Part 1: Next-Gen Machine Learning for Network Biology

Marinka Zitnik
Stanford University

Two Lectures

Part 1: May 15, 2019, 2:30 pm - 4:00 pm

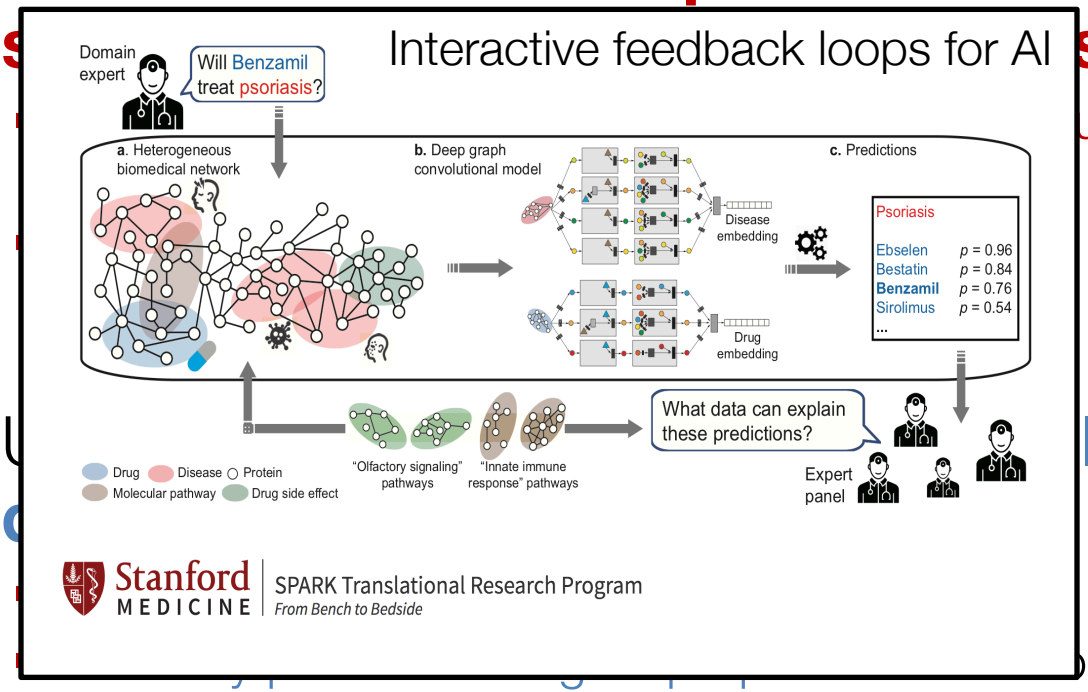
- Methodology: Shallow network embeddings:
  - Map nodes to low-dimensional features
- Resources: Data, tools, codebases
- Applications: PPIs, Disease pathways, Tissues

Part 2: May 16, 2019, 9:00 am – 10:30 am

- Methodology: Deep network embeddings:
  - Graph neural networks for rich biomedical graphs
- Resources: Data, practical advice and demos
- Applications: Polypharmacy, Drug repurposing

Preview of Tomorrow's Lecture

1. Used new methods to **predict safety,**



Two Lectures

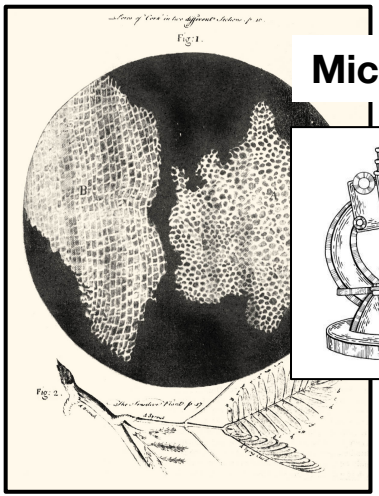
Part 1: May 15, 2019, 2:30 pm - 4:00 pm

- Methodology: Shallow network embeddings:
  - Map nodes to low-dimensional features
- Resources: Data, tools, codebases
- Applications: PPIs, Disease pathways, Tissues

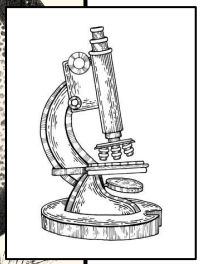
Part 2: May 16, 2019, 9:00 am – 10:30 am

- Methodology: Deep network embeddings:
  - Graph neural networks for rich biomedical graphs
- Resources: Data, practical advice and demos
- Applications: Polypharmacy, Drug repurposing

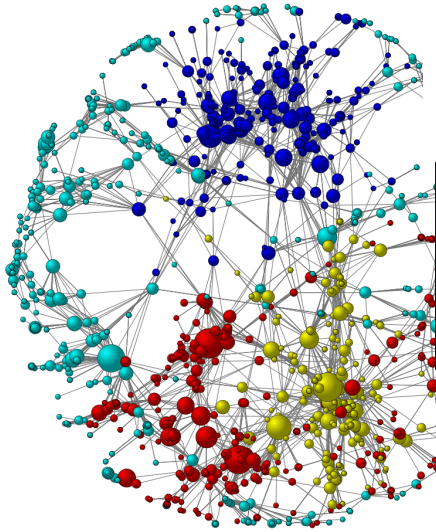
Science crucially depends on scientific instruments



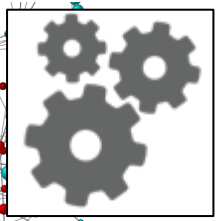
Microscope



Robert Hooke, *Micrographia*, 1665



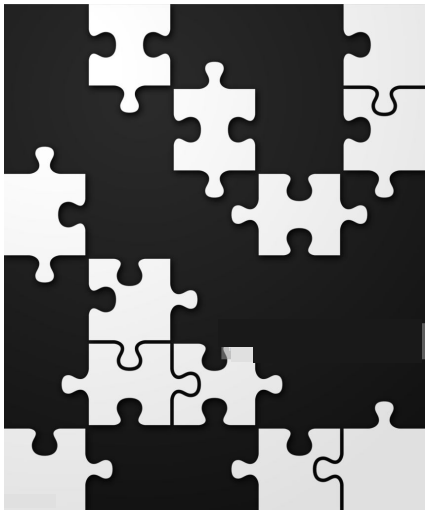
Knowledge discovery

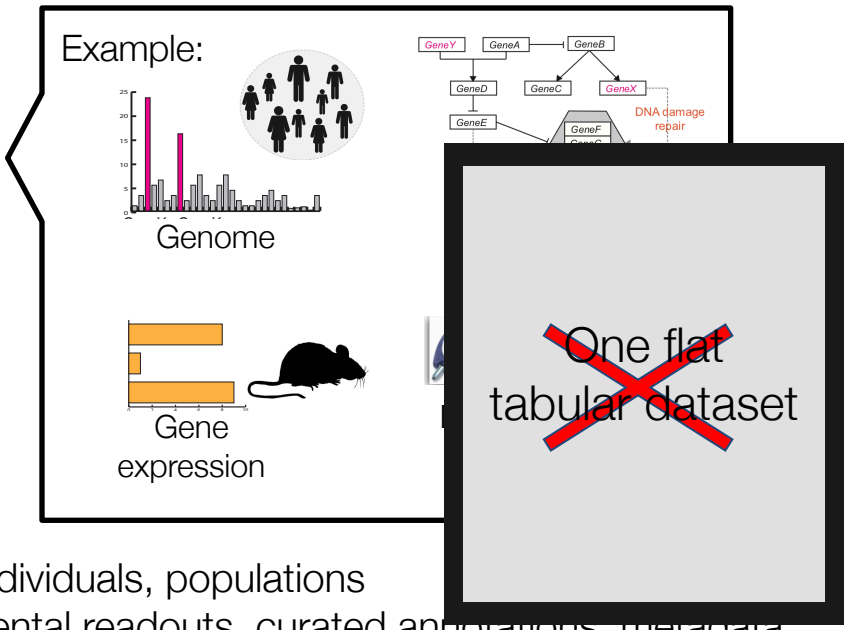


Physical instruments facilitate discoveries

Need instruments for modern, data intensive sciences

However: Biomedical data present challenges for knowledge discovery





Multi-scale: molecules, individuals, populations
Heterogeneous: experimental readouts, curated annotations, metadata
Confounded: data from different labs, biotech platforms, organisms

Zitnik et al. 2019. Machine learning for integrating data in biology and medicine: Principles, practice, and opportunities. *Information Fusion*.

Biomedical problems

Complex, multi-scale, heterogeneous datasets

Machine learning

Tabular, monolithic, flat matrix-like datasets

Significant gap between what ML can address and real-world biomedical problems

To close the gap one has to:

1. Develop a general mathematical representation to integrate heterogeneous data in their broadest sense
2. Develop methods for learning over such representation to open doors for new discoveries

Outline of this Lecture

1) Biological Networks

- Why networks? Why is learning on networks hard?

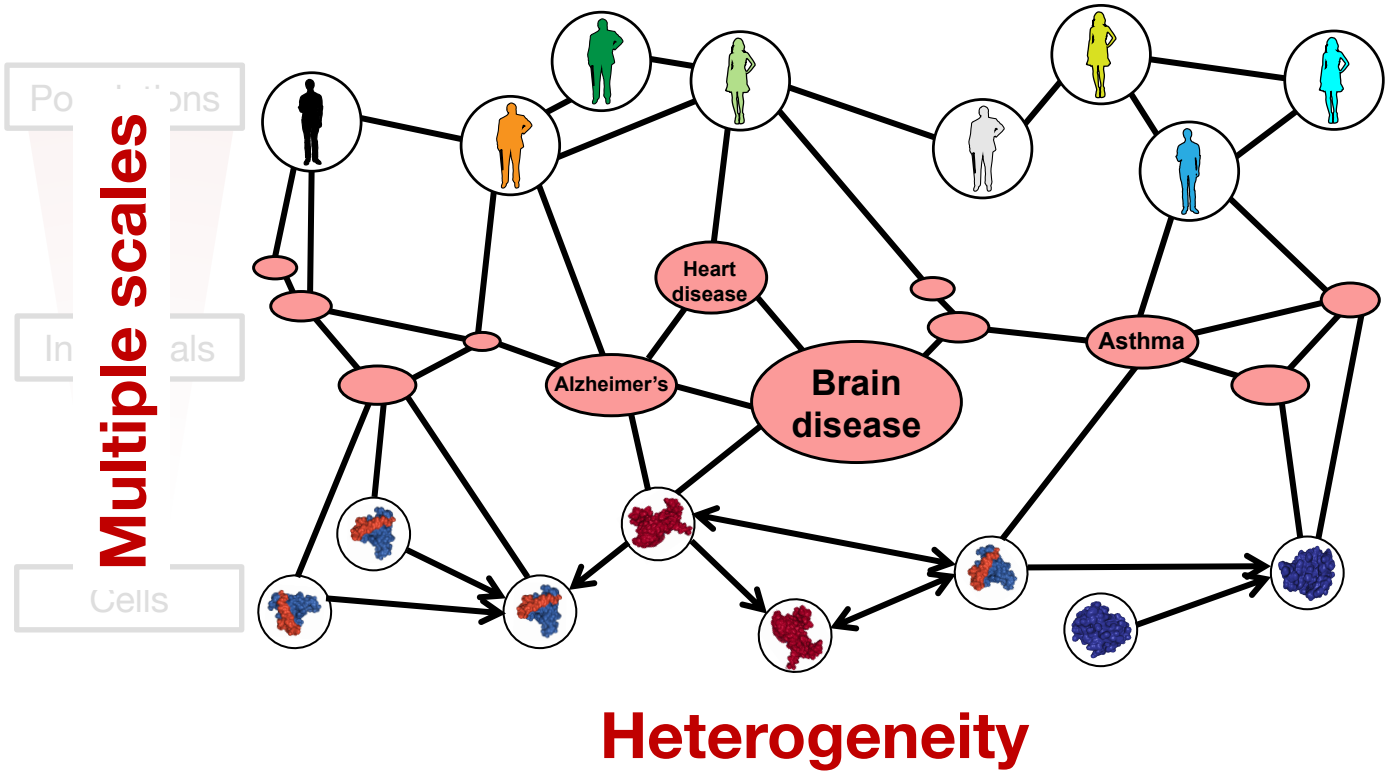
2) Node embeddings

- *Methodology:* Map nodes to vector representations
- *Applications:* PPIs, Disease pathways

3) Heterogeneous networks

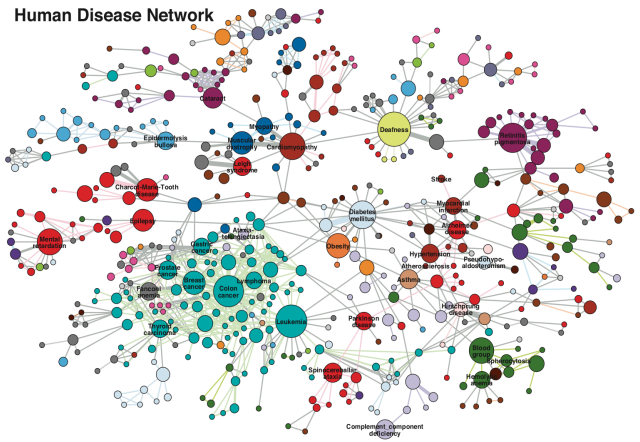
- Methodology: Embedding heterogeneous networks
- *Applications:* Human tissues

Networks allow for integration of biomedical data



Why Networks? Why Now?

- **Question:** How are human genetic diseases and the corresponding disease genes related to each other?
- **Findings:** Genes associated with similar diseases are likely to interact and have similar expression



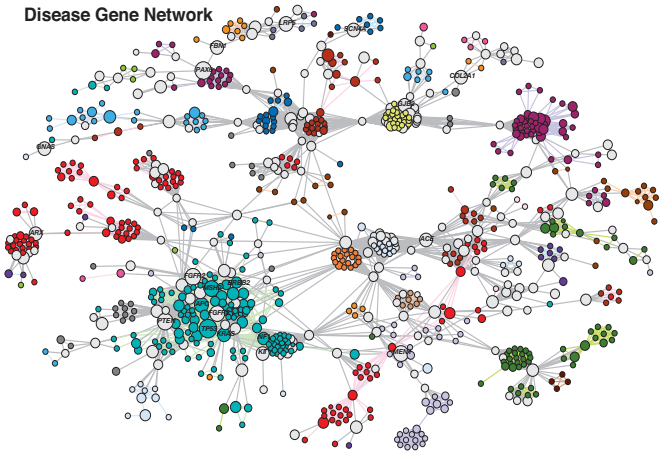


Image from: Goh et al. 2007. [The human disease network](). *PNAS*.

Why Networks? Why Now?

- **Question:** How to simulate a basic eukaryotic cell?
- **Findings:** Simulations reveal molecular mechanisms of cell growth, drug resistance and synthetic life

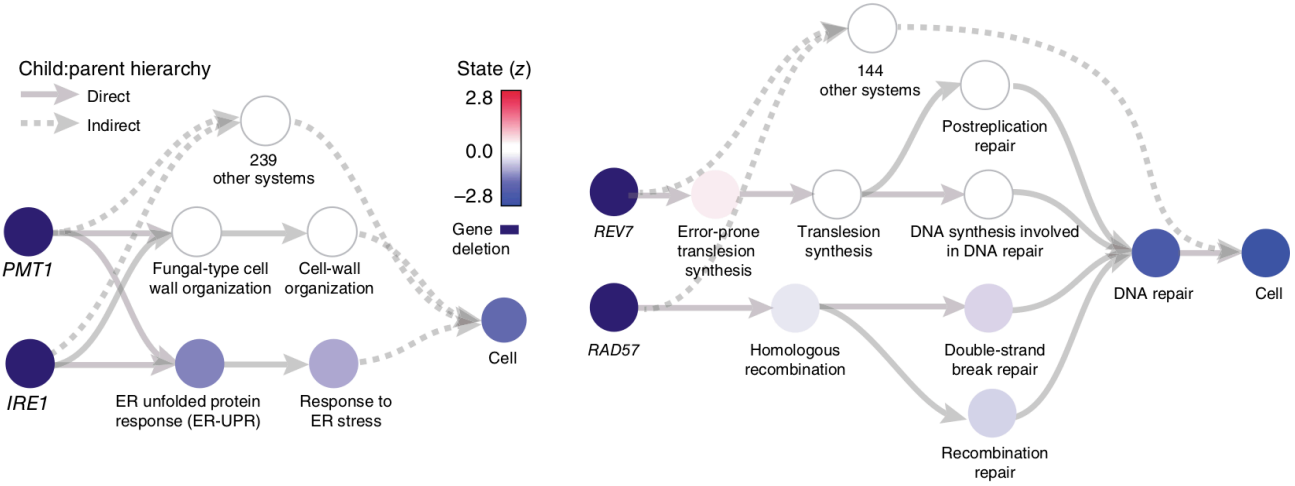


Image from: Ma et al. 2018. Using deep learning to model the hierarchical structure and function of a cell. *Nature Methods*.

Why Networks? Why Now?

- **Question:** How to discover heterogeneity of cancer?
- **Findings:** Analysis identifies new cancer subtypes with distinct patient survival

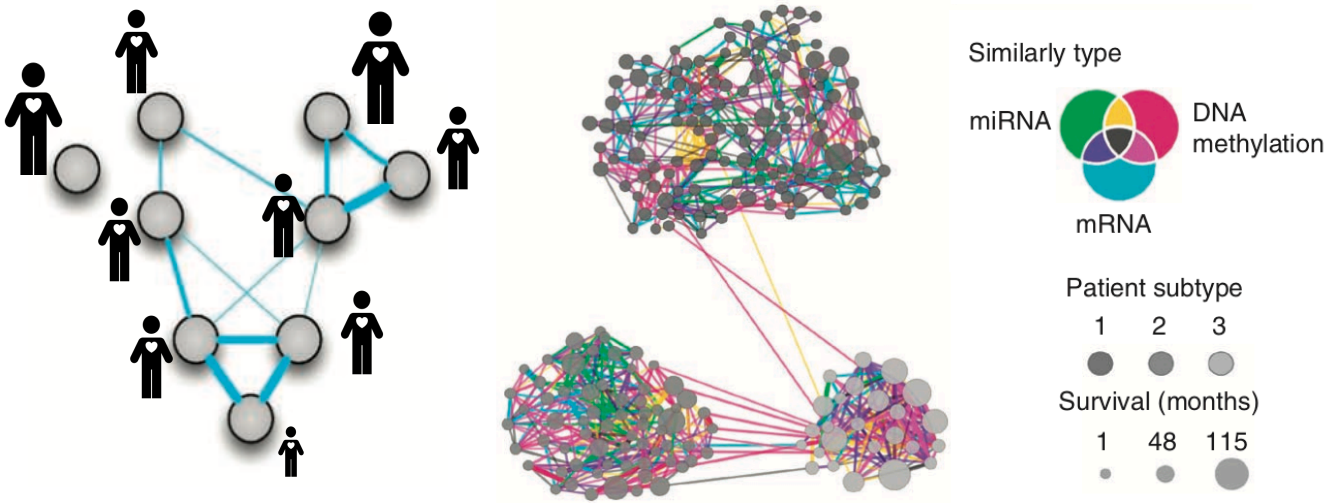
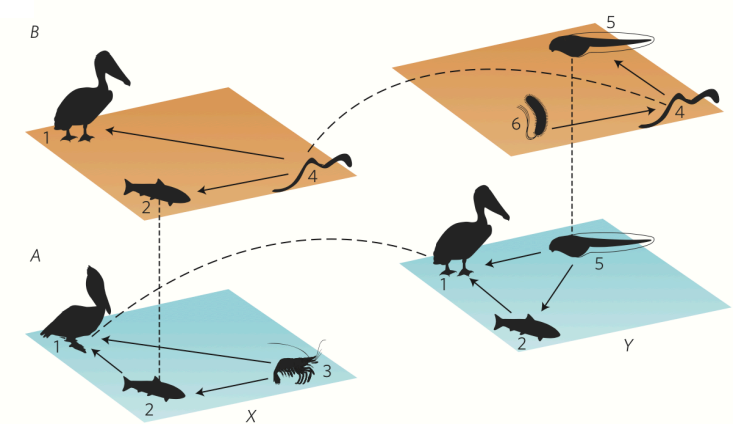


Image from: Wang et al. 2014. Similarity network fusion for aggregating data types on a genomic scale. *Nature Methods*.

Why Networks? Why Now?

- **Question:** How to study ecological systems?
- **Findings:** Pollinators interact with flowers in one season but not in another, and the same flower species interact with both pollinators and herbivores



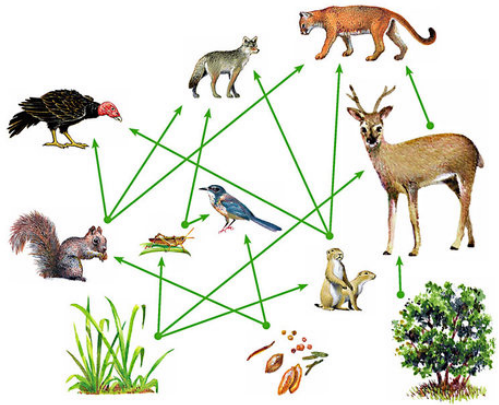


Image from: Pilosof et al. 2017. The multilayer nature of ecological networks. *Nature Ecology and Evolution*.

Why Networks? Why Now?

- **Question:** What are features of human microbiome?
- **Findings:** Microbiota reflects the seasonal availability of different types of food and differentiate industrialized and traditional populations

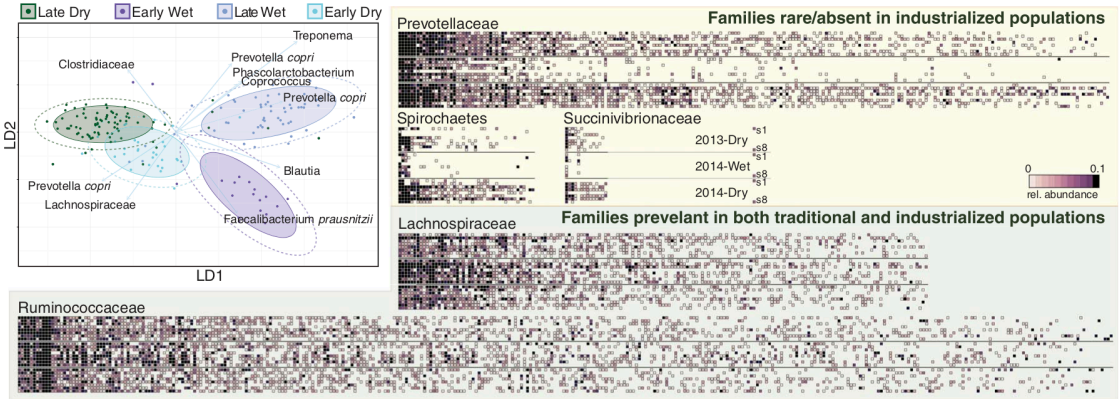
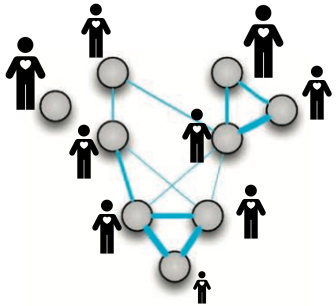
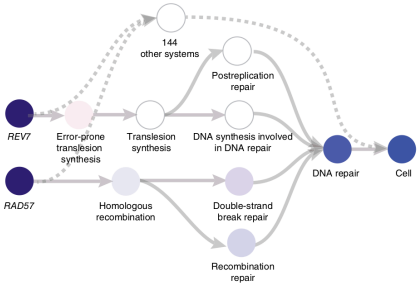


Image from: Smits et al. 2017. Seasonal cycling in the gut microbiome of the Hadza hunter-gatherers of Tanzania. *Science*.

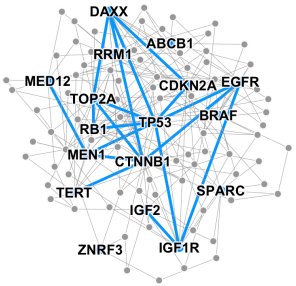
Many Data are Networks



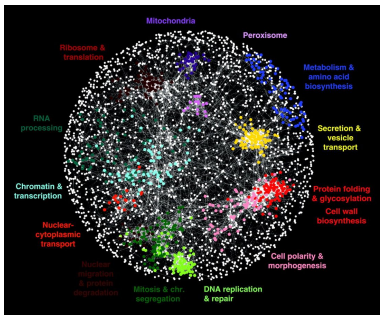
Patient networks



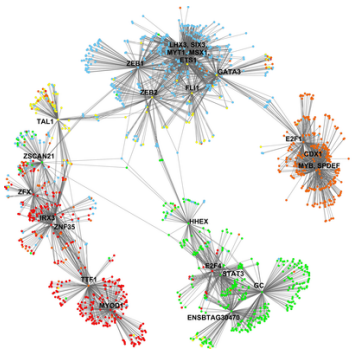
Hierarchies of cell systems



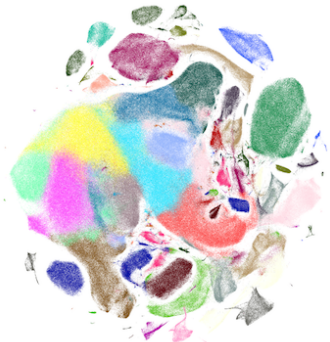
Disease pathways



Genetic interaction networks

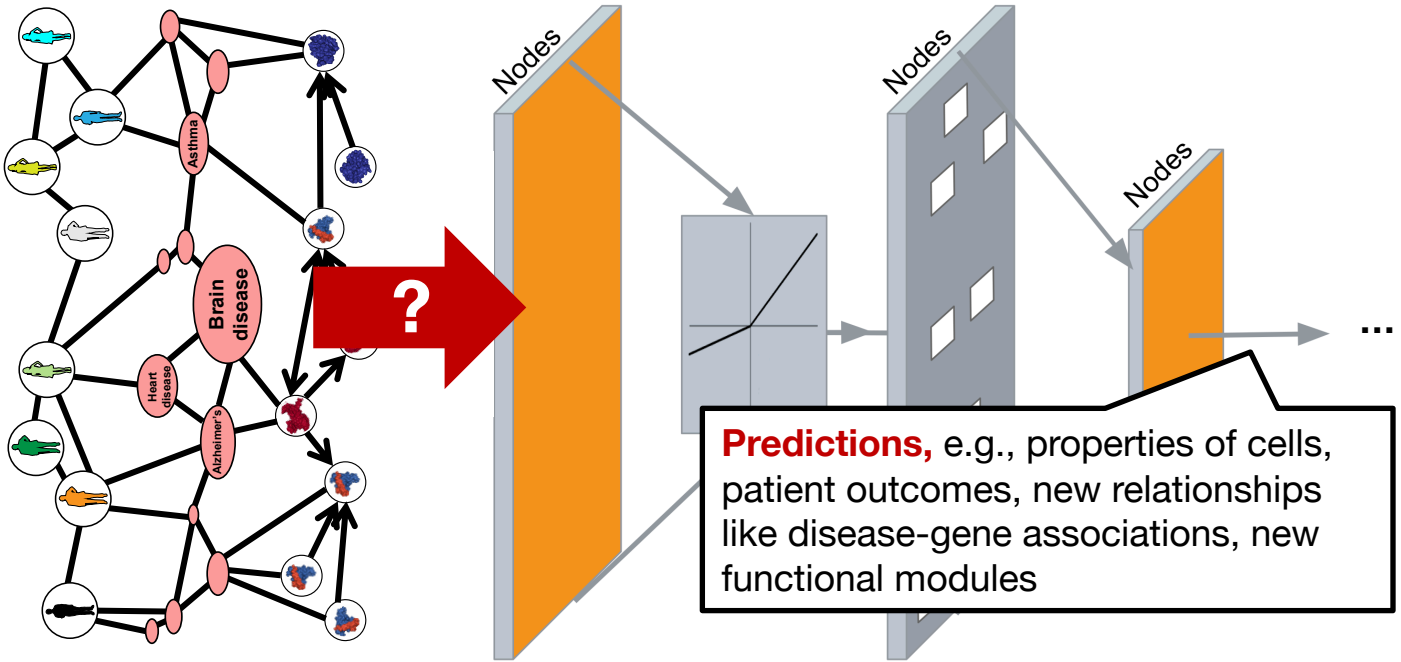


Gene co-expression networks



Cell-cell similarity networks

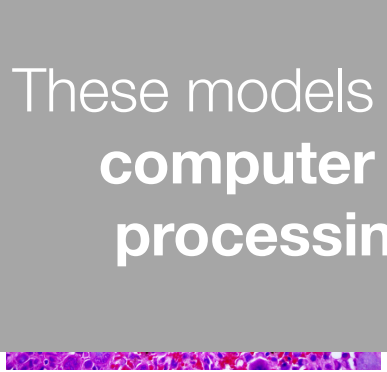
How to do machine learning on biomedical networks?



Networks are a powerful data representation, but are challenging to work with for prevailing deep models

Prevailing Deep Models

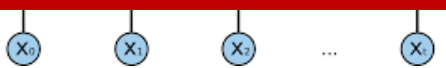
Primarily designed for grids or simple sequences:



These models brought extraordinary gains in **computer vision, natural language processing, speech,** and **robotics**

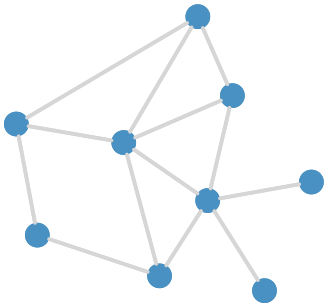
- RNNs for text/sequences

But are **unable to consider interactions**, the essence of biomedical networks

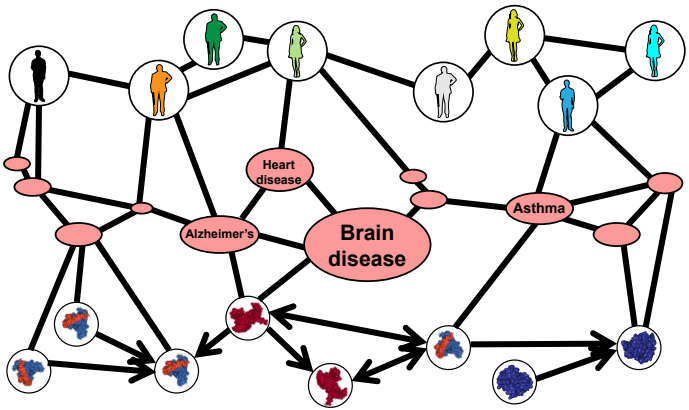


Biomedical Networks

Real-world networks look like this:



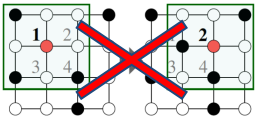
or this:



Examples:

Human contact networks, Disease networks, Patient networks, Cell similarity networks, Medical knowledge graphs

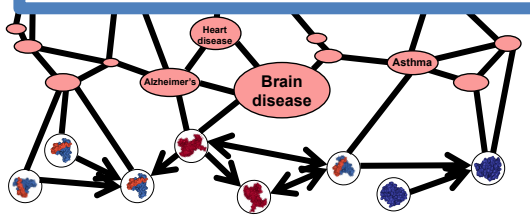
Why is deep learning on networks hard?



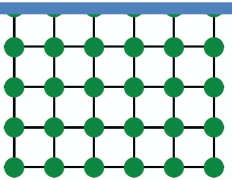
Biomedical networks are far more complex!

- Complex topographical structure (no spatial locality like grids)
- No fixed node ordering/reference point (isomorphism problem)

Need methods that generalize convolutions beyond simple lattices and learn and reason over rich networks



Biomedical networks



Images

Text

Outline of this Lecture

1) Biological networks

- Why networks? Why is learning on networks hard?

2) Node embeddings

- *Methodology:* Map nodes to vector representations
- *Applications:* PPIs, Disease pathways

3) Heterogeneous networks

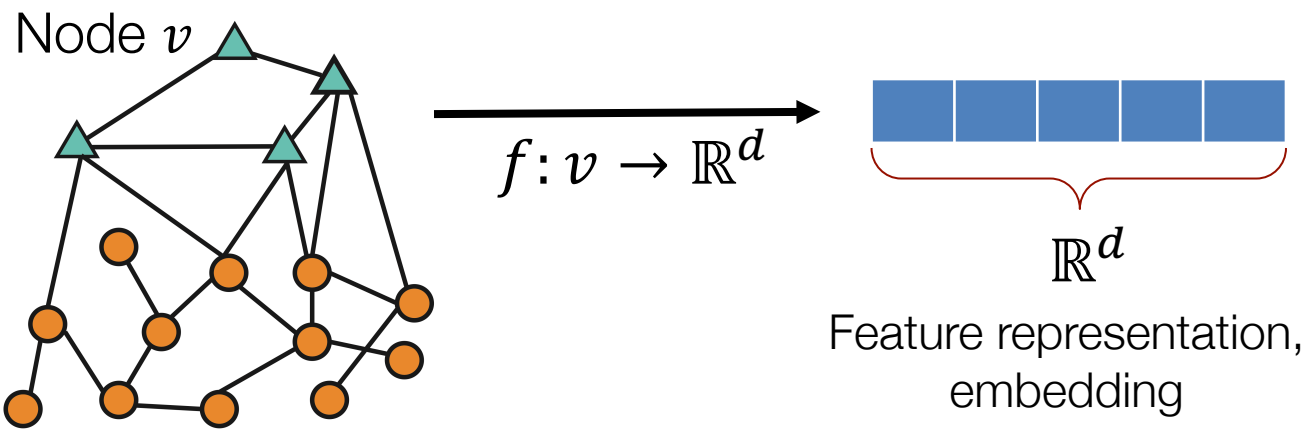
- Methodology: Embedding heterogeneous networks
- *Applications:* Human tissues

Part 2: Node Embeddings

Based on material from:

- Zitnik et al. 2018. Deep Learning for Network Biology. *ISMB.*
- Zitnik et al. 2018. Prioritizing Network Communities. *Nature Communications.*
- Nelson, Zitnik, et al., 2019. To embed or not: network embedding as a paradigm in computational biology. *Frontiers in Genetics.*
- Hamilton et al., 2017. Representation learning on graphs: Methods and applications. *IEEE Data Engineering Bulletin.*

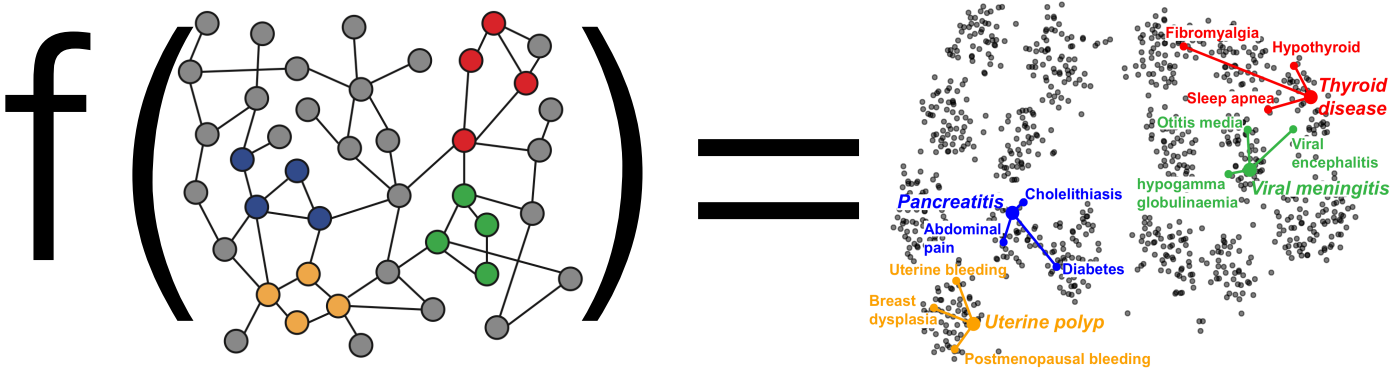
Embedding Nodes



Objective: Map nodes to d-dimensional embeddings such that nodes with similar network neighborhoods are embedded close together

How to learn mapping function f?

Example: Disease Similarity Network



Disease similarity network

2-dimensional node embeddings

Input

Output

Next: How to learn mapping function f?

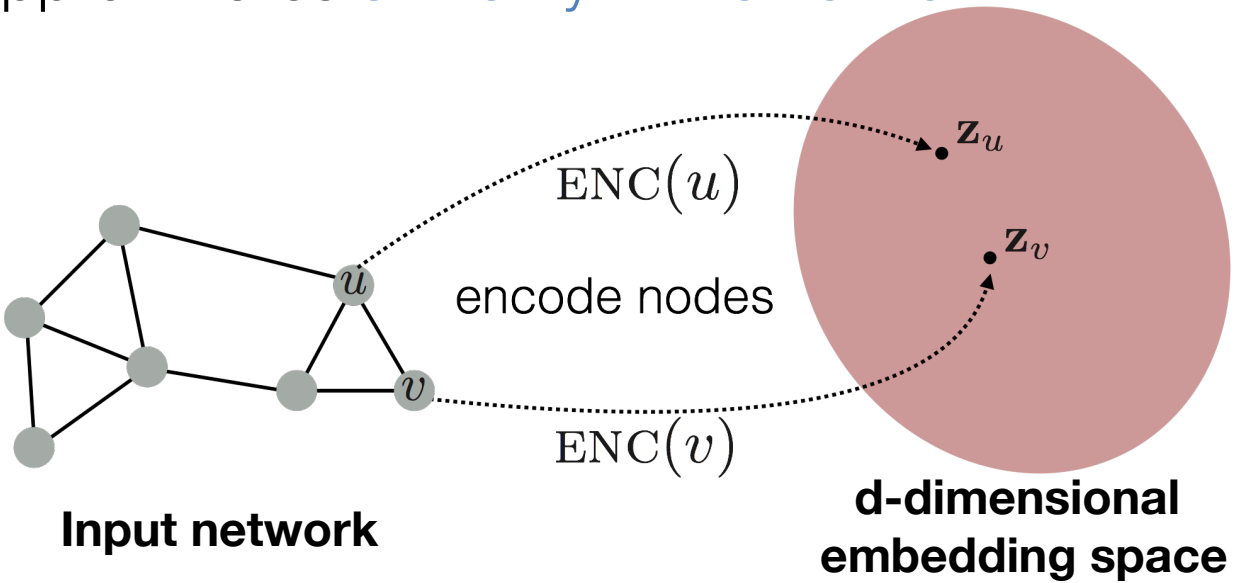
Setup

Assume we have a graph $\mathbf{G}$:

- $\mathbf{V}$ is the vertex set
- $\mathbf{A}$ is the adjacency matrix (binary):
  - Weighted, typed and dynamic graphs as well as multi-graphs (see next part & Thursday's lecture)
  - Integration of node/edge features, and extra information (see next part & Thursday's lecture)

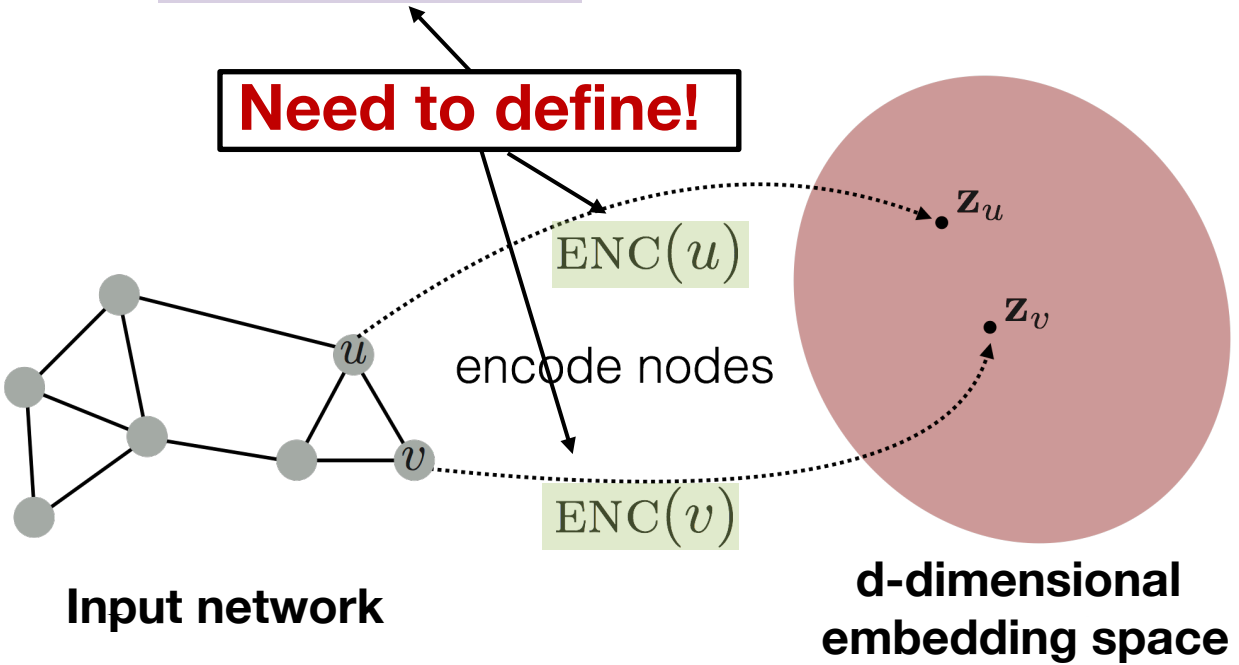
Embedding Nodes

Goal: Map nodes so that similarity in the embedding space (e.g., dot product) approximates similarity in the network



Embedding Nodes

Goal: $\text{similarity}(u, v) \approx \mathbf{z}_v^\top \mathbf{z}_u$



Learning Node Embeddings

1. **Define an encoder** (a function ENC that maps node u to embedding $\boldsymbol{z}_u$)
2. **Define a node similarity function** (measure of similarity in the network)
3. **Optimize parameters of the encoder so that:**

$$\text{similarity}(u, v) \approx \mathbf{z}_v^\top \mathbf{z}_u$$

Two Key Components

1. **Encoder** maps a node to a d-dimensional vector:

$$\text{ENC}(v) = \mathbf{z}_v$$

$\mathbf{d}$-dimensional embedding

node in the input graph

2. **Similarity function** defines how relationships in the input network map to relationships in the embedding space:

$$\text{similarity}(u, v) \approx \mathbf{z}_v^\top \mathbf{z}_u$$

Similarity of u and v in the network

dot product between node embeddings

Embedding Methods

- Many methods use similar encoders:
  - matrix factorizations, node2vec, DeepWalk, LINE, struc2vec
- These methods use different notions of node similarity:
  - Two nodes have similar embeddings if:
    - they are connected (i.e., matrix factorization)?
    - they share many neighbors?
    - they have similar local network structure?
    - etc.

Outline of This Section

1. Shallow node embeddings

2. Biomedical applications

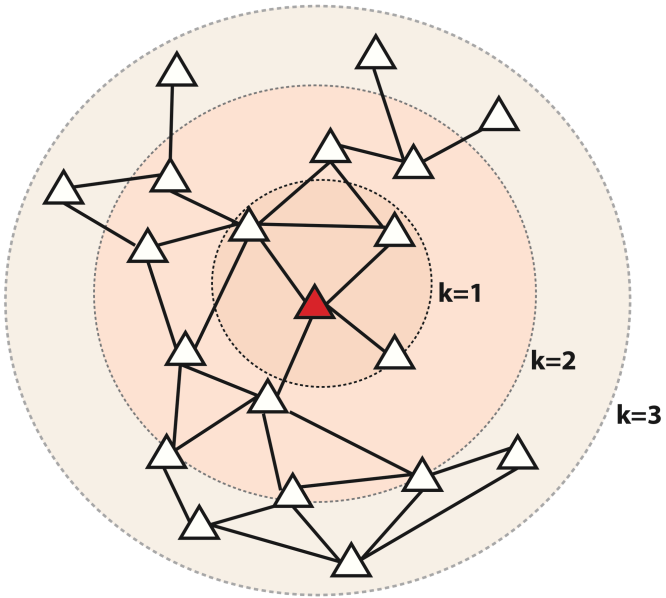
Shallow Node Embeddings

Based on material from:

- Perozzi et al. 2014. DeepWalk: Online Learning of Social Representations. *KDD.*
- Grover et al. 2016. node2vec: Scalable Feature Learning for Networks. *KDD.*
- Ribeiro et al. 2017. struc2vec: Learning Node Representations from Structural Identity. *KDD.*
- Donnat et al. 2018. Learning Structural Node Embeddings via Diffusion Wavelets. *KDD.*

Node Similarity

Idea: Define node similarity function based on higher-order neighborhoods



- **Red:** Target node
- **k=1:** 1-hop neighbors
  - $\mathbf{A}$ (i.e., adjacency matrix)
- **k= 2:** 2-hop neighbors
- **k=3:** 3-hop neighbors

How to stochastically define these higher-order neighborhoods?

Node Embeddings

- Intuition: Find embedding of nodes to d-dimensions that preserves similarity
- Idea: Learn node embedding such that nearby nodes are close together
- Given a node u, how do we define nearby nodes?
  - $N_R(u)$ … neighbourhood of u obtained by some strategy R

Optimization Task

- Given $G = (V, E)$
- Goal is to learn $f: u \rightarrow \mathbb{R}^d$
  - where f is a table lookup
    - We directly "learn" coordinates $\boldsymbol{z_u} = f(u)$ of u
- Given node u, we want to learn feature representation $f(u)$ that is predictive of nodes in u's neighborhood $N_{\mathrm{R}}(u)$

$$\max_f \sum_{u \in V} \log \Pr(N_{\mathrm{R}}(u) | \boldsymbol{z}_{\mathrm{u}})$$

Optimization Task

Goal: Find embedding $\boldsymbol{z_u}$ that predicts nearby nodes $\boldsymbol{N_R(u)}$:

$$\sum_{v \in V} \log(P(N_R(u)|\mathbf{z}_u))$$

Assume conditional likelihood factorizes:

$$P(N_R(u)|\mathbf{z}_u) = \prod_{n_i \in N_R(u)} P(n_i|\mathbf{z}_u)$$

Node Similarity Function Based on Random Walks

$$\mathbf{z}_u^\top \mathbf{z}_v \approx$$

Probability that u and v co-occur in a random walk over the network

Why Random Walks?

1. **Flexibility:** Stochastic definition of node similarity:
   - Local and higher-order neighborhoods

2. **Efficiency:** Do not need to consider all node pairs when training
   - Consider only node pairs that co-occur in random walks

Random Walk Optimization

1. Simulate many short random walks starting from each node using a strategy R
2. For each node u, get $\mathbf{N}_R(u)$ as a sequence of nodes visited by random walks starting at u
3. For each node u, learn its embedding by predicting which nodes are in $\mathbf{N}_R(u)$:

$$\mathcal{L} = \sum_{u \in V} \sum_{v \in N_R(u)} -\log(P(v|\mathbf{z}_u))$$

Random Walk Optimization

$$\mathcal{L} = \sum_{u \in V} \sum_{v \in N_R(u)} -\log\left(\frac{\exp(\mathbf{z}_u^\top \mathbf{z}_v)}{\sum_{n \in V} \exp(\mathbf{z}_u^\top \mathbf{z}_n)}\right)$$

sum over all nodes u

sum over nodes v seen on random walks starting from u

predicted probability of u and $\mathbf{v}$ co-occuring on random walk, i.e., use softmax to parameterize $P(v|\mathbf{z}_u)$

Random walk embeddings = z_u minimizing $\mathcal{L}$

Random Walk Optimization

But doing this naively is too expensive!

$$\mathcal{L} = \sum_{u \in V} \sum_{v \in N_R(u)} -\log\left(\frac{\exp(\mathbf{z}_u^\top \mathbf{z}_v)}{\sum_{n \in V} \exp(\mathbf{z}_u^\top \mathbf{z}_n)}\right)$$

Nested sum over nodes gives $O(|V|^2)$ complexity!

The problem is normalization term in the softmax function?

Solution: Negative Sampling

Solution: Negative sampling (Mikolov et al., 2013)

$$\log\left(\frac{\exp(\mathbf{z}_u^\top \mathbf{z}_v)}{\sum_{n\in V}\exp(\mathbf{z}_u^\top \mathbf{z}_n)}\right)$$

$$\approx \log(\sigma(\mathbf{z}_u^\top \mathbf{z}_v)) - \sum_{i=1}^{k}\log(\sigma(\mathbf{z}_u^\top \mathbf{z}_{n_i})), n_i \sim P_V$$

sigmoid function

random distribution over all nodes

i.e., instead of normalizing w.r.t. all nodes, just normalize against $\mathbf{k}$ random **negative samples**

Random Walks: Overview

1. Simulate many short random walks starting from each node using a strategy R
2. For each node u, get $\mathbf{N}_R(u)$ as a sequence of nodes visited by random walks starting at u
3. For each node u, learn its embedding by predicting which nodes are in $\mathbf{N}_R(u)$:

$$\mathcal{L} = \sum_{u \in V} \sum_{v \in N_R(u)} -\log(P(v|\mathbf{z}_u))$$

Can efficiently approximate using negative sampling

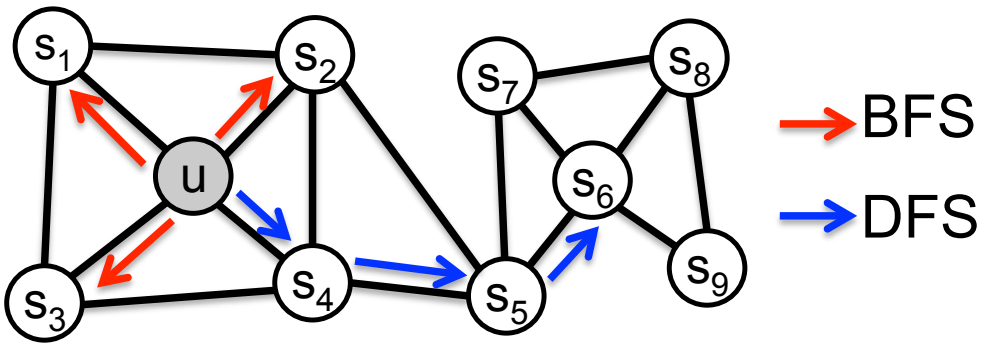
What is the strategy R?

- **So far:**
  - Given simulated random walks, we described how to optimize node embeddings

- **What strategies to use to get random walks?**
  - Simplest idea:
    - Fixed-length, unbiased random walks starting from each node (i.e., DeepWalk from Perozzi et al., 2013)
  - **Can we do better?**
    - Node2vec (Grover et al., 2016)
    - Struc2vec (Ribeiro et al., 2017)
    - Abu-El-Haija et al., 2017 and many others

node2vec: Biased Walks

Two classic strategies to define a neighborhood $N_R(u)$ of a given node u:



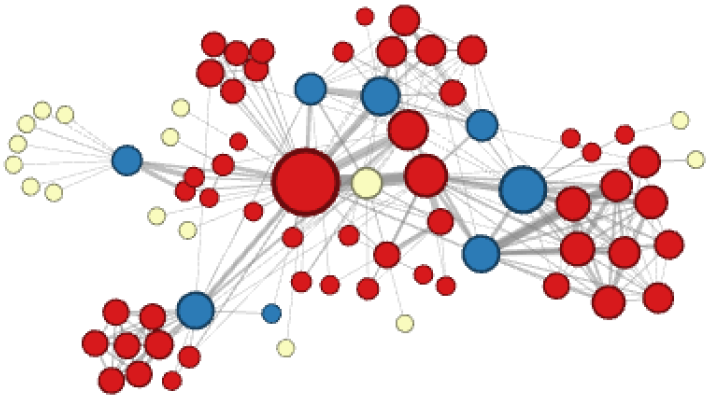
$N_{BFS}(u) = \{ s_1, s_2, s_3 \}$ Local microscopic view

$N_{DFS}(u) = \{ s_4, s_5, s_6 \}$ Global macroscopic view

Experiment: Local vs. Global

Local view of network

(Homophily)

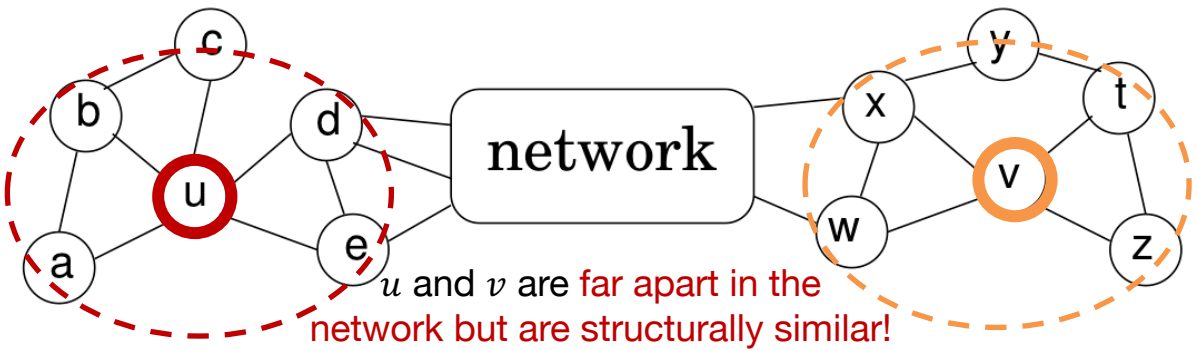


Global view of network

(Structural similarity)

struc2vec: Structural Similarity

- Goal: Nodes visited by random walks starting from node u should be structurally similar to u:
  - E.g., u and v are structurally similar, have similar local network structure



struc2vec: Three Main Steps

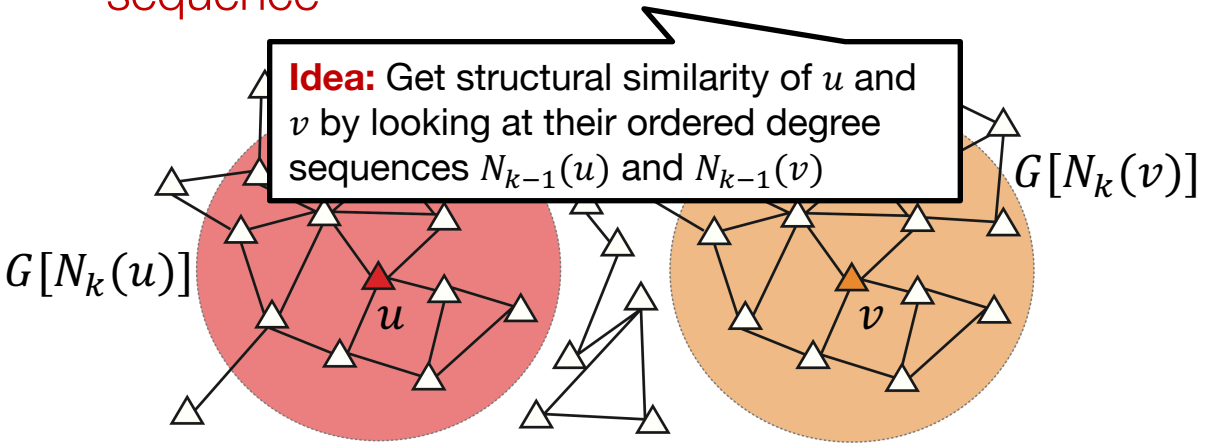
1. Compute structural similarity of nodes based on k-hop neighborhoods
2. Construct a new multilayer graph:
   - K-th layer measures structural similarity of nodes w.r.t. k-hop neighborhoods
3. Run weighted random walks on the multilayer graph to generate $\mathbf{N}_R(u)$

struc2vec: Step 1

Let $N_k(u)$ be nodes in k-hop neighborhood of u

Lemma: u and v are structurally equivalent considering k-hop neighborhoods:

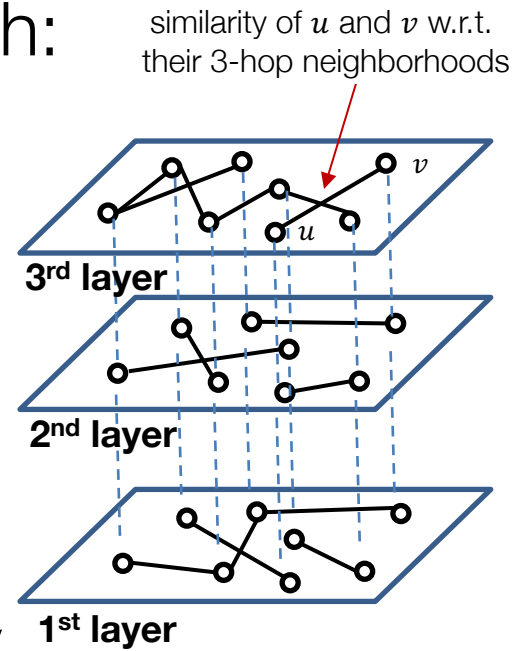
- $G[N_k(u)]$ and $G[N_k(v)]$ are isomorphic graphs
- $N_{k-1}(u)$ and $N_{k-1}(v)$ have identical ordered degree sequence



struc2vec: Step 2

Construct a multilayer graph:

- All nodes from the original network are in every layer
- **K-th layer**: Structural similarity of nodes w.r.t. k-hop neighborhoods
- **Edge weights:** Proportional to nodes' structural similarity



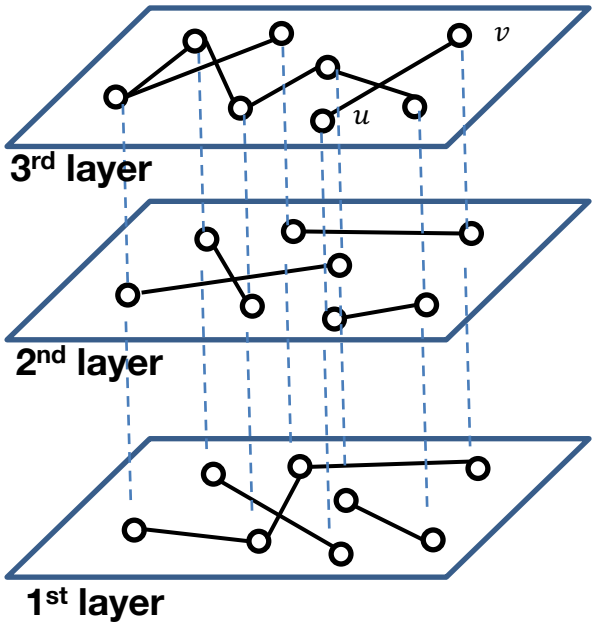
struc2vec: Step 3

Multilayer graph:

- K-th layer has structural similarity of nodes w.r.t. k-hop neighborhoods

Idea: Use the multilayer graph to get $\mathrm{N}_R(u)$:

- $\mathrm{N}_R(u)$ is a sequence of nodes visited by weighted random walk starting at u



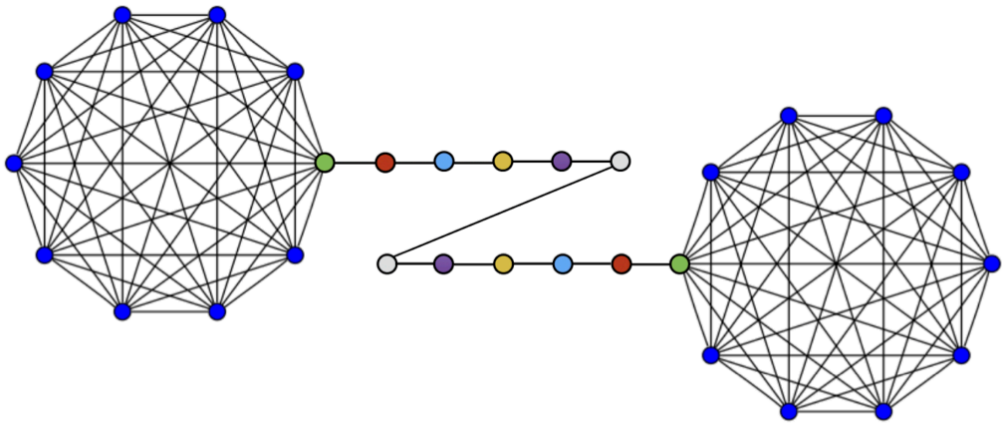
struc2vec: Overview

1. Construct the multilayer graph and simulate many random walks starting from each node
2. For each node u, get $\mathbf{N}_R(u)$ as a sequence of nodes visited by random walks starting at u
3. For each node u, learn its embedding by predicting which nodes are in $\mathbf{N}_R(u)$:

$$\mathcal{L} = \sum_{u \in V} \sum_{v \in N_R(u)} -\log(P(v|\mathbf{z}_u))$$

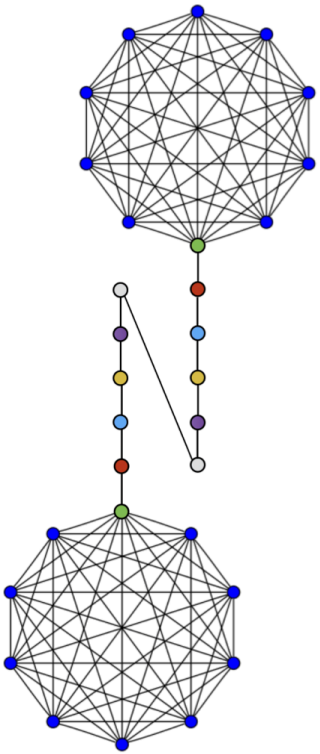
struc2vec: Experiment

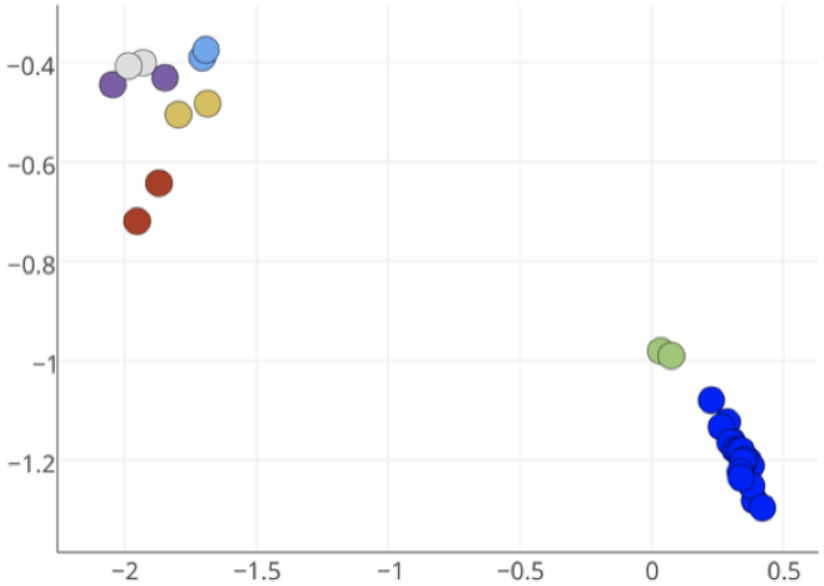
Barbell graph:



- Ground-truth: Nodes of the same color are structurally equivalent (i.e., their local network structure is the same)

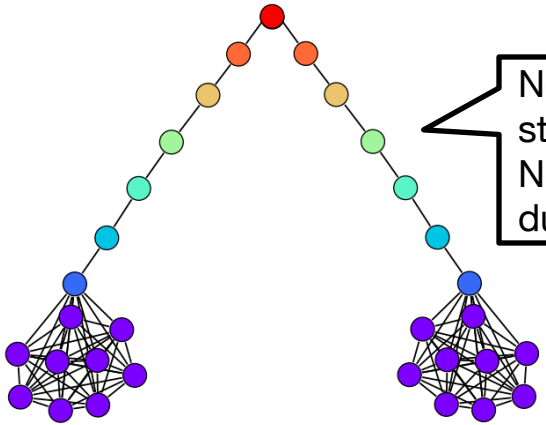
struc2vec: Experiment



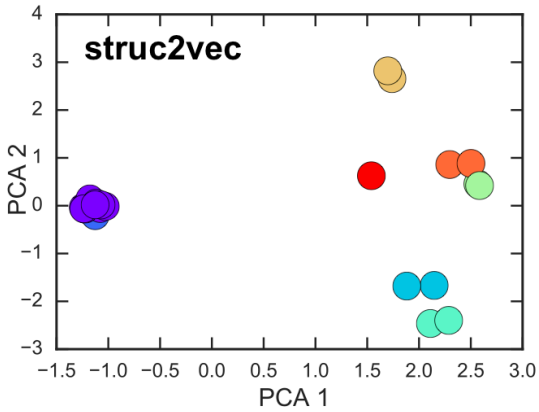


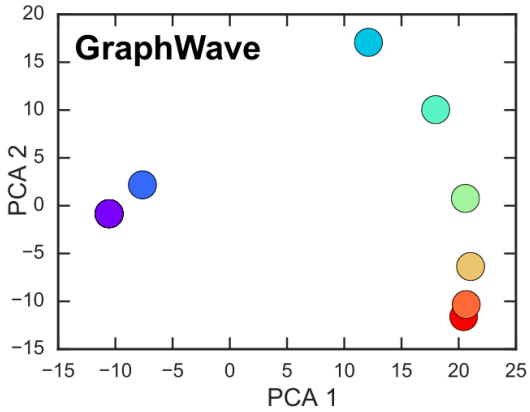
Nodes that are structurally similar are embeded close together!

Beyond struc2vec: GraphWave



Node colors indicate structural roles.
Not available to the algorithm during training.





Summary so Far

Approach: Embed nodes such that:

- Algebraic operations in the learned space reflect topology of the graph

Different notions of node similarity:

- Adjacency-based (i.e., similar if connected)
- Multi-hop similarity definitions
- Random walk approaches

In general: Must choose define node similarity that matches application!

Outline of This Section

1. Shallow node embeddings ✓
2. Biomedical applications 

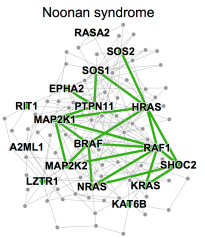
Biomedical Applications

Based on material from:

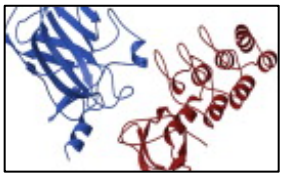
- Grover et al. 2016. node2vec: Scalable Feature Learning for Networks. *KDD.*
- Zitnik and Leskovec. 2017. Predicting Multicellular Function through Multilayer Tissue Networks. *Bioinformatics & ISMB.*
- Zitnik et al. 2018. Large-scale analysis of disease pathways in the human interactome. *PSB.*

Biomedical Applications

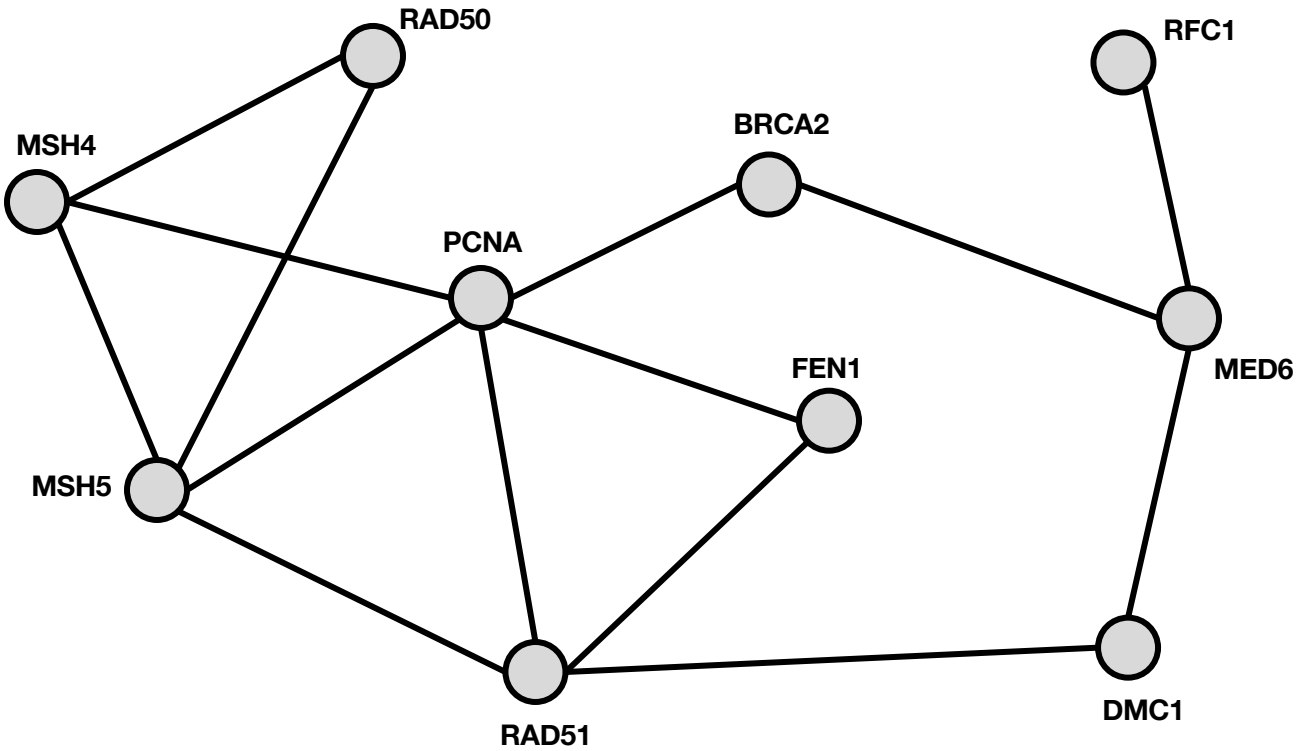
1. Disease pathway detection:
   - Identify proteins whose mutation is linked with a particular disease
   - **Task:** Multi-label node classification



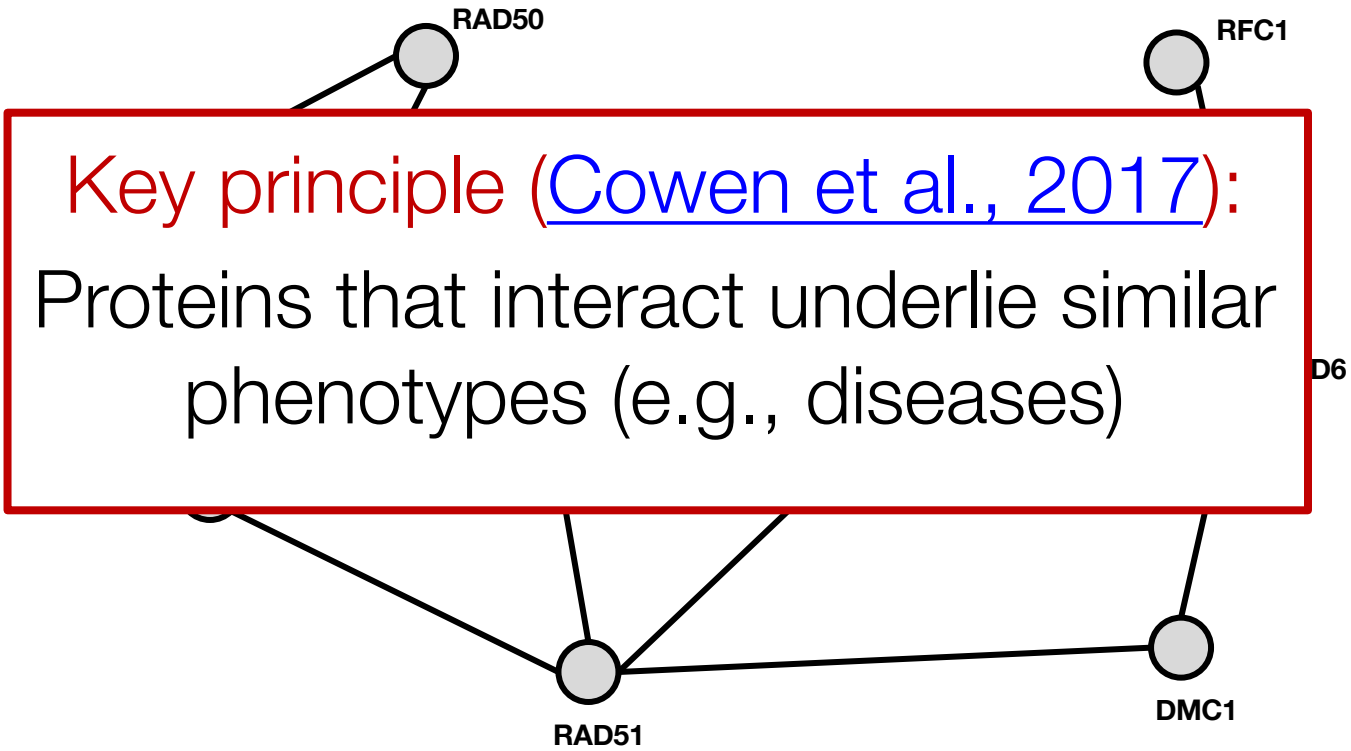
2. Protein interaction prediction:
   - Identify protein pairs that physically interact in a cell
   - **Task:** Link prediction



Human Interactome

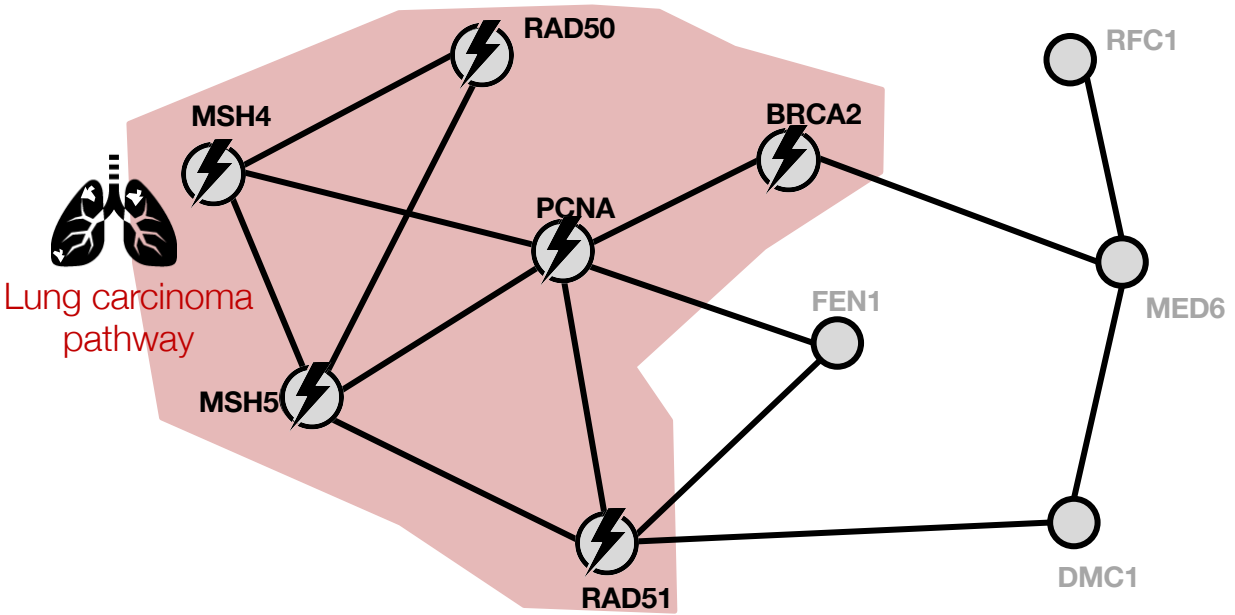


Human Interactome

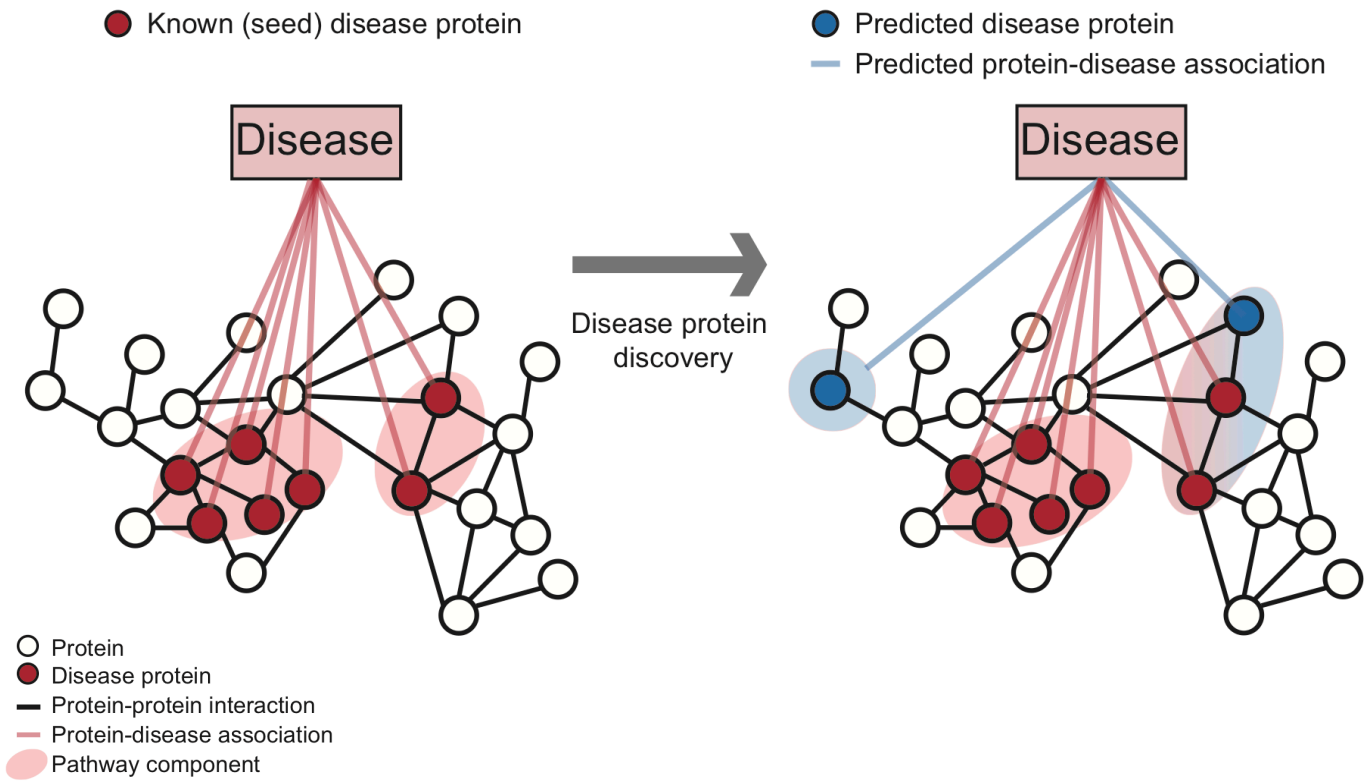


Disease Pathways

- Pathway: Subnetwork of interacting proteins associated with a disease



Disease Pathways: Task



Disease Pathway Dataset

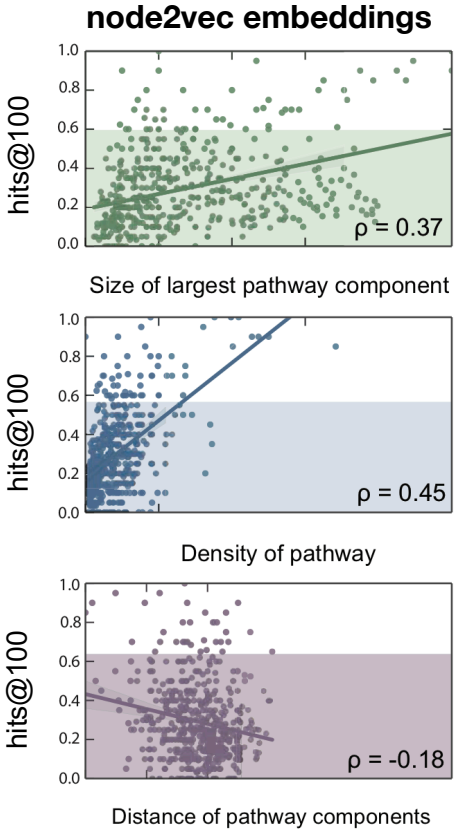
- Protein-protein interaction (PPI) network culled from 15 knowledge databases:
  - 350k physical interactions, e.g., metabolic enzyme-coupled interactions, signaling interactions, protein complexes
  - All protein-coding human genes (21k)
- Protein-disease associations:
  - 21k associations split among 519 diseases
- **Multi-label node classification:** every node (i.e., protein) can have 0, 1 or more labels (i.e., disease associations)

Experimental Setup

Two main stages:

1. Take the PPI network and use node2vec to learn node embeddings
2. For each disease:
   - Fit a classifier that predicts disease proteins based on the embeddings:
     - Train the classifier using training proteins
     - Predict a probability that a test protein is associated with the disease

Pathways: Results

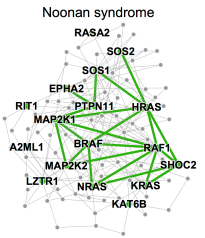


- Best performers:
  - node2vec embeddings
    $\text{hits@100} = 0.40$
  - DIAMOnD
    $\text{hits@100} = 0.30$
  - Matrix completion
    $\text{hits@100} = 0.29$
- Worst performer:
  - Neighbor scoring
    $\text{hits@100} = 0.24$

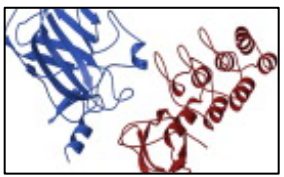
hits@100: fraction of all the disease proteins are ranked within the first 100 predicted proteins

Biomedical Applications

1. Disease pathway detection:
   - Identify proteins whose mutation is linked with a particular disease
   - **Task:** Multi-label node classification



2. Protein interaction prediction:
   - Identify protein pairs that physically interact in a cell
   - **Task:** Link prediction



Protein-Protein Interactions

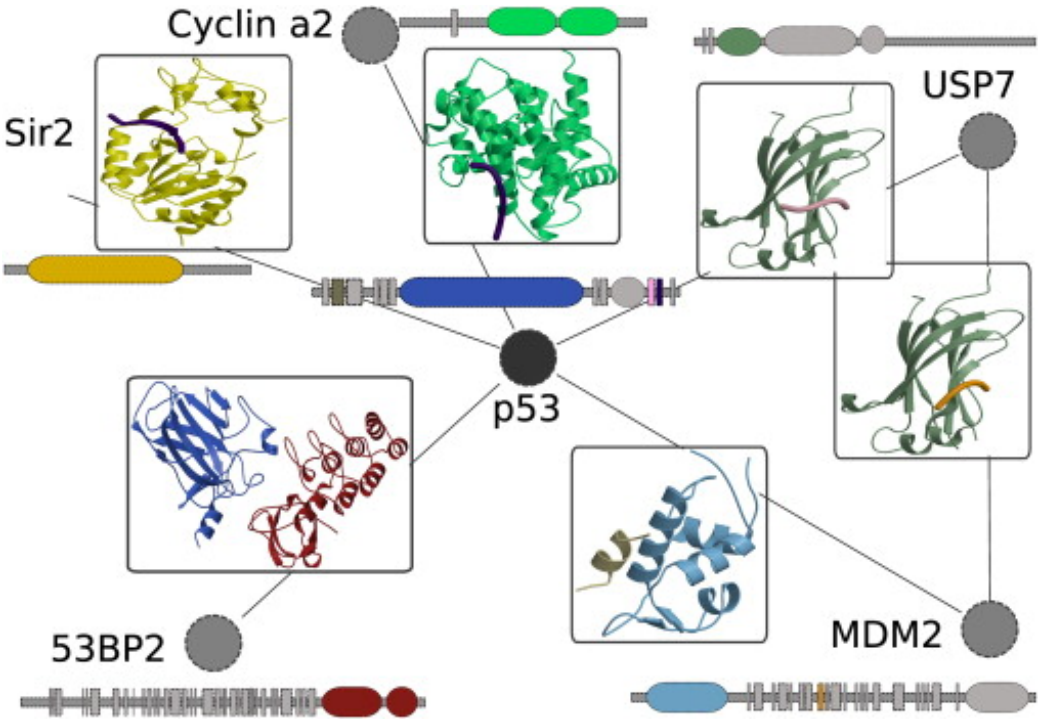
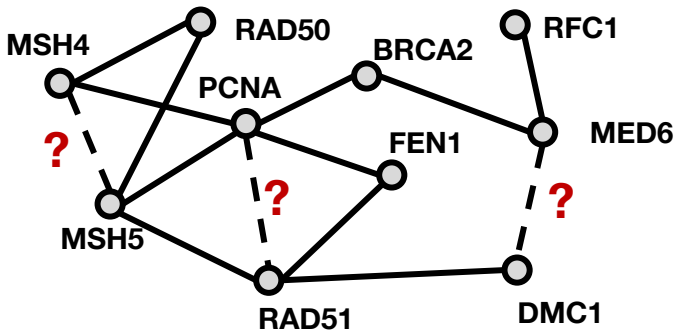


Image from: Perkins et al. Transient Protein-Protein Interactions: Structural, Functional, and Network Properties. *Structure*. 2010.

Network Data

- Human PPI network:
  - Experimentally validated physical protein-protein interactions
- **Link prediction:** Given two proteins, predict probability that they interact



Learning Edge Embeddings

- **So far:** Methods learn embeddings for nodes:
  - Great for tasks involving individual nodes (e.g., node classification)
- **Question:** How to address tasks involving pairs of nodes (e.g., link prediction)?
- **Idea:** Given u and v, define an operator g that generates an embedding for pair (u, v):

$$\boldsymbol{z}_{(u,v)} = g(u, v)$$

Learning Edge Embeddings

How to define operator g?

- **Desiderata:** The operator needs to be defined for any pair of nodes, even if the nodes are not connected
- We consider four choices for g:

Scoring node pairs	Definition
(a) Average	$[\mathbf{z}_u \boxplus \mathbf{z}_v]_i = \frac{\mathbf{z}_u(i)+\mathbf{z}_v(i)}{2}$
(b) Hadamard	$[\mathbf{z}_u \boxdot \mathbf{z}_v]_i = \mathbf{z}_u(i) * \mathbf{z}_v(i)$
(c) Weighted-L1	$\|\mathbf{z}_u \cdot \mathbf{z}_v\|_{\bar{1}i} = \|\mathbf{z}_u(i) - \mathbf{z}_v(i)\|$
(d) Weighted-L2	$\|\mathbf{z}_u \cdot \mathbf{z}_v\|_{\bar{2}i} = \|\mathbf{z}_u(i) - \mathbf{z}_v(i)\|^2$

Experimental Setup

- We are given a PPI network with a certain fraction of edges removed:
  - Remove about 50% of edges
  - Randomly sample an equal number of node pairs at random which have no edge connecting them
- Two main stages:
  1. Use node2vec to learn an embedding for every node in the filtered PPI network
  2. Predict a score for every protein pair in the test set based on the embeddings

PPI Prediction: Results

Op	Algorithm	Dataset		
		Facebook	PPI	arXiv
	Common Neighbors	0.8100	0.7142	0.8153
	Jaccard's Coefficient	0.8880	0.7018	0.8067
	Adamic-Adar	0.8289	0.7126	0.8315
	Pref. Attachment	0.7137	0.6670	0.6996
	Spectral Clustering	0.5960	0.6588	0.5812
(a)	DeepWalk	0.7238	0.6923	0.7066
	LINE	0.7029	0.6330	0.6516
	node2vec	0.7266	0.7543	0.7221
	Spectral Clustering	0.6192	0.4920	0.5740
(b)	DeepWalk	**0.9680**	0.7441	0.9340
	LINE	0.9490	0.7249	0.8902
	node2vec	**0.9680**	**0.7719**	**0.9366**
	Spectral Clustering	0.7200	0.6356	0.7099
(c)	DeepWalk	0.9574	0.6026	0.8282
	LINE	0.9483	0.7024	0.8809
	node2vec	0.9602	0.6292	0.8468
	Spectral Clustering	0.7107	0.6026	0.6765
(d)	DeepWalk	0.9584	0.6118	0.8305
	LINE	0.9460	0.7106	0.8862
	node2vec	0.9606	0.6236	0.8477

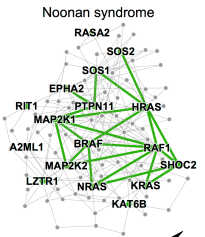
F1 – scores are in [0,1], higher is better

- Learned embeddings drastically outperform heuristic scores
- **Hadamard operator**:
  - Highly stable
  - Best average performance

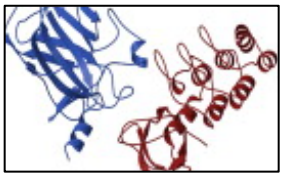
Scoring node pairs	Definition
(a) Average	$[\mathbf{z}_u \boxplus \mathbf{z}_v]_i = \frac{\mathbf{z}_u(i)+\mathbf{z}_v(i)}{2}$
(b) Hadamard	$[\mathbf{z}_u \boxdot \mathbf{z}_v]_i = \mathbf{z}_u(i) * \mathbf{z}_v(i)$
(c) Weighted-L1	$\|\mathbf{z}_u \cdot \mathbf{z}_v\|_{\bar{1}i} = \|\mathbf{z}_u(i) - \mathbf{z}_v(i)\|$
(d) Weighted-L2	$\|\mathbf{z}_u \cdot \mathbf{z}_v\|_{\bar{2}i} = \|\mathbf{z}_u(i) - \mathbf{z}_v(i)\|^2$

Biomedical Applications

1. Disease pathway detection:
   - Identify proteins whose mutation is linked with a particular disease
   - **Task:** Multi-label node classification



2. Protein interaction prediction:
   - Identify protein pairs that physically interact in a cell
   - **Task:** Link prediction



Outline of This Section

1. Random walk approaches ✓
2. Biomedical applications ✓

Outline of this Lecture

1) Biological networks

- Why networks? Why is learning on networks hard?

2) Node embeddings

- *Methodology:* Map nodes to vector representations
- *Applications:* PPIs, Disease pathways

3) Heterogeneous networks

- Methodology: Embedding heterogeneous networks
- *Applications:* Human tissues

Part 3: Heterogeneous Networks

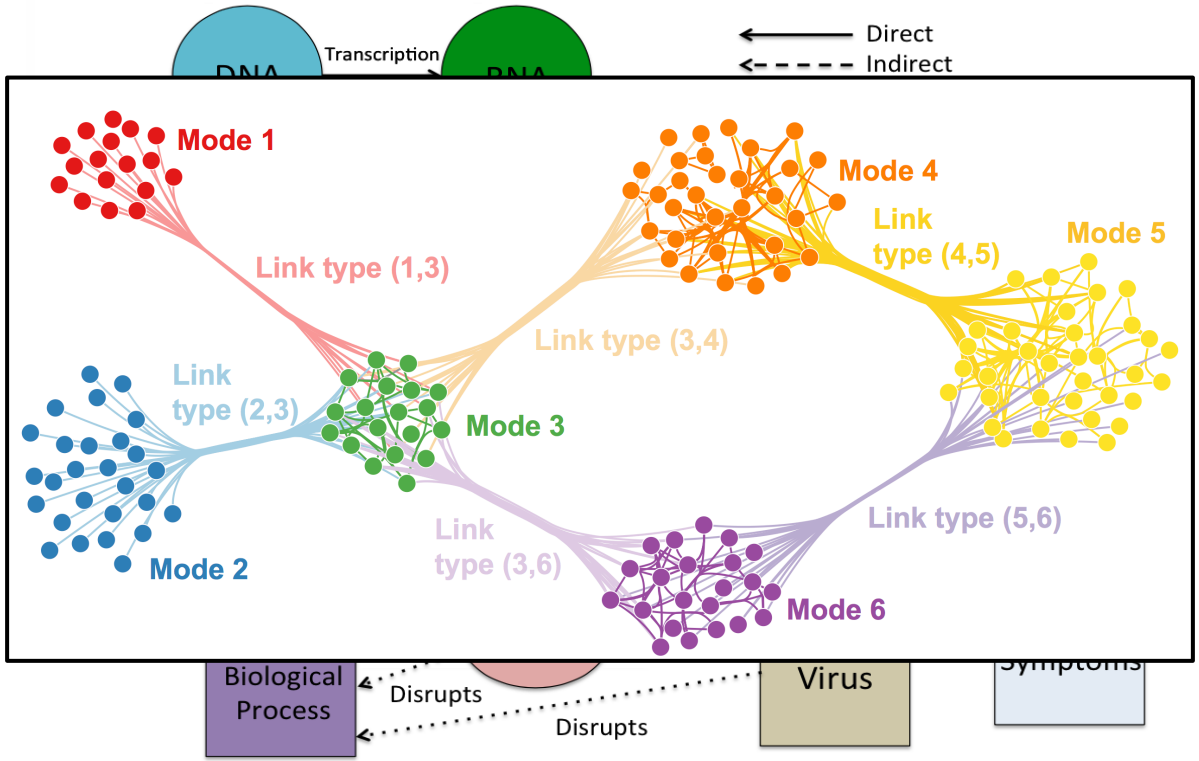
Based on material from:

- Zitnik et al., 2017. Predicting multicellular function through multi-layer tissue networks. *ISMB & Bioinformatics.*
- Zitnik et al., 2019. Evolution of resilience in protein interactomes across the tree of life. *Proceedings of the National Academy of Sciences* 116 (10), 4426-4433.

So far we focused on homogeneous networks!

Can we embed heterogeneous networks, i.e., het nets, knowledge graphs?

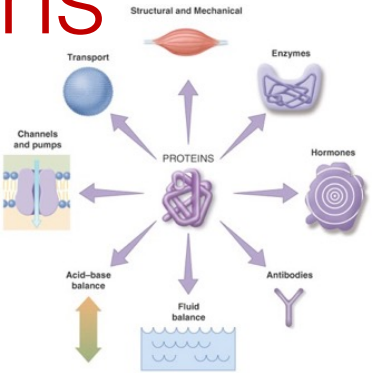
Many Het Nets in Biology



Motivating Problem: Prediction of Protein Functions

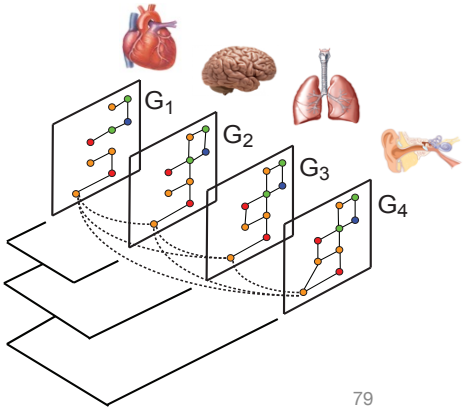
Proteins are worker molecules

- Biomedical and pharma implications



Functions depend on tissue context

- Proteins in similar tissues share similar features
- Functions in heart are different from functions in the brain, etc.



Why is protein function prediction across tissues hard?

1) Multiscale, hierarchical organization of tissues:
   - Tissues are related to each other
   - Proteins in similar tissues have similar functions

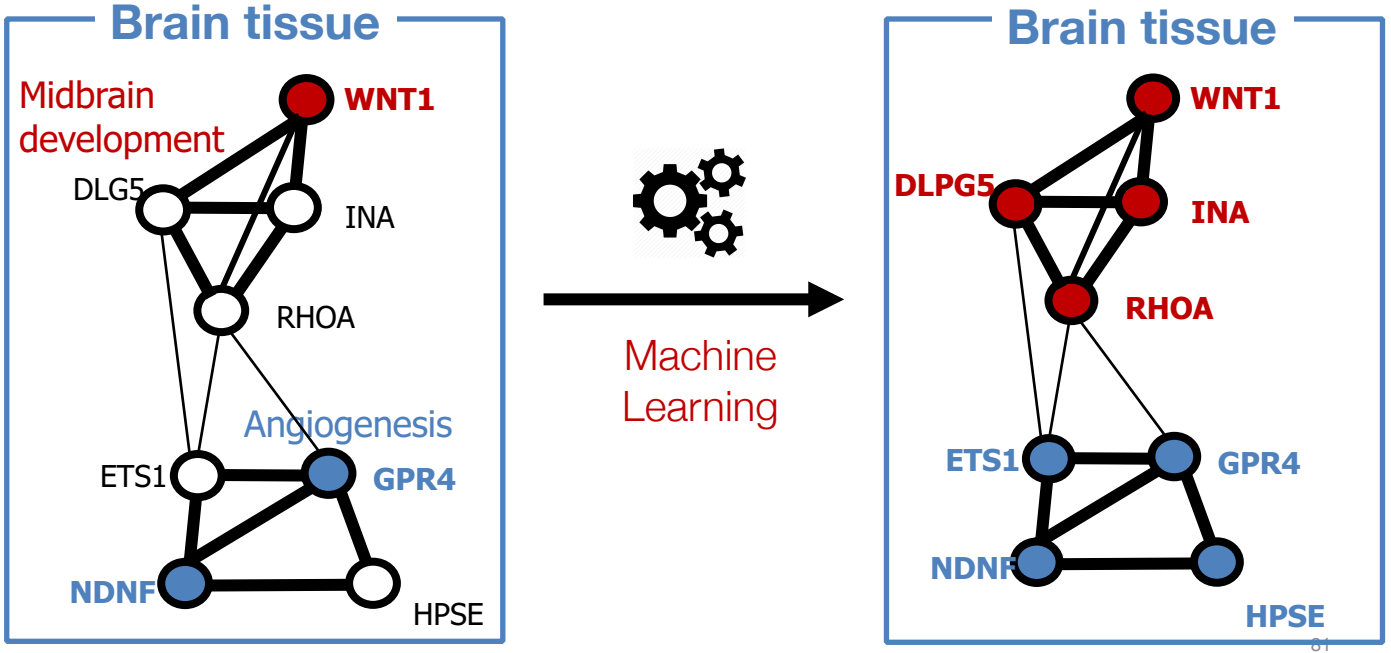
2) Many tissues have no annotations:
   - Need to predict functions in a tissue without any protein functions (node labels) in that tissue

3) Previous research [Radivojac et al.'13, Cho et al.'16; Kramer et al.'14; Yu et al.'15; etc.]
   - Protein functions assumed constant across tissues
     - Functions in heart are the same as in skin
     - Functions in the brain are the same as in skin

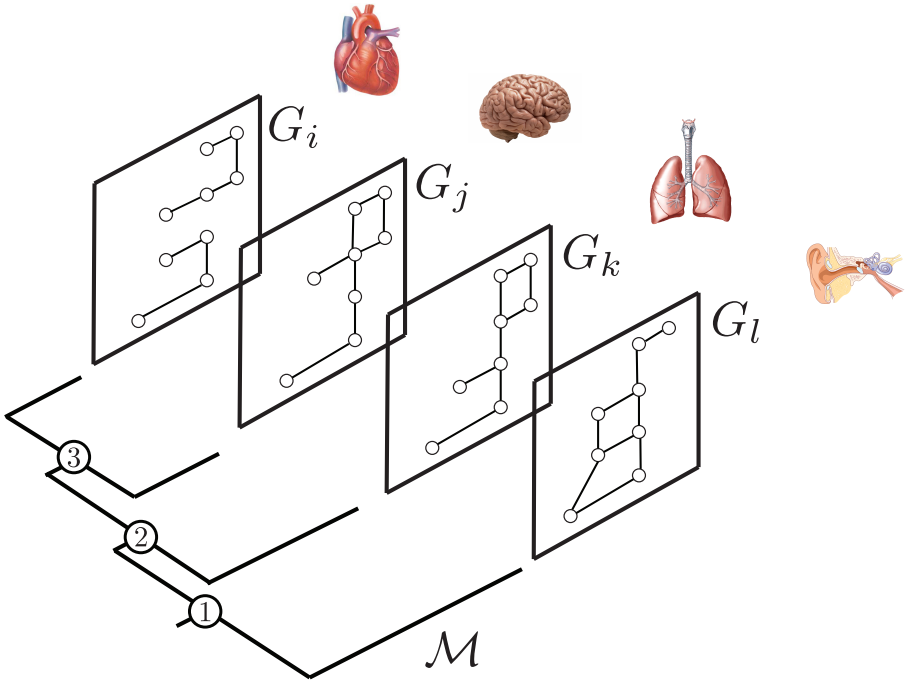
Motivating Problem: What Does My Protein Do?

Goal: Given a protein, a tissue, and a function, predict how likely the protein has that function in that tissue

$$\text{Protein} \times (\text{Function}, \text{Tissue}) \rightarrow [0,1]$$

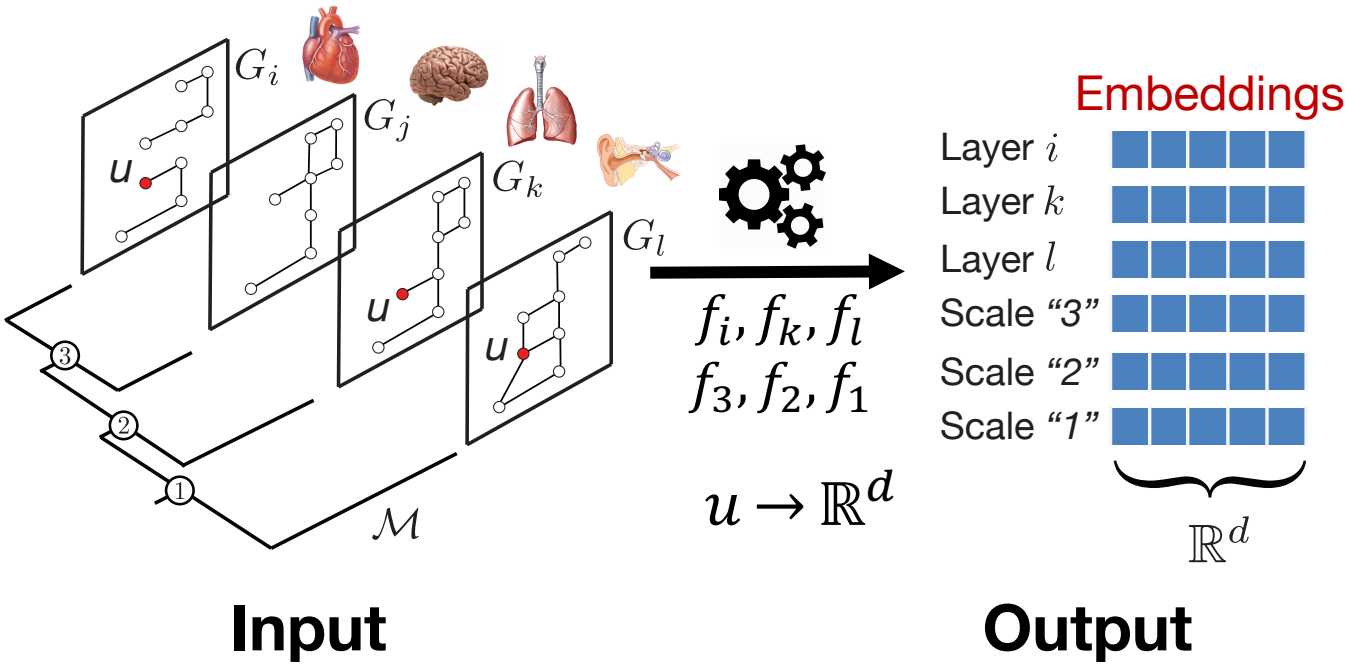


Multimodal Tissue Networks



Tissue-specific protein interaction networks + tissue hierarchy

Multimodal Tissue Networks



Input **Output**

How to learn mapping functions f_i?

Setup: Multimodal Networks

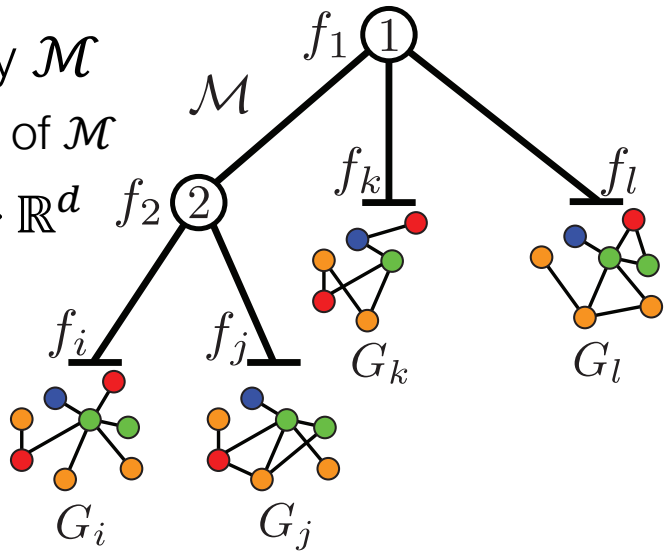
Input: Graphs $\{G_i\}_i$, hierarchy $\mathcal{M}$

- Graphs $\{G_i\}_{i=1..T}$ are in leaves of $\mathcal{M}$

Goal: Learn functions: $f_i: V_i \rightarrow \mathbb{R}^d$

Multi-scale model:

- Four layers: i, j, k, l
- Three scales: “3”, “2”, “1”



Output: Node embeddings:

- For each graph G_i
- For each sub-hierarchy

Embedding Approach

Two components:

1. For each graph G_i:

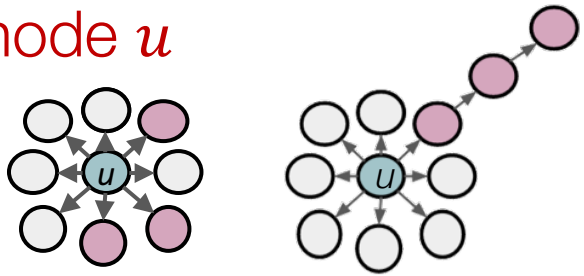
   Embed nodes with similar local topology close together

2. For hierarchy $\mathcal{M}$:

   Encourage nodes in similar graphs to share similar features

Single-Graph Objective

- Intuition: In each graph, embed nodes to d dimensions
- Approach: Nodes u and v are similar if their network neighborhoods are similar
- Given node u in graph G_i, neighborhood $N_i(u)$ is defined based on random walks starting at node u



Single-Graph Objective

- Given node u in graph $\mathbf{G}_i$, learn u's embedding such that it predicts nearby nodes $N_i(u)$:

$$\omega_i(u) = \log Pr(N_i(u)|f_i(u))$$

- Given T graphs $\{G_i\}_{i=1..T}$, maximize:

$$\Omega_i = \sum_{u \in V_i} \omega_i(u), \quad \text{for } i = 1, 2, \ldots, T$$

Summary so Far

We have not yet considered hierarchy $\mathcal{M}$:

- Node embeddings in different graphs are learned independently of each other

How to model dependencies between graphs when learning node embeddings?

Recall: Hierarchy of Graphs

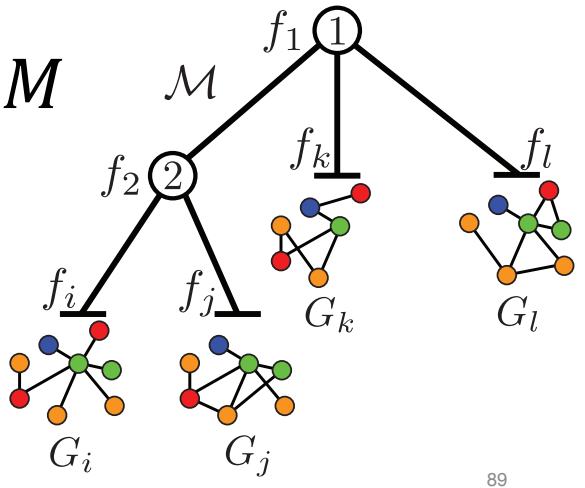
- Hierarchy M is a tree, given by the parent-child relationships:

$$\pi : M \rightarrow M$$

- $\pi(i)$ is parent of i in M

Example:

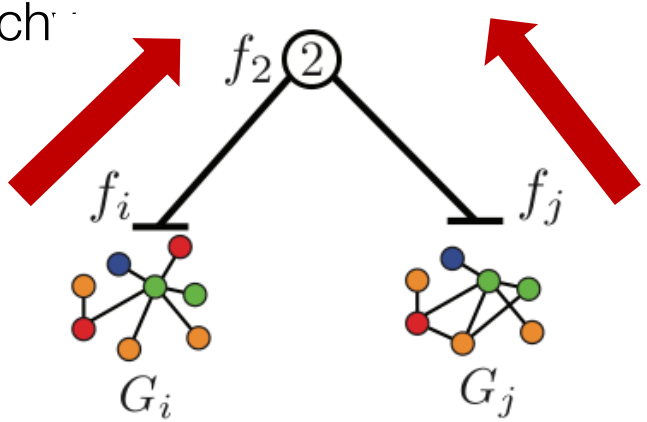
"2" is parent of G_i, G_j



Cross-Graph Objective

For hierarchy $\mathcal{M}$:

- Encode dependencies between graphs G_i
- **Recursive regularization:**
  - Embeddings at level i are encouraged to be similar to embeddings in i's parent in the hierarch



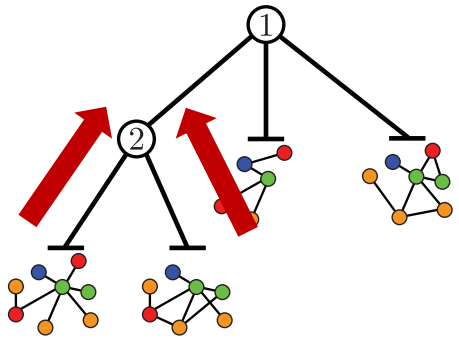
Cross-Graph Objective

- Given u, learn u's embedding in $\mathbf{G}_i$ to be close to u's embedding in parent $\pi(i)$:

$$c_i(u) = \frac{1}{2}\|f_i(u) - f_{\pi(i)}(u)\|_2^2$$

- Multi-scale: Repeat at every level of $\mathcal{M}$

$$C_i = \sum_{u \in L_i} c_i(u)$$



L_i has all graphs appearing in sub-hierarchy rooted at i

Embedding Approach: Optimization

Solve the maximum likelihood problem:

$$\max_{f_1, f_2, \ldots, f_{|M|}} \sum_{i \in \mathcal{T}} \Omega_i - \lambda \sum_{j \in \mathcal{M}} C_j$$

Single-graph objective

Cross-graph objective

Embedding Approach: Algorithm

1. For each graph G_i:
   - Sample fixed-length random walks starting from each node $u \in G_i$
2. Optimize the objective using stochastic gradient descent

Scalability: No pairwise comparison of nodes from different graphs:

$$O(\sum_{i,j}|V_i||V_j|) \quad \rightarrow \quad O(T\sum_i |V_i|)$$

Biomedical Application

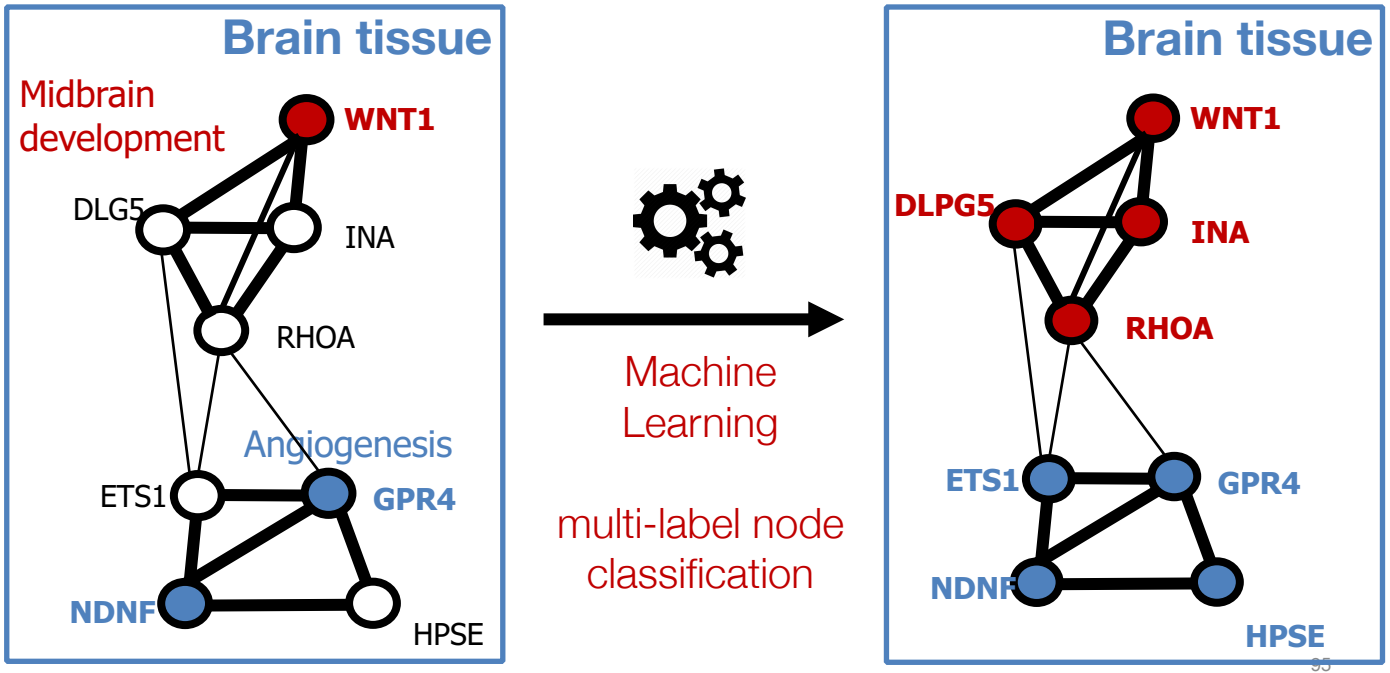
Based on material from:

- Zitnik et al., 2017. Predicting multicellular function through multi-layer tissue networks. *ISMB & Bioinformatics.*
- Wang, Pourshafeie, Zitnik, et al., 2018. Network Enhancement as a general method to denoise weighted biological networks. *Nature Communications.*

What Does My Protein Do?

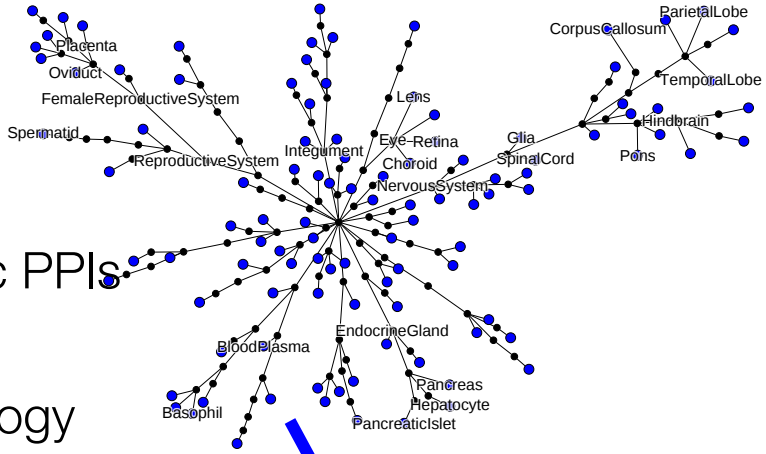
Goal: Given a protein, a tissue, and a function, predict how likely the protein has that function in that tissue

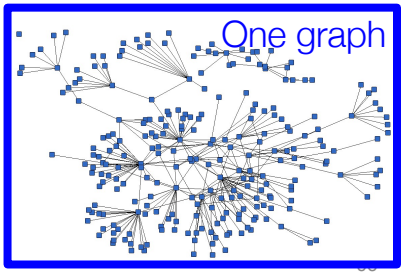
$$\text{Protein} \times (\text{Function}, \text{Tissue}) \rightarrow [0,1]$$



Data: 107 Tissue Graphs

- Graphs are PPI nets:
  - Nodes: proteins
  - Edges: tissue-specific PPIs
- Tissue hierarchy:
  - BRENDA tissue ontology
- Node labels:
  - Tissue-specific protein functions
  - GIANT / Human Base:
    - E.g., Function Cortex development in renal cortex tissue
    - E.g., Function Artery morphogenesis in artery tissue

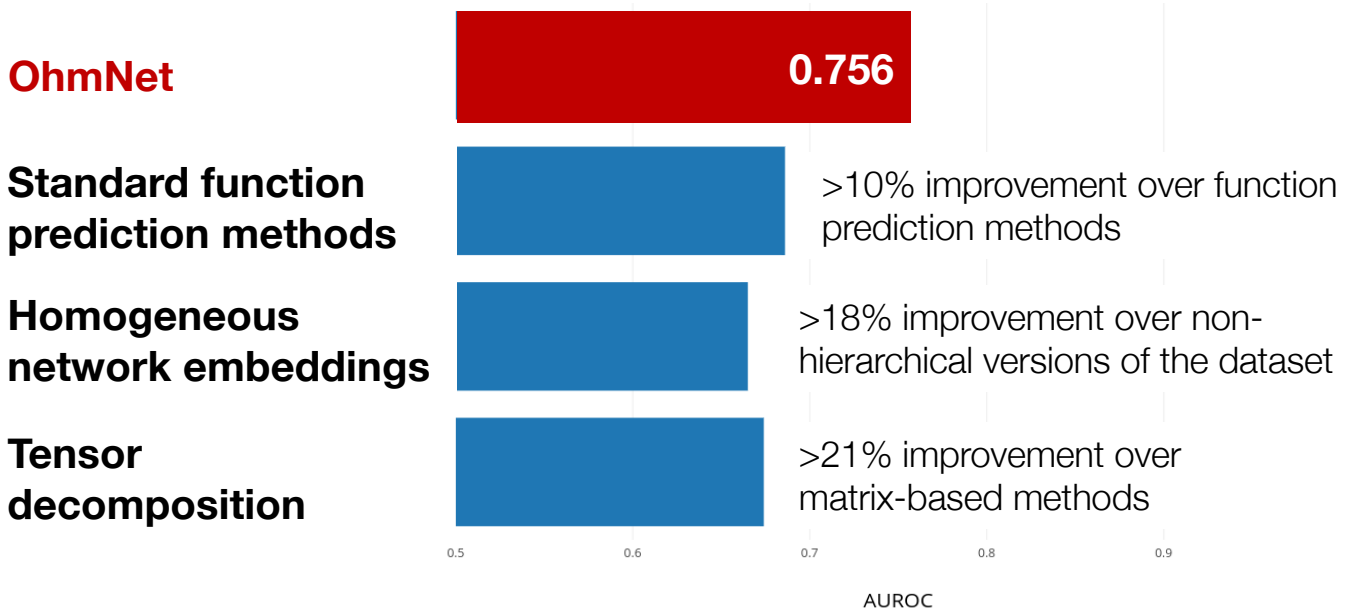




Experimental Setup

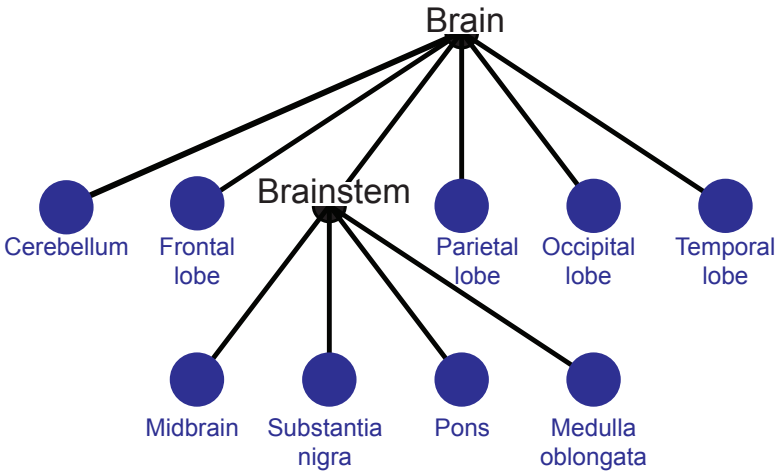
- Task: Multi-label node classification:
  - E.g., Does RPT1 play a role in angiogenesis in blood tissue?
- Every node (protein) is assigned one or more labels (functions)
- Setup:
  - Learn features for multimodal network
  - Train a classifier for each function based on a fraction of proteins and all their functions
  - Predict functions for new proteins

Results: Protein Function Prediction Across Tissues

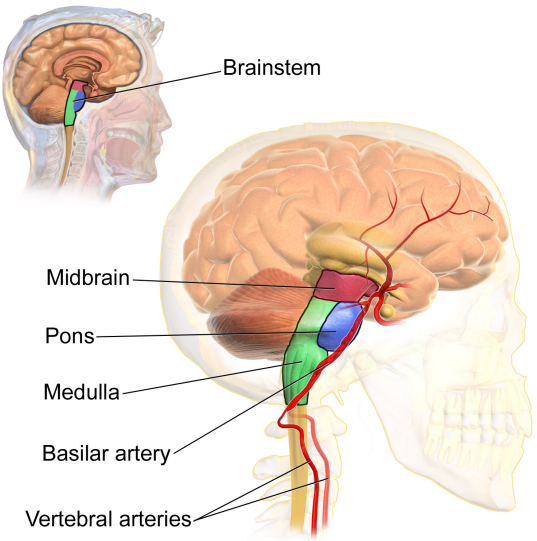


Substantial improvement over methods that ignore tissue-specific information

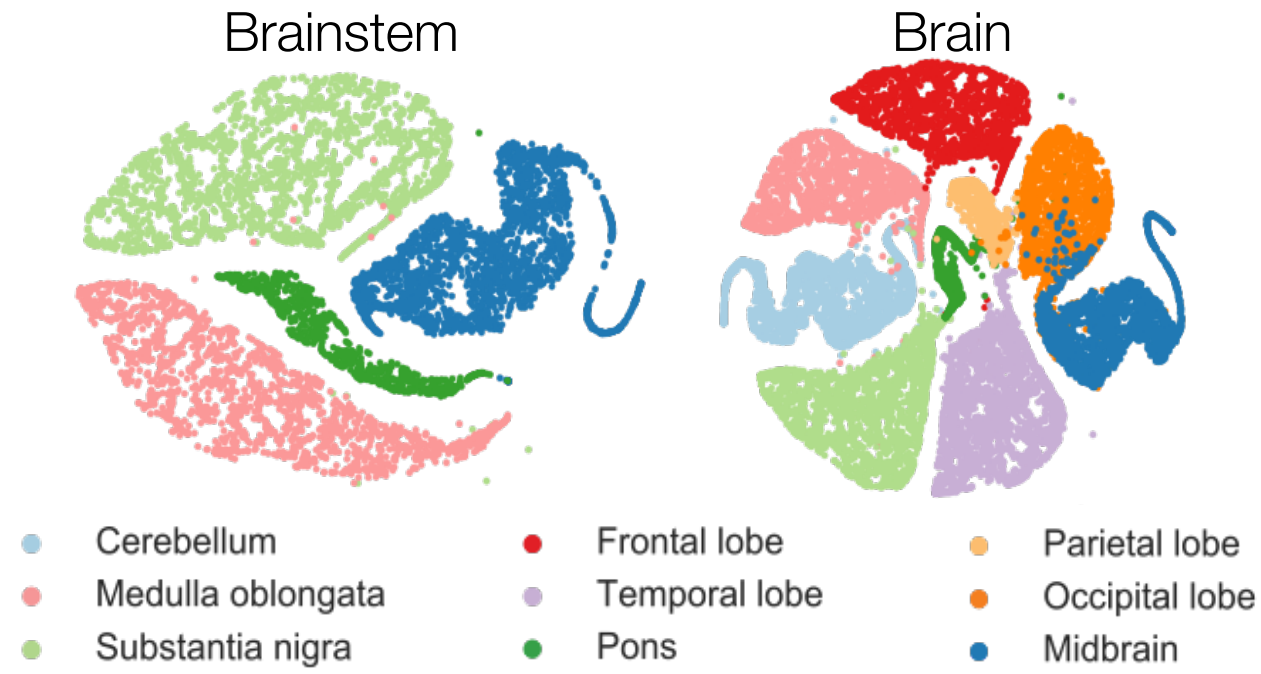
Case Study: 9 Brain Tissues



9 brain tissue PPI networks
in two-level hierarchy



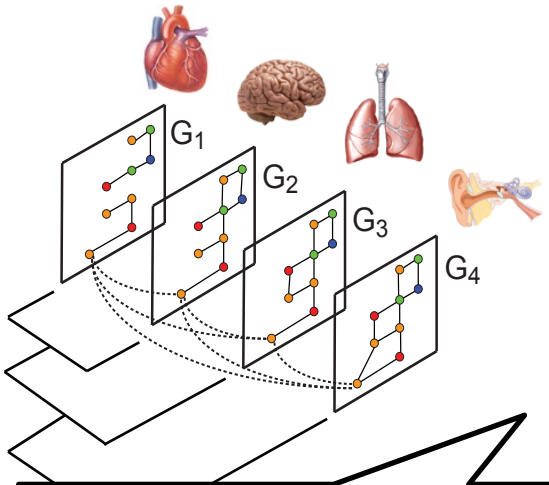
Multi-Scale Node Embeddings



Learned protein embeddings match human anotomy

Application: Transfer Learning

Predict protein function in a brand new functionally uncharacterized tissue



Target tissue	AUROC (Original)	AUROC (Transfer)
Natural killer cell	0.834 (± 0.076)	0.776 (± 0.063)
Placenta	0.830 (± 0.082)	0.758 (± 0.068)
Spleen	0.803 (± 0.030)	0.779 (± 0.043)
Liver	0.803 (± 0.047)	0.741 (± 0.025)
Forebrain	0.796 (± 0.036)	0.755 (± 0.037)
Macrophage	0.789 (± 0.037)	0.724 (± 0.024)
Epidermis	0.785 (± 0.030)	0.749 (± 0.032)
Hematopoietic stem c.	0.784 (± 0.035)	0.744 (± 0.036)
Blood plasma	0.784 (± 0.027)	0.703 (± 0.039)
Smooth muscle	0.778 (± 0.031)	0.729 (± 0.041)
Average	0.799	0.746

Task: Predict functions in target tissue without access to any annotation/label in that tissue

42% improvement over baselines

Outline of this Lecture

1) Biological networks

- Why networks? Why is learning on networks hard?

2) Node embeddings

- *Methodology:* Map nodes to vector representations
- *Applications:* PPIs, Disease pathways

3) Heterogeneous networks

- Methodology: Embedding heterogeneous networks
- *Applications:* Human tissues

Lecture Resources

- **MAMBO:** Multimodal biomedical networks
  - Scales to networks with 2.3 billion edges and over 2,000 modes
  - snap.stanford.edu/mambo
- **Network data:**
  - snap.stanford.edu/projects.html:
    - CRank, Decagon, MAMBO, NE, OhmNet, Pathways, Tree of Life, and many others
  - snap.stanford.edu/biodata
    - Algorithm benchmarking, method development
    - Easy to link entities across datasets

Networks and relationships

Name	Edges	Entities	Description
CC-Neuron	49,471,006	cell, cell	Similarity network between cells in embroyonic mouse brain
ChCh-Miner	96,137	drug, drug	Interactions between FDA-approved drugs
ChChSe-Decagon	4,649,441	drug, drug, side-effect	Side effects of drug combinations
ChG-InterDecagon	131,034	drug, gene	Chemical-gene interaction network

Two Lectures

Part1: May 15, 2019, 2:30 pm - 4:00 pm

- Methodology: Shallow network embeddings:
  - Map nodes to low-dimensional features
- Resources: Data, tools, codebases
- Applications: PPIs, Disease pathways, Tissues

Part2: May 16, 2019, 9:00 am – 10:30 am

- Methodology: Deep network embeddings:
  - Graph neural networks for rich biomedical graphs
- Resources: Data, practical advice and demos
- Applications: Polypharmacy, Drug repurposing