

PROTEIN STORM

A VISUALIZATION AND ANALYSIS TOOL FOR PROTEIN-PROTEIN INTERACTION NETWORKS

Pranav Srinivas

Biological Networks

- Biological Networks play an important role in cancer research and drug development
- Four types of Biological Network
 - Protein Signaling Network
 - Protein-Protein Interaction (PPI) network
 - Gene Regulatory network
 - Metabolic network
- Visualization of PPI network is explored in this project based on Processing 2.0
- PPI networks are undirected graphs
 - Similar to US road network or flight network

Network Properties

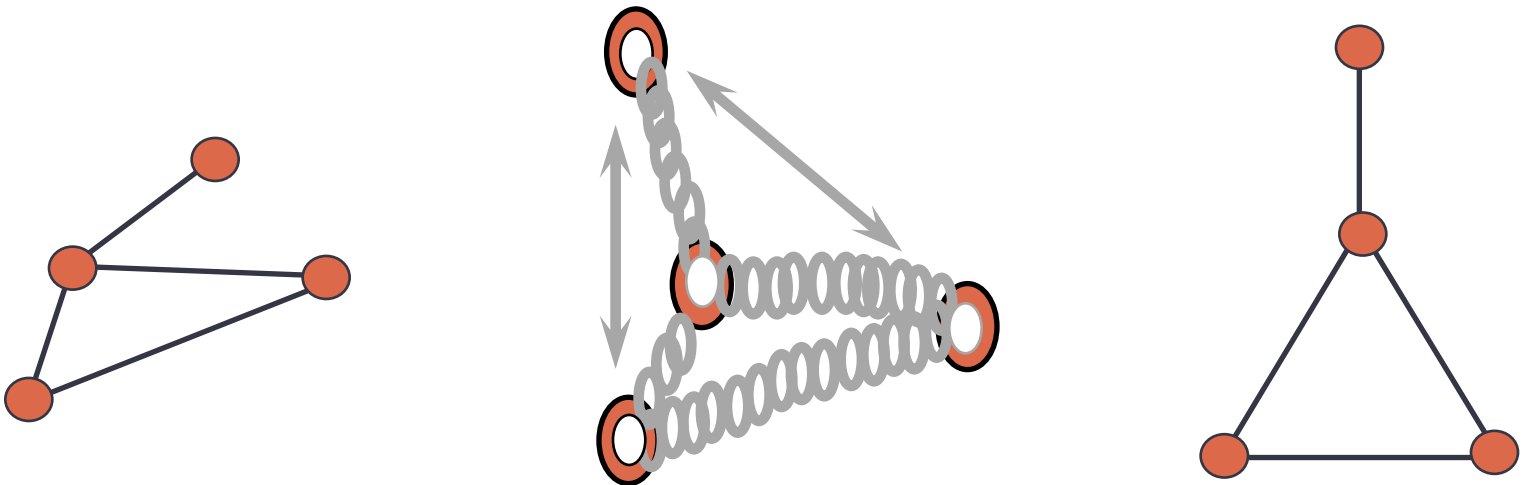
- Biological networks are small-world network
 - Most nodes can be reached via small number of hops
- Biological networks are scale-free network
 - Few nodes with very high degree called hub nodes
- Biological networks are organized as modules
 - Party hubs: within-module interaction
 - Date hubs: cross-module interaction
- Biological networks contain network motifs
 - Repeated topological structure
 - Example: Feed-forward, Positive and negative feed-back

PPI Network and Enrichment

- Complete PPI network is very large for elegant display
 - 19653 proteins, 39240 interactions
- Used as background network for enriching a user provided gene list
 - Extract sub-network from the background network that connects the user provided genes
 - Analogy: From flight network extract sub network that connects San Francisco, Boston, Dallas and Miami
 - Identify hub proteins and investigate shortest paths
- The background PPI network was obtained from Human Protein Reference Database (HPRD) (hprd.org)

Force Directed Visualization of Networks

- Proteins
 - Repel each other within a zone of influence
- Interactions
 - Spring that connect proteins
 - Repulsion force when shorter than the natural length
- Sum of the forces on each object is 0
- Reference: cs.brown.edu/~rt/gdhandbook/chapters/force-directed.pdf
- I implemented the most basic approach



Main Classes

- Protein

- class Protein {
 - private float velocityLimit; // Maximum velocity for force simulation
 - private float **repulsionStrength**; // -ve: Repulsion, +ve Attraction
 - private PVector **location**; // Position of this protein
 - private PVector **currentVelocity**; // Current protein velocity
 - private PVector **previousVelocity**; // Last protein velocity
 - private float radius; // Influence radius of protein
 - private float **dampingFactor**; // Force simulation damping factor
 - private float forceRampFactor; // Force simulation ramp factor
 - private ArrayList<Interaction> links; // All links of this protein (used in path queries)
- }

- Interaction

- class Interaction {
 - private Protein **sourceProtein**; // Source protein
 - private Protein **sinkProtein**; // Sink protein
 - private float **dampingFactor**; // Force damping factor
 - private float length; // length
 - private float **rigidity**; // Interaction rigidity (strength)
- }

Force Simulation and Visualization

- In the Processing draw() method
 - 1. Proteins repulse each other within influence zone
 - 2. Interactions apply their forces to source and sink proteins
 - 3. Update protein velocity and position
 - 4. Draw proteins and interactions at current position
- Zoom and Pan Control
 - MouseWheel events used for tracking zoom levels (in/out)
 - Processing provides a scale() method based on a zoom level
 - Arrow key events are processed for pan control (X or Y)
 - `translate(width/2 + panX, height/2 + panY);`
 - `scale(zoomLevel);`
 - `translate(-width/2 + panX, -height/2 + panY);`