Biological Networks

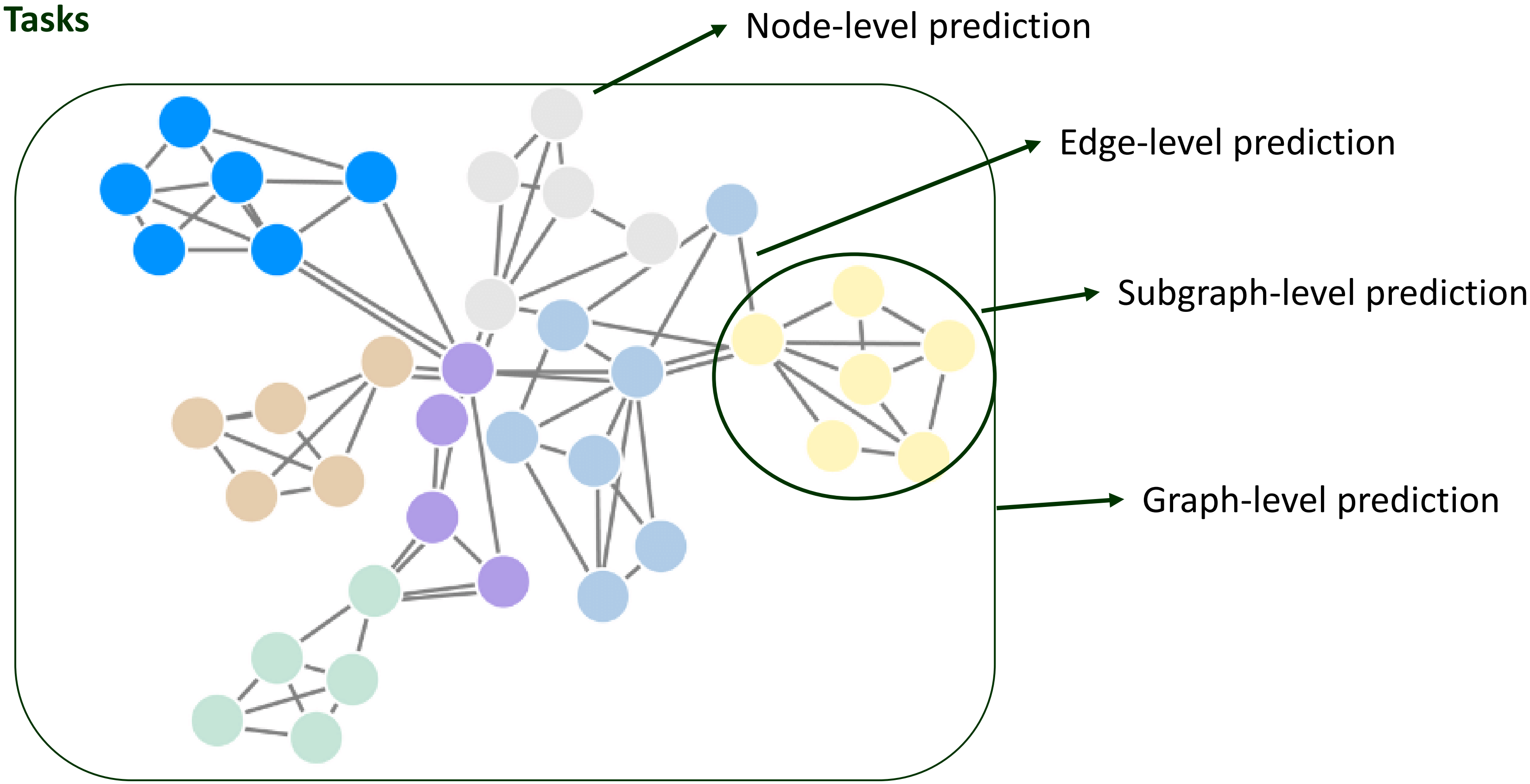
Young-Rae Cho, Ph.D.
Professor
Division of Software / Division of Digital Healthcare
Yonsei University – Mirae Campus

Network Analytics

❑ Representation

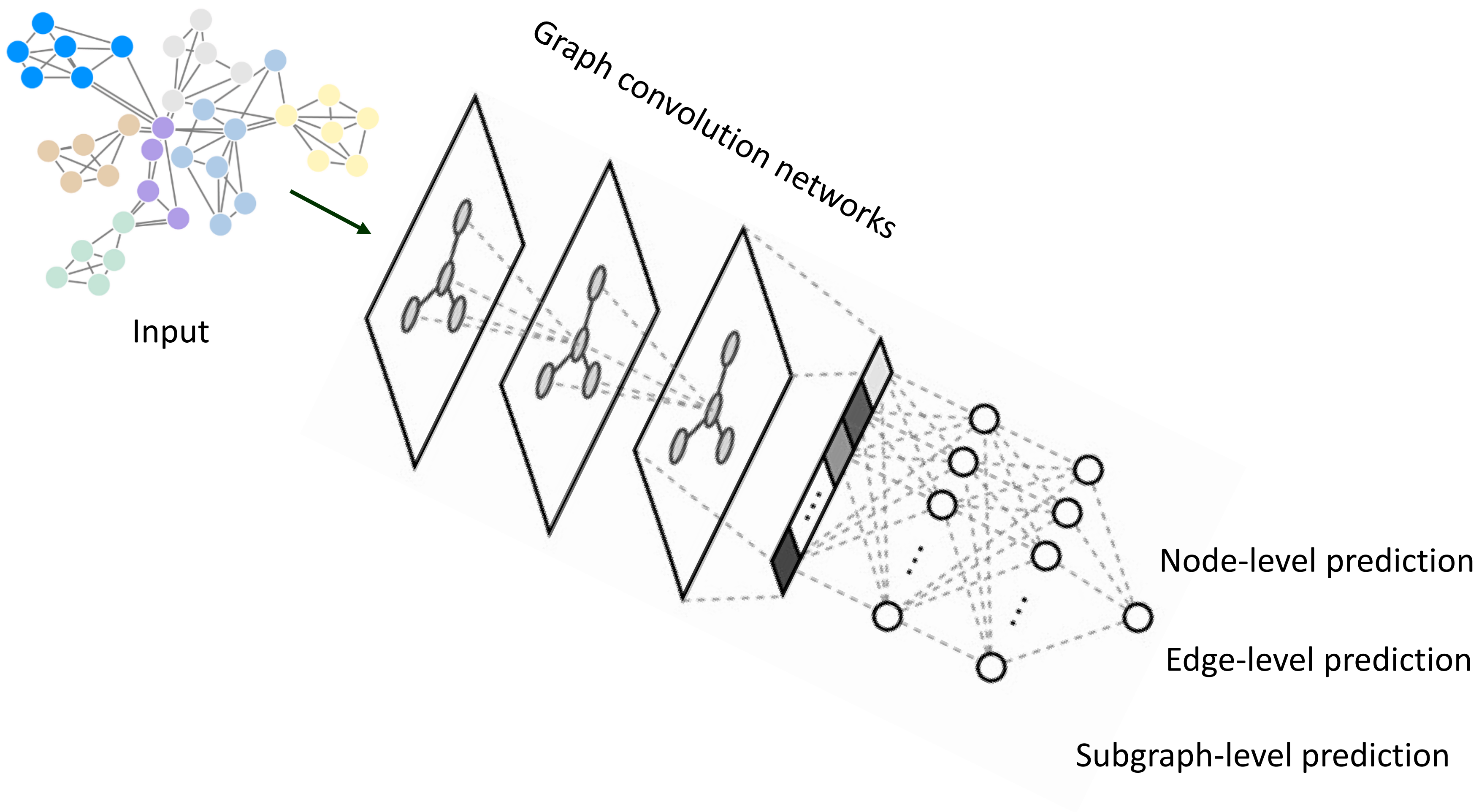
- Graph: an ordered pair G(V,E) with a set of vertices (or nodes) V and a set of edges E

❑ Tasks



Deep Learning with Networks

- Recent Tasks



Network Types

- **Extended Graph Representation**
  - directed graph vs. undirected graph
    - Whether each edge has a direction
  - weighted graph vs. unweighted graph vs. multi-graph
    - Whether each edge has a weight or allows multiple edges
  - labeled graph vs. unlabeled graph
    - Whether each vertex has a label
  - 0-D vs. 1-D vs. 2-D vs. 3-D graph representation
    - Whether each vertex has a specific coordinate
  - homogeneous network vs. heterogeneous network

    (nodes with node types, edges with relation types)

    (bipartite graph, tripartite graph, …)

Network Modeling – Random Graph Model

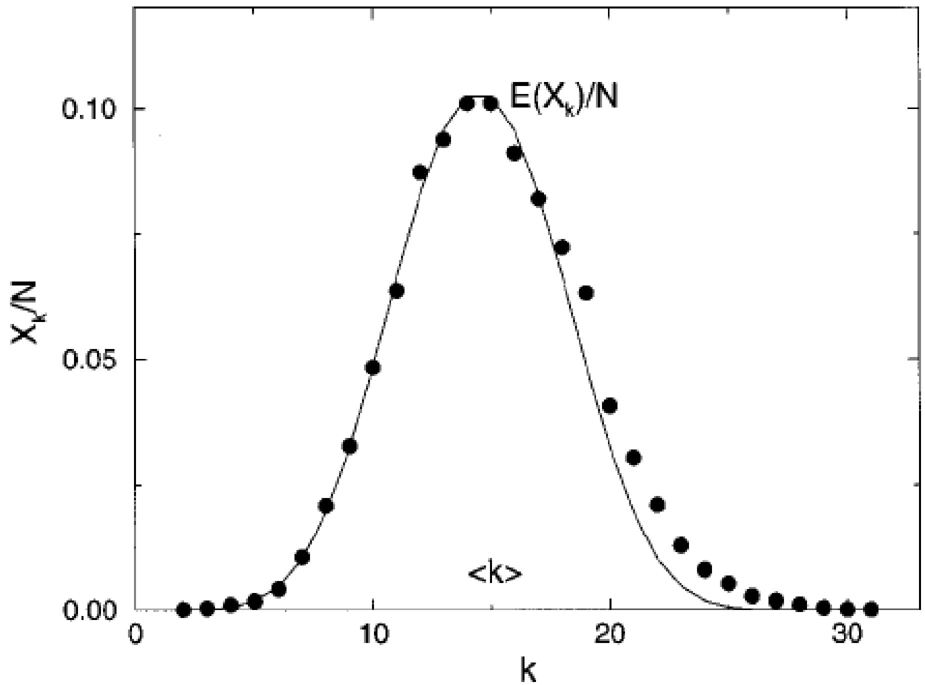
- **Erdós and Rényi, 1960**
  - Random graph as N nodes connected by m edges that are randomly chosen from N(N-1)/2 possible edges
  - m = p[N(N-1)/2] where p is the probability of each pair of nodes being connected
  - Degree distribution $P(k) = \binom{N-1}{k} p^k (1-p)^{N-1-k}$
    - Degree of v: the number of links from v to other nodes
    - Degree distribution P(k): probability that a node has k links
  - Expected number of nodes with degree k

$$E(X_k) = N\,P(k) = N \binom{N-1}{k} p^k (1-p)^{N-1-k} = \lambda_k \;\rightarrow\; P(X_k = r) = e^{-\lambda k} \lambda_k^{\,r} / r\,!$$

(Poisson distribution)

Examples of Random Graph Model

- **Example**
  - Poisson distribution with N = 1000 and p = 0.0015



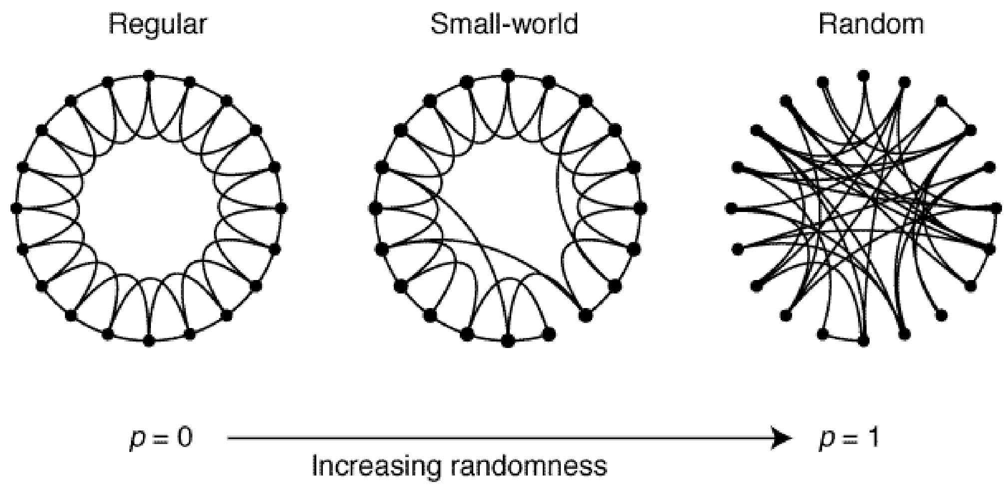
Topological Features in Random Graph Model

Topological Features

- Connectivity
  - ✓ Degree distribution → weighted graph? → directed graph?
  - ✓ Largest connected component → directed graph? (strongly connected component)
  - ✓ Density (Sparsity)
- Path
  - ✓ Average shortest path length → directed graph?
  - ✓ Diameter

Small-World Networks

- **Watts and Strogatz, 1998**
  - Average shortest path length grows to log of N; $L \propto \log N$



  - Hub-oriented structure
  - High clustering coefficient

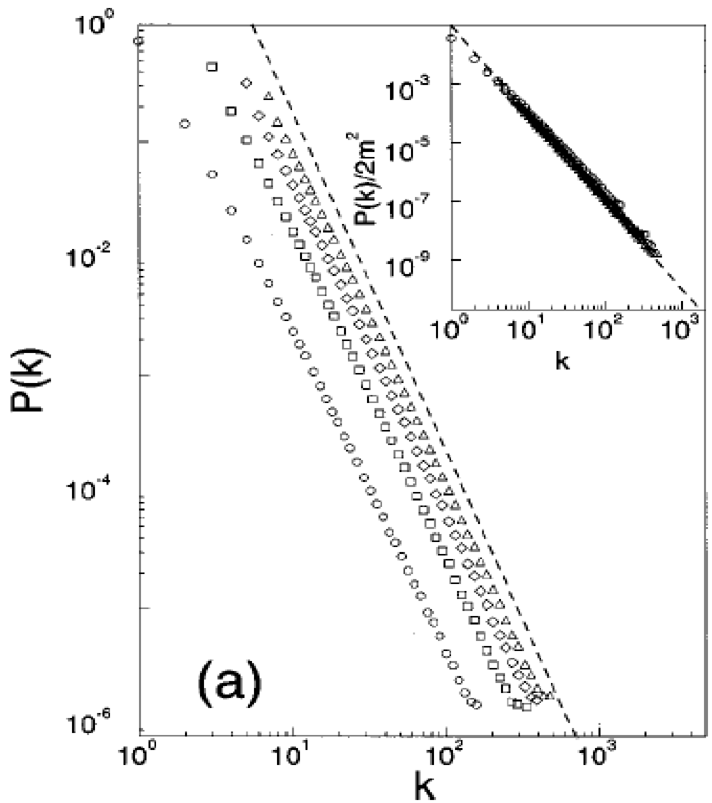
Network Modeling – Scale-Free Model

- **Barabasi and Albert, 1999**
  - Focused on network dynamics based on these two steps:
    - Growth: networks are continuously expanded by the addition of new nodes with a link to the nodes already present
    - Preferential attachment: the nodes are likely to be linked to high-degree nodes
  - Power-law degree distribution: $P(k) \sim k^{-\gamma}$ where $2 < \gamma < 3$
  - Features
    - A very few high-degree nodes and many low-degree nodes
      $\rightarrow$ scale-free degree distribution
    - Very low average shortest path length
      $\rightarrow$ small-world network
    - Hub-oriented structure
      $\rightarrow$ measuring hubness by centrality (e.g., degree, closeness)

Example of Scale-Free Model

➢ **Example**

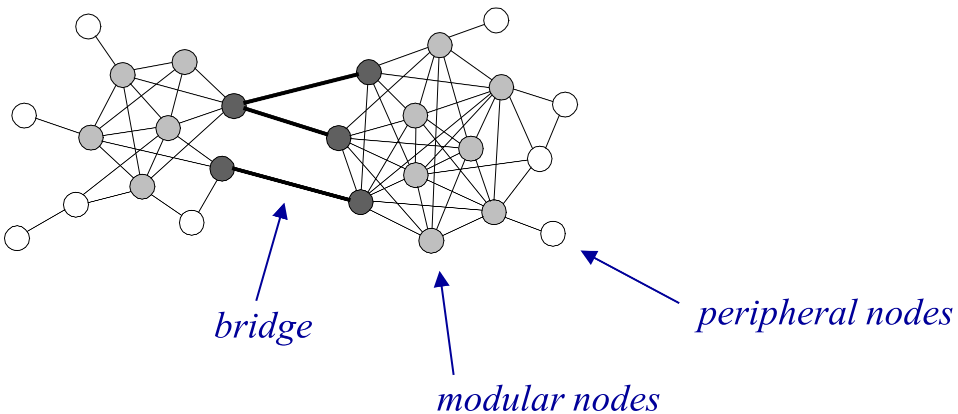
- Power-law degree distribution with the best fit of γ =2.9 on the dashed line



Network Modeling – Modular Networks

- **Modular Networks**
  - Dense intra-connections among the nodes in the same modules
  - Sparse inter-connections between two nodes in different modules

  - Verified by high average clustering coefficient
    - Clustering coefficient of a node v: the density of a subgraph with neighbors of v



Topological Features in Modular Network Model

- **Topological Features**
  - Centrality (Hubness) of a node
    - ✓ Degree
    - ✓ Closeness
    - ✓ Clustering coefficient
    - ✓ Eigenvector centrality

$$e_u = \frac{1}{\lambda} \sum_{v \in V} \mathbf{A}[u, v] e_v \; \forall u \in \mathcal{V},$$

  - Modularity of a graph
    - ✓ Density
    - ✓ Average clustering coefficient
  - Bridging factor of a node/an edge
    - ✓ Betweenness centrality
  - Subgraph pattern of a graph
    - ✓ Graphlet frequency

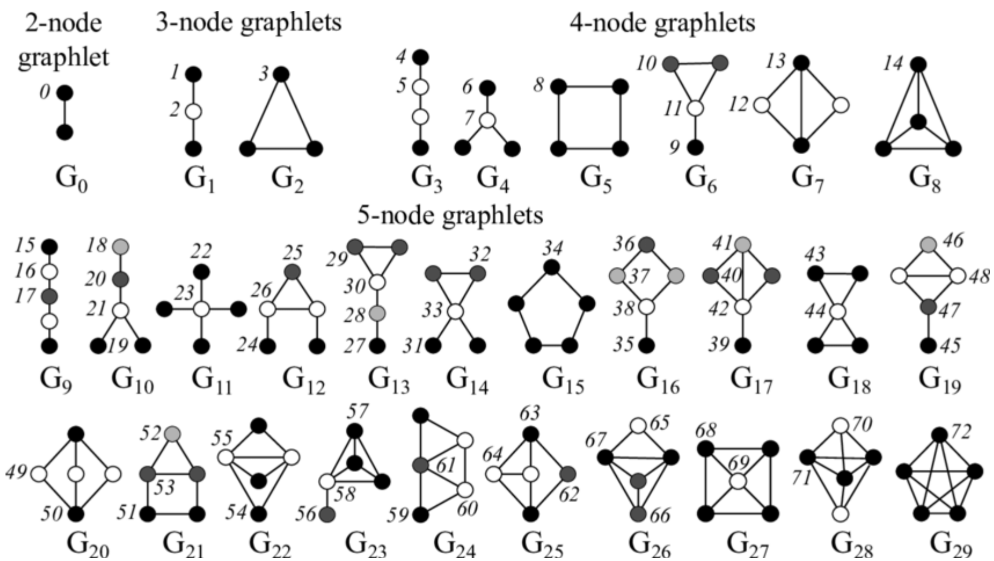
Topological Features in Modular Network Model

- **Betweenness centrality of nodes**
  - Betweenness of a vertex v_i, $C_B(v_i)$: the sum of the ratios of the shortest paths which pass through v_i
  - $C_B(v_i) = \sum_{s \neq v_i \neq t \in V} \frac{\sigma_{st}(v_i)}{\sigma_{st}}$ where σ_{st} is the number of shortest paths between s and t, and $\sigma_{st}(v_i)$ is the number of shortest paths between s and t, which pass through v_i
  - Detects the vertices located between two clusters

- **Betweenness centrality of edges**
  - Betweenness of an edge e_i, $C_B(e_i)$: the sum of the ratios of the shortest paths which pass through e_i

Topological Features in Modular Network Model

- **Graphlet (network motifs) frequency of a graph**
  - Graphlet: an induced, isomorphic subgraph

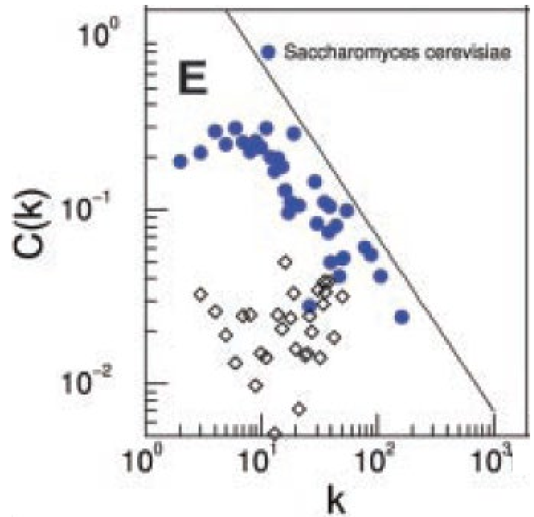


  - Relative graphlet frequency: $F_i(G) = -\log(N_i(G)/T(G))$ where $T(G) = \sum_{i=1}^{29} N_i(G)$
  - Graphlet-based distance: $D(G,H) = \sum_{i=1}^{29} |F_i(G) - F_i(H)|$

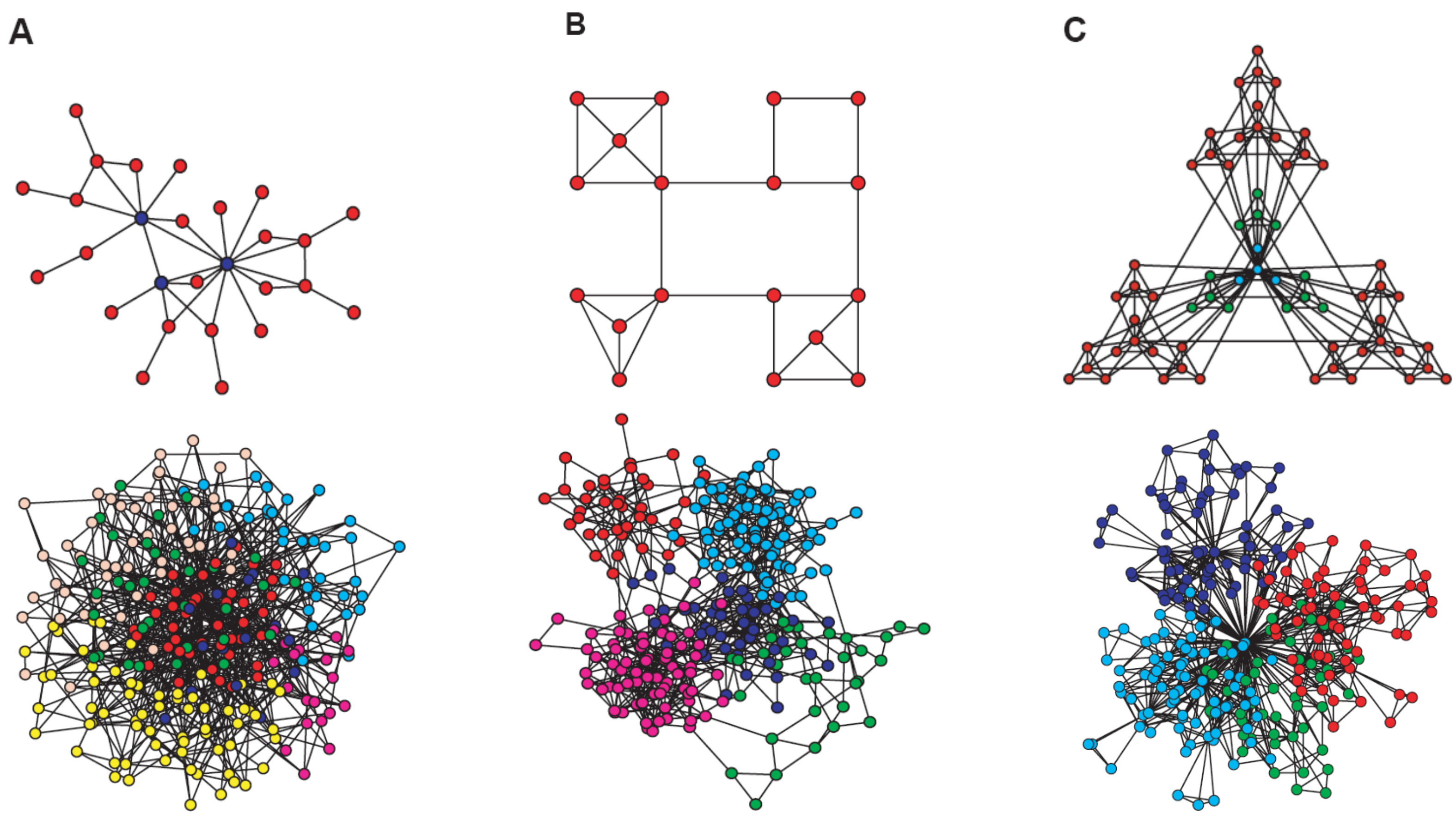
Network Modeling – Hierarchical Networks

Hierarchical Networks

- Integrated of scale-free topology with modular structure
- Hierarchy of modules is controlled by hubs
- Clustering coefficient distribution C
  - Scale-free network: C is independent of degree *k*
  - Hierarchical network: C ~ k^{-1}



Schematic View



Scale-free Networks **Modular Networks** **Hierarchical Networks**

1. Link Prediction (Association Prediction)

❑ Definition

- An association between two nodes is represented as a “true” link
- In G(V,E), E denotes a set of observed links
- The goal of link prediction is to identify the unobserved true links.

❑ Topology-based Methods

- Jaccard index of common neighbors
- Adamic-Adar measure

$$A(x,y) = \sum_{u \in N(x) \cap N(y)} \frac{1}{\log |N(u)|}$$

- Katz measure

$$\mathbf{S}_{\text{Katz}}[u,v] = \sum_{i=1}^{\infty} \beta^i \mathbf{A}^i [u,v]$$

❑ Feature-based Methods

- Cosine similarity of feature vectors

$$\frac{A \cdot B}{||A||\,||B||} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

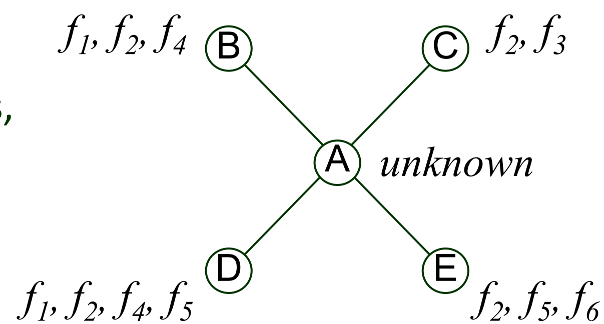
2. Node Classification (Function Prediction)

❑ Definition

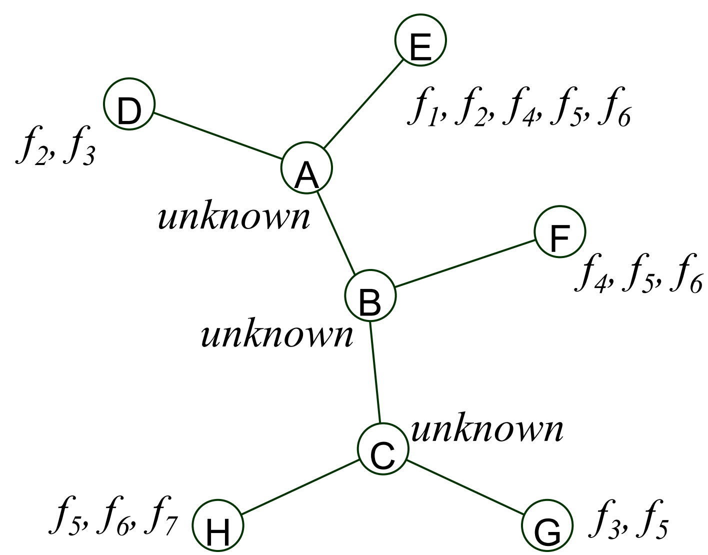
- The goal of node classification is to identify the class labels of unknown nodes.

❑ Topology-based Methods

- Majority voting of neighbors' classes, called "guilt-by-association"



- Random walk (for global optimization)



3. Graph Clustering (Module Detection)

❑ Definition

- The goal of graph clustering is to identify a set of subgraphs that are densely connected within each subgraph (adding periphery) and are sparsely connected between subgraphs.
- Functional module: a group of entities having the similar functions

❑ Topology-based Methods

- Density-Based Clustering
  - ✓ Searching for densely connected sub-graphs (local optimum)
- Partition-Based Clustering
  - ✓ Searching for optimal partition of a graph (global optimum)
- Hierarchical Clustering
  - ✓ Bottom-up approaches: Merging the closest nodes iteratively
  - ✓ Top-down approaches: Dividing a graph into two or more sub-graphs recursively

Examples of Graph Clustering

- Examples of **Density-Based Clustering**
  - Clique search / Clique percolation
  - k-core decomposition
  - Seed-growth approaches
    - ✓ Expanding seed clusters by density functions for local optimum

- Examples of **Partition-Based Clustering**
  - Restricted neighborhood search
    - ✓ Updating the partition repeatedly by moving restricted neighbors

- Examples of **Hierarchical Clustering**
  - Bottom-up approaches
  - Top-down approaches
    - ✓ Minimum cut / Edge betweenness cut

Examples of Density-Based Graph Clustering

- **MCODE**
  1. Vertex weighting
     - Uses density of k-core of the neighboring subgraph → core-clustering coefficient
  2. Module prediction
     - Seeds the highest weighted vertex
     - Includes recursively the vertices whose weight is above a given threshold
  3. Post-processing
     - "fluff" option: when a vertex is included, set the neighborhood density parameter
     - "haircut" option: when a vertex is included, remove low k vertex

Examples of Density-Based Graph Clustering

- **Graph Entropy**
  1. Seed cluster shrinking
     - Removes vertices in the neighboring subgraph of a seed based on graph entropy
  2. Seed cluster expansion
     - Adds vertices outside the current cluster based on graph entropy

- **Definition** of Graph Entropy
  - Inner links, Outer links
    - ✓ Inner links of v in G'(V',E'): edges from v to the vertices in V'
      - → $p_i(v)$: probability of v having inner links
    - ✓ Outer links of v in G'(V',E'): edges from v to the vertices not in V'
      - → $p_o(v)$: probability of v having outer links
  - Vertex entropy: $e(v) = - p_i(v) \log_2 p_i(v) - p_o(v) \log_2 p_o(v)$
  - Graph entropy : $e(G(V,E)) = \Sigma_{v \in V}\, e(v)$

4. Path Search (Pathway Identification)

❑ Definition

- (Signaling) Pathway: a series of linked nodes

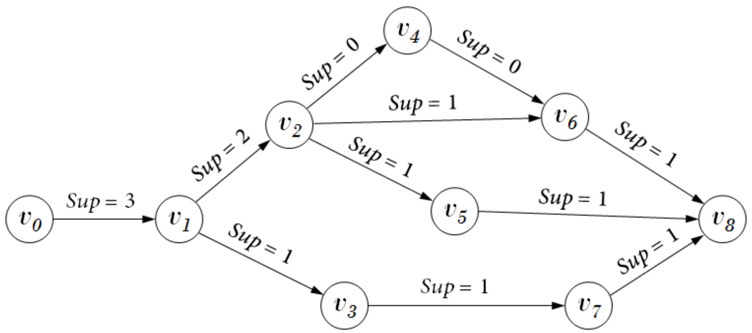
  - a series of genes having signaling and response relationship
- Signaling network: a combined form of linear signaling pathways

  (a directed acyclic graph)

❑ Topology-based Methods

- Strongest path search
  - ✓ Listing all possible paths to select the strongest path
  - ✓ Needs a heuristic algorithm (greedy search)

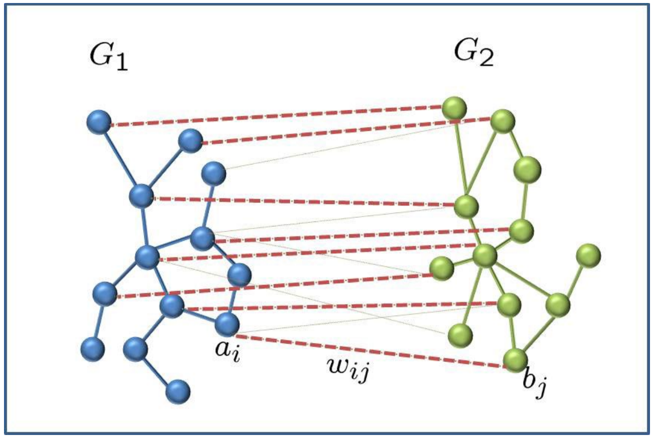
- Most frequent path search
  - ✓ Computing the number of shortest paths towards the target recursively



5. Network Alignment (Node Mapping)

❑ **Definition**

- Aligning two or more networks
- Mapping nodes that belong to the same entity from different networks



❑ **Applications**

- Aligning two or more protein-protein interaction networks to
  - Find ortholog pairs
  - Predict cellular functions
  - Predict conserved interactions
  - Measure evolutionary distance between PPI networks

Sequence Alignment vs. Network Alignment

- Sequence Alignment
  - Aligning two or more sequences
  - Searches matches (identical letters), mismatches (non-identical letters), and gaps
  - Returns alignment in 2-row representation including gaps

- Network Alignment
  - Aligning two or more networks
  - Searches matches (orthologs), mismatches (non-orthologs), and gaps
  - Returns an alignment network having ortholog pairs as nodes AND/OR conserved interactions as edges
  - Types
    - ✓ Global network alignment vs. Local network alignment
    - ✓ Pairwise network alignment vs. Multiple network alignment
    - ✓ 1-to-1 mapping vs. m-to-n mapping

Examples of Network Alignment Algorithms

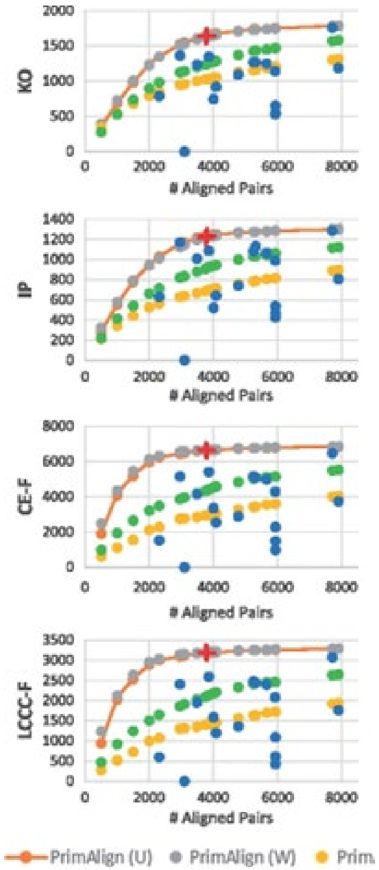
❑ Technical Issues

- Which features are used
- How to optimize the alignment network for multiple orthologs
- How to improve efficiency of network alignment

❑ PrimAlign (PageRank-Inspired Markovian Network Alignment)

- Algorithm
  1. Edge weighting
  2. Transition matrix building
  3. PageRank-inspired stationary distribution computation
  4. Inter-network traversal probability thresholding

- Experimental Results



Questions?

- Lecture Slides on the Course Website, "https://ads.yonsei.ac.kr/faculty/bioinformatics"