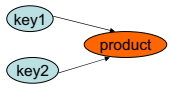
Biological networks

Construction and Analysis

Recap

- Gene regulatory networks
  - Transcription Factors: special proteins that function as “keys” to the “switches” that determine whether a protein is to be produced
  - Gene regulatory networks try to show this “key-product” relationship and understand the regulatory mechanisms that govern the cell.

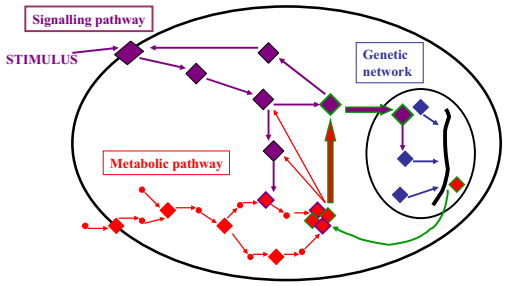


  - We went over a simple algorithm for detecting significant patterns in these networks

Other networks?

- Apart from regulation there are other events in a cell that require interaction of biological molecules
- Other types of molecular interactions that can be observed in a cell
  - enzyme – ligand
    - **enzyme:** a protein that catalyzes, or speeds up, a chemical reaction
    - **ligand:** extracellular substance that binds to receptors
    - metabolic pathways
  - protein – protein
    - cell signaling pathways
    - proteins interact physically and form large complexes for cell processes

Pathways are inter-linked

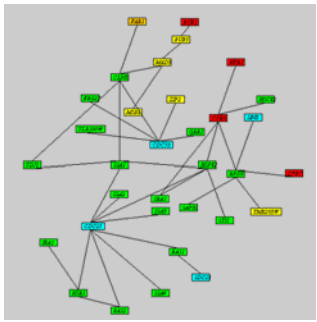


Interactions → Pathways → Network

- A collection of interactions defines a network
- Pathways are subsets of networks
  - All pathways are networks of interactions, however not all networks are pathways!
  - Difference in the level of annotation or understanding
- We can define a pathway as a biological network that relates to a **known** physiological process or complete function

The “interactome”

- The complete wiring of a proteome.
- Each vertex represents a protein.
- Each edge represents an “interaction” between two proteins.



An edge between two proteins if...

- The proteins interact physically and form large complexes
- The proteins are enzymes that catalyze two successive chemical reactions in a pathway
- One of the proteins regulates the expression of the other

Sources for interaction data

- Literature: research labs have been conducting small-scale experiments for many years!
- Interaction dabases:
  - MIPS (Munich Information center for Protein Sequences)
  - BIND (Biomolecular Network Interaction Database)
  - GRID (General Repository for Interaction Datasets)
  - DIP (Database of Interacting Proteins)
- Experiments:
  - Y2H (yeast two-hybrid method)
  - APMS (affinity purification coupled with mass spectrometry)

- These methods provide the ability to perform genome/proteome-scale experiments.
  - For yeast: 50,000 unique interactions involving 75% of known open reading frames (ORFs) of yeast genome
  - However, for *C. elegans* they provide relatively small coverage of the genome with ~5600 interactions.
- Problems with high-throughput experiments:
  - Low quality, false positives, false negatives
  - Fraction of biologically relevant interactions: 30%-50% (Deane *et al.* 2002)

Solution:

- User other indirect data sources to create a probabilistic protein network.
- Other sources include:
  - Genome data:
    - Existence of genes in multiple organisms
    - Locations of the genes
  - Bio-image data
  - Gene Ontology annotations
  - Microarray experiments
  - Sub-cellular localization data

Probabilistic network approach

- Each “interaction” link between two proteins has a posterior probability of existence, based on the quality of supporting evidence.

Bayesian Network approach

- Jansen *et al.* (2003) *Science*. Lee *et al.* (2004) *Science*.
- Combine individual probabilities of likelihood computed for each data source into a single likelihood (or probability)
- Naive Bayes:
  - Assume independence of data sources
  - Combine likelihoods using simple multiplication

Bayesian Approach

- A scalar score for a pair of genes is computed separately for each information source.
- Using gold positives (known interacting pairs) and gold negatives (known non-interacting pairs) interaction likelihoods for each information source is computed.
- The product of likelihoods can be used to combine multiple information sources
  - Assumption: A score from a source is independent from a score from another source.

Computing the likelihoods

- Partition the pair scores of an information source into bins and provide likelihoods for score-ranges
- E.g. Using the microarray information source and using Pearson correlation for scoring protein pairs you may get scores between -1 and 1. You want to know what is the likelihood of interaction for a protein pair that gets a Pearson correlation of 0.6.

Partitioning the scores

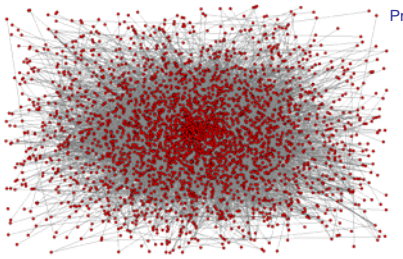
pearson corr.	likelihood
(0.8,1.0]	
(0.6,0.8]	
(0.4,0.6]	
(0.2,0.4]	
(0.0,0.2]	
(-0.2,0.0]	
(-0.4,-0.2]	
(-0.6,-0.4]	
(-0.8,-0.6]	
[-1.0,-0.8]	

Computing the likelihood

- $$L = \frac{P(\text{Interaction} \mid \text{Score}) / P(\text{Interaction})}{P(\sim\text{Interaction} \mid \text{Score}) / P(\sim\text{Interaction})}$$
- Example

Protein interaction networks

- Large scale (genome wide networks):



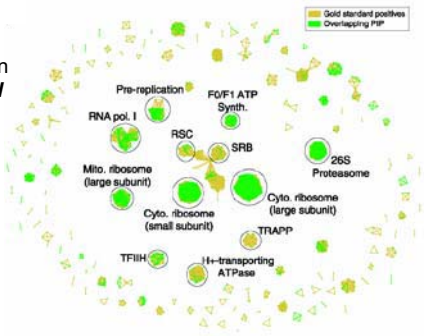
ProNet (Asthana et al.)
Yeast
3,112 nodes
12,594 edges

Analyzing Protein Networks

- Predict members of a partially known protein complex/pathway.
- Infer individual genes' functions on the basis of linked neighbors.
- Find strongly connected components, clusters to reveal unknown complexes.
- Find the best interaction path between a source and a target gene.

Simple analysis

The network can be ***thresholded*** to reveal clusters of interacting proteins



Complex/Pathway membership problem

- E.g.,
  - *C. elegans* cell death (apoptosis) pathway
  - Identified ~50 genes involved in the pathway.
  - Are there other genes involved in the pathway? Biologists would like to know:
    - Which genes (out of ~15K genes) should be tested in the RNAi screens next?

Complex/pathway membership problem

- Given a a set of proteins identified as the core complex (query), rank the remaining proteins in the network according to the probability that they "connect" to the core complex.
- This problem is very similar to the "network reliability" problem in communication networks.

Network reliability

- Two terminal network reliability problem:
  - Given a graph of connections between terminals:
    - Each connection weighted by the probability that the corresponding wire is functioning at a given time
  - What is the probability that some path of functioning wires connects two terminals at a given time?

Exact solution: NP-hard
Several approximation methods exist

Monte Carlo simulation

- Monte Carlo simulation (ProNet: Asthana *et al.* 2004)
  - Create a sample of **N** binary networks from the probabilistic network (according to a Bernoulli trial on each edge based on its probability).
- Use breadth-first search to determine the existence of a path between the nodes (i.e., the two terminals).
- The fraction of sampled networks in which there exists a path between the two nodes is an approximation to the exact network reliability.

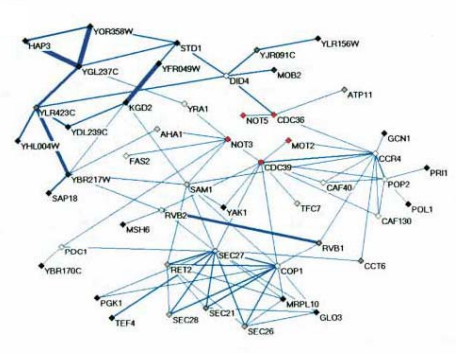
Parameters

- Number of binary networks (samples) to be sampled from the probabilistic network
  - 1000, 5000, 10000 ?
- The depth of the breadth-first search: complexity increases as you search for the existence of a path to a distant node.
  - 4, 10, 20 ?

ProNet

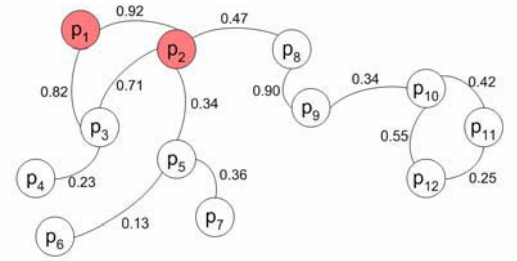
- Generate 10,000 binary networks from a probabilistic network (according to a Bernoulli trial on each edge based on its probability)
- Use breadth-first search to determine the existence of a path between two nodes
  - Limit the maximum depth to 4 to reduce computation
- For each protein *i* in the network, count the fraction C_i of sampled networks in which there exists a path between *i* and the core complex.
- Report proteins ranked by C_i

ProNet: example



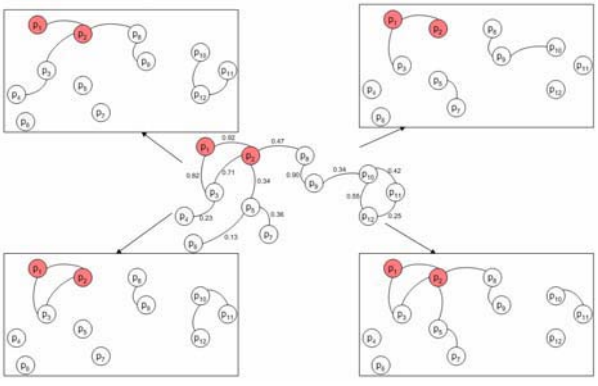
Example

- Complex nodes: p_1 and p_2



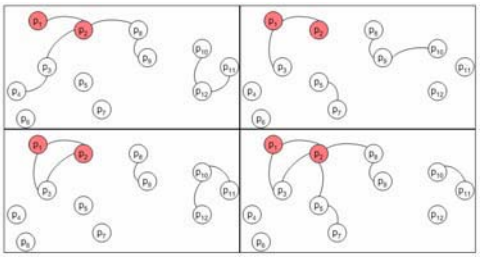
Example

- Sample size: 4, maximum search depth: 3



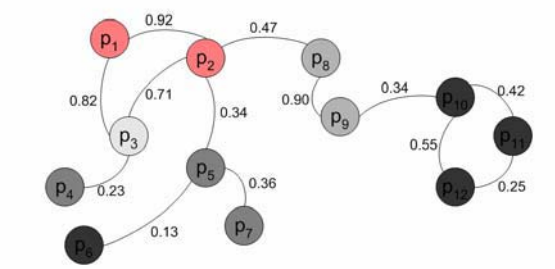
Example

- Sample size: 4, maximum search depth: 3

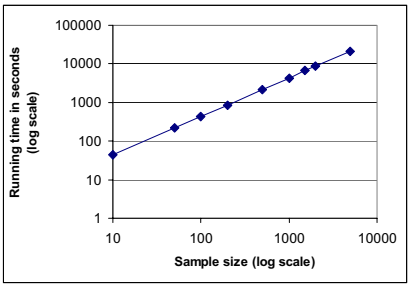


$C_{p3} = 4/4 = 1.0$
$C_{p4} = 1/4 = 0.25$
$C_{p5} = 1/4 = 0.25$
$C_{p6} = 0/4 = 0.0$
$C_{p7} = 1/4 = 0.25$
$C_{p8} = 2/4 = 0.5$
$C_{p9} = 2/4 = 0.5$
$C_{p10} = 0/4 = 0.0$
$C_{p11} = 0/4 = 0.0$
$C_{p12} = 0/4 = 0.0$

Results



Running time vs. sample size



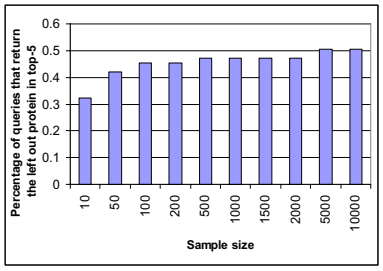
What about accuracy of the technique? Is it able to give a good ranking for the nodes of the network, based on their closeness to the core?

Leave-one-out benchmark

- Use known complexes to evaluate the accuracy of the method
- Leave one member (in turn) from each complex/pathway.
- Use the rest of the complex/pathway as the starting, i.e., query, set.
- Examine the rank of the left-out protein.
  - What do we expect from a good technique?

Accuracy vs. sample size

- How does the sample size effect returned results?



Monte Carlo simulation

- Disadvantages:
  - What is the best choice for the number of samples?
  - What should be the maximum depth for breadth-first search? (Need a cutoff to decrease running time)
  - Scalability issues: May need a lot of computation time for large networks

Random Walks

- Random Walks on graphs
  - Google's page rank

Google's PageRank

- Assumption: A **link** from page A to page B is a **recommendation** of page B by the author of A (we say B is *successor* of A)

➔Quality of a page is related to its in-degree

- Recursion: Quality of a page is related to
  - its in-degree, and to
  - the *quality* of pages linking to it

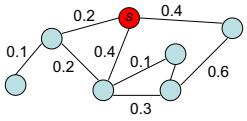
➔**PageRank** [BP '98]

Definition of PageRank

- Consider the following infinite random walk (surf):
  - Initially the surfer is at a random page
  - At each step, the surfer proceeds
    - to a randomly chosen web page with probability d
    - to a randomly chosen successor of the current page with probability 1-d
- **The PageRank of a page p is *the fraction of steps the surfer spends at p in the limit.***

Random walks **with restarts** on interaction networks

- Consider a random walker that starts on a source node, s. At every time tick, the walker chooses randomly among the available edges (based on edge weights), or goes back to node s with probability $\boldsymbol{c}$.



Random walks on graphs

- The probability $p_s(v)^{(t)}$, is defined as the probability of finding the random walker at node v at time t.
- The steady state probability $p_s(v)$ gives a measure of affinity to node s, and can be computed efficiently using iterative matrix operations.

Computing the steady state **p** vector

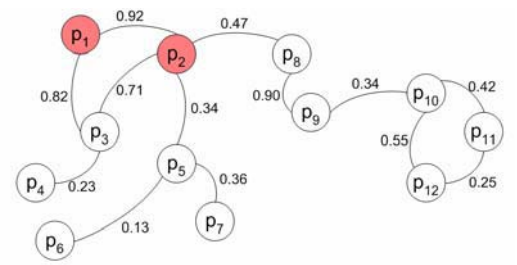
- Let **s** be the vector that represents the source nodes (i.e., $\mathbf{s}_i = 1/n$ if node i is one the n source nodes, and 0 otherwise).
- Compute the following until **p** converges:

$$\mathbf{p} = (1\text{-}c)\mathbf{A}\mathbf{p} + c\mathbf{s}$$

where **A is the column normalized adjacency matrix** and c is the restart probability.

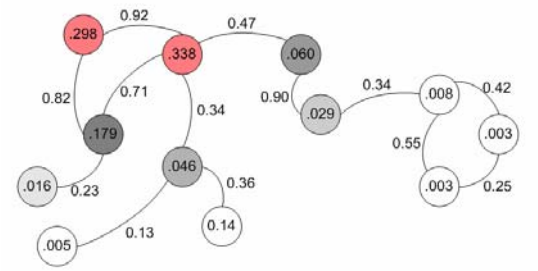
Same example

- Start nodes: p_1 and p_2



Random walk results

- Restart probability, c = 0.3



Experiments

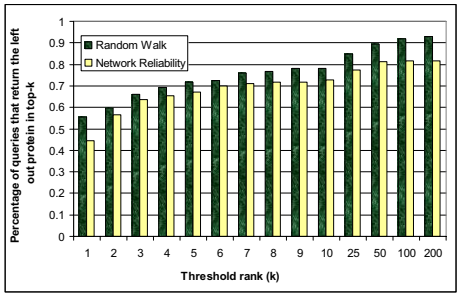
- Conducted complex/pathway membership queries on a probabilistic Yeast network:
  - ConfidentNet (Lee *et al.*, 4,681 nodes, 34,000 edges)
- Assembled a test set of 27 MIPS complexes and 10 KEGG pathways.

Leave-one-out benchmark

- Leave one member (in turn) from each complex/pathway.
- Use the rest of the complex/pathway as the starting, i.e., query, set.
- Examine the rank of the left-out protein.

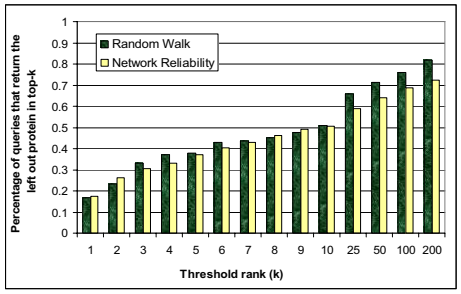
Leave-one-out on ConfidentNet

- MIPS complex queries



Leave-one-out on ConfidentNet

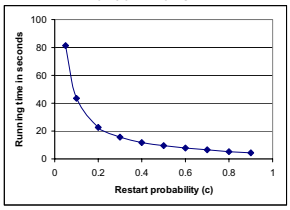
- KEGG pathway queries



Running time

- Total time to complete 121 MIPS complex queries

Random Walks



Network Reliability by Monte Carlo Sampling