Bioinformatics, 37(6), 2021, 793–799
doi: 10.1093/bioinformatics/btaa881
Advance Access Publication Date: 14 October 2020
Original Paper

Systems biology

Learning graph representations of biochemical networks and its application to enzymatic link prediction

Julie Jiang [1], Li-Ping Liu[1,*] and Soha Hassoun[1,2,*]

[1]Department of Computer Science and [2]Department of Chemical and Biological Engineering, Tufts University, Medford 02155, USA

*To whom correspondence should be addressed.

Associate Editor: Pier Luigi Martelli

Received on February 25, 2020; revised on August 1, 2020; editorial decision on September 28, 2020; accepted on September 29, 2020

Abstract

Motivation: The complete characterization of enzymatic activities between molecules remains incomplete, hindering biological engineering and limiting biological discovery. We develop in this work a technique, enzymatic link prediction (ELP), for predicting the likelihood of an enzymatic transformation between two molecules. ELP models enzymatic reactions cataloged in the KEGG database as a graph. ELP is innovative over prior works in using graph embedding to learn molecular representations that capture not only molecular and enzymatic attributes but also graph connectivity.

Results: We explore transductive (test nodes included in the training graph) and inductive (test nodes not part of the training graph) learning models. We show that ELP achieves high AUC when learning node embeddings using both graph connectivity and node attributes. Further, we show that graph embedding improves link prediction by 30% in area under curve over fingerprint-based similarity approaches and by 8% over support vector machines. We compare ELP against rule-based methods. We also evaluate ELP for predicting links in pathway maps and for reconstruction of edges in reaction networks of four common gut microbiota phyla: actinobacteria, bacteroidetes, firmicutes and proteobacteria. To emphasize the importance of graph embedding in the context of biochemical networks, we illustrate how graph embedding can guide visualization.

Availability and implementation: The code and datasets are available through https://github.com/HassounLab/ELP.

Contact: liping.liu@tufts.edu or soha.hassoun@tufts.edu

1 Introduction

Characterizing enzymes through sequencing, annotation and homology has enabled the creation of complex system models that have played a critical role in advancing many biomedical and bioengineering applications. Insufficient characterization of enzymes, however, fundamentally limits our understanding of metabolism and creates knowledge gaps across many applications. For example, while nearly 300 β-glucuronidases (gut-bacterial enzymes that hydrolyze glucuronate-containing polysaccharides such as heparin and hyaluronate as well as small-molecule drug glucuronides) have been cataloged, functional information is available for only a small fraction (<10%) (Pellock *et al.*, 2019), thus limiting our ability to analyze host–microbiota interactions. Importantly, most enzymes if not all are promiscuous, acting on substrates other than the enzymes' natural substrates (Hult and Berglund, 2007; Khersonsky and Tawfik, 2010). At least one-third of protein superfamilies are functionally diverse, each superfamily catalyzing multiple reactions (Almonacid and Babbitt, 2011). Despite progress in functional annotation, the complete characterization or curation of enzyme function and the reactions they catalyze remains elusive. Computational prediction of enzymatic transformations promises to complement existing databases and provide new opportunities for biological discovery.

A common predictor of enzyme–compound interaction is compound and/or enzyme similarity to those within known enzymatic reactions. In biological engineering, molecular similarity between a query molecule and native substrates that are known to be catalyzed by the enzyme inform putative enzymatic transformations (Pertusi *et al.*, 2015). A high similarity score indicates a likely transformation. In drug–protein interaction analysis, molecular similarity and machine learning are utilized to predict the likelihood of interactions (Kurgan and Wang, 2018). Some techniques quantify similarity between reactions. EC-BLAST quantifies similarities between enzymatic reactions based on the similarities of bond changes, reaction centers and substrates and products (Rahman *et al.*, 2014). SimCAL computes reaction similarity at different levels such as the transformation region between substrates and products, or the similarity across all products-substrates within a reaction (Sivakumar *et al.*, 2018).

In addition to predicting aspects of enzymatic reaction similarities, there are rule-based methods to predict products of promiscuous reactions. Typically, such rules specify how a substrate molecule can be transformed to a product molecule. The rules can be hand

curated based on common biotransformations (Li *et al.*, 2004; Morreel *et al.*, 2014), extracted from existing sources, e.g. the KEGG RPAIR (Kotera *et al.*, 2004) RCLASS database (Kotera *et al.*, 2014b) or automatically extracted from reactions (Sivakumar *et al.*, 2016). As each rule is associated with a particular set of reactions, the presence of a rule directly correlates with the ability of predicting its associated enzymatic transformations.

In this article, we address the problem of predicting enzymatic transformations (links) between two molecules, a problem known as 'link prediction', where a link is a connection between two nodes within a network graph (Liben-Nowell and Kleinberg, 2007). Earlier work used the Tanimoto coefficient to score the maximum common substructure (MCS) between two molecules (Kotera *et al.*, 2008). Citing the computational inefficiency of MCS, an NP-hard problem, Kotera *et al.* (2013b, 2014a) utilized support vector machines (SVMs) to predict such links. Compound pairs in the KEGG RPAIR data were used as positive examples, while unknown interactions between compound pairs were utilized as negative examples. Feature vectors were constructed using either common or differential features based on various fingerprints. The use of SVMs along with additional substructures in the format of KCF-S [KEGG Chemical Function (KCF)-and-Substructures (Kotera *et al.*, 2013a)] and of aligned molecular graphs (Yamanishi *et al.*, 2015) further improved link prediction. Tabei *et al.* (2016) utilized joint-learning classifiers for link prediction and for predicting enzyme orthologs that could catalyze predicted transformations between compound pairs.

We present in this article a novel technique, enzymatic link prediction (ELP), for predicting enzymatic transformations between two molecules. ELP advances over the state of the art in two ways. First, ELP maps known enzymatic reactions already cataloged in databases [here, the KEGG database (Kanehisa and Goto, 2000)] to a graph structure, where compounds are represented as graph nodes while reactions are represented as graph edges. While snippets of such graph structures have been previously utilized as training data for multi-step pathway reconstruction (Kotera *et al.*, 2014a) and exploited during synthesis pathway construction (Yousofshahi et al., 2011), ELP utilizes all graph connectivity when predicting enzymatic links. Second, ELP uses graph embeddings (Cai *et al.*, 2018; Goyal and Ferrara, 2018) to learn molecular representations that reflect not only molecular structural properties but also relationships with other molecules in the network graph. Such embeddings have proven effective in predicting missing information, identifying spurious interactions, predicting links appearing in future evolving network, and analyzing biomedical networks (Cai *et al.*, 2018; Goyal and Ferrara, 2018; Yue *et al.*, 2019). We analyze both transductive (test nodes included in the training graph) and inductive (test nodes not part of the training graph) models. We evaluate ELP when learning node embeddings using both graph connectivity and node attributes and compare to similarity-based approaches.

2 Materials and methods

2.1 Constructing graph from the KEGG database

While proteins can interact with other proteins, the focus of this article is on enzymatic transformations between small molecules (those with masses less than 1000 Da). Such transformations form the backbone of metabolic networks. The KEGG database catalogs such enzymatic reactions and can be used to construct a data graph. Molecules in the KEGG database are represented as nodes. Each substrate-product pair within a reaction is modeled as an edge in the graph. As most KEGG reactions are reversible, we construct a non-directional graph. Biochemical networks have cofactor molecules (e.g. NADP, H_2O) that participate in many reactions, forming high-connectivity hub nodes within the graph (Ravasz *et al.*, 2002). As we aim to predict connectivity between non-cofactor metabolites, such high-connectivity nodes and their edges are excluded from the graph.

Nodes are assigned molecular fingerprints as attributes. The fingerprints are encoded as binary vectors of fixed length K. We select two fingerprints that reflect the presence or absence of pre-defined structural molecular fragments: the MACCS fingerprint with $K = 166$ structural keys (Durant *et al.*, 2002), and the PubChem fingerprint with $K = 881$ structural keys (Kim *et al.*, 2016).

Enzymatic reaction data are assigned as edge attributes. Each edge is assigned the enzyme commission (EC) number that catalyzes the associated chemical reaction. EC numbers are represented as four numbers separated by periods (e.g. L-lactate dehydrogenase is assigned EC number 1.1.1.27). Each edge is also assigned an RCLASS label (Kotera *et al.*, 2014b), five digit label. Each such label is associated with a group of reactions that share the same localized structural change between a substrate and a product (e.g. the addition or removal of a hydroxyl group). Although a reaction may be associated with one or more RC labels, each substrate-product pair is associated with only one RC label. If a reaction has no label, we assign it a null label. Thus, each edge in the graph is associated with an EC label and a RC label. A graph $G = (V, E)$ therefore consists of a set of vertices V and a set of edges E. Every node $i \in V$ represents a molecule and every edge $(i, j) \in E$ for some $(i, j) \in V$ represents an enzymatic reaction connecting two molecules i and j.

2.2 The ELP method

ELP has two steps (Fig. 1): (A) learning embedding vectors of graph nodes, and (B) predicting interaction between a pair of nodes from their embedding vectors. Embeddings are low-dimensional vector representations of each node. An embedding is characteristically similar to molecular fingerprint in the sense that they both quantitatively describe the molecules. Unlike fingerprints, however, entries in an embedding vector cannot be directly interpreted, but rather can be decoded by a suitable learning algorithm. Importantly, embeddings capture the inherent structure of the graph as well as attributes of the nodes and edges in the graph, which allow them to be used as input for downstream tasks such as link prediction. For the first step, we use the embedding propagation (EP) algorithm (García-Durán and Niepert, 2017). EP was selected because it almost consistently outperformed several other methods in the presence of node attributes on several datasets. Further, EP has the advantage of fewer parameters and hyperparameters when compared to other methods (e.g. Grover and Leskovec, 2016; Perozzi *et al.*, 2014; Tang *et al.*, 2015). For the second step, we train a neural network that takes pairs of learned embedding vectors as input and predicts the connectivity of two molecules.

2.2.1 Connectivity-based learned embeddings

The simplest form of EP is to learn a set of node embedding vectors $\mathbf{U} = \{\mathbf{u}_i \in \mathbb{R}^d : i \in V\}$, where d is the embedding size. Embeddings are randomly initialized prior to training. Node embeddings are learned via an iterative process, by propagating forward (representations of nodes) and backward (gradients) messages between neighboring nodes. The iterative process repeats until a convergence threshold is reached. Suppose $N(i) = \{j \in V : (i, j) \in E\}$ is the set of neighboring nodes of node i. The model aims to reconstruct embeddings $\mathbf{u}_i$ from the embeddings of i's neighbors. The reconstructed node embedding $\mathbf{u}_i$ for node i is:

$$\tilde{\mathbf{u}}_i = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \mathbf{u}_j \tag{1}$$

The learning objective of EP is to maximize the similarly between $\tilde{\mathbf{u}}_i$ and $\mathbf{u}_i$. Instead of maximizing the absolute values of inner products for all such nodes, EP maximizes their values in a relative sense: the reconstruction should be more similar to the corresponding embedding vector than any other embedding vectors. The error in

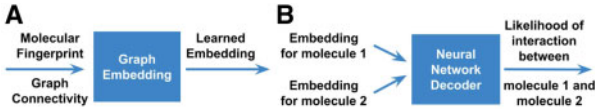

Fig. 1 ELP Overview. (**A**) Molecular representations are learned using graph embedding. (**B**) Learned embeddings are used to predict links

reconstruction is therefore minimized through a margin-based ranking loss (García-Durán and Niepert, 2017):

$$\mathcal{L} = \sum_{i \in V} \sum_{j \in V, j \neq i} \max\{\gamma - \tilde{\mathbf{u}}_i^\top \mathbf{u}_i + \tilde{\mathbf{u}}_i^\top \mathbf{u}_j, 0\}, \quad (2)$$

where $\gamma > 0$ is a chosen margin hyperparameter. The objective is optimized by stochastic gradient descent. However, summing over all nodes as indicated by the inner sum is very expensive. For performance, we randomly select one node as the negative example for each real node in every iteration to compute an estimation of ℓ and its gradient, as was done in García-Durán and Niepert (2017).

2.2.2 Attribute-based learned embeddings

To incorporate information from edge attributes, EP learns embedding vectors $\mathbf{Z} = \{\mathbf{z}_c \in \mathbb{R}^d : c = 1, \ldots, C\}$ for the C reaction labels. The reconstructed node embedding $\mathbf{u}_i$ for node i is modified as follows:

$$\tilde{\mathbf{u}}_i = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \mathbf{u}_j + \alpha \mathbf{z}_{r(i,j)}, \quad (3)$$

where $r_{i,k}$ is the edge label of the edge (i, j) and $\mathbf{z}_r(i, j)$ is the corresponding edge embedding. The hyperparameter $\alpha \in \{0, 1\}$ weights the importance of edge features. The vector $\mathbf{z}_0$ corresponding to null edge attributes is fixed to zero to avoid affecting the reconstruction. Embeddings based on edge attributes can be learned simultaneously while learning connectivity-based embeddings. While the edge embeddings are used during training, they are not used to compute the final embeddings of nodes after EP training.

EP can also learn K fingerprint embedding vectors $\mathbf{V} = \{\mathbf{v}_k \in \mathbb{R}^d : k = 1, \ldots, K\}$. Specifically, the node-attribute-based embedding $\mathbf{u}_i$ of a node i is the mean of fingerprint embeddings $\mathbf{v}_k$ corresponding to positive fingerprint entries in the fingerprint vector $\mathbf{f}_i \in \{0, 1\}^K$:

$$\mathbf{u}_i^{fp} = \frac{1}{\sum_{k=1}^K f_{ik}} \sum_{k=1}^K f_{ik} \mathbf{v}_k \quad (4)$$

When computing embeddings based on node attributes, we optimize the fingerprint embeddings $\mathbf{V}$, instead of $\mathbf{U}$, through the learning objective in Equation (2). An advantage of the EP algorithm is its ability to learn only one of the node embedding types or all. If both node attributes and connectivity embeddings are trained, we simply concatenate $\mathbf{u}_i$ and $\mathbf{u}_i^{fp}$ to form the final node embedding vector of node i before applying the link prediction model. L2 regularization is applied to all variables $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{Z}$.

2.2.3 Link prediction

The trained node embeddings are used as inputs to a logistics link prediction model. Pairs of embeddings of nodes involved in a known reaction are positive examples; pairs of embeddings of nodes that have no or unknown interaction are treated as negative examples. To make link predictions, the neural network outputs the likelihood of an edge for every pair of input node embeddings. The final result of the model is evaluated based on the area under curve (AUC) metric, wherein the false positive rate and true positive rate are evaluated at every threshold to compute the area under the receiver operating characteristic (ROC) curve.

2.3 Training and testing

We explore two learning scenarios—transductive and inductive—that we apply to ELP and our baseline methods. In the transductive setting, we train on all available nodes and evaluate the edge recovery for a set of test edges that were withheld from training. Hence, the graph is split into training and testing sets by partitioning on the edges. During training, all non-training edges are considered negative examples, including those that are test edges. During testing, we evaluate the AUC using the withheld test edges as positive examples and an equal-sized sample of the negative edges as the negative examples. In the inductive scenario, the model predicts interactions for *out-of-sample* nodes excluded from the training set. In the case of ELP, we compute embeddings for out-of-sample nodes from their attributes and predict possible enzymatic reactions for them. Due to the lack of prior connectivity information for out-of-sample nodes, only embeddings based solely on node attributes are learned during training for the ELP method. To generate the training and testing sets, we reserve a certain portion of nodes and their incident edges for the test graph. All other nodes and edges are included in the training graph. Similar to the evaluation of the transductive learning scenario, we sample a set of negative edges equal in size with the test edges.

For all experiments, the embedding dimension is set to 128. The learning rate for the EP framework is set to 0.01, the regularization to 0.0002, and the γ margin is set to 10. The embedding vectors are trained batch size of 2048 for 500 epochs or until convergence, whichever one comes earliest. The deep neural network decoder predicts the connectivity of two molecules based on their embeddings consists of two hidden layers of sizes 32 and 16. It is trained for 40 epochs on a batch size of 2048 with a learning rate of 0.01. The margin hyperparameter $\gamma >$ is set to 10. In experiments using edge features, α is set to 1.

3 Results

Once cofactors were excluded, and MACCS and PubChem fingerprints were generated for all our nodes, our dataset representing the biochemical network underlying the KEGG database consisted of 7049 nodes and 12 507 edges, with an average node degree of 3.5. We evaluate both scenarios and all techniques using 5-fold cross-validation.

3.1 Transductive link prediction

Results for several transductive scenarios are reported [Table 1, partitions (A)–(D)]. When performing connectivity-based prediction (partition A), we compare ELP against different variants of node2-vec (Grover and Leskovec, 2016), an algorithm for learning node embeddings based on random walks of the graph. Node2vec maximizes the probability of occurrence of nearby nodes in fixed-length random walks, thus preserving higher-order proximity between nodes. The characteristics of the random walks can be specified using the return (backtrack) parameter p, the in–out parameter q, the length of the walks l and the context window k, which controls for the neighborhood of nodes considered as nearby. The embedding dimension of node2vec is fixed to be the same as the one used in ELP. We compare ELP to several node2vec variants. The first variant (default) is the node2vec model with default parameters $p = 1$, $q = 1$, $l = 80$, and $k = 10$; the second variant (short walks) reduces the length of the walk to 10 and the context size to 5. Based on the improvement in AUC from 0.80 to 0.82 using shorter walks, we fix $l = 10$ and $k = 5$ and explore the differences between a DFS-style random walk and a BFS-style random walk, which is defined as $p = 1$, $q = 0.5$ and $p = 1$, $q = 2$, respectively. We found a DFS-style random walk led to poorer results (0.75 AUC) but a BFS-style random walk gave the best node2vec result (0.83 AUC), suggesting that localized neighborhoods are more effective in learning node representations than larger neighborhoods. In contrast, ELP, which explicitly considers only the immediate neighbors of every node, outperforms all node2vec variants with an AUC of 0.88.

Partition (B) explores the effects of using only the MACCS or PubChem fingerprints without utilizing graph-connectivity embeddings. As a baseline, we apply the Jaccard similarity model on the fingerprints of every substrate-product pair in the test set. The Jaccard AUC results are 0.67 using MACCS and 0.65 using PubChem. Another baseline for this partition, denoted L2SVM, is a link prediction model similar to Kotera *et al.* (2013b) based on molecular fingerprints. It uses the similarities and differences between the two fingerprints of a given pair of molecules as inputs to an SVM. We chose the L2-regularized SVM as it was the best performing model in Kotera *et al.* (2013b). The original model proposed by

Table 1 Link prediction results for the transductive learning scenario

Model	Connectivity	Fingerprint	Enzyme label	AUC
A. Connectivity only				
node2vec (default)	Yes	—	—	0.80±.011
node2vec (short walks)	Yes	—	—	0.83±.020
node2vec (DFS)	Yes	—	—	0.75±.152
node2vec (BFS)	Yes	—	—	0.83±.004
ELP	Yes	—	—	0.88±.003
B. Fingerprints only				
Jaccard	No	MACCS	—	0.67±.006
Jaccard	No	PubChem	—	0.65±.006
L2SVM	No	MACCS	—	0.89±.002
ELP	No	MACCS	—	0.93±.004
ELP	No	PubChem	—	0.93±.002
C. Connectivity and # fingerprint				
ELP	Yes	MACCS	—	0.97±.003
ELP	Yes	PubChem	—	0.97±.001
D. Connectivity, # fingerprint, and # enzyme labels				
ELP	Yes	MACCS	EC	0.97±.001
ELP	Yes	PubChem	RC	0.97±.001

The AUC results and the standard deviations obtained using 5-fold cross-validation. We partitioned the experiments to facilitate comparisons. (A) Using only network connectivity to learn embeddings. (B) Using only MACCS or Pubchem fingerprints. ELP still uses network connectivity to indirectly learn fingerprint embeddings. (C) Using both network connectivity and fingerprints. (D) Using network connectivity, fingerprints and enzyme labels.

Kotera *et al.* (2013b) uses reactant pairs based on an earlier definition of "main" type transformations that was present in the RPAIR database within KEGG (Kotera *et al.*, 2013b). To facilitate a meaningful comparison, we apply Kotera *et al.* (2013b)'s model directly on our described network, maintaining the assumption that all substrate-product pairs are reversible. Using L2SVM on the MACCS fingerprints yields a mean AUC of 0.89. L2SVM requires significant compute time and memory to process all possible ordered pairs of molecules. As such, the experiment using L2SVM on the PubChem fingerprints required over 1 terabyte of memory on a CPU Processor. The experiment could not be completed using our available resources. As for the ELP model, both the MACCS and PubChem fingerprints achieve a mean AUC of 0.93. Using ELP with MACCS fingerprints leads to slightly larger variabilities (+0.002 std) across 5-fold cross-validation than with the PubChem fingerprints.

Per partition (C), using a combination of both connectivity information and fingerprint attributes with ELP yields the best results, with both MACCS and PubChem fingerprints achieving an AUC of 0.97. We see again that the MACCS fingerprints lead to 0.002 more variation in terms of standard deviation, but the computational advantages of using MACCS fingerprints is its smaller size ($K = 166$) compared to PubChem fingerprints ($K = 881$). In partition (D) we incorporate enzyme labels as edge labels. This addition does little to enhance predictive accuracy but decreases the standard deviation, indicating that the inclusion of enzyme labels leads to more stable results. There is little difference in AUC when utilizing the two enzyme labels.

Figure 2 presents plots for two scenarios using ELP: (A) connectivity only and (B) connectivity with MACCS fingerprints as node attributes. The plot reveals that the lower AUC performance is mostly attributed to having a higher FPR when there is a higher TPR. In other words, we can achieve an almost 0.50 TPR at little cost (little sacrifice in FPR), but as the need to observe improvement in TPR increases, the FPR rises dramatically.

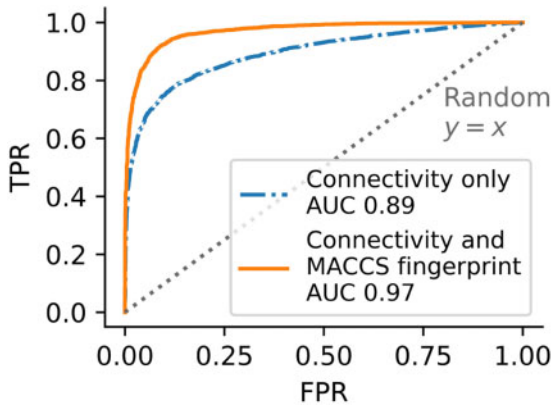

Fig. 2 ROC curve plot for transductive learning with and without MACCS fingerprint as node attributes

Table 2 Link prediction results for the inductive learning scenario

Model	Connectivity	Fingerprint	AUC
Jaccard	No	MACCS	0.68 ± 0.004
Jaccard	No	Pubchem	0.67 ± 0.014
ELP	No	MACCS	0.93 ± 0.005
ELP	No	Pubchem	0.94 ± 0.005

The AUC results and the standard deviations obtained using five random sample of held-out test nodes (5% of all nodes). All models are tested on recovering edges incident to the test nodes.

3.2 Inductive link prediction

Several inductive scenarios were investigated (Table 2). In these scenarios, 5% of all nodes were removed from the graph during training. ELP based on MACCS node attributes achieves an AUC of 0.93 and ELP based on PubChem node attributes achieves an AUC of 0.94. This performance is nearly identical to ELP's performance in the transductive learning scenarios, wherein an AUC of 0.93 is achieved using either MACCS or PubChem fingerprints. Despite the out-of-sample nodes in the test set not being part of the training graph, ELP robustly leveraged fingerprint information for nodes within the training graph to achieve higher AUC. Similarity analyses based on the Jaccard similarity scores are much lower, with 0.68 and 0.67 AUCs for MACCS and PubChem, respectively.

3.3 Pathway reconstruction

To evaluate how ELP recovers links within metabolic pathways, we reconstruct pathway edges that are omitted during training. We select the same set of pathway groups that was used in Kotera *et al.* (2013b). For each pathway, we reserve its edges as the test graph and use an equal-size random sample of negative edges in the test graph. We evaluate the ability of the ELP model with the MACCS fingerprints to recover edges within each individual pathway.

We illustrate our results in Figures 3 and 4. Similar to earlier findings (Kotera *et al.*, 2013b), the results for individual pathways are overall lower than a random 5-fold split (Fig. 3), with a mean AUC of 0.88. The results for the glycan biosynthesis and metabolism pathway group spans a wide range from 0.5 (same as pure chance) to 1 (perfect prediction). The variability is due to the small number of edges within these pathways, where some have as little as 2 true edges. In general, there is little difference in the reconstruction of different pathway maps using ELP. To further benchmark our results, we applied the Jaccard similarity scoring on the same task. Figure 4

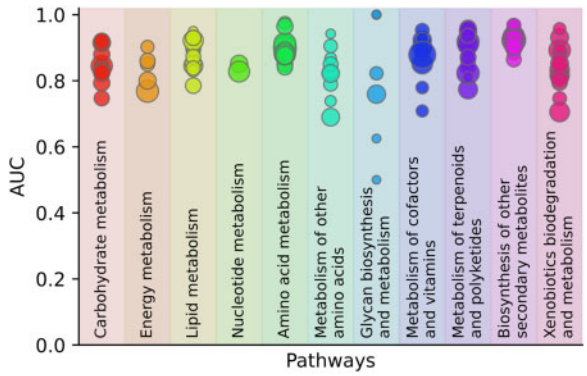

Fig. 3 AUC results using the ELP model for each pathway within each pathway group, shown for various functional pathway groups. The size of each marker is proportional to the number of edges being tested for each pathway

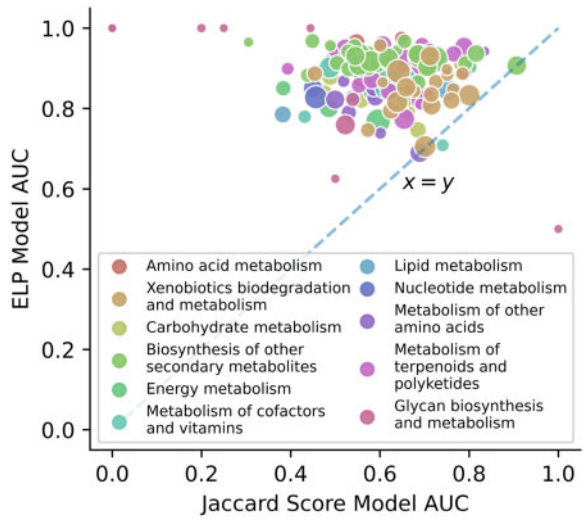

Fig. 4 AUC results using the Jaccard score model and the ELP model shown for various functional pathways groups. Markers are color-coded by pathway groups and the size of each marker is proportional to the number of edges being tested

Table 3 Organism reconstruction result (mean AUC scores and the standard deviation) for each phylum

Phylum	# Test Edges	Jaccard AUC	ELP AUC
Proteobacteria	4982	0.77 ± 0.007	0.89 ± 0.001
Firmicutes	5454	0.77 ± 0.010	0.91 ± 0.004
Bacteroidetes	3674	0.76 ± 0.009	0.90 ± 0.005
Actinobacteria	5968	0.77 ± 0.010	0.91 ± 0.001

Each phylum is repeated five times, each with five different random samples of 1024 test edges.

shows that our results are almost always better than results given by Jaccard scores (above the $x = y$ line).

3.4 Organism reconstruction

We assess how ELP recovers enzymatic reactions at the organism level. We explore link reconstruction for four gut microbiota phyla: actinobacteria, bacteroidetes, firmicutes and proteobacteria (Rinninella *et al.*, 2019). For each phyla, we retrieved available corresponding organisms from the KEGG database. Due to the high number of affiliated edges for each phylum (Table 3) and the conservation of metabolism across many organisms, we tested link construction for a large subset of such edges. We test the reconstruction of 1024 randomly sampled edges per phylum. We report the average AUC and standard deviation across five 1024-edge reconstructions for each phylum using the Jaccard similarity model and the ELP model with MACCS fingerprints. For all phyla, the Jaccard similarity model consistently yields AUCs between 0.76 and 0.77 with standard deviations around 0.01. The ELP model achieves higher AUCs ranging between 0.89 and 0.91, and the results show smaller variations.

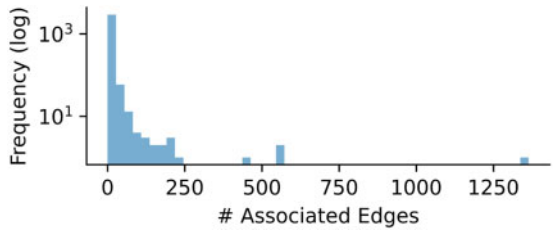

Fig. 5 Histogram distribution of the number of graph edges associated with reactions in each RCLASS

Table 4 Rule reconstruction results of the top 20 rules with the most number of associated test edges

Rule	# Test edges	Jaccard AUC	ELP AUC
RC00003	1364	0.79	0.95
RC00006	565	0.74	0.89
RC00392	550	0.66	0.93
RC00049	438	0.74	0.82
RC00014	279	0.40	0.76
RC00171	210	0.61	0.75
RC00021	207	0.77	0.79
RC00041	205	0.71	0.77
RC00010	171	0.58	0.82
RC00055	169	0.71	0.77
RC00020	152	0.71	0.83
RC00046	141	0.80	0.86
RC00062	136	0.67	0.87
RC00008	124	0.62	0.76
RC00279	121	0.14	0.69
RC00460	102	0.66	0.90
RC00523	101	0.62	0.79
RC00466	92	0.64	0.93
RC00059	89	0.76	0.83
RC00661	81	0.60	0.85

3.5 Rule reconstruction

In contrast to a data-driven machine learning framework like ELP, rule-based models [e.g. PROXIMAL (Yousofshahi *et al.*, 2015) and ReactPred (Sivakumar *et al.*, 2016)] rely on transformation rules. The accuracy of rule-based methods depends entirely on the availability of transformation rules and the ability to apply such rules to query substrate molecules. We design a rule reconstruction experiment with the goal of evaluating how ELP recover edges associated with the most prevalent rules.

To this end, we compile a list of reaction rules (RCLASSES) in the KEGG database. Given that cofactors were not included in our graph, we remove RLCASSES associated with cofactors, which accounts for less than 5% of all RCLASSES. For each RCLASS, we find the list of associated graph edges through all reactions linked to the RCLASS. Figure 5 depicts the distribution of the number of associated edges of RCLASSES. We observe that the distribution is heavy tailed, where the majority of rules have very few associated edges and a few rules have many associated edges. To test the reconstruction of one rule, we hide all of its associated edges and train on the

rest of the edges using the ELP model with the MACCS fingerprints. Similar to previous testing frameworks, we evaluate the reconstruction of the associated edges against a randomly sampled set of negative edges of equal size. We evaluate ELP's performance when hiding the top 20 most popular RCLASSES, one RCLASS at a time.

We report our ELP results along with the Jaccard similarity scoring results as a baseline comparison (Table 4). The AUC results over the top 20 RCLASSES using the Jaccard similarity have a mean of 0.65 and standard deviation of 0.15. In contrast, the results using ELP are higher and vary considerably less with a mean and standard deviation of 0.83 and 0.07, respectively. The AUC results using ELP are consistently above 0.75, with the exception of RC00279 (0.69). This RCLASS is associated with enzymes with E.C. numbers 2.5.1.* and denotes the transfer of alkyl or aryl groups other than methyl groups. All edges in the KEGG graph associated with this RCLASS involve isopentenyl diphosphate and a larger molecule, resulting in low similarity between substrate and product. This RCLASS therefore presents an ostensibly difficult rule to predict, as demonstrated by its Jaccard AUC of 0.14. Importantly, ELP performance is not negatively impacted by the prevalence of rules. This result shows that even with the lack of certain rules all together, ELP is successful at recovering missing reactions and its performance is stable, suggesting that ELP is a promising framework in place of rule-based prediction models that completely fail to recover missing reactions if the relevant rule is not considered.

3.6 Biochemical network visualization

To further illustrate the importance of graph embedding in the context of biochemical networks, we show how graph embedding can be used for visualization. Figure 6 presents a visualization of the embeddings for two reference pathways, the citrate cycle (TCA) cycle and Glycolysis/Gluconeogenesis, as documented in the KEGG database. The resulting subgraph for the TCA cycle consists of 25 nodes and 70 edges, while the subgraph for glycolysis/gluconeogenesis pathway consists of 46 nodes and 126 edges. Twelve compounds are common to both pathways, and include phosphate, diphosphate, pyruvate, thiamine diphosphate, lysine, oxaloacetate and phosphoenolpyruvate. These compounds contribute to 23 edges that overlap in both pathways. To visualize embeddings of these metabolites, we reduce the dimensionality of the embeddings to 2 via t-SNE (Maaten and Hinton, 2008). For the connectivity only plot (top), we observe tight clustering of metabolites within each pathway, while we observe looser clustering when using MACCS fingerprints as node attributes (bottom). Nodes that are embedded far away from the clusters, phosphate, diphosphate, and carbon dioxide, exhibit high connectivity within the KEGG graph, with node degrees of 460, 398, and 545, respectively. On the contrary, nodes within the KEGG graph have an average degree of 3.5, and nodes within the two reference pathways have an average degree of 5.5.

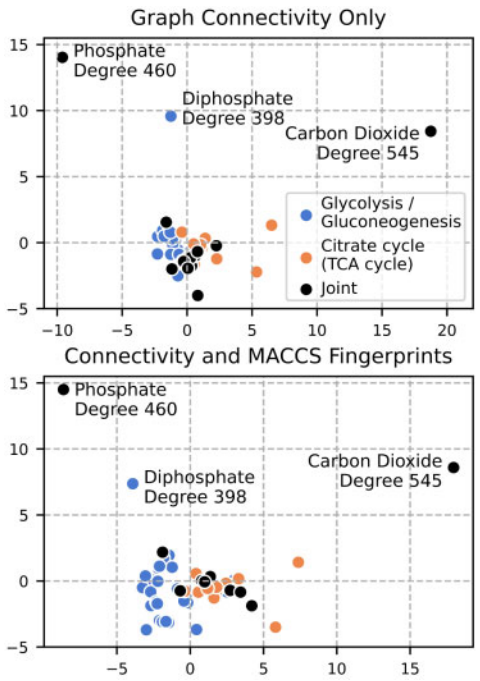

Fig. 6 2D t-SNE (Maaten and Hinton, 2008) visualization of embeddings for two transductive scenarios using ELP. Top: using graph connectivity only. Bottom: using both graph connectivity and MACCS fingerprints

3.7 Runtime

A single reconstruction using embeddings in ELP has worst case time complexity linearly proportional to the maximum degree of a node (García-Durán and Niepert, 2017). Combined with negative sampling, a single iteration of ELP takes $O(K|V|\deg_{\max}(V))$, where K is the length of fingerprints. The space complexity of ELP is $O(dKn)$, where d is the embedding dimension. For benchmarking purposes, our experiments using ELP with MACCS fingerprints completed in under 40 min of wall-clock time with 50 GB of available memory. In contrast, each experiment using L2SVM with the MACCS fingerprints took $\bar{7}5$ min with 500 GB of available memory. SVM is space bound due to the large number of n^2 ordered compound pairs and requires $O(Kn^2)$ memory.

4 Conclusion

This work uses EP to learn molecular representations that capture both graph connectivity, enzymatic properties and structural molecular properties. We show that link prediction using only graph connectivity is on par with using molecular similarity. Importantly, we show high accuracy in link prediction when using both graph connectivity and molecular attributes. Link prediction outperforms prior techniques based on similarity methods, SVMs and rule-based methods. Link prediction was shown effective in reconstructing metabolic pathways and reactions within the gut microbiota. This work has broader and practical impact. ELP can be used to guide many biological discoveries and engineering applications such as identifying catalyzing enzymes when constructing novel synthesis pathways or predicting interaction between microbes and human hosts. Graph embedding can be used for other applications such as biochemical network visualization, as demonstrated herein. Further, while our approach is applied to biochemical enzymatic networks, it can enhance link prediction in chemical networks, where rule-based and path-based link prediction respectively yielded 52.7% and 67.5% prediction accuracy (Segler and Waller, 2017).

Funding

This research was supported by National Science Foundation [Award 1909536]. Li-Ping Liu was supported by National Science Foundation [Award 1850358, 1908617]. The research was also supported by the NIGMS of the National Institutes of Health [Award R01GM132391]. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

Conflict of Interest: none declared.

References

Almonacid,D.E. and Babbitt,P.C. (2011) Toward mechanistic classification of enzyme functions. *Curr. Opin. Chem. Biol.*, **15**, 435–442.

Cai,H. *et al.* (2018) A comprehensive survey of graph embedding: problems, techniques, and applications. *IEEE Trans. Knowl. Data Eng.*, **30**, 1616–1637.

García-Durán,A. and Niepert,M. (2017) Learning graph representations with embedding propagation. In: *Proceedings of the International Conference on Neural Information Processing Systems, NIPS'17*, pp. 5119–5130, Red Hook, NY, USA. Curran Associates Inc.

Durant,J.L. *et al.* (2002) Reoptimization of mdl keys for use in drug discovery. *J. Chem. Inform. Comput. Sci.*, **42**, 1273–1280.

Goyal,P. and Ferrara,E. (2018) Graph embedding techniques, applications, and performance: a survey. *Knowl. Based Syst.*, **151**, 78–94.

Grover,A. and Leskovec,J. (2016) Node2vec: scalable feature learning for networks. In: Proceedings of 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'16, pp. 855–864, New York, NY, USA. ACM.

Hult,K. and Berglund,P. (2007) Enzyme promiscuity: mechanism and applications. *Trends Biotechnol.*, **25**, 231–238.

Kanehisa,M. and Goto,S. (2000) Kegg: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.*, **28**, 27–30.

Khersonsky,O. and Tawfik,D.S. (2010) Enzyme promiscuity: a mechanistic and evolutionary perspective. *Annu. Rev. Biochem.*, **79**, 471–505.

Kim,S. *et al.* (2016) PubChem substance and compound databases. *Nucleic Acids Res.*, **44**, D1202–D1213.

Kotera,M. *et al.* (2004) Computational assignment of the EC numbers for genomic-scale analysis of enzymatic reactions. *J. Am. Chem. Soc.*, **126**, 16487–16498.

Kotera,M. *et al.* (2008) Eliciting possible reaction equations and metabolic pathways involving orphan metabolites. *J. Chem. Inform. Model.*, **48**, 2335–2349.

Kotera,M. *et al.* (2013a) KCF-S: KEGG Chemical Function and Substructure for improved interpretability and prediction in chemical bioinformatics. *BMC Syst. Biol.*, **7**, S2.

Kotera,M. *et al.* (2013b) Supervised *de novo* reconstruction of metabolic pathways from metabolome-scale compound sets. *Bioinformatics*, **29**, i135–i144.

Kotera,M. *et al.* (2014a) Metabolome-scale prediction of intermediate compounds in multistep metabolic pathways with a recursive supervised approach. *Bioinformatics*, **30**, i165–i174.

Kotera,M. *et al.* (2014b) Predictive genomic and metabolomic analysis for the standardization of enzyme data. *Perspect. Sci.*, **1**, 24–32.

Kurgan,L. and Wang,C. (2018) Survey of similarity-based prediction of drug-protein interactions. *Curr Med Chem.*, 27, 5856–5886.

Li,C. *et al.* (2004) Computational discovery of biochemical routes to specialty chemicals. *Chem. Eng. Sci.*, **59**, 5051–5060.

Liben-Nowell,D. and Kleinberg,J. (2007) The link-prediction problem for social networks. *J. Am. Soc. Inform. Sci. Technol.*, **58**, 1019–1031.

Maaten,L. v d. and Hinton,G. (2008) Visualizing data using t-SNE. *J. Mach. Learn. Res.*, **9**, 2579–2605.

Morreel,K. *et al.* (2014) Systematic structural characterization of metabolites in Arabidopsis via candidate substrate-product pair networks. *Plant Cell*, **26**, 929–945.

Pellock,S.J. *et al.* (2019) Discovery and characterization of fmn-binding β-glucuronidases in the human gut microbiome. *J. Mol. Biol.*, **431**, 970–980.

Perozzi,B. *et al.* (2014) Deepwalk: online learning of social representations. In: *Proceedings of 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'14*, pp. 701–710, New York, NY, USA, ACM.

Pertusi,D.A. *et al.* (2015) Efficient searching and annotation of metabolic networks using chemical similarity. *Bioinformatics*, **31**, 1016–1024.

Rahman,S.A. *et al.* (2014) Ec-blast: a tool to automatically search and compare enzyme reactions. *Nat. Methods*, **11**, 171–174.

Ravasz,E. *et al.* (2002) Hierarchical organization of modularity in metabolic networks. *Science*, **297**, 1551–1555.

Rinninella,E. *et al.* (2019) What is the healthy gut microbiota composition? A changing ecosystem across age, environment, diet, and diseases. *Microorganisms*, **7**, 14.

Segler,M.H. and Waller,M.P. (2017) Modelling chemical reasoning to predict and invent reactions. *Chemistry*, **23**, 6118–6128.

Sivakumar,T.V. *et al.* (2016) ReactPRED: a tool to predict and analyze biochemical reactions. *Bioinformatics*, **32**, 3522–3524.

Sivakumar,T.V. *et al.* (2018) Simcal: a flexible tool to compute biochemical reaction similarity. *BMC Bioinformatics*, **19**, 254.

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

Tang,J. *et al.* (2015) LINE: Large-scale information network embedding. In: Proceedings of 24th International Conference on World Wide Web, *WWW' 15*, pp. 1067–1077, Republic and Canton of Geneva, CHE.

Yamanishi,Y. *et al.* (2015) Metabolome-scale *de novo* pathway reconstruction using regioisomer-sensitive graph alignments. *Bioinformatics*, **31**, i161–i170.

Yousofshahi,M. *et al.* (2011) Probabilistic pathway construction. *Metabol. Eng.*, **13**, 435–444.

Yousofshahi,M. *et al.* (2015) PROXIMAL: a method for prediction of xenobiotic metabolism. *BMC Syst. Biol.*, **9**, 94.

Yue,X. et al. (2019) Graph embedding on biomedical networks: methods, applications and evaluations. *Bioinformatics*, 36, 1241–1251.