Bioinformatics

, 37(19), 2021, 3328–3336

doi: 10.1093/bioinformatics/btab224

Advance Access Publication Date: 6 April 2021

Original Paper

Data and text mining

HPOFiller: identifying missing protein–phenotype associations by graph convolutional network

Lizhi Liu1, Hiroshi Mamitsuka2,3 and Shanfeng Zhu  4,5,6,7,8,\*

1 2

School of Computer Science, Fudan University, Shanghai 200433, China, Bioinformatics Center, Institute for Chemical Research, Kyoto University, Uji, Kyoto Prefecture 611-0011, Japan, 3Department of Computer Science, Aalto University, Espoo, Finland, 4Institute of Science and Technology for Brain-Inspired Intelligence and Shanghai Institute of Artificial Intelligence Algorithms, Fudan University, Shanghai 200433, China, 5Key Laboratory of Computational Neuroscience and Brain-Inspired Intelligence (Fudan University), Ministry of Education, Shanghai 200433, China, 6Shanghai Key Lab of Intelligent Information Processing, Fudan University, Shanghai 200433, China, 7MOE Frontiers Center for Brain Science, Fudan University, Shanghai 200433, China and 8Zhangjiang Fudan International Innovation Center, Shanghai 200433, China.

\*To whom correspondence should be addressed.

Associate Editor: Jonathan Wren

Received on October 14, 2020; revised on February 20, 2021; editorial decision on March 24, 2021; accepted on April 5, 2021

# Abstract

Motivation: Exploring the relationship between human proteins and abnormal phenotypes is of great importance in the prevention, diagnosis and treatment of diseases. The human phenotype ontology (HPO) is a standardized vocabulary that describes the phenotype abnormalities encountered in human diseases. However, the current HPO annotations of proteins are not complete. Thus, it is important to identify missing protein–phenotype associations. Results: We propose HPOFiller, a graph convolutional network (GCN)-based approach, for predicting missing HPO annotations. HPOFiller has two key GCN components for capturing embeddings from complex network structures: (i) S-GCN for both protein–protein interaction network and HPO semantic similarity network to utilize network weights; (ii) Bi-GCN for the protein–phenotype bipartite graph to conduct message passing between proteins and phenotypes. The core idea of HPOFiller is to repeat run these two GCN modules consecutively over the three networks, to refine the embeddings. Empirical results of extremely stringent evaluation avoiding potential information leakage including cross-validation and temporal validation demonstrates that HPOFiller significantly outperforms all other state-of-the-art methods. In particular, the ablation study shows that batch normalization contributes the most to the performance. The further examination offers literature evidence for highly ranked predictions. Finally using known disease-HPO term associations, HPOFiller could suggest promising, unknown disease–gene associations, presenting possible genetic causes of human disorders.

Downloaded from https://academic.oup.com/bioinformatics/article/37/19/3328/6212045 by National Science & Technology Library user on 03 October 2022

# 1 Introduction

Uncovering phenotypic correlations of gene mutations have long been an essential task in genetics research. The Human Phenotype Ontology (HPO) (Ko¨hler et al., 2019) provides a standardized vocabulary of phenotype abnormalities encountered in human diseases and of their semantic relationships. The HPO annotations of human genes can facilitate disease gene identification and prioritization and hence assist clinical diagnostics (Ko¨hler et al., 2009).

|  |
| --- |
| Figure 1 shows the average number of HPO annotations including ancestors of the specific annotations over proteins. We keep track of proteins that already exist in the database released in March 2018 and count how many annotations each protein has on average as time goes on. This figure indicates an around 20% increase of the average number in the past two years, implying that a large number of missing associations still exist between proteins and phenotypes. The incomplete HPO annotations would degrade the performance of phenotype prediction tools (Liu et al., 2020) and affect the analysis of genetic causes of disorders. Thus it would be imperative to develop a computational method for identifying missing proteinHPO term associations. |

However, filling the missing HPO annotations is a very challenging task: (i) The annotations are highly sparse. For HPO annotations

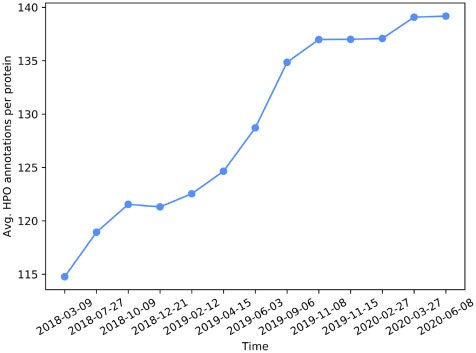


Fig. 1. For proteins that already existed in the HPO annotations released on March 9, 2018, the average number of annotations (including ancestors of the specific annotations) per protein increased over time

released by June 2020, annotations (positive examples) are only 1.58% among all possible pairs of proteins and HPO terms. (ii) The distribution of HPO annotations is skewed. In HPO released by June 2020, more than 1700 out of 15 054 HPO terms are used to annotate only one protein, while over 3700 terms are associated with more than 10 proteins. (iii) HPO terms are not independent of each other but organized hierarchically as a directed acyclic graph (DAG). The directed edge between two terms represents an ‘is-a’ relationship, keeping the ‘true-path-rule’. That is, a protein, which is annotated with a given term, can be annotated with all ancestor terms in the DAG.

The importance of protein–protein interaction (PPI) network for prediction of HPO annotations is broadly recognized, due to an assumption that strongly interacted proteins are more likely to be associated with similar phenotypes (Goh et al., 2007; Oti et al., 2006). Thus, taking the PPI network as input data, can be of great help to identify missing HPO annotations. Besides, the similarity between HPO terms providing quantitative measures of phenotype relationships would be useful likewise. Accordingly, we have three input graphs: two types of similarity networks separately for proteins and HPO terms, and a bipartite network by annotations between proteins and HPO terms.

Recently graph convolutional networks (GCNs) (Defferrard et al., 2016; Kipf and Welling, 2017), the extension of convolutional neural networks (CNNs) for specifically graph-based data, has achieved great success in many applications. GCN with non-linear activations is suitable for capturing the complex structures behind the input networks. In addition, stacking multiple GCN layers leads to the expressive modeling of high-order connectivity which makes the model not limited to focus on local-structure. We thus, for predicting missing HPO annotations, present a GCN-based approach, termed HPOFiller, to utilize three types of input networks. In particular, we design two kinds of GCN blocks: (i) S-GCN on PPI network and HPO similarity network, respectively, that aggregates feature information from neighbors, considering edge weights, to obtain better representations; (ii) Bi-GCN on the protein–phenotype bipartite network that allows feature information interchanging between proteins and HPO terms. It is noteworthy that we adopt HPO semantic similarity rather simply HPO binary hierarchy to enable the information to flow between the siblings or ancestor-descendant more than strict parent–child, in order that the model can find similar HPO terms more broadly and deeply. During the training stage, we propose an enhanced annotation matrix as the objective goal to relax the label sparsity.

We extensively evaluated the performance of HPOFiller through cross-validation and temporal validation. Specially, we designed an extremely strict cross-validation procedure avoiding any potential information leakage between training and test sets. Experimental results demonstrated that HPOFiller outperformed state-of-the-art methods by large margins under both cross-validation and temporal validation. Particularly, the ablation study revealed that batch normalization contributed the most to the performance. In addition, we confirmed literature evidence for predictions highly ranked by HPOFiller but not yet been added to the latest HPO annotations database, implying potentially performance under-estimation. Finally, using disease-HPO term associations, HPOFiller found promising, unknown disease–gene associations, presenting the predictability of our method for possible genetic causes of human diseases.

Downloaded from https://academic.oup.com/bioinformatics/article/37/19/3328/6212045 by National Science & Technology Library user on 03 October 2022

# 2 Related work

Completing protein-HPO term associations has been tackled mainly by label propagation-based and matrix completion-based approaches.

A well-known assumption is that similar proteins tend to be related to similar abnormal phenotypes, which is consistent with smoothness assumption in label propagation (LP) (Zhu et al., 2003). Petegrosso et al. (2017) extended vanilla LP (Zhou et al., 2003) to dual label propagation (DLP) by coupling smoothness term imposing smoothness in PPI network and another term imposing smoothness in the HPO hierarchy, which encouraged directly connected phenotypes to be associated with the same protein. DLP was further extended to tlDLP (Petegrosso et al., 2017) by adopting transfer learning. It incorporated GO annotations and let proteins with similar functions be likely to be associated with similar phenotypes.

Matrix completion captured intrinsic relationships between proteins and phenotypes in a latent space. Typically, AiProAnnotator (Gao et al., 2018) imposed graph Laplacians on both PPI network and HPO similarity to standard matrix completion (SMC) over the protein–phenotype matrix to find better low-rank approximation solution.

In general, the above methods were not competent enough to capture non-linear relations underlying protein–phenotype associations. Recently, GCN (Defferrard et al., 2016; Kipf and Welling, 2017) has opened a new paradigm for graph learning and achieved great success in numerous fields, such as disease gene prioritization (Han et al., 2019), polypharmacy side effects prediction (Zitnik et al., 2018) and drug repurposing (Wang et al., 2020), etc. We here apply GCN to identifying missing protein–phenotype associations, and to the best of our knowledge, this is the first work based on GCN for this problem.

# 3 Materials and methods

## 3.1 Problem statement

Given m proteins P ¼ fp1;p2;...;pmg and n HPO terms

T ¼ ft1;t2;...;tng, the known associations between them are represented by a binary matrix Y~, where Y~ ij ¼ 1 if protein pi is annotated by HPO term tj, otherwise Y~ ij ¼ 0. However, Y~ ij ¼ 0 does not mean that there must be no relation between pi and tj, but only that this link has not been observed yet. Our objective is to identify those missing HPO annotations (Fig. 2). Specifically, for protein pi, we want to find the HPO term tj that Y~ ij ¼ 0 but tj may potentially be related to pi. It is noteworthy that we are not to predict annotations of novel proteins (i.e. proteins without any known annotations) but rather to identify the missing annotations of those proteins with known (but incomplete) annotations.

## 3.2 Key idea

We have two types of building blocks: proteins and HPO terms. Our main procedure has two steps: (i) The two types of building blocks are first combined together as a bipartite graph through HPO annotations to preliminarily estimate the embeddings in the latent space, (ii) which are then further refined by using similarity networks separately for each type of building blocks. To be more specific, HPOFiller has two GCN modules: Bi-GCN and S-GCN. Bi-GCN first merges the information from both proteins and phenotypes through HPO annotation bipartite network to estimate latent representations, which are further refined by S-GCN separately for proteins and HPO terms, particularly by using edge weights over protein (and HPO terms) similarity network. We repeat this main procedure and the resultant embeddings are transformed into lowdimensional vectors through multi-layer perceptron, separately for proteins and HPO terms, to be taken for prediction. Figure 3 illustrates the pipeline of this process.

## 3.3 Graph construction

### 3.3.1 Protein-HPO term bipartite graph

We construct a bipartite graph with m protein nodes and n HPO term nodes for describing protein-HPO term associations. If a protein has been annotated with an HPO term, an edge is added to link them. Formally, we denote its adjacency matrix A 2 f0;1gðmþnÞðmþnÞ as

0 Y~

A ¼ ½ T : (1) Y~ 0

where Y~ 2 f0;1gmn is the known annotation matrix, and 0 is allzero matrix.

### 3.3.2 Similarity of proteins

The PPI network has been demonstrated as one of the most informative data sources in the HPO prediction problem (Kahanda et al., 2015; Liu et al., 2020). We utilize STRING (Szklarczyk et al., 2019) to quantify the similarity of two proteins. The protein similarity graph is denoted by Sp 2 Rmm with entries being interacting scores.

### 3.3.3 Similarity of HPO terms

The HPO terms are organized as a Directed Acyclic Graph (DAG), where each term can have multiple parents and multiple children. Petegrosso et al. (2017) assumed that the connected phenotypes (parent–child pairs) were likely to be associated with the same protein. However, the flow of information was strictly restricted to these parent–child edges, and hence it hinders from finding similar phenotypes in different branches. To address this issue, we compute the semantic similarity between HPO terms by using the information coefficient (SimIC) measure (Li et al., 2010), which is based on the Information Content (IC) (Resnik, 1995) for HPO term t:

Downloaded from https://academic.oup.com/bioinformatics/article/37/19/3328/6212045 by National Science & Technology Library user on 03 October 2022

Nt

ICðtÞ ¼ log ; (2) N

where N is the total number of proteins and Nt is the number of proteins annotated by term t and all its descendants. Then the SimIC is defined as:

2 ICðt Þ 1

simICðt1;t2Þ ¼ MICA 1 ; (3)

ICðt1Þ þ ICðt2Þ 1 þ ICðtMICAÞ

where tMICA is the Most Informative Common Ancestor (MICA) (Resnik, 1999) of t1 and t2, i.e. the common ancestor with the high-

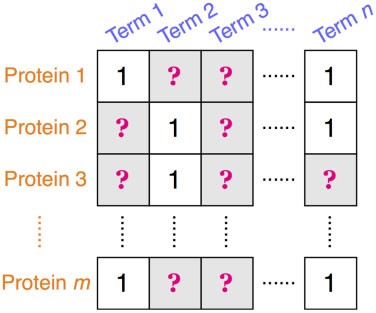


Fig. 2. An illustration of the prediction of missing HPO annotations problem. An entry of 1 indicates the association between the corresponding protein and HPO term is known, and an entry filled with question mark means an unobserved relationship. Our goal is to figure out which unidentified annotations may be true

est IC. Here, we denote the HPO semantic similarity graph as St 2 Rnn with entries being their information coefficients (Fig. 4).

## 3.4 Feature generation

Each row of protein similarity matrix Sp and HPO term similarity St is able to act as the feature vector in fact, however, they may not sufficiently capture the network structure, especially non-neighbouring, higher-order connectivity. On this account, we run Random Walk with Restart (RWR) (Tong et al., 2006) separately on Sp and St to introduce topological context of each node into their initial vector representations. The procedure can be formulated as the following recurrence equation:

pitþ1 ¼ ð1 aÞptiS^ þ aei; (4)

|  |
| --- |
| Fig. 3. The overall framework of HPOFiller. The input features generated by random walk with restart are transformed into low-dimensional representations at first. After that, we stack two modules to be comprised of Bi-GCN and S-GCN to refine the feature vectors. Finally, three fully connected layers are used to reduce the dimensions and output the final representations. The prediction is made by multiplying protein’s and HPO term’s representations. Batch normalization is added between two consecutive layers |

where pti is a row vector of node i, whose kth entry indicates the probability of reaching node k after t steps. The initial probabilities p0i is one-hot vector ei where ei;i ¼ 1 and 0 otherwise. a is the restart probability. S^ is the one-step probability transition matrix obtained from S (i.e. Sp or St) by row-wise normalization. Here, S refers to Sp (or St). After obtaining the steady state, we set feature vector xpi ¼

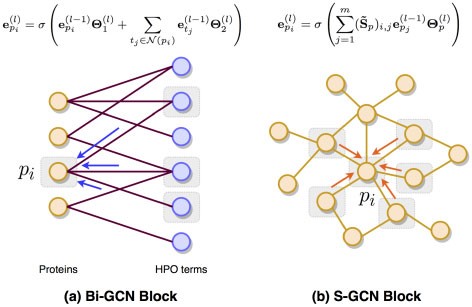


Fig. 4. Schematic information propagation in two types of GCN blocks. (a) Bi-GCN block is run on protein-HPO term association graph with bipartite structure. The embedding for protein pi (yellow node highlighted by gray box) is generated by aggregation of incoming messages from the connected HPO terms. (b) S-GCN block is run over similarity graph. In the illustration of message propagation on protein similarity graph, the output embedding of protein pi (central node with gray box) is obtained by the weighted sum over the messages from its connected proteins

p1i on Sp (or xtj ¼ p1j on St) for protein pi (or HPO term tj), capturing high-order interactions of network nodes.

## 3.5 GCN blocks

### 3.5.1Bi-GCN layer

As shown in Fig. 4(A), Bi-GCN refines the embeddings of proteins and HPO terms by communicating information between proteins and HPO terms. That is, Bi-GCN propagates the embedding over protein-HPO term bipartite graph A. Let us take protein pi as an example, in the lth layer, the process can be formulated as:

eðpliÞ ¼ repðli1ÞH1ðlÞ þ X eðtjl1ÞH2ðlÞ: (5)

tj2NðpiÞ

The above equation can be viewed as two steps. First, we construct messages for pi’s neighboring nodes (i.e. its annotated HPO

terms) and itself, namely etðjl1ÞH2ðlÞ and epðli1ÞH1ðlÞ, respectively. Here, epðli1Þ 2 RdGCNðl1Þ;etðjl1Þ 2 RdGCNðl1Þ denote the node embeddings of pi and tj in the ðl 1Þth layer, respectively. H1ðlÞ 2 RdðGCNl1ÞdðGCNlÞ ;

HðlÞ dðGCNl1ÞdðGCNlÞ are the trainable weight matrices. Then we aggre-

2 R

2

gate the incoming messages by summing over all neighbors NðpiÞ and pi itself, and pass the accumulated message to an activation function rðÞ. Note that we take the self-connection of pi into consideration in order to retain the information of original features.

The representation etðjlÞ for term tj can be obtained analogously. To summarize, Bi-GCN allows to combine the information of proteins and HPO terms explicitly.

#### 3.5.2 S-GCN layer

Unlike Bi-GCN running on unweighted graph, S-GCN is designed to make better use of the information lying in the weights on the similarity network (Fig. 4(B)). Specifically, taking PPI network Sp as an example, we define the lth S-GCN layer for protein pi as:

### Xm

eðpliÞ ¼ r ðS~pÞi;jepðlj1ÞHpðlÞ ; (6)

j¼1

where S~p ¼ Dp012S0pDp012 is the symmetric normalized adjacency matrix of S0p ¼ Sp þ I with inserted self-loops, and ðD0pÞii ¼ Pj ðS0pÞij is diagonal degree matrix. Eq. (6) can also be viewed as two-steps operation: the message of pi’s neighboring node pj is generated by epðlj1ÞHðplÞ at first, and then those messages are summed up with the edge weights of S0p and fed into an activation function rðÞ.

HðplÞ 2 RdðGCNl1ÞdðGCNlÞ is the parameter matrix to learn. If we stack proteins’ embeddings vertically, we can rewrite Eq. (6) in matrix form:

EðplÞ ¼ r Dp012S0pDp012Eðl1ÞHpðlÞÞ: (7)

This equation is consistent with that in (Kipf and Welling, 2017). We can produce embedding eðtjlÞ for term tj in an analogous way. The S-GCN layer combines the information of neighbors based on their contributions, which allows the weights (the most important part of similarity graph) to incorporate into our model.

Downloaded from https://academic.oup.com/bioinformatics/article/37/19/3328/6212045 by National Science & Technology Library user on 03 October 2022

## 3.6 Model architecture

Figure 3 shows the entire architecture of HPOFiller. Taking the input of proteins’ feature vectors Xp 2 Rmm derived from RWR on protein similarity graph Sp and HPO terms’s feature vectors Xt 2 Rnn derived from RWR on phenotype similarity graph St, we first feed them to a dense layer, respectively, to reduce the dimension to the same. The resultant vectors are fed into Bi-GCN and then SGCN. In Bi-GCN, the two separate information of proteins and HPO terms can be combined by passing messages across them along the edges (i.e. known annotations) in the bipartite graph. This process of the lth layer can be written as follows:

½EðplÞ;EðtlÞ ¼ BNðlÞ Bi GCNðlÞ ½Epðl1Þ;Etðl1Þ : (8)

Those vectors are then fed to corresponding S-GCN blocks, respectively, to refine the embeddings by leveraging the weights of similarity network. Formally speaking, the output is computed as follows:

### 

EðplÞ ¼ BNðplÞ S GCNðplÞ EðplÞ ;

(9)

EðtlÞ ¼ BNðtlÞ S GCNðtlÞ EðtlÞ :

By repeatedly performing above operations, we finally obtain the embeddings of proteins and HPO terms that can sufficiently capture the information on two similarity networks and known proteinHPO term annotations. Subsequently, those GCN-generated embeddings are fed to a three-layers perceptron to distill low-dimensional representations, separately for proteins and HPO terms. Specifically, we have final representations as follows:

Uð3Þ ¼ Denseðp3Þ BNðp2Þ Denseðp1Þ Epð2Þ ;

(10)

Vð3Þ ¼ Denseðt3Þ BNtð2Þ Denseðt1Þ Etð2Þ :

Lastly, the probability of protein pi being annotated with HPO term tj can be predicted by

yi;j ¼ uði3Þvjð3ÞT: (11)

It is noteworthy that we add Batch Normalization (BN) module (Ioffe and Szegedy, 2015) between two consecutive layers in order to mitigate internal covariate shift and thus increase the stability. Given a batch of input vectors B ¼ fx1;...;xmg, the empirical mean and variance are computed as:

|  |  |
| --- | --- |
| B i B m i B m  i¼1 i¼1  After normalization by re-centering and re-scaling: |  |
| x x^i ¼ ~~q~~i~~ffiffiffiffiffiffiffiffiffiffiffiffiffi~~ lB ; | (13) |

l ¼ 1 Xm 2 1 Xm x l Þ2: (12)

x and r ¼ð

r2B þ

where is an arbitrarily small constant for numerical stability, the BN transformed output is obtained by

x0i ¼ cx^i þ b BNðxiÞ; (14)

where c and b are subsequently learned in the optimization process.

From our experiments, we could see that BN contributes greatly to our model (see Section 4.4.2).

## 3.7 Model training

We adopt classical loss function that minimizes the Frobenius norm of the difference between known annotation matrix and predicted matrix, while the high sparsity of HPO annotations hinders the straight-forward application of Y~. To alleviate the problem, inspired by (Han et al., 2019), we propose the -enhanced loss function which controls the margin between the predicted score and the label with hyper-parameter . That is, we use the enhanced annotation matrix Y~ 0 as the target:

|  |  |
| --- | --- |
| Y~ 0i;j ¼ f0 otherwiseif Y~ i;j ¼ 1:;  Accordingly, the loss function can be written as: | (15) |
| ‘ ¼k XðYY~Þjj2F þ kjjHjj22; | (16) |

where X is the mask of observed entries: Xij ¼ 1 when Y~ i;j is in the training set and 0 otherwise, 8 denotes the Hadamard product (a.k.a element-wise product), and k is the decay factor to balance the regularization term of all trainable model parameters H in order to prevent overfitting. Through properly tuning by grid search, we enlarge the margin between the predicted score and the label to improve the influence of relatively few positive samples.

# 4 Experiments

## 4.1 Data

We examined the performance of HPOFiller by two evaluation manners: (i) cross-validation and (ii) temporal validation.

### 4.1.1 Data preparation for cross-validation

We downloaded human gene-HPO term associations released by February 12, 2019 from HPO project website ([http://compbio.char ite.de/jenkins/job/hpo.annotations/)](http://compbio.charite.de/jenkins/job/hpo.annotations/). Then the genes in raw HPO annotations were mapped into proteins using the UniProt ID mapping tool (<https://www.uniprot.org/mapping/>). To keep a high data quality, we filtered out proteins that were not stored in Swiss-Prot. The true-path-rule was applied to propagate annotations. In this work, we focused on the biggest sub-ontology in HPO, Phenotypic Abnormality (PA). Therefore, only HPO terms belonging to PA remained. After processing, the dataset consisted of 3 884 proteins and 8 289 HPO terms. Note that we only remained the terms currently used to annotate at least one protein.

We conducted 10-fold cross-validation in this work. Specifically, we randomly split all protein-HPO term pairs into ten equal-sized parts, where one was held out for test set, and the remaining nine parts constituted the training set. However, it would lead to potential information leakage: if a known HPO annotation (p, t) appears in the test set, while the descendant of t named t0 is put into the training set, then we can imply the relation between p and t by simply propagating the known annotation ðp;t0Þ according to the truepath-rule. To plug the loophole, for each pair of annotations between protein p and HPO term t in the test set, the associations between p and all the descendants of t were all removed from the training set. Despite such processing in (Gao et al., 2018; Petegrosso et al., 2017), the information leakage still existed. Considering a negative sample (p, t) in the test set that protein p has no known relation with HPO term t, if there exists a negative annotation ðp;t0Þ in the training set where t0 is the ancestor of t, then we can derive the negative associations (p, t) by propagating negative annotation downward. Therefore, for each negative association (p, t) in the test set, we further removed the negative annotations between p and the ancestors of t in the training set. It is noteworthy that in the crossvalidation, we set Xij ¼ 1 for the training set and Xij ¼ 0 for the test set.

Downloaded from https://academic.oup.com/bioinformatics/article/37/19/3328/6212045 by National Science & Technology Library user on 03 October 2022

Table 1. Statistics of dataset used for temporal validation

|  |  |  |  |
| --- | --- | --- | --- |
| Proteins | HPO terms | Training set | Test set |
| 3884 | 8797 | Before February 12,  2019 | February 12, 2019 to  June 8, 2020 |
|  |  | 474 487 pos.  (1.39%) | 71 835 pos. (0.21%)  33 621 226 neg. (98.40%) |

Note: ‘pos.’ refers to positive sample, while ‘neg.’ refers to negative sample.

For the PPI network, we downloaded STRING v11 ([https:// string-db.org/)](https://string-db.org/) released by January 19, 2019. For the HPO semantic similarity, to avoid information leakage, we calculated based on the HPO annotations in the training set rather than the whole set.

4.1.2 Data preparation for temporal validation In the temporal validation, we adopted a similar strategy as proposed in the CAFA challenge (Jiang et al., 2016; Radivojac et al., 2013). The training set comprised HPO annotations released by February 12, 2019, and the test set comprised the new annotations added from February 12, 2019 to June 8, 2020. HPO annotations in the test set were aligned to February 12, 2019 version and thus the newly created HPO terms were discarded. The statistics of the dataset are shown in Table 1. In order to avoid information leakage, we adopted STRING v11 which released before February 12, 2019 and computed HPO similarity using the training dataset. Note that we set Xij ¼ 1 for all i ¼ 1;...;m and j ¼ 1;...;n.

## 4.2 Evaluation metrics

We used three metrics for evaluating pairs of protein-HPO term associations in the test set: (i) AUC: area under the receiver-operating characteristic curve, (ii) AUPR: area under the precision-recall curve and (iii) AP@K: average precision at the top K measuring the precision at all ranks before position K that hold a relevant item, which was adopted by (Krichene and Rendle, 2020; Li et al., 2019; Zitnik et al., 2018), that is,

1 XK

AP@K ¼ dði 2 RÞP@K; (17) minðjRj;KÞ i¼1

where R refers to the set of rankings of all relevant items, jRj is the size of R, dði 2 RÞ ¼ 1 if the ith prediction is correct and 0 otherwise, P@K is the precision at position K measuring the fraction of relevant items among the top K predicted items:

jfr 2 R : r Kgj

P@K ¼ : (18)

K

Additionally, we evaluated the performance separately on the leaf HPO terms (i.e. the specific annotations) and the internal HPO terms (i.e. the ancestors of specific terms), named AUC-leaf, AUPRleaf, AUC-internal and AUPR-internal.

## 4.3 Competing methods and implementation details

We evaluated the performance of HPOFiller against six state-of-theart methods which were introduced in Section 2: LP, DLP

(Petegrosso et al., 2017), tlDLP-BP and tlDLP-MF (Petegrosso et al., 2017), SMC and AiProAnnotator (Gao et al., 2018). Note that tlDLP used GO annotations of either biological process (tlDLP-BP) or molecular function (tlDLP-MF). Hyperparameters of each method were determined by internal 10-fold cross-validation with grid search. We were extremely careful of information leakage, so the versions of data sources utilized in temporal validation were all early than February 12, 2019.

For our method, we set a ¼ 0:9 in the RWR step empirically (Long et al., 2020). The dimensions of embeddings generated by GCNs were fixed to 800, i.e. dGCNðlÞ ¼ 800 (l¼0, 1, 2), while the sizes of embeddings produced by MLP were set as follows: dMLPð1Þ ¼

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 2. Performance comparison under 10-fold cross-validation   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Method | AUC | AUPR | AUC-leaf | AUPR-leaf | AUC-internal | AUPR-internal | AP@5k | AP@10k | AP@20k | AP@50k | | LP | 0.9318 | 0.3776 | 0.7903 | 0.2837 | 0.9353 | 0.4643 | 0.6426 | 0.5198 | 0.3976 | 0.2446 | | DLP | 0.9319 | 0.3823 | 0.7904 | 0.2872 | 0.9355 | 0.4694 | 0.6570 | 0.5304 | 0.4051 | 0.2492 | | tlDLP-BP | 0.8855 | 0.3557 | 0.7881 | 0.2753 | 0.8797 | 0.4158 | 0.6137 | 0.5051 | 0.3906 | 0.2406 | | tlDLP-MF | 0.9260 | 0.3903 | 0.8169 | 0.2941 | 0.9317 | 0.4765 | 0.6640 | 0.5426 | 0.4181 | 0.2588 | | SMC | 0.8636 | 0.3857 | 0.7542 | 0.3093 | 0.8445 | 0.4179 | 0.7638 | 0.6641 | 0.4858 | 0.2617 | | AiProAnnotator | 0.9461 | 0.3711 | 0.8014 | 0.2960 | 0.9433 | 0.4119 | 0.6600 | 0.5678 | 0.4146 | 0.2212 | | HPOFiller | 0.9288 | 0.4345\* | 0.7693 | 0.3311\* | 0.9356 | 0.5244\* | 0.8347\* | 0.7138\* | 0.5423\* | 0.3109\* |   \*Statistical significance (P<0.001) by pairwise t-test. The boldface items in the table represent the best performance. |

400; dMLPð2Þ ¼ 200 and dMLPð3Þ ¼ 100. We optimized our model with RmsProp optimizer for 3 000 epochs with initial learning rate as 0.0001 and weight decay factor as 1.0, and the learning rate would decrease by half every 1000 epochs. We adopted LeakyReLU activations with negative slope being 0.01 for GCN blocks and ReLU for MLP. We set ¼ 5 in MSE loss function. The model was implemented by PyTorch and PyTorch Geometric.

## 4.4 Results of cross-validation

### 4.4.1 Performance comparison

Table 2 shows the results of 10-fold cross-validation. HPOFiller achieves the best prediction performance with AUPR of 0.4345, which is 11.3% higher than that of the second-best method tlDLPMF which utilizes more information (i.e. GO annotations of MF). The inferior performance of tlDLP implies that multiple data sources might not be properly integrated into their models. Moreover, the AUPR of HPOFiller is 13.7% and 17.1% higher than DLP and AiProAnnotator, respectively, which both use the PPI network and HPO term similarity (despite calculated in different ways). This result demonstrates the effectiveness of GCN to exploit network information. In addition, HPOFiller outperforms others for predicting not only internal annotations but also specific annotations that are more informative. Furthermore, Supplementary Figure S1 shows that HPOFiller keeps the highest precision, except for extremely low recall, indicating that HPOFiller can accurately return the relevant results. Regarding AUC, HPOFiller is moderate, which might be caused by drastic label imbalance (Saito and Rehmsmeier, 2015). As for AP@K where we choose K ¼ 5k;10k;20k;50k, HPOFiller consistently outperforms the competing methods at all K with significant margins.

### 4.4.2 Ablation study

To investigate the effectiveness and necessity of components of HPOFiller, we conduct ablation study by removing one component from the model and evaluate the performance using 10-fold crossvalidation. The results are shown in Figure 5. We notice that AUPR drastically drops by 88.0% without batch normalization, implying its importance on stability. What’s more, removing the output MLP layers also results in performance degradation, indicating the role of distillation. Finally, we observe that models without GCN blocks are all defeated, which again demonstrates the ability to refine highquality representations from the network which in turn benefits the overall performance.

### 4.4.3 Parameter analysis

Downloaded from https://academic.oup.com/bioinformatics/article/37/19/3328/6212045 by National Science & Technology Library user on 03 October 2022

There are many hyper-parameters in our model, each of which will have an influence on the performance. Thus, we conduct parameter sensitivity analysis by varying one hyper-parameter with others fixing. Supplementary Figure S2 shows the test AUPR and AUC w.r.t epoch of HPOFiller. We can see that, as the number of iterations increase, AUPR has converged but AUC begins to slowly decrease, implying the over-fitting. Therefore, we terminate the learning process at 3000 iterations to avoid performance degradation. Supplementary Figure S3 presents the performance changes w.r.t

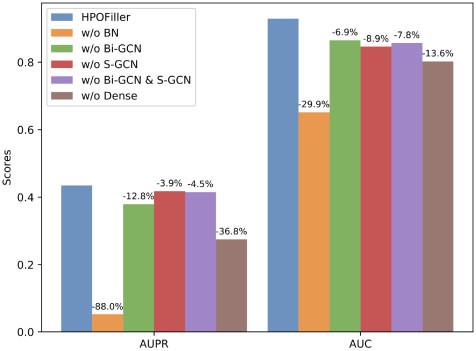


Fig. 5. Ablation study between HPOFiller and its variants derived by removing one of component from the model. The percentage refers to the rate of change in AUPR or AUC by leaving out the particular component relative to that obtained by the full model

multiple hyper-parameters. In particular, continuously increasing the depth of GCN layers leads to over-fitting. This might be caused by applying a too deep architecture might introduce more noises on graph to the representation learning and causes over-smoothing. Moreover, stacking multiple output MLP layers can consistently enhance the predictive performance, but further appending layers also leads to over-fitting. Similar trend also appears in the dimension of the embeddings. All these indicate that too complex models do not necessarily lead to the best performance. As for the restart probability a in RWR, a larger a yields better performance, suggesting that capturing local context may be needed at the preliminary stage. Finally, we investigate the effect of in Eq. (15). From Supplementary Figure S3c, we find that appropriate enhancement on positive samples can mitigate the impact of label imbalance and hence boost the performance.

## 4.5 Results of temporal validation

### 4.5.1 Performance comparison

Table 3 summarizes the results of temporal validation. HPOFiller outperforms all other methods in terms of AUC and AUPR, demonstrating the advantage of our model in predicting missing HPO annotations. Compared with matrix completion-base methods, label propagation-based methods achieve better performance. The matrix completion integrates information underlying networks into lowrank matrices with loss, while GCN would make better use of the network information and hence help improve the performance. It is noteworthy that AUPRs in temporal validation are relatively lower than those in cross-validation. It might be attributed to a lot of annotations in the test set that are still missing, and as a result, the performance is potentially under-estimated (Liu et al., 2020).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 3. Performance comparison under temporal validation   |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Method | | | AUC | | AUPR | | AUC-leaf | | AUPR-leaf | AUC-internal AUPR-internal | | | LP | | | 0.8916 | | 0.0461 | | 0.7800 | | 0.0387 | 0.8694 0.0534 | | | DLP | | | 0.8913 | | 0.0472 | | 0.7797 | | 0.0392 | 0.8694 0.0540 | | | tlDLP-BP | | | 0.8900 | | 0.0472 | | 0.7997 | | 0.0397 | 0.8747 0.0549 | | | tlDLP-MF | | | 0.8885 | | 0.0471 | | 0.8016 | | 0.0391 | 0.8729 0.0540 | | | SMC | | | 0.8326 | | 0.0224 | | 0.7262 | | 0.0194 | 0.8241 0.0246 | | | AiProAnnotator | | | 0.8404 | | 0.0211 | | 0.7329 | | 0.0181 | 0.8306 0.0238 | | | HPOFiller | | | 0.9013 | | 0.0483 | | 0.8046 | | 0.0401 | 0.8804 0.0550 | | | Note: The boldface items in the table represent the best performance.  Table 4. Top predictions of protein–phenotype associations with literature eviden | | | | | | | | | ce |  |  | | Rank | UniProt ID | Gene | | Protein name | | HPO term ID | | HPO term name | | Reference | Evidence | | 32  45  47 | P04637 | TP53 | | Cellular tumor antigen p53 | | HP:0000153  HP:0031816  HP:0000163 | | Abnormality of the mouth  Abnormal oral morphology  Abnormal oral cavity morphology | | Pandya et al. (2018) | ‘Progressive accumulation of genetic errors (including mutations in TP53 and CDKN1A) is associated with the initiation and progression of potentially malignant oral lesions toward frank malignancy.’ | | 4  6  41  94 | P00533 | EGFR | | Epidermal growth factor receptor | | HP:0000707  HP:0012638  HP:0012639  HP:0002011 | | Abnormality of the nervous system  Abnormality of nervous system physiology  Abnormality of nervous system morphology  Morphological abnormality of the central nervous system | | Ahluwalia et al. (2018) | ‘Central nervous system (CNS) metastases are a common complication in patients with epidermal growth factor receptor (EGFR)-mutated non-small cell lung cancer (NSCLC), resulting in a poor prognosis and limited treatment options.’ | | 4263 | P35222 | CTNNB1 | | Catenin beta-1 | | HP:0010461 | | Abnormality of the male genitalia | | Lin et al. (2008) | ‘The fact that both endodermal and ectodermal b-Catenin | | 4665  5280 |  |  | |  | | HP:0000811  HP:0000032 | | Abnormal external genitalia  Abnormality of male external genitalia | |  | knockout animals develop severe hypospadias in both sexes raises the possibility that deregulation of any of these functions can contribute to the etiology of congenital external genital defects in humans.’ | | 4759 | Q6PI48 | DARS2 | | Aspartate-tRNA  ligase,  mitochondrial | | HP:0001252 | | Muscular hypotonia | | Ko¨hler et al. (2015) | ‘At the age of 10months, he showed... no active moving with muscular hypotonia.... A homozygous mutation in the DARS2 gene is most probably the cause of the disease (LBSL).’ | |

4.5.2 Case study 1: top predictions with literature evidence The annotations in the test set for temporal validation are still incomplete, and so a lot of predictions by HPOFiller might be true annotations even if they are not annotated yet. Table 4 presents several top predictions that are not in the HPO annotations released by June 2020 but supported by literature.

The TP53 gene provides instructions for making a protein called cellular tumor antigen p53, which acts as a tumor suppressor to regulate cell division. Pandya et al. (2018) found that p53 protein over-expression and p53 mutations were responsible for dysplastic oral lesions. Recently, Caponio et al. (2020) reports that mutations of TP53 are the most frequent somatic genomic alterations in head and neck squamous cell carcinoma (HNSCC), and more than 90%

Downloaded from https://academic.oup.com/bioinformatics/article/37/19/3328/6212045 by National Science & Technology Library user on 03 October 2022

Table 5. Runtime comparison of different methods under temporal validation

|  |  |
| --- | --- |
| Method | Runtime |
| LP | 1.90s |
| DLP | 588.11s |
| tlDLP-BP | 2764.68s |
| tlDLP-MF | 2914.81s |
| SMC | 1932.56s |
| AiProAnnotator | 3199.52s |
| HPOFiller | 1041.41s |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 6. Top disease–gene associations found by HPOFiller that are newly added to the latest OMIM database   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | Rank | Protein ID | Gene | Protein name | HPO term ID | HPO term name | Disease ID | Disease name | | 114 | P05231 | IL6 | Interleukin-6 | HP:0002408 | Cerebral arteriovenous | OMIM:108010 | Arteriovenous malformations | |  |  |  |  |  | malformation |  | of the brain (BAVM) | | 1323 | Q30201 | HFE | Hereditary hemochromatosis protein | HP:0000726 | Dementia | OMIM:104300 | Alzheimer disease (AD) | | 4032 | P05164 | MPO | Myeloperoxidase | HP:0002423 | Long-tract signs |  |  |   Note: ‘HPO term’ refers to the predicted missing HPO annotation of corresponding protein by HPOFiller. |

of HNSCCs involve the mucosal surfaces of the oral cavity, oropharynx and larynx.

The epidermal growth factor receptor (EGFR) is a transmembrane protein that regulates cell proliferation, apoptosis, angiogenesis, adhesion and metastasis. Ahluwalia et al. (2018) suggested that patients with EGFR-mutated non-small cell lung cancer (NSCLC) were more likely to suffer central nervous system (CNS) metastases.

In addition to the cancers, there are some predictions related to rare phenotypic abnormalities. b-catenin is a dual function protein, involved in regulation and coordination of cell-cell adhesion and gene transcription. Lin et al. (2008) implied that the deregulation of b-catenin could contribute to the etiology of congenital external genital defects in humans based on the experiments on the mice.

The mitochondrial aspartyl-tRNA synthetase is an important enzyme in the synthesis of mitochondria, the energy-producing centers in cells. Ko¨hler et al. (2015) reported a 2.5-year-old baby suffering from leukoencephalopathy with brainstem and spinal cord involvement and lactate elevation (LBSL). He showed muscular hypotonia at the age of 10months. The authors believed that a homozygous mutation in the DARS2 gene is most probably the cause of LBSL.

### 4.5.3 Case study 2: typical example

To demonstrate the practical advantage of HPOFiller, we use menin (UniProt ID: O00255) as a typical example. Menin is the protein product encoded by MEN1 gene, which serves as a putative tumor suppressor associated with multiple endocrine neoplasia type 1. Although menin is believed to be likely implicated in several important cell functions, the exact role of menin is yet to be elucidated (Kamilaris and Stratakis, 2019). Supplementary Figure S4 presents the HPO annotations predicted by different methods. There are 47 newly added HPO annotations of menin, and HPOFiller successfully predicts 17 of them (36.2%), comparing to only 8 for the next-best method. Furthermore, from Supplementary Figure S4, we observe that HPOFiller can find more specific HPO terms, implying the highly positive effect of GCN to capture the semantic relationships between HPO terms from HPO semantic similarity network.

### 4.5.4 Runtime analysis

The runtime of comparing methods is given in Table 5. The experiments are conducted on CentOS 7.5.1804 with Intel(R) Xeon(R) Silver CPU and 256GB RAM, and our model is run on NVIDIA(R) GeForce(R) GTX 1080 Ti GPU. HPOFiller needs half an hour to finish the computation, which is two to three times faster than four out of all six competing methods.

4.5.5 Application to find disease–gene associations We present a further usage of HPOFiller: using known disease-HPO term associations as well as predicted HPO annotations can identify new disease–gene/proteins relationships. We obtain predicted disease-related genes/proteins by building a bridge between HPO annotations of diseases released in February 2019 and predicted proteinHPO term associations generated by HPOFiller. Table 6 lists three top predictions that are added to the latest OMIM gene-disease relationships database. It demonstrates that by using the standardized description of the abnormal phenotypes of the disease by clinicians and other biocurators, HPOFiller can reveal possible genetic causes of diseases.

Downloaded from https://academic.oup.com/bioinformatics/article/37/19/3328/6212045 by National Science & Technology Library user on 03 October 2022

# 5 Conclusion

We presented HPOFiller, a GCN-based approach for identifying missing HPO annotations. The key idea of HPOFiller is to repetitively integrate the information between proteins and HPO terms through protein-HPO term bipartite network by Bi-GCN to provide preliminary embeddings in the latent space, which are then refined by S-GCN on PPI network and HPO semantic similarity network separately. Empirical experiments under stringent conditions showed that HPOFiller significantly outperformed state-of-the-art methods. Besides, we could show evidence from literature for some predicted (unknown) associations, implying the under-estimation of performance. Furthermore, HPOFiller could discover potential disease–gene associations by using known disease-HPO term associations. Due to the utilization of PPI network, HPOFiller was currently limited to gene-coded proteins. Since HPO included a number of disease related non-coding RNAs, we could extend our work to identify missing annotations of non-coding RNAs. Additionally, exploring a more efficient architecture using GCN for predicting HPO annotations would be interesting future work.

# Funding

S.Z. was supported by National Natural Science Foundation of China [61872094], Shanghai Municipal Science and Technology Major Project [2017SHZDZX01] and Information Technology Facility, CAS-MPG Partner Institute for Computational Biology, Shanghai Institute for Biological Sciences, Chinese Academy of Sciences. L.L has been supported by National Natural Science Foundation of China (No. 61872094), Shanghai Municipal Science and Technology Major Project (No.2018SHZDZX01), ZJ Lab, and Shanghai Center for BrainScience and Brain-Inspired Technology. H.M. has been supported partially by Academy of Finland [315896], JST ACCEL [JPMJAC1503], NEXT KAKENHI [19H04169].

Conflict of Interest: none declared.

# References

Ahluwalia,M. et al. (2018) Epidermal growth factor receptor tyrosine kinase inhibitors for central nervous system metastases from non-small cell lung cancer. Oncologist, 23, 1199–1209.

Caponio,V. et al. (2020) Computational analysis of TP53 mutational landscape unveils key prognostic signatures and distinct pathobiological pathways in head and neck squamous cell cancer. Br. J. Cancer, 123, 1302–1314.

Defferrard,M. et al. (2016) Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5–10, 2016, Barcelona, Spain, pp. 3837–3845.

Gao,J. et al. (2018) AiProAnnotator: low-rank Approximation with network side information for high-performance, large-scale human Protein abnormality Annotator. In: IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2018, December 3–6, 2018. IEEE Computer Society, Madrid, Spain, pp. 13–20.

Goh,K. et al. (2007) The human disease network. Proc. Natl. Acad. Sci. USA, 104, 8685–8690.

Han,P. et al. (2019) GCN-MF: Disease–Gene Association Identification By Graph Convolutional Networks and Matrix Factorization. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4–8, 2019. ACM, pp. 705–713.

Ioffe,S. and Szegedy,C. (2015) Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6–11 July 2015, volume 37 of JMLR Workshop and

Conference Proceedings. JMLR.org, pp. 448–456, Lille, France.

Jiang,Y. et al. (2016) An expanded evaluation of protein function prediction methods shows an improvement in accuracy. Genome Biol., 17, 1–19.

Kahanda,I. et al. (2015) PHENOstruct: prediction of human phenotype ontology terms using heterogeneous data sources. F1000Res, 4, 259.

Kamilaris,C. and Stratakis,C. (2019) Multiple Endocrine Neoplasia Type 1 (MEN1): an update and the significance of early genetic and clinical diagnosis. Front. Endocrinol., 10, 339.

Kipf,T. and Welling,M. (2017) Semi-Supervised Classification with Graph Convolutional Networks. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Conference Track Proceedings. OpenReview.net.

Ko¨hler,C. et al. (2015) Infantile manifestation of a mitochondriopathy due to a homozygous mutation in DARS2 gene. Neuropediatrics, 46, FV02–FV07.

Ko¨hler,S. et al. (2009) Clinical diagnostics in human genetics with semantic similarity searches in ontologies. Am. J. Hum. Genet., 85, 457–464.

Ko¨hler,S. et al. (2019) Expansion of the Human Phenotype Ontology (HPO) knowledge base and resources. Nucleic Acids Res., 47, D1018–D1027.

Krichene,W. and Rendle,S. (2020) On sampled metrics for item recommendation. In: KDD ’20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23–27, 2020. ACM, pp. 1748–1757.

Li,B. et al. (2010) Effectively integrating information content and structural relationship to improve the GO-based similarity measure between proteins. In: International Conference on Bioinformatics & Computational Biology, BIOCOMP 2010, July 12–15, 2010, 2 Volumes. CSREA Press, Las Vegas Nevada, USA, pp. 166–172.

Li,Y. et al. (2019) PGCN: disease gene prioritization by disease and gene embedding through graph convolutional neural networks. bioRxiv. (https: //www.biorxiv.org/content/10.1101/532226v1)

Lin,C. et al. (2008) Tissue-specific requirements of b-catenin in external genitalia development. Development, 135, 2815–2825.

Liu,L. et al. (2020) HPOLabeler: improving prediction of human protein–phenotype associations by learning to rank. Bioinformatics, 36, 4180–4188.

Downloaded from https://academic.oup.com/bioinformatics/article/37/19/3328/6212045 by National Science & Technology Library user on 03 October 2022

Long,Y. et al. (2020) Predicting human microbe-drug associations via graph convolutional network with conditional random field. Bioinformatics, 36, 4918–4927.

Oti,M. et al. (2006) Predicting disease genes using protein–protein interactions. J. Med. Genet., 43, 691–698.

Pandya,J. et al. (2018) A correlation of immunohistochemical expression of TP53 and CDKN1A in oral epithelial dysplasia and oral squamous cell carcinoma. J. Cancer Res. Ther., 14, 666.

Petegrosso,R. et al. (2017) Transfer learning across ontologies for phenome-genome association prediction. Bioinformatics, 33, 529–536.

Radivojac,P. et al. (2013) A large-scale evaluation of computational protein function prediction. Nat. Methods, 10, 221–227.

Resnik,P. (1995) Using information content to evaluate semantic similarity in a taxonomy. In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, IJCAI 95, August 20–25 1995, 2 Volumes, pp. 448–453. Morgan Kaufmann, Montre´al Que´bec, Canada.

Resnik,P. (1999) Semantic similarity in a taxonomy: an information-based measure and its application to problems of ambiguity in natural language. J. Artif. Intell. Res., 11, 95–130.

Saito,T. and Rehmsmeier,M. (2015) The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLoS One, 10, e0118432.

Szklarczyk,D. et al. (2019) STRING v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. Nucleic Acids Res., 47, D607–D613.

Tong,H. et al. (2006) Fast Random Walk with Restart and Its Applications. In Proceedings of the 6th IEEE International Conference on Data Mining (ICDM 2006), 18–22 December 2006. IEEE Computer Society, Hong Kong, China, pp. 613–622.

Wang,Z. et al. (2020) Toward heterogeneous information fusion: bipartite graph convolutional networks for in silico drug repurposing.

Bioinformatics, 36, i525–i533.

Zhou,D. et al. (2003) Learning with Local and Global Consistency. In: Advances in Neural Information Processing Systems 16. Neural Information Processing Systems, NIPS 2003, December 8–13, 2003. MIT Press, Vancouver and Whistler, British Columbia, Canada, pp. 321–328.

Zhu,X. et al. (2003) Semi-supervised learning using gaussian fields and harmonic functions. In Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21–24, 2003. AAAI Press, Washington, DC, USA, pp. 912–919.

Zitnik,M. et al. (2018) Modeling polypharmacy side effects with graph convolutional networks. Bioinformatics, 34, i457–i466.