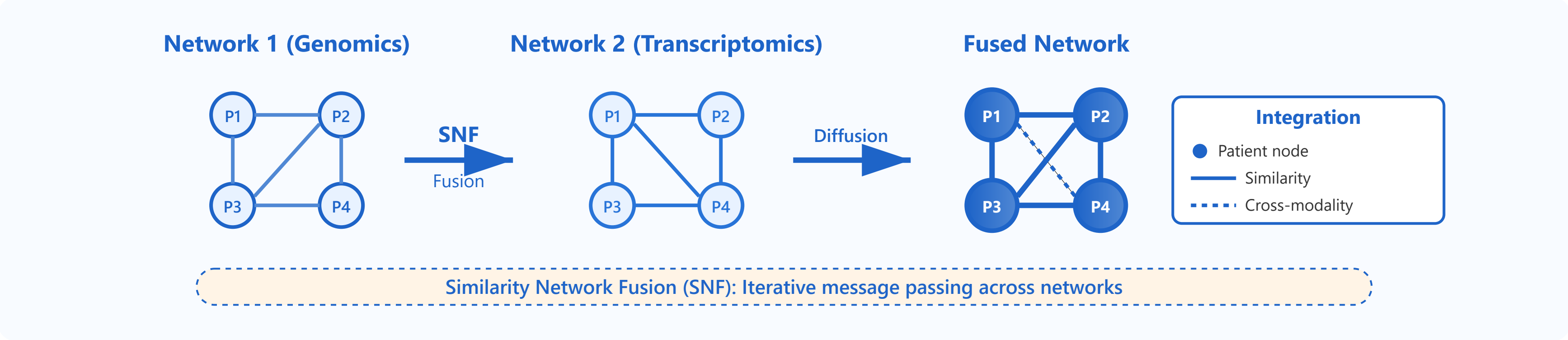
Graph-based Integration



Similarity Networks

Patient or feature similarity graphs

Network Fusion

SNF: fusing multiple similarity networks

Random Walk

Diffusion-based integration on networks

Graph Neural Networks

Deep learning on graph-structured data

Multiplex Networks

Multi-layer network representations

1. Similarity Networks

Overview

Similarity networks represent relationships between patients or features based on their shared characteristics across omics datasets. Each node represents a sample (patient) or feature (gene, protein), and edges represent similarity scores.

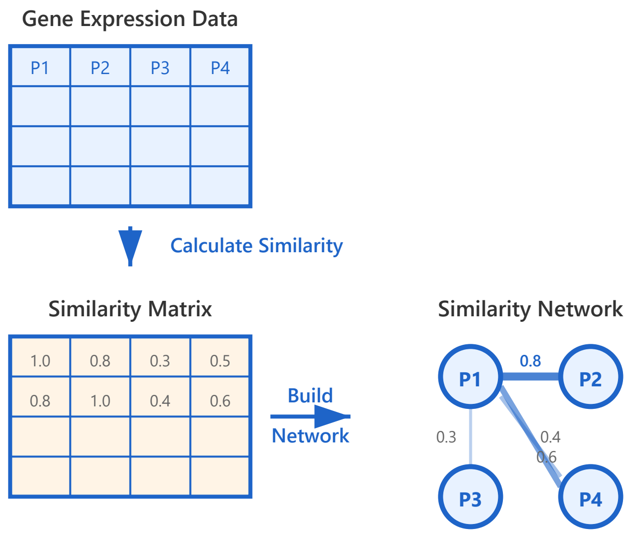
Construction Methods

- **K-Nearest Neighbors (KNN):** Connect each node to its k most similar neighbors
- **Gaussian Kernel:** Weight edges using exponential decay based on distance
- **Correlation-based:** Use Pearson or Spearman correlation coefficients

Applications

- Patient stratification and subtype discovery
- Feature selection and biomarker identification
- Disease trajectory modeling

Patient Similarity Network Construction



1.0	0.8	0.3	0.5
0.8	1.0	0.4	0.6

Key Points:

- Edge weights represent similarity strength between patients
- Thresholding removes weak connections for clearer network structure
- Multiple similarity metrics can be combined for robust networks

2. Network Fusion (SNF)

Similarity Network Fusion Algorithm

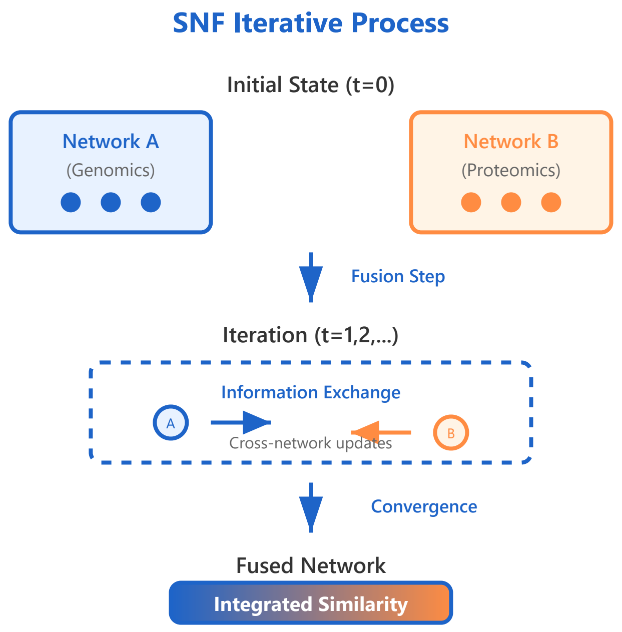
SNF integrates multiple patient similarity networks by iteratively updating each network using information from other networks. This cross-network information exchange creates a unified representation that captures complementary information.

Algorithm Steps

- **Step 1:** Construct individual similarity networks for each data type
- **Step 2:** Normalize networks to make them comparable
- **Step 3:** Iteratively update each network using local and global information
- **Step 4:** Converge to a fused network representation

Advantages

- Handles incomplete data naturally
- Emphasizes concordant patterns across data types
- No feature-level alignment required



SNF Iterative Process

Initial State (t=0)

Network A
(Genomics)

Network B
(Proteomics)

Fusion Step

Iteration (t=1,2,...)

Information Exchange

A

B

Cross-network updates

Convergence

Fused Network

Integrated Similarity

Mathematical Formulation:

- P(t+1) = S × [P(t) + Σ(P_k(t))] / (m-1) × S^T
- S: Local similarity matrix (KNN structure)
- P: Full similarity matrix being updated
- Converges typically within 10-20 iterations

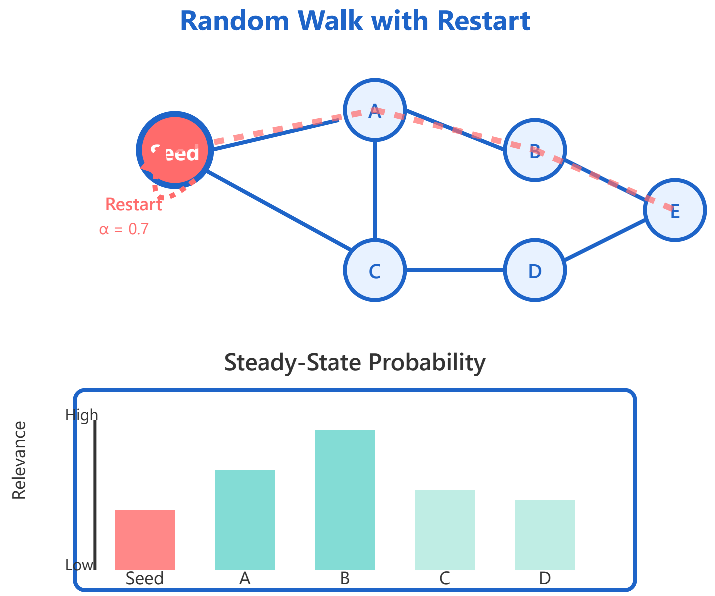
3. Random Walk with Restart (RWR)

Concept

Random walk algorithms simulate a particle moving randomly across network edges, with a probability of restarting at seed nodes. This diffusion process captures both direct and indirect relationships in the network.

Algorithm

- **Initialization:** Start at seed node(s) representing genes, patients, or features of interest
- **Walk:** Move to neighboring nodes with probability proportional to edge weights
- **Restart:** With probability α (typically 0.7), return to seed nodes
- **Convergence:** Iterate until steady-state probabilities stabilize



Applications in Multi-Omics

- Disease gene prioritization across omics layers
- Drug-target prediction in biological networks
- Pathway enrichment analysis

Key Advantages:

- Captures global network topology, not just local neighborhoods
- Naturally handles heterogeneous networks with multiple node types
- Robust to noise and missing edges in biological networks
- Steady-state probabilities provide quantitative ranking of nodes

4. Graph Neural Networks (GNNs)

Overview

Graph Neural Networks extend deep learning to graph-structured data, learning node representations by aggregating information from neighboring nodes. GNNs are particularly powerful for multi-omics integration as they can model complex relationships.

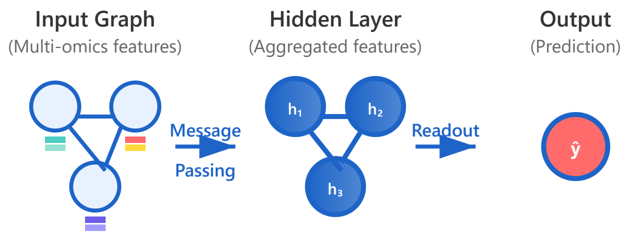
Key Architectures

- **Graph Convolutional Networks (GCN):** Aggregate neighbor features using spectral convolutions
- **Graph Attention Networks (GAT):** Learn importance weights for each neighbor
- **GraphSAGE:** Sample and aggregate from fixed-size neighborhoods
- **Message Passing Neural Networks:** Generalized framework for node updates

Multi-Omics Applications

- Patient outcome prediction using heterogeneous networks
- Drug response modeling with gene-gene interactions
- Biological pathway discovery

Graph Neural Network Architecture



Message Passing Mechanism

1. Aggregate — Collect neighbor features
2. Combine — Mix with own features
3. Update — Apply neural network

Mathematical Form:

$$h'_i = \sigma(W \cdot \Sigma_j(a_{ij} \cdot h_j) + b)$$

a_{ij}: attention weight, σ: activation function

GNN Advantages for Multi-Omics:

- End-to-end learning of representations from raw data
- Automatic feature extraction from graph structure
- Can handle heterogeneous node and edge types
- Scalable to large biological networks

5. Multiplex Networks

Concept

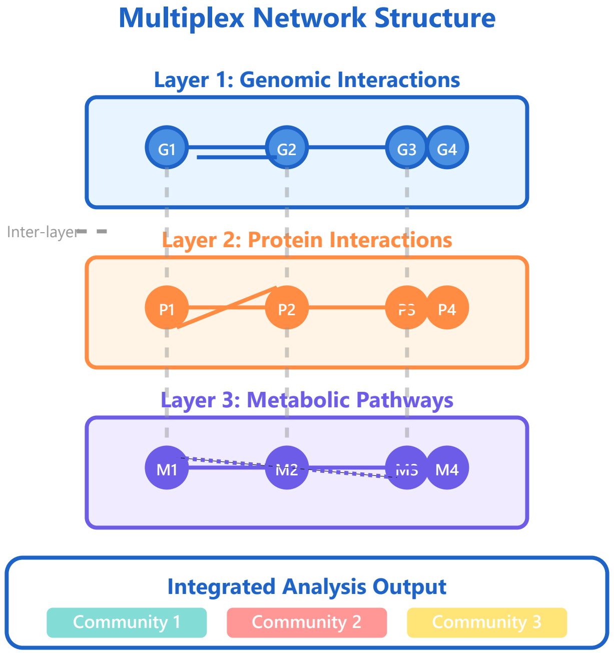
Multiplex networks represent multi-omics data as multiple layers where each layer corresponds to a different data type or interaction type. Nodes are shared across layers, but edges differ, capturing complementary relationships.

Layer Types in Multi-Omics

- **Genomic Layer:** Gene-gene interactions, co-expression
- **Protein Layer:** Protein-protein interactions
- **Metabolic Layer:** Metabolite-metabolite relationships
- **Regulatory Layer:** TF-gene, miRNA-gene interactions

Analysis Methods

- Multi-layer centrality measures
- Cross-layer community detection
- Multiplex random walks
- Layer coupling analysis



Multiplex Network Structure

Layer 1: Genomic Interactions

Layer 2: Protein Interactions

Layer 3: Metabolic Pathways

Integrated Analysis Output

Key Features of Multiplex Networks:

- Each layer captures a different biological relationship type
- Nodes represent the same biological entities across layers
- Inter-layer connections represent cross-omics relationships
- Enables discovery of layer-specific and cross-layer patterns
- Useful for identifying versatile biomarkers active across multiple omics layers