# Section 2: Overview of graph embedding methods

## Graph Embedding

1. Random-walk
2. Neural Network
3. Matrix-Factorization Based
4. Node proximity for signed networks

A definition that is generalized from the second-order node proximity for unsigned networks (e.g. LINE) [1st condition]

nSNE introduced a 2nd condition, that is, if two nodes in a signed network are similar, they not only should satisfy the second-order node proximity but also have similar sign context.

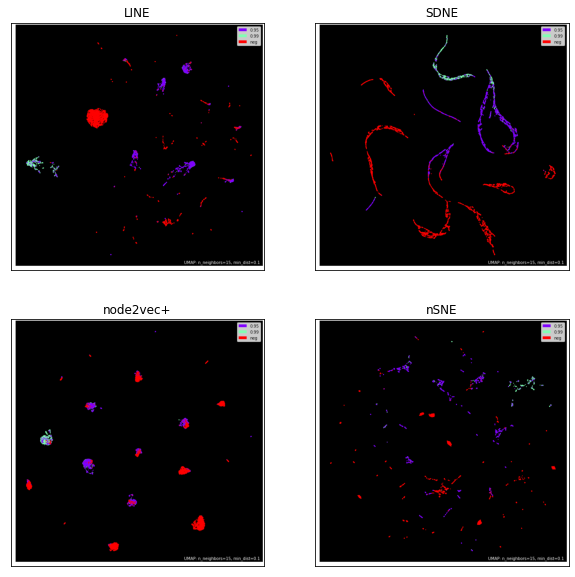
[Demonstrated by tradeoff parameter Beta]

Provides a unified objective function that can preserve both the node and edge pattern of the network

nSNE: Embeddings of nodes and mapping functions learned via back-propagation algorithm

## Protein Embedding

**Network Reconstruction Problem**

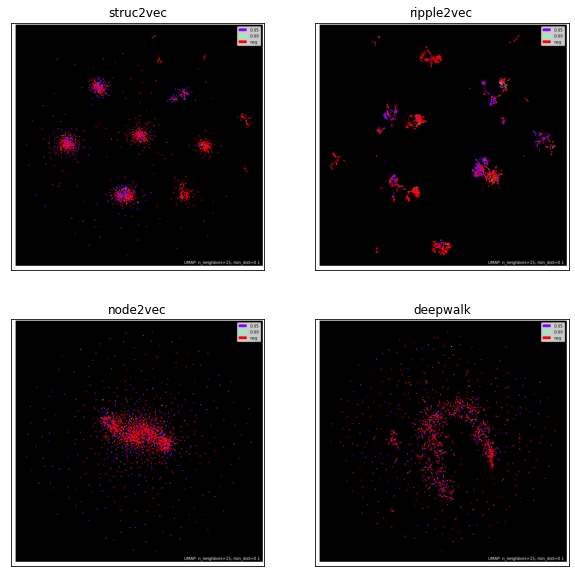


[Plot edges of training set]

These 4 methods able to clearly distinguish highly interacting protein pairs (in green) from remaining protein pairs and the negative pairs

However, only SDNE performance (w/ tune hyperparamters) remained consistent, even on other datasets (without having to retune hyperparameters).

Due to Homogeneity, completeness and v-measure score?



[Random-walk methods]

For node2vec and deepwalk, there is only 1 distinct cluster, with no clear separation between the positive and negative protein pairs.

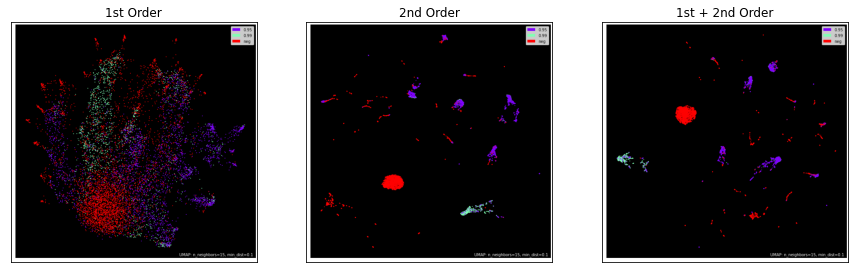
Although clusters were formed, the positive and negative pairs overlap and are not able to distinguish the highly interacting protein pairs.

Methods that did not perform well in the network reconstruction problem was “discarded”.

A good network embedding method should ensure that the learned embeddings can preserve the original network structure {Wang, 2016 #10} \cite{RN10}.

Therefore, in subsequent experiments, (i.e., unbalanced dataset and experimentally verified), we only tested on these 4 methods (i.e., LINE, SDNE, node2vec+ and nSNE).

LINE (Importance of having 2nd order proximity)



* Plot nSNE embeddings

K = 128, B = 0, 0.005 and 1

# Section 3: Experiments

## Chosen IAV strains

**[Insert Table]**

## Datasets constructed/used

* **Explored 3 problems:**

1. Network reconstruction problem
2. Testing on an experimentally verified dataset
3. Link prediction on an unbalanced dataset problem

* Formally introduce HVPPI

Threshold Score range (between 0 and 1)

0.99 >= 0.375

0.95 >= 0.212 and < 0.375

0.90 >= 0.143 and < 0.212

Negative < 0.143

* Overview of complete network graph

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Data | Complete | Edgelist1 |  | | | Link Prediction  (Unbalanced dataset)3 | |
| Train | Test 1  (Network Reconstruction) | Test 2  (Experimentally Verified)2 | Train | Test |
| # nodes | 15, 685 (? IAV, ? Human) | 15, 685 |  |  |  |  |  |
| # interacting pairs | 91217  (?0.99, ?0.95, ?0.90) | (?%) |  |  |  |  |  |
| # non-interacting pairs | 550187 |  |  |  |  |  |  |
| HVPPI Score (MIN) | 0.000 |  |  |  | N/A |  |  |
| HVPPI Score (MAX) | 0.990 |  |  |  | N/A |  |  |
| Degree ? |  |  |  |  |  |  |  |

1 As HVPPI is a prediction tool, only edges with consistent prediction scores on 2 runs were considered.

2 Positive from HPIDB 3.0 \cite{} and Negative from negatome 2.0 \cite{} database.

3 All remaining protein pairs in edgelist was used to construct this dataset. (This allows further testing of performance/robustness of different embeddings methods.)

Demonstrate robustness on incomplete interaction network

## 3.3 Experiments

### **Models**

#### Feedforward neural network (FNN)

FNN with early stopping

#### Skip-GNN

Use same code and hyperparameters as original paper

Experiment using different NRL methods to train input features (embedding) for the original graph

### **Network Representation Learning (NRL) methods**

Review of network representation learning (NRL) methods to learn node embedding from an (unsigned) network

* Weighted network (using HVPPI scores (score ranges from 0 to 1) as the edge weights): node2vec+

[Mention that edge weights were changed to -1 and 1 based on threshold of 0.143]

* Signed network (-1 and 1): Neural network signed network embedding (nSNE) {Song, 2018 #50}

## 3.4 Present results for each method + dataset

1. Mention that hyperparameters were selected following guidelines provided by {Yue, 2020 #3}

Comparison of Feed-forward neural network VS Skip-GNN

* Why I decided to “abandon” Skip-GNN

2. Choosing of mapping functions to compute edge embedding

* Concatenation, Sum, Average, Hadamard Product
* No choosing required for nSNE

3. Comparison with protein embedding

* For unbalanced and experimentally verified datasets

### Clustering evaluation

* Insert metrics explanation + equation

(Homogeneity, Completeness, V-measure, Fowlkes-Mallows score)

Better scores = Better performance on a “new” unseen dataset (i.e., experimentally verified) [SDNE & nSNE performs better than LINE and node2vec+]

From **Specificity** scores, lower false positive rate

Even though it’s human-human interactions instead of iav-human interactions

# Section 4: Discussion

Comment on overall performance of graph embedding and its performance against protein embedding

* Visualizations (UMAP) and evaluation metrics to support discussion of graph embedding methodologies
* Link clustering
* Additional points:
* Node2vec VS node2vec +
* LINE (1st order proximity VS 2nd and 1st + 2nd order proximity)

In biological networks, similarity between nodes that do not directly interact has proven incredibly useful in the last decade across a variety of interaction networks. {Huang, 2020 #6}

## Higher homogeneity, complet

## Limitations

# Section 5: Future work

* This work can be extended to include other IAV strains.
* Probability instead of binary

# Section 6: Conclusion