Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Original Paper

OXFORD

Systems

GNE: A deep learning framework for gene network inference by aggregating biological information

Kishan KC [1,*], Rui Li [1], Feng Cui [2] and Anne R. Haake [1]

[1]Golisano College of Computing and Information Sciences, Rochester Institute of Technology, Rochester, 14623, USA and
[2]Gosnell School of Life Sciences, Rochester Institute of Technology, Rochester, 14623, USA

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: The topological landscape of gene interaction networks provides a rich source of information for inferring functional patterns of genes or proteins. However, it is still a challenging task to aggregate heterogeneous biological information such as gene expression and gene interactions to achieve more accurate inference for prediction and discovery of new gene interactions. In particular, how to generate a unified vector representation to integrate diverse input data is a key challenge addressed here.
Results: We propose a scalable and robust deep learning framework to learn embedded representations to unify known gene interactions and gene expression for gene interaction predictions. These low-dimensional embeddings derive deeper insights into the structure of rapidly accumulating and diverse gene interaction networks and greatly simplify downstream modeling. We compare the predictive power of our deep embeddings to the state-of-the-art machine learning methods. The results suggest that our deep embeddings achieve significantly more accurate predictions. Moreover, a set of novel gene interaction predictions are validated by up-to-date literature-based database entries.
Availability: Source code and preprocessed datasets are available at https://github.com/kckishan/GNE under the GNU General Public License.
Contact: kk3671@rit.edu

1 Introduction

A comprehensive study of gene interactions (GIs) provides means to identify the functional relationship between genes and their corresponding products, as well as insights into underlying biological phenomena that are critical to understanding phenotypes in health and disease conditions (Mani *et al.*, 2008; Boucher and Jenna, 2013; Lage, 2014). Since advancements in measurement technologies have led to numerous high-throughput datasets, there is a great value in developing efficient computational methods capable of automatically extracting and aggregating meaningful information from heterogeneous datasets to infer gene interactions.

Although a wide variety of machine learning models have been developed to analyze high-throughput datasets for GI prediction (Madhukar *et al.*, 2015), there are still some major challenges, such as efficient analysis of large heterogeneous datasets, integration of biological information, and effective feature engineering. To address these challenges, we propose a novel deep learning framework to integrate diverse biological information for GI network inference.

Our proposed method frames GI network inference as a problem of network embedding. In particular, we represent gene interactions as a network of genes and their interactions and create a deep learning framework to automatically learn an informative representation which integrates both the topological property and the gene expression property. A key insight behind our gene network embedding method is the "guilt by association" assumption (Oliver, 2000), that is, genes that are co-localized or have similar topological roles in the interaction network are likely to be functionally correlated. This insight not only allows us to discover similar genes and proteins but also to infer the properties of unknown ones. Our network embedding generates a lower-dimensional vector representation of the gene topological characteristics. The relationships between genes including higher-order topological properties are captured by the distances between genes in the embedding space. The new low-dimensional representation of a GI network can be used for various

downstream tasks, such as gene function prediction, gene interaction prediction, and gene ontology reconstruction (Cho *et al.*, 2016).

Furthermore, since the network embedding method can only preserve the topological properties of a GI network, and fails to generalize for genes with no interaction information, our scalable deep learning method also integrates heterogeneous gene information, such as expression data from high throughput technologies, into the GI network inference. Our method projects genes with similar attributes closer to each other in the embedding space, even if they may not have similar topological properties. The results show that by integrating additional gene information in the network embedding process, the prediction performance is improved significantly.

GI prediction is a long-standing problem. The proposed machine learning methods include statistical correlation, mutual information (Marbach *et al.*, 2012), data imputation, matrix completion and network-based methods (e.g. common neighborhood, network embedding) (Madhukar *et al.*, 2015; Cui *et al.*, 2017). Among these methods, some methods such as statistical correlation and mutual information consider only gene expression whereas other methods use only topological properties to predict GIs.

Data imputation techniques, such as weighted KNNImpute and Local Least Squares (LLSImpute), are used for gene expression data for the imputation of missing GIs (Liew *et al.*, 2010). A more systematic matrix approximation method for data imputation is proposed to improve the imputation process on a relatively small GI dataset (26 * 26 matrix) (Järvinen *et al.*, 2008). These methods do not take GI network structure into consideration.

Network-based methods have been proposed to leverage topological properties of GI networks (Lei *et al.*, 2012). Neighborhood-based methods quantify the proximity between genes, based on common neighbors in GI network (Alanis-Lobato *et al.*, 2013). The proximity scores assigned to a pair of genes rely on the number of neighbors that the pair has in common. Adjacency matrix, representing interaction network, or proximity matrix, obtained from neighborhood-based methods, are processed with network embedding methods to learn embeddings that preserve the structural properties of the network. Structure-preserving network embedding methods such as Isomap (Tenenbaum *et al.*, 2000) are proposed as a dimensionality reduction technique. Since the goal of these methods is solely for graph reconstruction, the embedding space may not be suitable for GI network inference. In addition, these methods construct the graphs from the data features where proximity between genes is well defined in the original feature space (Cui *et al.*, 2017). On the other hand, in GI networks, gene proximities are not explicitly defined, and they depend on the specific analytic tasks and application scenarios.

Our deep learning method allows incorporating gene expression data with GI network topological structure information in order to preserve both structural and attribute proximity in the low-dimensional representation for GI predictions. Moreover, the scalable architecture will enable us to incorporate additional attributes. Topological properties of GI network and expression profiles are transformed into two separate embeddings: ID embedding (which preserves the topological structure proximity) and attribute embedding (which preserves the attribute proximity) respectively. With a multilayer neural network, we then aggregate the complex statistical relationships between topology and attribute information to improve GI predictions.

In summary, our contributions are as follows:

- We propose a novel deep learning framework to learn lower dimensional representations while preserving structural and attribute proximity of GI networks.
- We evaluate the prediction performance on the datasets of two organisms based on the embedded representation and achieve significantly better predictions than the state-of-the-art competitor approaches.
- Our method can predict new gene interactions which are validated on an up-to-date GI database.

2 Preliminaries

We formally define the problem of gene network inference as a network embedding problem using the concepts of structural and attribute proximity as demonstrated in Figure 1.

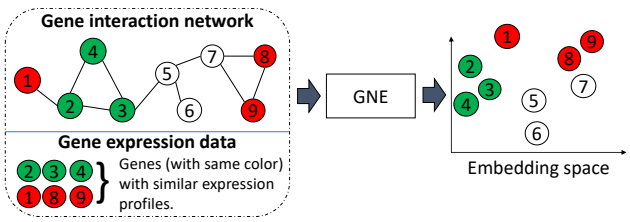

Fig. 1. An illustration of gene network embedding (GNE) to integrate gene interaction network and gene expression data to learn a lower-dimensional representation. The nodes represent genes, and the genes with the same color have similar expression profiles. GNE groups genes with similar network topology, which are connected or have a similar neighborhood in the graph, and attribute similarity (similar expression profiles) in the embedded space.

DEFINITION 1: (**Gene network**) Gene network can be represented as a network structure, which represents the interactions between genes within an organism. The interaction between genes corresponds to either a physical interaction through their gene products, e.g., proteins, or one of the genes alters or affects the activity of other gene of interest. We denote gene network as $G = (V, E, A)$ where $V = \{v_i\}$ denotes genes or proteins, $E = \{e_{ij}\}$ denotes edges that correspond to interactions between genes v_i and v_j, and $A = \{A_i\}$ represents the attributes of gene v_i. Edge e_{ij} is associated with a weight $w_{ij} \geq 0$ indicating the strength of the connection between gene v_i and v_j. If gene v_i and v_j is not linked by an edge, $w_{ij} = 0$. We name interactions with $w_{ij} > 0$ as positive interactions and $w_{ij} = 0$ as negative interactions. In this paper, we consider weights w_{ij} to be binary, indicating whether genes v_i and v_j interact or not.

Genes directly connected with a gene v_i in gene network denote the local network structure of gene v_i. We define local network structures as the first-order proximity of a gene.

DEFINITION 2: (**First-order proximity**) The first-order proximity in a gene network is the pairwise interactions between genes. Weight w_{ij} indicates the first-order proximity between gene v_i and v_j. If there is no interaction between gene v_i and v_j, their first-order proximity w_{ij} is 0.

Genes are likely to be involved in the same cellular functions if they are connected in the gene network. On the other hand, even if two genes are not connected, they may be still related in some cellular functions. This indicates the need for an additional notion of proximity to preserve the network structure. Studies suggest that genes that share a similar neighborhood are also likely to be related (Cho *et al.*, 2016). Thus, we introduce second-order proximity that characterizes the global network structure of the genes.

DEFINITION 3: (**Second-order proximity**) Second order proximity denotes the similarity between the neighborhood of genes. Let $N_i = \{s_{i,1},\ldots, s_{i,i-1}, s_{i,i+1},\ldots, s_{i,M-1}\}$ denotes the first-order proximity

Table 1. Terms and Notations

Symbol	Definitions
M	Total number of genes in gene network
E	Number of expression values for each gene
N_i	Set of the neighbor genes of gene v_i
$\mathbf{v}_i^{(s)}$	Structural representation of gene v_i
$\mathbf{v}_i^{(a)}$	Attribute representation of gene v_i
$\widetilde{\mathbf{v}}_i$	Neighborhood representation of gene v_i
$\mathbf{v}_i$	Concatenated representation of topological properties and expression data
k	Number of hidden layers to transform concatenated representation into embedding space
$\mathbf{h}^{(k)}$	Output of k^{th} hidden layer
$\mathbf{W}_k$	Weight matrix for k^{th} hidden layer
$\mathbf{W}_{id}$	Weight matrix for topological structure embedding
$\mathbf{W}_{att}$	Weight matrix for attribute embedding
$\mathbf{W}_{out}$	Weight matrix for output layer

of gene v_i, where $s_{i,j}$ is w_{ij} if there is direct connection between gene v_i and gene v_j, otherwise 0. Then, the second order proximity is the similarity between N_i and N_j. If there is no path to reach gene v_i from gene v_j, the second proximity between these genes is 0.

Integrating first-order and second-order proximities simultaneously can help to preserve topological properties of the gene network. To generate a more comprehensive representation of the genes, it is crucial to integrate gene expression data as gene attributes with their topological properties. Besides preserving topological properties, gene expression provides additional information to predict the network structure.

DEFINITION 4: (**Attribute proximity**) Attribute proximity denotes the similarity between the expression of genes.

We thus investigate both structural and attribute proximity for gene network embedding, which is defined as follows:

DEFINITION 5: (**Gene network embedding**) Given a gene network denoted as $G = (V, E, A)$, gene network embedding aims to learn a function *f* that maps gene network structure and their attribute information to a d-dimensional space where a gene is represented by a vector $y_i \in \mathbb{R}^d$ where $d \ll M$. The low dimensional vectors y_i and y_j for genes v_i and v_j preserve their relationships in terms of the network topological structure and attribute proximity.

3 Gene Network Embedding (GNE)

Our deep learning framework as shown in Figure 2 jointly utilizes gene network structure and gene expression data to learn a unified representation for the genes. Embedding of a gene network projects genes into a lower dimensional space, known as the embedding space, in which each gene is represented by a vector. The embeddings preserve both the gene network structure and statistical relationships of gene expression. We list the variables to specify our framework in Table 1 .

3.1 Gene Network Structure Modeling

GNE framework preserves first-order and second-order proximity of genes in the gene network. The key idea of network structure modeling is to estimate the pairwise proximity of genes in terms of the network structure. If two genes are connected to each other or share the similar neighborhood genes, they tend to be related and should be placed closer to each other in the embedding space. Inspired by the Skip-gram model (Mikolov *et al.*, 2013), we use one hot encoded representation to represent topological information of a gene. Each gene v_i in the network is represented as an M-dimensional vector where only the i^{th} component of the vector is 1.

To model topological similarity, we define the conditional probability of gene v_j on gene v_i using a softmax function as:

$$p(v_j|v_i) = \frac{\exp(f(v_i, v_j))}{\sum_{j'=1}^{M} \exp(f(v_i, v_{j'}))} \tag{1}$$

which measures the likelihood of gene v_i being connected with v_j. Let function f represents the mapping of two genes v_i and v_j to their estimated proximity score. Let $p(N|v)$ be the likelihood of observing a neighborhood N for a gene v. By assuming conditional independence, we can factorize the likelihood so that the likelihood of observing a neighborhood gene is independent of observing any other neighborhood gene, given a gene v_i:

$$p(N_i|v_i) = \prod_{v_j \in N_i} p(v_j|v_i) \tag{2}$$

where N_i represents the neighborhood genes of the gene v_i. Global structure proximity for a gene v_i can be preserved by maximizing the conditional probability over all genes in the neighborhood. Hence, we can define the likelihood function that preserve global structure proximity as:

$$L = \prod_{i=1}^{M} p(N_i|v_i) = \prod_{i=1}^{M} \prod_{v_j \in N_i} p(v_j|v_i) \tag{3}$$

Let $\mathbf{v}_i^{(s)}$ denotes the dense vector generated from one-hot gene ID vector, which represents topological information of that gene. GNE follows direct encoding methods (Mikolov *et al.*, 2013; Grover and Leskovec, 2016) to map genes to vector embeddings, simply known as embedding lookup:

$$\mathbf{v}_i^{(s)} = \mathbf{W}_{id} v_i \tag{4}$$

where $\mathbf{W}_{id} \in \mathbb{R}^{d \times M}$ is a matrix containing the embedding vectors for all genes and $v_i \in \mathbb{I}_M$ is a one-hot indicator vector indicating the column of $\mathbf{W}_{id}$ corresponding to gene v_i.

3.2 Gene Expression Modeling

GNE encodes the expression data from microarray experiments to the dense representation using a non-linear transformation. The amount of mRNA produced during transcription measured over a number of experiments helps to identify similarly expressed genes. Since expression data have inherent noise (Tu *et al.*, 2002), transforming expression data using a non-linear transformation can be helpful to uncover the underlying representation. Let x_i be the vector of expression values of gene v_i measured over E experiments. Using non-linear transformation, we can capture the non-linearities of expression data of gene v_i as:

$$\mathbf{v}_i^{(a)} = \delta_a(\mathbf{W}_{att} \cdot x_i) \tag{5}$$

where $\mathbf{v}_i^{(a)}$ represents the lower dimensional attribute representation vector for gene v_i. $\mathbf{W}_{att}$, and δ_a represents the weight matrix, and activation function of attribute transformation layer respectively.

We use the deep model to approximate the attribute proximity by capturing complex statistical relationships between attributes and introducing non-linearities, similar to structural embedding.

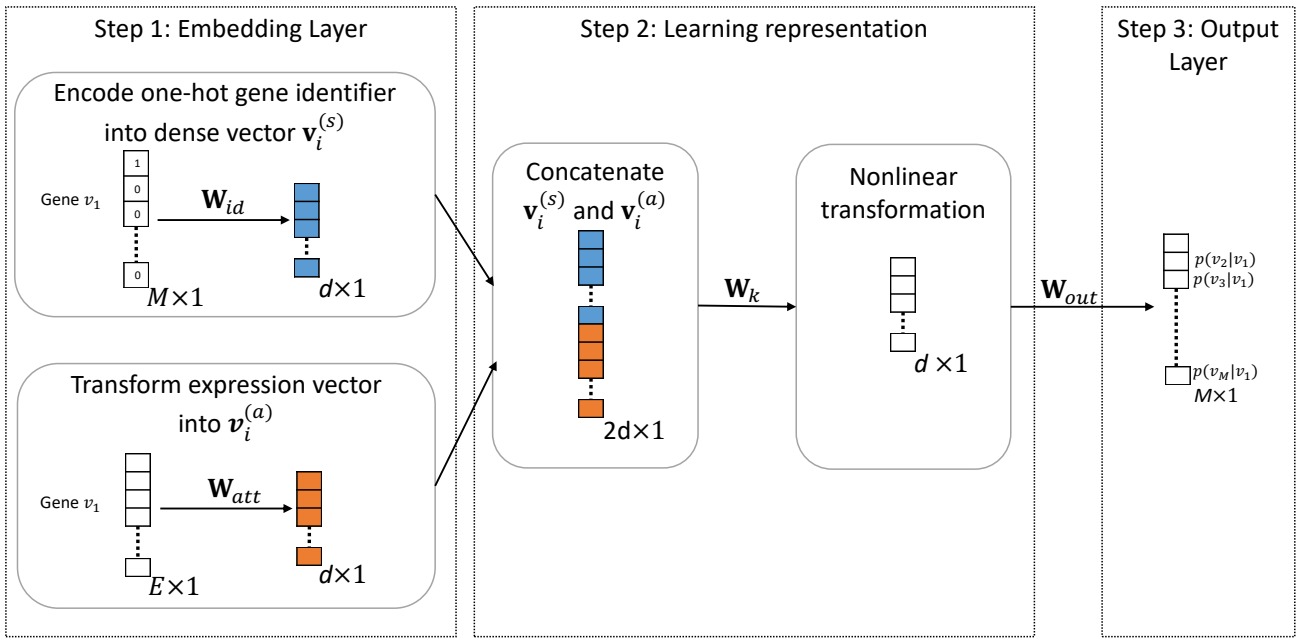

Fig. 2. Overview of Gene Network Embedding (GNE) Framework for gene interaction prediction. On the left,one-hot encoded representation of gene is encoded to dense vector $\mathrm{v}_i^{(s)}$ of dimension $d \times 1$ which captures topological properties and expression vector of gene is transformed to $\mathrm{v}_i^{(a)}$ of dimension $d \times 1$ which aggregates the attribute information (Step 1). Next, concatenation of two embedded vectors (creates vector with dimension $2d \times 1$) allows to combine strength of both network structure and attribute modeling. Then, nonlinear transformation of concatenated vector enables GNE to capture complex statistical relationships between network structure and attribute information and learn better representations (Step 2). Finally, these learned representation of dimension $d \times 1$ is transformed into a probability vector of length $M \times 1$ in output layer, which contains the predictive probability of gene v_i to all the genes in the network. Conditional probability $p(v_j|v_i)$ on output layer indicates the likelihood that gene v_j is connected with gene v_i (Step 3).

3.3 GNE Integration

GNE models the integration of network structure and attribute information to learn more comprehensive embeddings for gene networks. GNE takes two inputs: one for structural information of a gene as one hot gene ID vector and another for its expression as an attribute vector. Each input is encoded to its respective embeddings. One hot representation for a gene v_i is projected to the dense vector $\mathbf{v}_i^{(s)}$ which captures the topological properties. Non-linear transformation of attribute vector generates compact representation vector $\mathbf{v}_i^{(a)}$.

Previous work (Tang *et al.*, 2015) combines heterogeneous information using the late fusion approach. However, the late fusion approach is the approach of learning separate models for heterogeneous information and integrating the representations learned from separate models. On the other hand, the early fusion combines heterogeneous information and train the model on combined representations (Snoek *et al.*, 2005). We thus propose to use the early fusion approach to combine them by concatenating. As a result, learning from structural and attribute information can complement each other, allowing the model to learn their complex statistical relationships as well. Embeddings from structural and attribute information are concatenated into a vector as:

$$\mathbf{v}_i = [\mathbf{v}_i^{(s)} \quad \lambda \mathbf{v}_i^{(a)}] \tag{6}$$

where λ is the importance of gene expression information relative to topological information.

The concatenated vectors are fed into a multilayer perceptron with k hidden layers. The hidden representations from each hidden layer in GNE are denoted as $\mathbf{h}_i^{(0)}, \mathbf{h}_i^{(1)}, ..., \mathbf{h}_i^{(k)}$, which can be defined as :

$$\begin{aligned} \mathbf{h}_i^{(0)} &= \delta(\mathbf{W}_0 \mathbf{v}_i + b^{(0)}), \\ \mathbf{h}_i^{(k)} &= \delta_k(\mathbf{W}_k \mathbf{h}_i^{(k-1)} + b^{(k-1)}) \end{aligned} \tag{7}$$

where δ_k represents the activation function of layer k. $\mathbf{h}_i^{(0)}$ represents initial representation and $\mathbf{h}_i^{(k)}$ represents final representation of the input gene v_i. Transformation of input data using multiple non-linear layers has shown to improve the representation of input data (He *et al.*, 2016). Moreover, stacking multiple layers of non-linear transformations can help to learn high-order statistical relationships between topological properties and attributes.

At last, final representation $\mathbf{h}_i^{(k)}$ of a gene v_i from the last hidden layer is transformed to probability vector, which contains the conditional probability of all other genes to v_i:

$$\mathbf{o}_i = [p(v_1|v_i), p(v_2|v_i), \ldots, p(v_M|v_i)] \tag{8}$$

where $p(v_j|v_i)$ represents the probability of gene v_i being related to gene v_j and $\mathbf{o}_i$ represents the output probability vector with the conditional probability of gene v_i being connected to all other genes.

Weight matrix $\mathbf{W}_{out}$ between the last hidden layer and the output layer corresponds to the abstractive representation of neighborhood of genes. A j^{th} row from $\mathbf{W}_{out}$ refers to the compact representation of neighborhood of gene v_j, which can be denoted as $\widetilde{\mathbf{v}}_j$. The proximity score between gene v_i and v_j can be defined as:

$$f(v_i, v_j) = \widetilde{\mathbf{v}}_j \cdot \mathbf{h}_i^{(k)} \tag{9}$$

which can be replaced into Equation 1 to calculate the conditional probability:

$$p(v_j|v_i) = \frac{\exp(\widetilde{\mathbf{v}}_j \cdot \mathbf{h}_i^{(k)})}{\sum_{j'=1}^{M} \exp(\widetilde{\mathbf{v}}_{j'} \cdot \mathbf{h}_i^{(k)})} \tag{10}$$

Our model learns two latent representations $\mathbf{h}_i^{(k)}$ and $\widetilde{\mathbf{v}}_i$ for a gene v_i where $\mathbf{h}_i^{(k)}$ is the representation of gene as a node and $\widetilde{\mathbf{v}}_i$ is the

representation of the gene v_i as a neighbor. These two representations are added to get final representation for a gene as:

$$y_i = \mathbf{h}_i^{(k)} + \widetilde{\mathbf{v}}_i \tag{11}$$

For an edge connecting gene v_i and v_j, we create feature vector by combining embeddings of those genes using Hadamard product. Empirical evaluation shows features created with Hadamard product gives better performance over concatenation (Grover and Leskovec, 2016). Then, we train a logistic classifier on these features to classify whether genes v_i and v_j interact or not.

3.4 Parameter Optimization

To optimize GNE, the goal is to maximize objective function mentioned in Equation 10 as a function of all parameters. Let Θ be the parameters of GNE which includes $\{\mathbf{W}_{id}, \mathbf{W}_{att}, \mathbf{W}_{out}, \Theta_h\}$ and Θ_h represents weight matrices $\mathbf{W}_k$ of hidden layers. We train our model to maximize the objective function with respect to all parameters Θ :

$$\begin{aligned}
\Theta^* &= \underset{\Theta}{\operatorname{argmax}} \left[\log \prod_{i=1}^{M} \prod_{v_j \in N_i} p(v_j|v_i) \right] \\
&= \underset{\Theta}{\operatorname{argmax}} \left[\sum_{i=1}^{M} \sum_{v_j \in N_i} \log\ p(v_j|v_i) \right] \\
&= \underset{\Theta}{\operatorname{argmax}} \left[\sum_{i=1}^{M} \sum_{v_j \in N_i} \log \frac{\exp(\widetilde{\mathbf{v}}_j \cdot \mathbf{h}_i^{(k)})}{\sum_{j'=1}^{M} \exp(\widetilde{\mathbf{v}}_{j'} \cdot \mathbf{h}_i^{(k)})} \right] \\
&= \underset{\Theta}{\operatorname{argmax}} \sum_{i=1}^{M} \sum_{v_j \in N_i} \left[(\widetilde{\mathbf{v}}_j \cdot \mathbf{h}_i^{(k)}) - \log\ Z_i \right]
\end{aligned} \tag{12}$$

where $Z_i = \sum_{j'=1}^{M} \exp(\widetilde{\mathbf{v}}_{j'} \cdot \mathbf{h}_i^{(k)})$ is the per-gene partition function.

Using Equation 11, the objective function can be simplified as :

$$\Theta^* = \underset{\Theta}{\operatorname{argmax}} \sum_{i=1}^{M} \sum_{v_j \in N_i} \left[f(v_i, v_j) - \log\ Z_i \right] \tag{13}$$

Maximizing this objective function with respect to parameters Θ is computationally expensive, which requires the calculation of partition function Z_i for each gene. In order to calculate a single probability, we need to aggregate all genes in the network. To address this problem, we adopt the approach of negative sampling (Mikolov *et al.*, 2013) which samples the negative interactions, interactions with no evidence of their existence, according to some noise distribution for each edge e_{ij}. This approach allows us to sample small subset of genes from the network as negative samples for a gene, considering that the genes on selected subset don't fall in the neighborhood N_i of the gene. Above objective function enhances the similarity of a gene v_i with its neighborhood genes $v_j \in N_i$ and weakens the similarity with genes not in its neighborhood genes $v_j \notin N_i$. It is inappropriate to assume that two genes in the network are not related if they are not connected. It may be the case that there is not enough experimental evidence to support that they are related yet. Thus, forcing the dissimilarity of a gene with all other genes, not in its neighborhood N_i seems to be inappropriate.

We adopt Adam optimization (Kingma and Ba, 2014), which is an extension to stochastic gradient descent, for optimizing Equation 13. Adam computes adaptive learning rate for each parameter. In each step, Adam algorithm samples mini-batch of interactions and then updates GNE's parameters. To address the issue of overfitting, regularization like dropout (Srivastava *et al.*, 2014) and batch normalization (Ioffe and Szegedy, 2015) is added to hidden layers. Proper optimization of GNE gives the final representation for each gene.

4 Experimental setup

We evaluate our model using two real organism datasets. We take gene interaction network data from BioGRID database[1] (Stark *et al.*, 2006) and gene expression data from DREAM5 challenge (Marbach *et al.*, 2012). We use two interaction datasets from BioGRID database (2017 released version 3.4.153 and 2018 released version 3.4.158) to evaluate the predictive performance of our model. Self-interactions and redundant interactions are removed from interaction datasets. The statistics of the datasets are shown in Table 2.

Table 2. Statistics of the datasets from BioGRID (Stark et al., 2006) and DREAM5 challenge (Marbach et al., 2012).

Datasets	#(Genes)	#(Interactions) 2017 version	#(Interactions) 2018 version	#(Experiments)
Yeast	5,950	544,652	557,487	536
E. coli	4,511	148,340	159,523	805

We evaluate the learned embeddings to infer gene network structure. We randomly hold out a fraction of interactions as the validation set for hyper-parameter tuning. Then, we divide the remaining interactions randomly into training and testing dataset with the equal number of interactions. Since validation and test set contains only positive interactions, we randomly sample an equal number of gene pairs from the network, considering the missing edge between the gene pairs represents the absence of interactions. Given the gene network G with a fraction of missing interactions, the task is to predict these missing interactions.

We compare the GNE model with five competing methods. Correlation directly predicts the interactions between genes based on the correlation of expression profiles. Then, the following three baselines (Isomap, LINE, and node2vec) are network embedding methods. Specifically, node2vec is the state-of-the-art deep learning method for structural network embedding. We evaluate the performance of GNE against the following methods:

- **Correlation (Butte and Kohane, 1999)**
  It computes Pearson's correlation coefficient between all genes and the interactions are ranked via correlation scores, i.e. highly correlated gene pairs receive higher confidence.
- **Isomap (Lei *et al.*, 2012)**
  It computes all-pairs shortest-path distances to create a distance matrix and performs singular-value decomposition of that matrix to learn a lower-dimensional representation. Genes separated by the distance less than threshold ϵ in embedding space are considered to have the connection with each other and the reliability index, a likelihood indicating the interaction between two genes, is computed using FSWeight (Chua *et al.*, 2006).
- **LINE (Tang *et al.*, 2015)**
  Two separate embeddings are learned by preserving first-order and second-order proximity of the network structure respectively. Then, these embeddings are concatenated to get final representations for each node.
- **node2vec (Grover and Leskovec, 2016)**
  It learns the embeddings of the node by applying Skip-gram model

[1] Interaction dataset is downloaded from `http://theBioGRID.org/`.

to node sequences generated by biased random walk. We tuned two hyper-parameters p and q that control the random walk.

Note that the competing methods such as Isomap, LINE, and node2vec are designed to capture only the topological properties of the network. For the fair comparison with GNE that additionally integrates expression data, we concatenate attribute feature vector with learned gene representation to extend baselines by including the gene expression. We name these variants as Isomap+, LINE+, and node2vec+.

The parameter settings for GNE are determined by its performance on the validation set. We randomly initialize GNE's parameters, optimizing with mini-batch Adam. We test the batch size of [8, 16, 32, 64, 128, 256] and learning rate of [0.1, 0.01, 0.005, 0.002, 0.001, 0.0001]. We set the number of negative samples to be 10. The embedding dimension d is set to be 128 for all methods. Table 3 summarizes the optimal parameters tuned on validation data sets.

Table 3. Optimal parameter settings

Dataset	Learning rate	Batch size	Embedding dimension (d)	Epoch
Yeast	0.005	256	128	20
E. coli	0.002	128	128	20

To capture the non-linearity of gene expression data, we choose ELU (Clevert *et al.*, 2015) activation function, which corresponds to δ_a in Equation 5, based on empirical evaluation. We use single hidden layer (k = 1) with hyperbolic tangent activation (tanh) to model complex statistical relationships between topological properties and attributes of the gene.

We use the area under ROC curve (AUROC) and area under precision-recall curve (AUPR) (Davis and Goadrich, 2006) to evaluate the rankings generated by the model for interactions in the test set. These metrics are widely used in evaluating the ranked list of predictions in gene interaction (Madhukar *et al.*, 2015).

5 Results and Discussion

We present empirical results of our proposed method against other methods.

5.1 Analysis of gene embeddings

We visualize the embedding vectors of genes learned by GNE. We take the learned embeddings, which specifically model the interactions by preserving topological and attribute similarity. We embed these embeddings into a 2D space using t-SNE package (Maaten and Hinton, 2008) and visualize them in Figure 3. For comparison, we also visualize the embeddings learned by structure-preserving deep learning methods, such as LINE, and node2vec.

For each pair of genes in interaction network of E.coli, we compute the number of shared neighborhood genes and also the correlation of their expression. From all gene pairs, we select the fully-connected subnetwork with five genes (PHEM, YJJB, CSPI, YABI, and YFEA), having the high number of shared neighborhood genes and also strongly correlated expression. Because of their similarity in topological properties and attributes, we expect them to be close to each other in the embedding space. These five genes are marked in Figure 3.

Figure 3 demonstrates that GNE places these marked genes closer to each other than other methods. Furthermore, this analysis indicates that

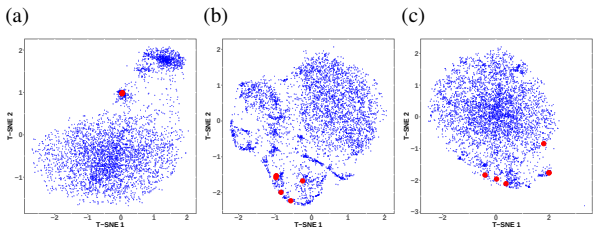

Fig. 3. Visualization of learned embeddings for genes on E.coli using (a) GNE, (b) LINE, and (c) node2vec. Genes are mapped to the 2D space using the t-SNE package (Maaten and Hinton, 2008) with learned representations from each method as input. Each point in the 2D space represents a gene. Selected genes are PHEM, YJJB, CSPI, YABI and YFEA are marked to point their position in embedding space. Visualization of embeddings learned by GNE shows these genes as overlapping with each other in the 2D space.

GNE learns similar representations for genes having similar topological properties and expression.

5.2 Gene Interaction Prediction

We randomly remove 50% of interactions from the network and compare various methods to evaluate their predictions for 50% missing interactions. Table 4 shows the performance of GNE and other methods on gene interaction prediction across different datasets. As our method significantly outperforms other competing methods, it indicates the informativeness of gene expression in predicting missing interactions. Also, our model is capable of integrating attributes with topological properties to learn better representations.

Table 4. Area under ROC curve (AUROC) and Area under PR curve (AUPR) for gene Interaction Prediction. + indicates the concatenation of expression data with learned embeddings to create final representation. ** denotes that GNE significantly outperforms node2vec at 0.01 level paired t-test.

Methods	Yeast		E.coli	
	AUROC	AUPR	AUROC	AUPR
Correlation	0.582	0.579	0.537	0.557
IsoMap	0.507	0.588	0.559	0.672
LINE	0.726	0.686	0.897	0.851
node2vec	0.739	0.708	0.912	0.862
IsoMap+	0.653	0.652	0.644	0.649
LINE+	0.745	0.713	0.899	0.856
node2vec+	0.751	0.716	0.871	0.826
GNE (our model)	**0.825****	**0.821****	**0.940****	**0.939****

We compare our model with a correlation-based method, that takes only expression data into account. Our model shows significant improvement of 0.243 (AUROC), 0.242 (AUPR) on yeast and 0.403 (AUROC), 0.382 (AUPR) on E.coli over correlation-based methods. This improvement suggests the significance of topological properties of the gene network.

The network embedding method, Isomap, performs poorly in comparison to correlation-based methods on yeast because of its limitation on network inference. Deep learning based network embedding methods such as LINE, and node2vec show the significant gain over Isomap and correlation-based methods. node2vec outperforms LINE across two datasets. GNE trained only with topological properties outperforms these

structured-based deep learning methods. However, these methods don't consider the attributes of the gene that we suggest to contain useful information for gene interaction prediction. By adding expression data with topological properties, GNE outperforms structure-preserving deep embedding methods across both datasets.

Focusing on the results corresponding to the integration of expression data with topological properties, we find that the method of integrating the expression data plays an important role in the performance. Performance of node2vec+ (LINE+, Isomap+) shows little improvement with the integration of expression data on yeast. However, node2vec+ (LINE+, Isomap+) has no improvement or decline in performance on E.coli. The decline in performance indicates that simply concatenating the expression vector with learned representations for the gene is insufficient to capture the rich information in expression data. The late fusion approach of combining the embedding vector corresponding to the topological properties of gene network and the feature vector representing expression data have no significant improvement in the performance (except Isomap). In contrast, our model incorporates gene expression data with topological properties by the early fusion method and shows significant improvement over other methods.

5.3 Impact of network sparsity

We investigate the robustness of our model with respect to network sparsity. We hold out 10% interactions as the test set and change the sparsity of the remaining network by randomly removing a portion of remaining interactions. Then, we train GNE to predict interactions in the test set and evaluate the change in performance with respect to network sparsity. We evaluate two versions of our implementations: GNE with only topological properties and GNE with topological properties and expression data. The result is shown in Figure 4.

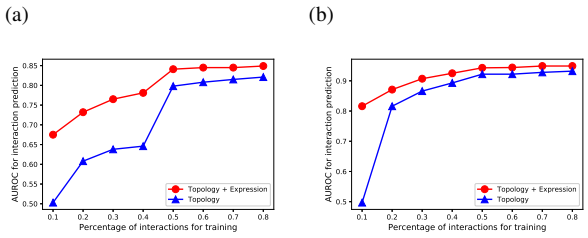

Fig. 4. Comparison of our method's performance with respect to network sparsity and addition of expression information. (a) yeast (b) E.coli. Integration of expression data with topological properties of the gene network improves the performance for both datasets.

Figure 4 shows that our method's performance improves with an increase in the number of training interactions across datasets. In addition, our method's performance improves when expression data is integrated with topological structure. Specifically, GNE trained on 10% of total interactions and attributes of yeast shows a significant gain of 0.172 AUROC (from 0.503 to 0.675) over GNE trained only with 10% of total interactions as shown in Figure 4(a). Similarly, GNE improves the AUROC from 0.497 to 0.816 for E.coli with the same setup as shown in Figure 4(b). The integration of gene expression data results in less improvement when we train GNE on a relatively large number of interactions.

Moreover, the performance of GNE trained with 50% of total interactions and expression data is comparable to be trained with 80% of total interactions without gene expression data as shown in Figure 4.The integration of expression data with topological properties into GNE model has more improvement on E.coli than yeast when we train with 10% of total interactions for each dataset. The reason for this is likely the difference in the number of available interactions for yeast and E.coli as shown in Table 2. This indicates the informativeness of gene expression when we have few interactions and supports the idea that the integration of expression data with topological properties improves gene interaction prediction.

5.4 Impact of λ

GNE involves the parameter λ that controls the importance of gene expression information relative to topological properties of gene network as shown in Equation 6. We examine how the choice of the parameter λ affects our method's performance. Figure 5 shows the comparison of our method's performance with different values of λ when GNE is trained on varying percentage of total interactions.

We evaluate the impact of λ on range [0, 0.2, 0.4, 0.6, 0.8, 1]. When λ becomes 0, the learned representations model only topological properties. In contrast, setting the high value for λ makes GNE learn only from attributes and degrades its performance. Therefore, our model performs well when λ is within [0, 1].

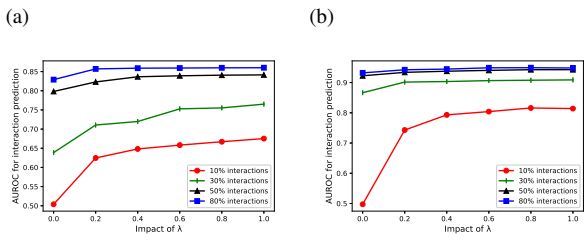

Fig. 5. Impact of λ on performance of our method trained with different percentages of interactions for training GNE. (a) yeast (b) E.coli. Different lines indicate different percentages of interactions.

Figure 5 shows that integration of expression data improves the performance of GNE to predict gene interactions. Impact of λ depends on the number of interactions used to train GNE. If GNE is trained with few interactions, integration of expression data with topological properties plays an important role in predicting missing interactions. As the number of training interactions increases, integration of expression data has less impact but still improves the performance over only topological properties.

Figure 4 and 5 demonstrate that the expression data contributes the increase in AUROC by nearly 0.14 when interactions are less than 40% for yeast and about 0.32 when interactions are less than 10% for E. coli. More topological properties and attributes are required for yeast than E.coli. It may be related to the fact that yeast is a more complex species than E.coli. Moreover, we can speculate that more topological properties and attributes are required for higher eukaryotes like humans. In humans, GNE that integrates topological properties with attributes may be more successful than the methods that only use either topological properties or attributes.

This demonstrates the sensitivity of GNE with respect to parameter λ. This parameter λ has a huge impact on our method's performance and should be selected properly.

5.5 Investigation of GNE's predictions

We investigate the predictive ability of our model in identifying new gene interactions. For this aim, we consider two versions (2017 and 2018 version) of BioGRID interaction datasets. The 2018 version contains 12,835 new interactions for yeast and 11,185 new interactions for E.coli than the 2017 version. Section 5.3 suggests that GNE's performance trained with 50% and 80% of total interactions are comparable for both yeast and E.coli. We thus train our model with 50% of total interactions from the 2017 version to learn the embeddings for genes and demonstrate

Table 5. New gene interactions on 2018 version that are assigned high probability by GNE after integration of expression data. We provide probability predicted by GNE (with/without expression data) for new interactions in the 2018 version and evidence supporting the existence of predicted interactions.

Organism	Probability		Gene i	Gene j	Experimental Evidence Code	Evidence
	Topology	Topology + expression				
Yeast	0.287	0.677	TFC8	DHH1	Affinity Capture-RNA	(Miller *et al.*, 2018)
	0.394	0.730	SYH1	DHH1	Affinity Capture-RNA	(Miller *et al.*, 2018)
	0.413	0.746	CPR7	DHH1	Affinity Capture-RNA	(Miller *et al.*, 2018)
	0.253	0.551	MRP10	DHH1	Affinity Capture-RNA	(Miller *et al.*, 2018)
	0.542	0.835	RPS13	ULP2	Affinity Capture-MS	(Liang *et al.*, 2017)
E.coli	0.014	0.944	ATPB	RFBC	Affinity Capture-MS	(Babu *et al.*, 2018)
	0.012	0.941	NARQ	CYDB	Affinity Capture-MS	(Babu *et al.*, 2018)
	0.013	0.937	PCNB	PAND	Affinity Capture-MS	(Babu *et al.*, 2018)
	0.015	0.939	FLIF	CHEY	Affinity Capture-MS	(Babu *et al.*, 2018)
	0.017	0.938	YCHM	PROB	Affinity Capture-MS	(Babu *et al.*, 2018)

the impact of integrating expression data with topological properties. We create the test set with new interactions from 2018 version of BioGRID as positive interactions and the equal number of negative interactions randomly sampled. We make predictions for these interactions using learned embeddings and create a list of (Gene v_i, Gene v_j, probability), ranked by the predicted probability. Our model assigns high probabilities to positive interactions over negative interactions which are demonstrated by Figure 6.

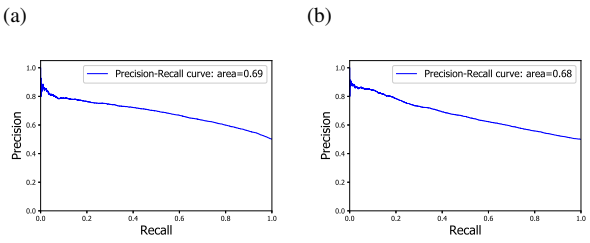

Fig. 6. Precision-Recall curve demonstrating our method's performance in predicting new interactions. (a) yeast (b) E.coli

Table 5 shows top 5 interactions with the significant increase in predicted probability for both yeast and E.coli after expression data is integrated. We also provide literature evidence with experimental evidence code[2] obtained from the BioGRID database (Stark *et al.*, 2006) supporting these predictions. BioGRID compiles interaction data from numerous publications through comprehensive curation efforts. Taking new interactions added to BioGRID into consideration, we evaluate the probability of these interactions predicted by GNE trained with and without expression data. Specifically, integration of expression data increases the probability of 8,331 (out of 11,185) interactions for E.coli (improving AUROC from 0.606 to 0.662) and 6,010 (out of 12,835) interactions for yeast (improving AUROC from 0.685 to 0.707). Table 5 shows that integration of topology and expression data significantly increases the probabilities of true interactions between genes.

This analysis demonstrates the potential of our method in the discovery of gene interactions.

[2] Experimental evidence codes supporting the interactions are referenced on `https://wiki.thebiogrid.org/doku.php/experimental_systems`.

5.6 Generalization to new genes

GNE uses gene expression data which allows it to learn embeddings for genes that were not the part of network structure used in training. To evaluate how our model generalizes to unseen genes, we randomly remove 10%/25%/50% of genes from network structure and train GNE to learn embeddings for rest of the genes. GNE learns weight matrix $\mathbf{W}_{att}$ which transforms gene expression data to dense vector i.e. their embeddings. Given the network structure with 10%/25%/50% missing genes, our aim is to predict the interactions between these missing genes using attribute embeddings. To evaluate the performance of the model, we take interactions between these missing genes from BioGRID as positive samples and randomly sample an equal number of missing gene pairs that have no interactions.

Table 6. Area under ROC curve (AUROC) and Area under PR curve (AUPR) for interaction prediction task with respect to varying number of missing genes.

Methods	% unseen	Yeast		E.coli	
		AUROC	AUPR	AUROC	AUPR
Correlation	10%	0.568	0.568	0.537	0.558
	25%	0.580	0.580	0.540	0.549
	50%	0.578	0.580	0.535	0.553
GNE	10%	0.636	0.620	0.631	0.627
	25%	0.646	0.630	0.583	0.599
	50%	0.632	0.618	0.581	0.586

Table 6 shows that GNE outperforms correlation-based methods in predicting gene interactions between unseen genes. The result is also true when the half of the genes are missing from network structure. Structure embedding methods such as LINE, node2vec and Isomap cannot be applied to this scenario, since they require topological properties of unseen genes to generate embeddings. Integration of attributes and flexibility of GNE to learn embeddings only based on these attributes help to generalize to unseen genes.

6 Conclusion

We developed a novel deep learning framework, namely GNE to perform gene network embedding. Specifically, we design deep neural network architecture to model the complex statistical relationships between gene interaction network and expression data. GNE is flexible to the addition

of different types and number of attributes. Experimental results show that GNE can learn informative representations for the gene network and achieve better performance in gene interaction prediction over other state-of-the-art methods.

As future work, we aim to study the impact of integrating other sources of information about gene such as transcription factor binding sites, functional annotations (from gene ontology), gene sequences, metabolic pathways etc. into GNE in predicting gene interaction.

Funding

This work was supported by the National Science Foundation [1062422 to A.H.] and the National Institutes of Health [R15GM116102 to F.C.].

References

Alanis-Lobato, G., Cannistraci, C. V., and Ravasi, T. (2013). Exploitation of genetic interaction network topology for the prediction of epistatic behavior. *Genomics*, **102**(4), 202–208.

Babu, M., Bundalovic-Torma, C., Calmettes, C., Phanse, S., Zhang, Q., Jiang, Y., Minic, Z., Kim, S., Mehla, J., Gagarinova, A., *et al.* (2018). Global landscape of cell envelope protein complexes in escherichia coli. *Nature biotechnology*, **36**(1), 103.

Boucher, B. and Jenna, S. (2013). Genetic interaction networks: better understand to better predict. *Frontiers in genetics*, **4**, 290.

Butte, A. J. and Kohane, I. S. (1999). Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. In *Biocomputing 2000*, pages 418–429. World Scientific.

Cho, H., Berger, B., and Peng, J. (2016). Compact integration of multi-network topology for functional analysis of genes. *Cell systems*, **3**(6), 540–548.

Chua, H. N., Sung, W.-K., and Wong, L. (2006). Exploiting indirect neighbours and topological weight to predict protein function from protein–protein interactions. *Bioinformatics*, **22**(13), 1623–1630.

Clevert, D.-A., Unterthiner, T., and Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*.

Cui, P., Wang, X., Pei, J., and Zhu, W. (2017). A survey on network embedding. *arXiv preprint arXiv:1711.08752*.

Davis, J. and Goadrich, M. (2006). The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM.

Grover, A. and Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Järvinen, A. P., Hiissa, J., Elo, L. L., and Aittokallio, T. (2008). Predicting quantitative genetic interactions by means of sequential matrix approximation. *PLoS One*, **3**(9), e3284.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lage, K. (2014). Protein–protein interactions and genetic diseases: the interactome. *Biochimica et Biophysica Acta (BBA)-Molecular Basis of Disease*, **1842**(10), 1971–1980.

Lei, Y.-K., You, Z.-H., Ji, Z., Zhu, L., and Huang, D.-S. (2012). Assessing and predicting protein interactions by combining manifold embedding with multiple information integration. In *BMC bioinformatics*, volume 13, page S3. BioMed Central.

Liang, J., Singh, N., Carlson, C. R., Albuquerque, C. P., Corbett, K. D., and Zhou, H. (2017). Recruitment of a sumo isopeptidase to rdna stabilizes silencing complexes by opposing sumo targeted ubiquitin ligase activity. *Genes & development*, **31**(8), 802–815.

Liew, A. W.-C., Law, N.-F., and Yan, H. (2010). Missing value imputation for gene expression data: computational techniques to recover missing data from available information. *Briefings in bioinformatics*, **12**(5), 498–513.

Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, **9**(Nov), 2579–2605.

Madhukar, N. S., Elemento, O., and Pandey, G. (2015). Prediction of genetic interactions using machine learning and network properties. *Frontiers in bioengineering and biotechnology*, **3**, 172.

Mani, R., Onge, R. P. S., Hartman, J. L., Giaever, G., and Roth, F. P. (2008). Defining genetic interaction. *Proceedings of the National Academy of Sciences*, **105**(9), 3461–3466.

Marbach, D., Costello, J. C., Küffner, R., Vega, N. M., Prill, R. J., Camacho, D. M., Allison, K. R., Aderhold, A., Bonneau, R., Chen, Y., *et al.* (2012). Wisdom of crowds for robust gene network inference. *Nature methods*, **9**(8), 796.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Miller, J. E., Zhang, L., Jiang, H., Li, Y., Pugh, B. F., and Reese, J. C. (2018). Genome-wide mapping of decay factor–mrna interactions in yeast identifies nutrient-responsive transcripts as targets of the deadenylase ccr4. *G3: Genes, Genomes, Genetics*, **8**(1), 315–330.

Oliver, S. (2000). Proteomics: guilt-by-association goes global. *Nature*, **403**(6770), 601.

Snoek, C. G., Worring, M., and Smeulders, A. W. (2005). Early versus late fusion in semantic video analysis. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 399–402. ACM.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, **15**(1), 1929–1958.

Stark, C., Breitkreutz, B.-J., Reguly, T., Boucher, L., Breitkreutz, A., and Tyers, M. (2006). Biogrid: a general repository for interaction datasets. *Nucleic acids research*, **34**(suppl_1), D535–D539.

Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., and Mei, Q. (2015). Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077. International World Wide Web Conferences Steering Committee.

Tenenbaum, J. B., De Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *science*, **290**(5500), 2319–2323.

Tu, Y., Stolovitzky, G., and Klein, U. (2002). Quantitative noise analysis for gene expression microarray experiments. *Proceedings of the National Academy of Sciences*, **99**(22), 14031–14036.