,S. *et al.* (2018) Watch your step: learning node embeddings via graph attention. In: *Advances in Neural Information Processing Systems*. pp. 9180–9190. Montreal, Canada.

Ansorge,W.J. (2009) Next-generation DNA sequencing techniques. *N. Biotechnol.*, **25**, 195–203.

Arthur,D. and Vassilvitskii,S. (2007) k-means++: the advantages of careful seeding. In: *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. pp. 1027–1035. Society for Industrial and Applied Mathematics. New Orleans, Louisiana, USA.

Carbonell,P. *et al.* (2018) Selenzyme: enzyme selection tool for pathway design. *Bioinformatics*, **34**, 2153–2154.

Caspi,R. *et al.* (2016a) BioCyc: online resource for genome and metabolic pathway analysis. *FASEB J.*, **30**, lb192.

Caspi,R. *et al.* (2016b) The metaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. *Nucleic Acids Res.*, **44**, D471–D480.

Cohen,T. *et al.* (2019) Gauge equivariant convolutional networks and the icosahedral CNN. In: *International Conference on Machine Learning*. pp. 1321–1330. Long Beach, California, USA.

Dale,J.M. *et al.* (2010) Machine learning methods for metabolic pathway prediction. *BMC Bioinformatics*, **11**, 15.

Dong,Y. *et al.* (2017) metapath2vec: scalable representation learning for heterogeneous networks. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. pp. 135–144. ACM. Halifax, NS, Canada.

Eady,R.R. (1996) Structure- function relationships of alternative nitrogenases. *Chem. Rev.*, **96**, 3013–3030.

Fortunato,S. (2010) Community detection in graphs. *Phys. Rep.*, **486**, 75–174.

Fu,T-y. *et al.* (2017) HIN2Vec: explore meta-paths in heterogeneous information networks for representation learning. In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. pp. 1797–1806. ACM. Singapore.

Grover,A., and Leskovec,J. (2016) node2vec: scalable feature learning for networks. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. pp. 855–864. ACM. San Francisco, CA, USA.

Henderson,K. *et al.* (2012) RolX: structural role extraction & mining in large graphs. In: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. pp. 1231–1239. ACM. Beijing, China.

Hussein,R. *et al.* (2018) Are meta-paths necessary? Revisiting heterogeneous graph embeddings. In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. pp. 437–446. ACM. Torino, Italy.

Jiao,D. *et al.* (2013) Probabilistic inference of biochemical reactions in microbial communities from metagenomic sequences. *PLoS Comput. Biol.*, **9**, e1002981.

Kanehisa,M. *et al.* (2017) KEGG: new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Res.*, **45**, D353–D361.

Karp,P.D. *et al.* (2016) Pathway tools version 19.0 update: software for pathway/-genome informatics and systems biology. *Brief. Bioinform.*, **17**, 877–890.

Karp,P.D. *et al.* (2018) The EcoCyc Database. *EcoSal Plus*, **8**. pp. 1–19.

Lawson,C.E. *et al.* (2019) Common principles and best practices for engineering microbiomes. *Nat. Rev. Microbiol.*, **17**, 725–741.

M.A.Basher,A.R. *et al.* (2020) Metabolic pathway inference using multi-label classification with rich pathway features. *PLoS Comput. Biol.*, **16**, e1008174.

McInnes,L. *et al.* (2018) UMAP: uniform manifold approximation and projection. *J. Open Source Softw.*, **3**, 861.

Mikolov,T. *et al.* (2013) Distributed representations of words and phrases and their compositionality. In: *Advances in Neural Information Processing Systems*. pp. 3111–3119. Lake Tahoe, Nevada, USA.

Newman,M.E. (2006) Modularity and community structure in networks. *Proc. Natl. Acad. Sci. USA*, **103**, 8577–8582.

Pei,H. *et al.* (2020) Geom-GCN: geometric graph convolutional networks. *In International Conference on Learning Representations, Addis Ababa, Ethiopia.*

Perozzi,B. *et al.* (2014) DeepWalk: online learning of social representations. In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. pp. 701–710. ACM. NY, USA.

Shafiei,M. *et al.* (2014) BiomeNet: a Bayesian model for inference of metabolic divergence among microbial communities. *PLoS Comput. Biol.*, **10**, e1003918.

Shi,C. *et al.* (2017) A survey of heterogeneous information network analysis. *IEEE Trans. Knowl. Data Eng.*, **29**, 17–37.

Sun,Y. *et al.* (2011) PathSim: meta path-based top-K similarity search in heterogeneous information networks. *Proc. VLDB Endow.*, **4**, 992–1003.

Tabei,Y. *et al.* (2016) Simultaneous prediction of enzyme orthologs from chemical transformation patterns for de novo metabolic pathway reconstruction. *Bioinformatics*, **32**, i278–i287.

Toubiana,D. *et al.* (2019) Combined network analysis and machine learning allows the prediction of metabolic pathways from tomato metabolomics data. *Commun. Biol.*, **2**, 214.

Wang,D. *et al.* (2016) Structural deep network embedding. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. pp. 1225–1234. ACM. San Francisco, CA, USA.

Ye,Y. and Doak,T.G. (2009) A parsimony approach to biological pathway reconstruction/inference for genomes and metagenomes. *PLoS Comput. Biol.*, **5**, e1000465.