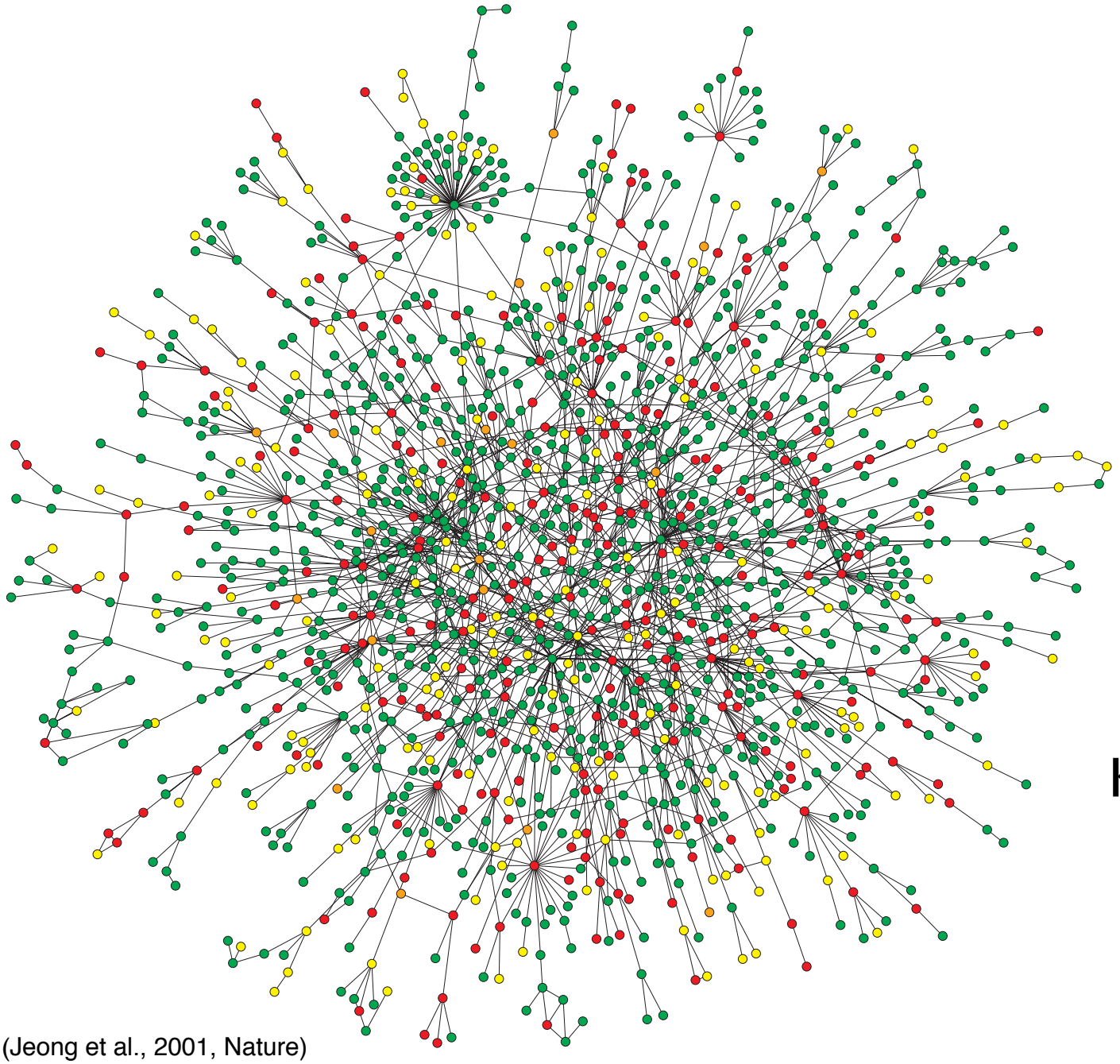
Comparative Network Analysis

BMI/CS 776
www.biostat.wisc.edu/bmi776/
Spring 2012
Colin Dewey
cdewey@biostat.wisc.edu

Protein-protein Interaction Networks



(Jeong et al., 2001, Nature)

- Yeast protein interactions from yeast two-hybrid experiments
- Largest cluster in network contains 78% of proteins

Knock-out phenotype

- lethal
- non-lethal
- slow growth
- unknown

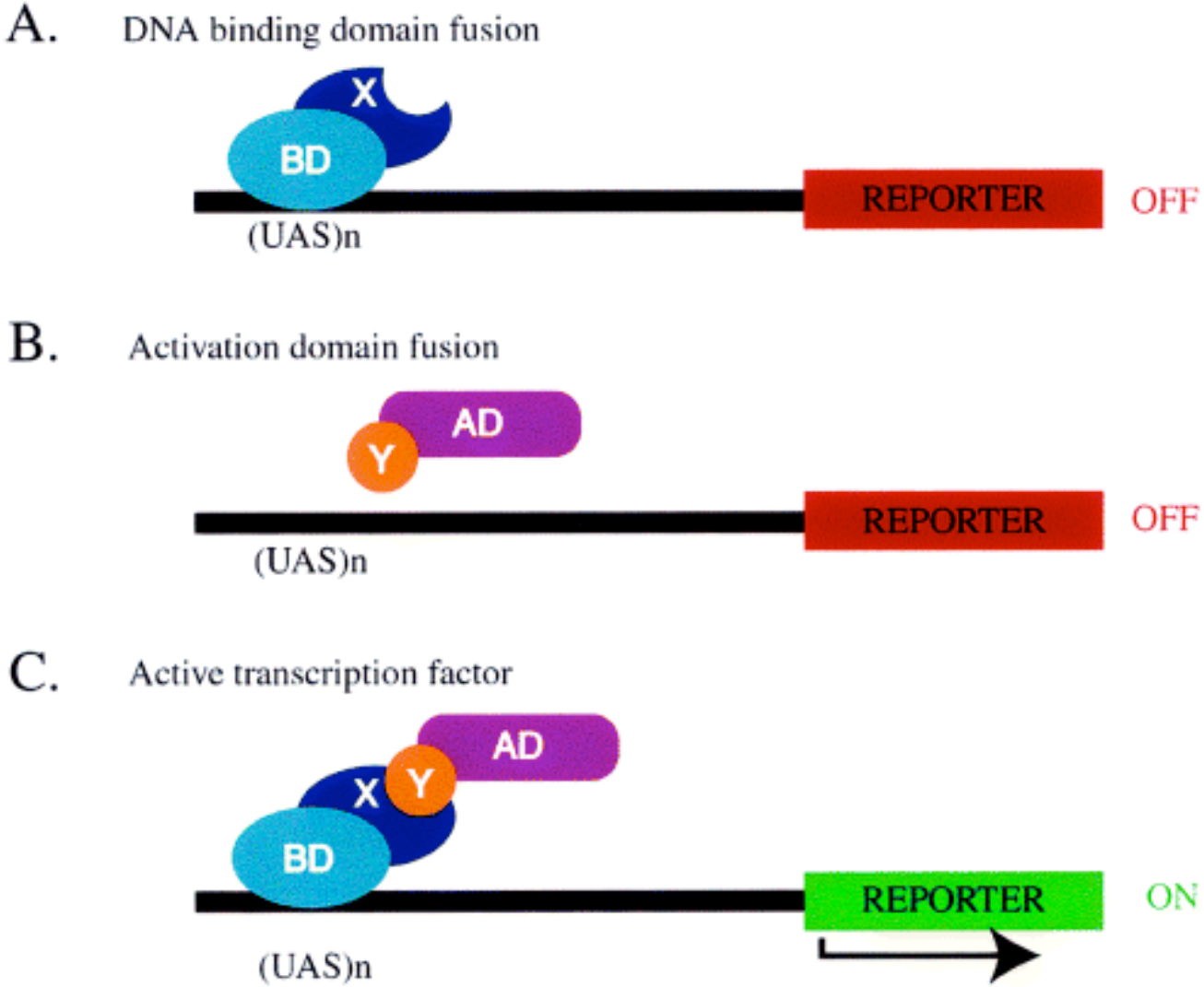
Overview

- Experimental techniques for determining networks
- Properties of biological networks
- Comparative network tasks

Experimental techniques

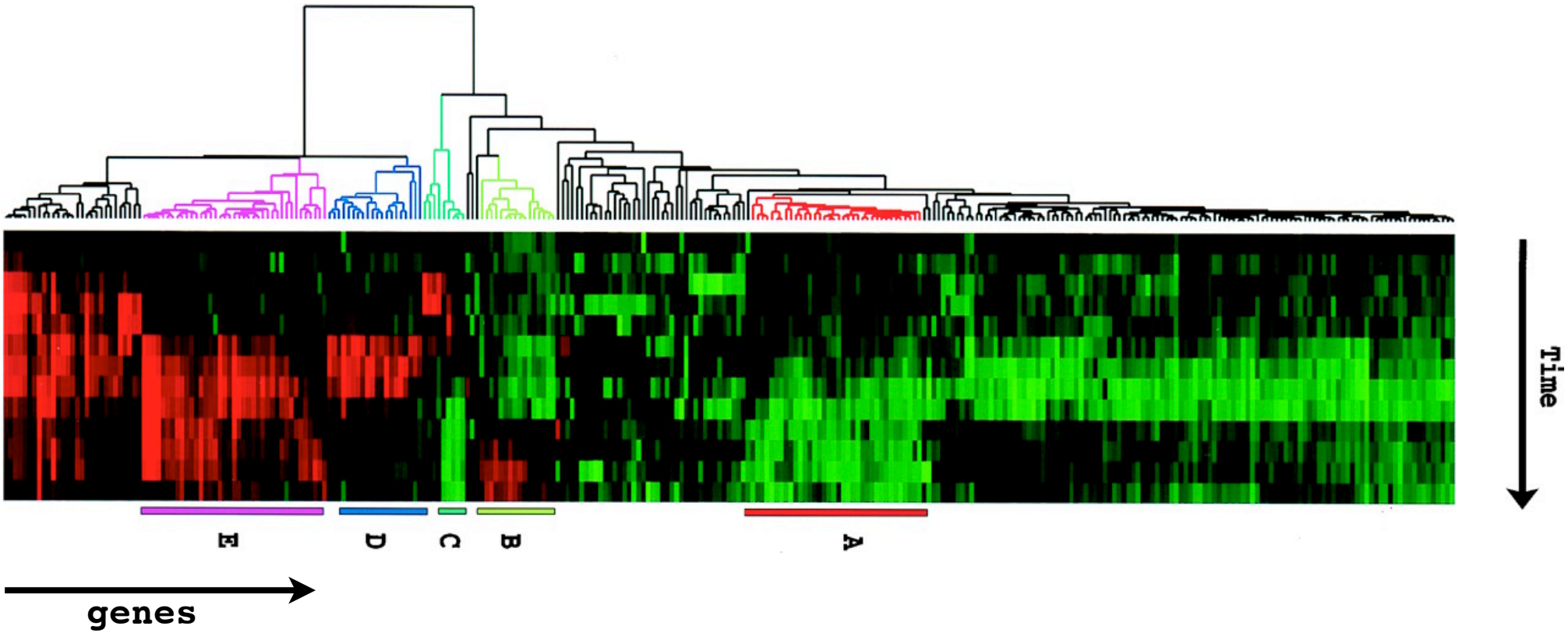
- Yeast two-hybrid system
  - Protein-protein interactions
- Microarrays
  - Expression patterns of mRNAs
  - Similar patterns imply involvement in same regulatory or signaling network
- Knock-out studies
  - Identify genes required for synthesis of certain molecules

Yeast two-hybrid system



(Stephens & Banting, 2000, Traffic)

Microarrays

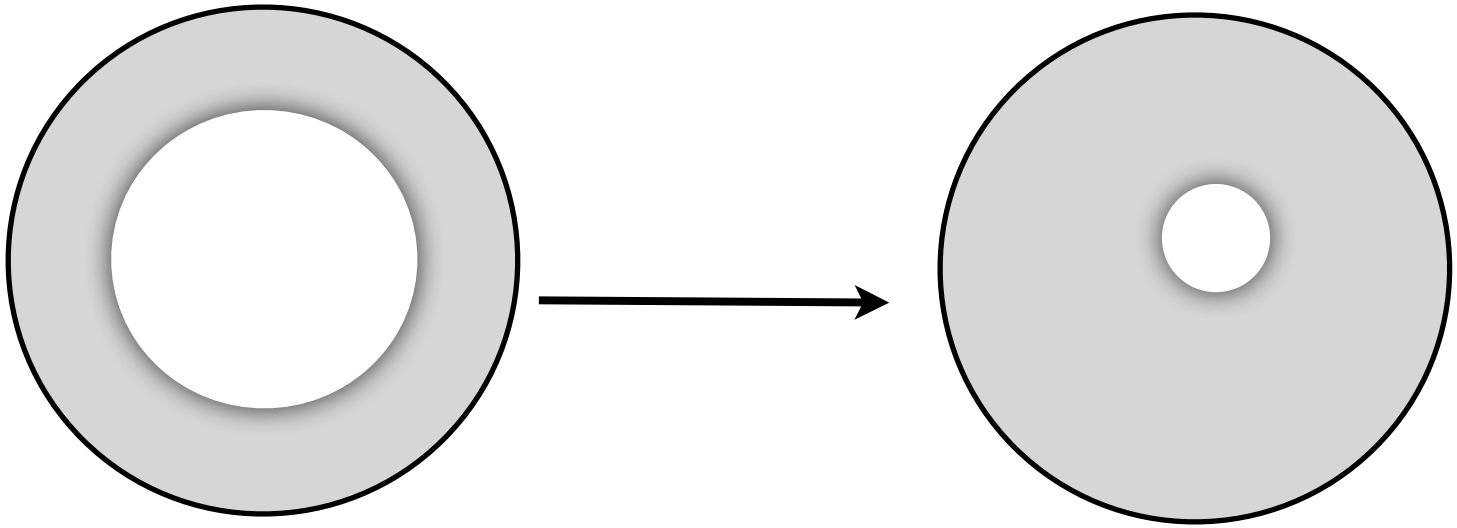


(Eisen et al., 1998, PNAS)

Knock-out studies

Yeast with one gene deleted

Growth?



Rich media

His^- media

Topological properties of networks

- Degree: number of edges in/out of a node
  - Average degree
  - Degree distribution: *P(k)*, fraction of nodes with degree *k*
- Clustering coefficient: measure of grouping in graph
- Path length: shortest path between two nodes
  - Average path length

Clustering coefficient

$$C_i = \frac{2n_i}{k_i(k_i - 1)}$$

C_i: clustering coefficient of node i

n_i: number of edges between neighbors of node i (number of triangles involving node i)

k_i: degree of node i

Interesting to look at C(k): average clustering coefficient of nodes with degree k

Erdös & Rényi random graphs

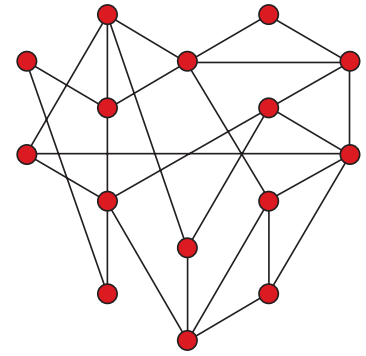
- Erdös & Rényi (1960): *On the evolution of random graphs*
- Construction
  - Start with N vertices, zero edges
  - Add each possible edge with probability p
- Expected number of edges: pN(N - 1)/2
- Expected degree: p(N-1)

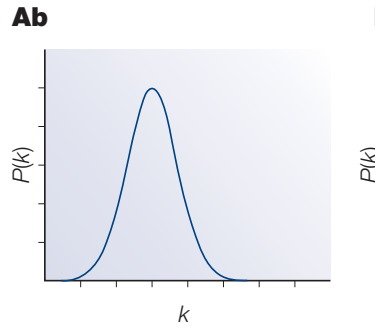
Properties of ER graphs

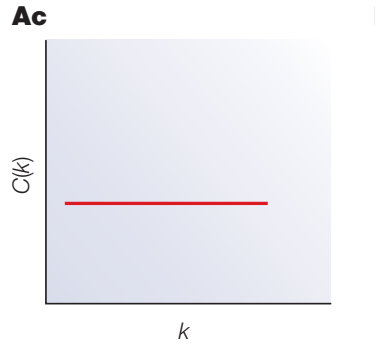
- Degree of nodes ~ Poisson distribution
- Most nodes have degree close to average degree
- Average path length ~ log n
- Clustering coefficient does not depend on degree k

$$P(k) \approx \frac{e^{-\lambda}\lambda^{k}}{k!}$$

$$\lambda = p(N-1)$$







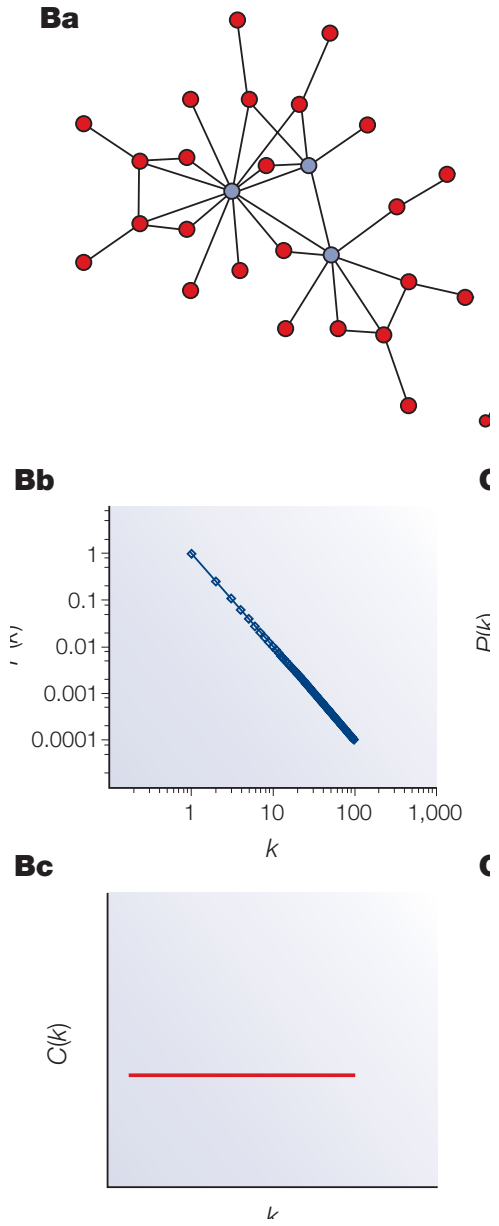
(Barabási & Oltvai, 2004)

Scale-free networks

- Barabási & Albert (1999): *Emergence of scaling in random networks*
- Random construction:
  - Start with a few connected nodes
  - Add nodes one at a time
  - Add *m* edges between new node and previous nodes
    - For each edge, probability of being incident to node *i* is $\frac{k_i}{\sum_j k_j}$ ← degree of node j

Properties of scale-free networks

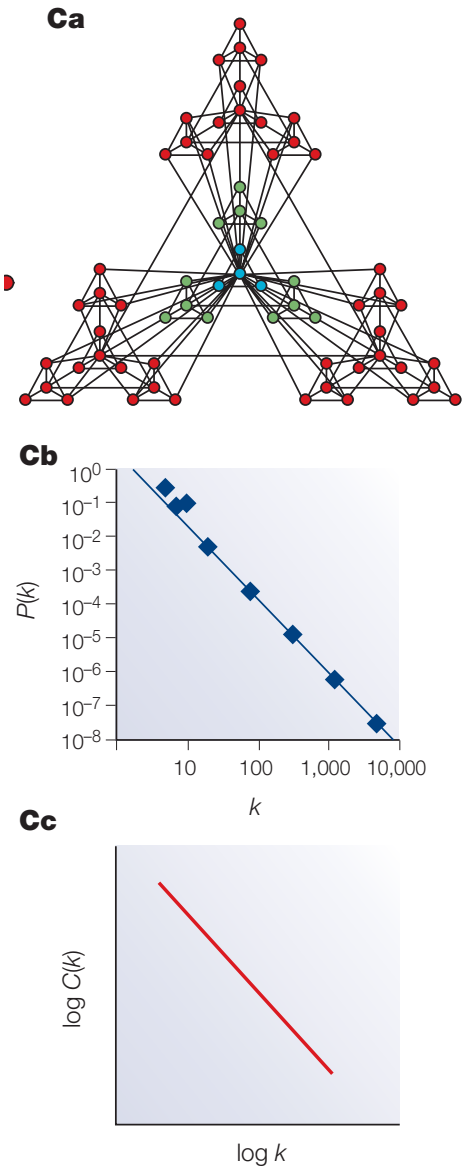
- Degrees:
  - $P(k) \sim k^{-\gamma}$ (power law distribution)
  - Most nodes have very small degree
  - A few nodes (hubs) with large degree
- Average path length ~ log log n
- Flat C(k)
- Properties depend on value of γ



(Barabási & Oltvai, 2004)

Hierarchical network

- Recursive generation
- Scale-free
- Clustering coefficient dependent on degree: $C(k) \sim k^{-1}$



(Barabási & Oltvai, 2004)

Classifying networks

- Metabolic networks
  - scale-free
- PPI networks
  - scale-free
- Regulatory networks
  - mixed
    - out-degree of transcription factors is scale-free
    - in-degree of regulated genes is exponential

Paths in biological networks

- Path length between two vertices is often very small
- random graph gives expected path length as log N
- scale-free graph has log log N expected path length
- However, hubs not often connected to each other: *disassortative*

Small-world networks

- Small-world networks are graphs with small average path length
- ER graphs are small-world: log n average path length
- Scale-free graphs often very small: log log n (for some values of γ)
- However, biological networks are both small-world and *disassortative*: hubs are not often connected to each other

Evolving networks

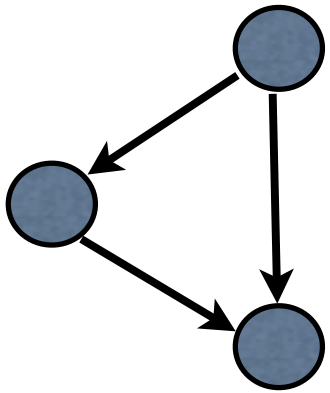
- Growth
  - Early nodes have more links
- Preferential attachment
  - As new nodes added, more likely to be connected to already highly-connected nodes
  - Leads to scale-free networks
- Gene duplication
  - Major force in protein network evolution
  - Highly-connected nodes more likely to have neighbors duplicate and add more edges

Network problems

- Network inference
  - Given raw experimental data
  - Infer network structure
- Motif finding
  - Identify common subgraph topologies
- Module detection
  - Identify subgraphs that perform same function
- Conserved modules
  - Identify modules that are shared in networks of multiple species

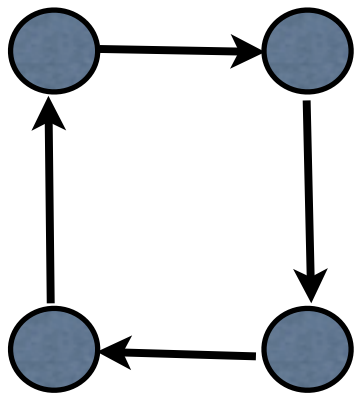
Network motifs

- Problem: Find subgraph topologies that are statistically more frequent than expected
- Brute force approach
  - Count all topologies of subgraphs of size m
  - Randomize graph (retain degree distribution) and count again
  - Output topologies that are over/under represented



Feed-forward loop: over-represented in regulatory networks

not very common



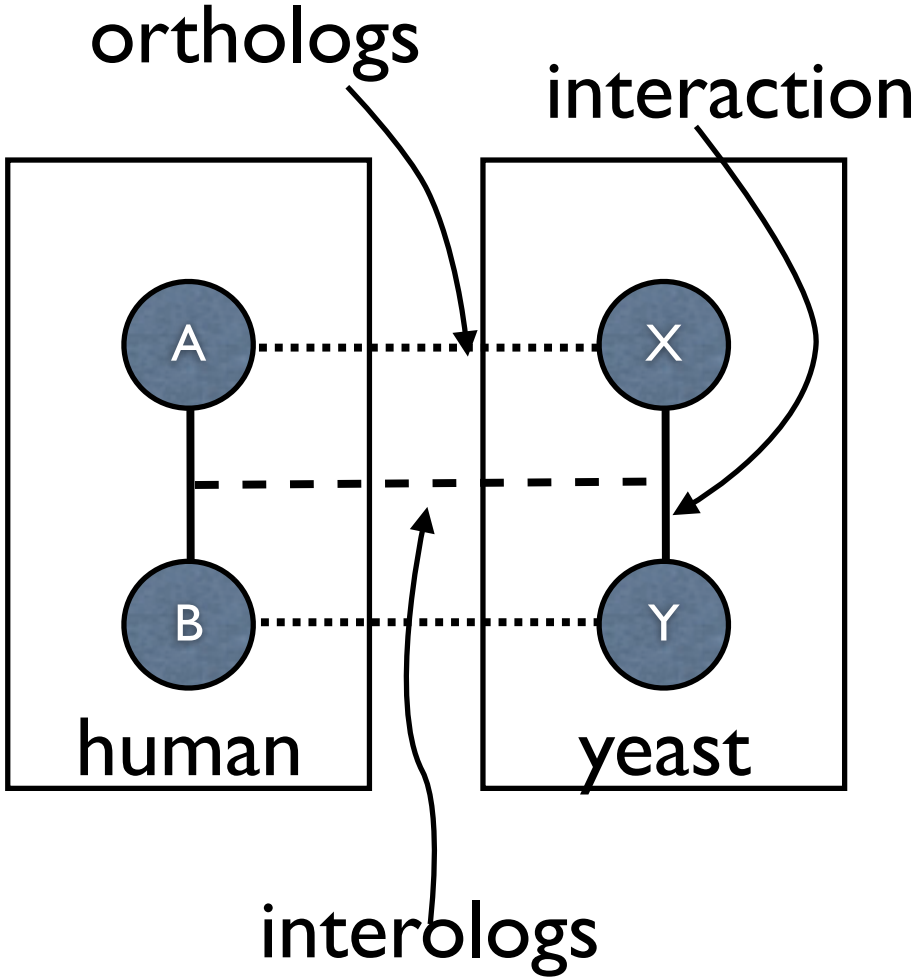
Network modules

- Modules: dense (highly-connected) subgraphs (e.g., large cliques or partially incomplete cliques)
- Problem: Identify the component modules of a network
- Difficulty: definition of module is not precise
  - Hierarchical networks have modules at multiple scales
  - At what scale to define modules?

Conserved modules

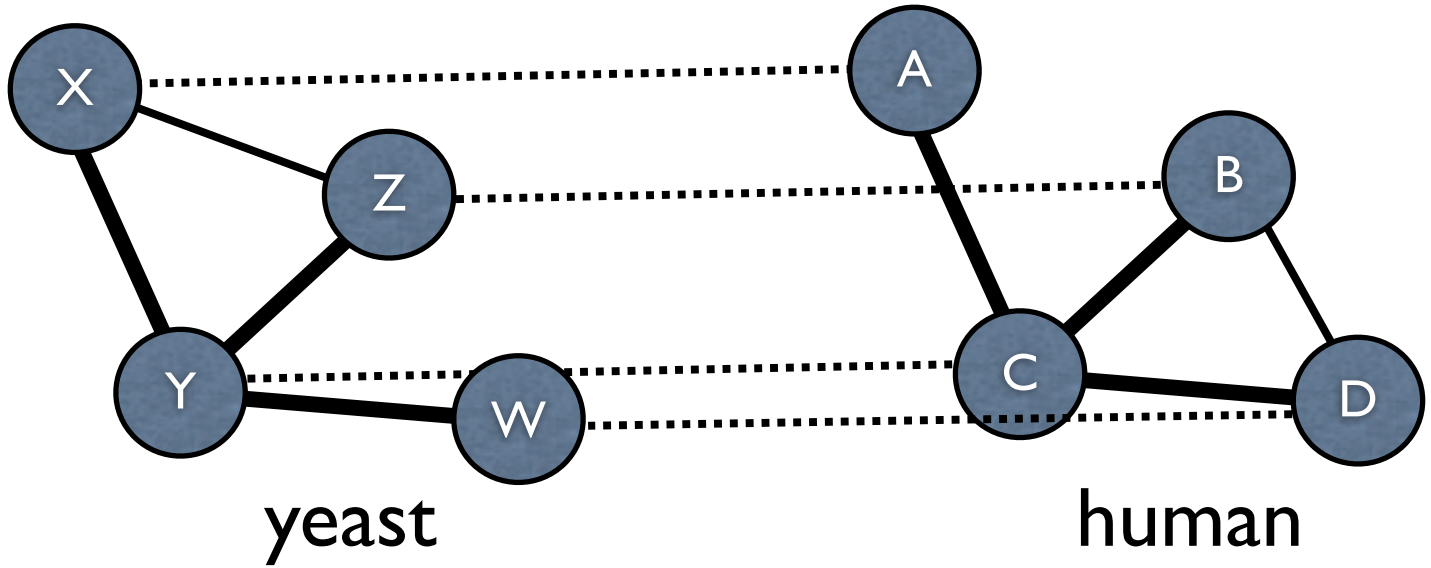
- Identify *modules* in multiple species that have "conserved" topology
- Typical approach:
  - Use sequence alignment to identify homologous proteins and establish correspondence between networks
  - Using correspondence, output subsets of nodes with similar topology

Conserved interactions



- Network comparison between species also requires sequence comparison
- Protein sets compared to identify orthologs
- Common technique: highest scoring BLAST hits used for establishing correspondences

Conserved modules



- Conserved module: orthologous subnetwork with significantly similar edge presence/absence

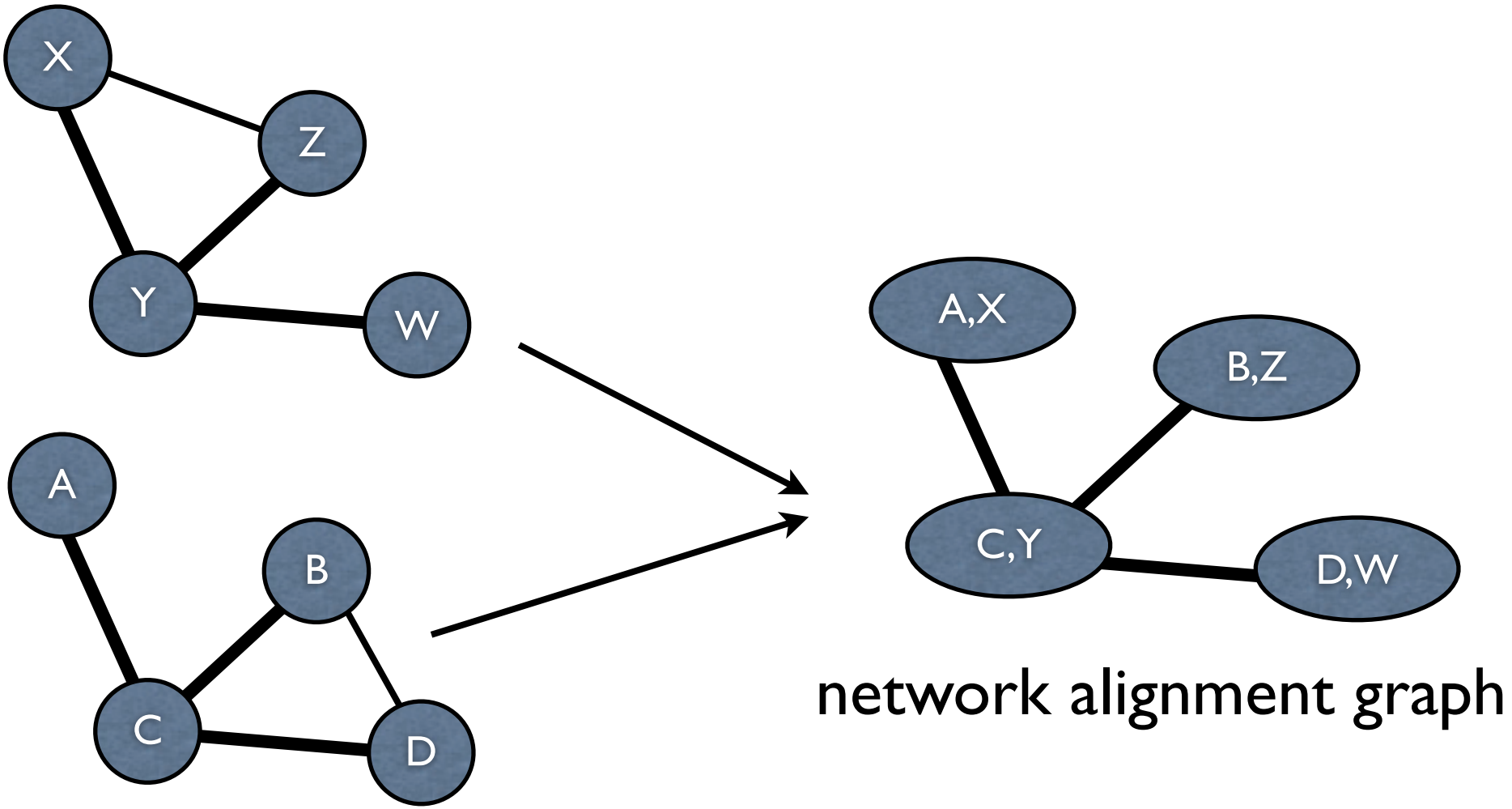
Comparative network analysis

- Compare networks from different...
  - interaction detection methods
    - yeast 2-hybrid, mass spectrometry, etc.
  - conditions
    - heat, media, other stresses
  - time points
    - development, cell cycle
  - species

Comparative tasks

- Integration
  - Combine networks derived from different methods (e.g. experimental data types)
- Alignment
  - Identify nodes, edges, modules common to two networks (e.g., from different species)
- Database query
  - Identify subnetworks similar to query in database of networks

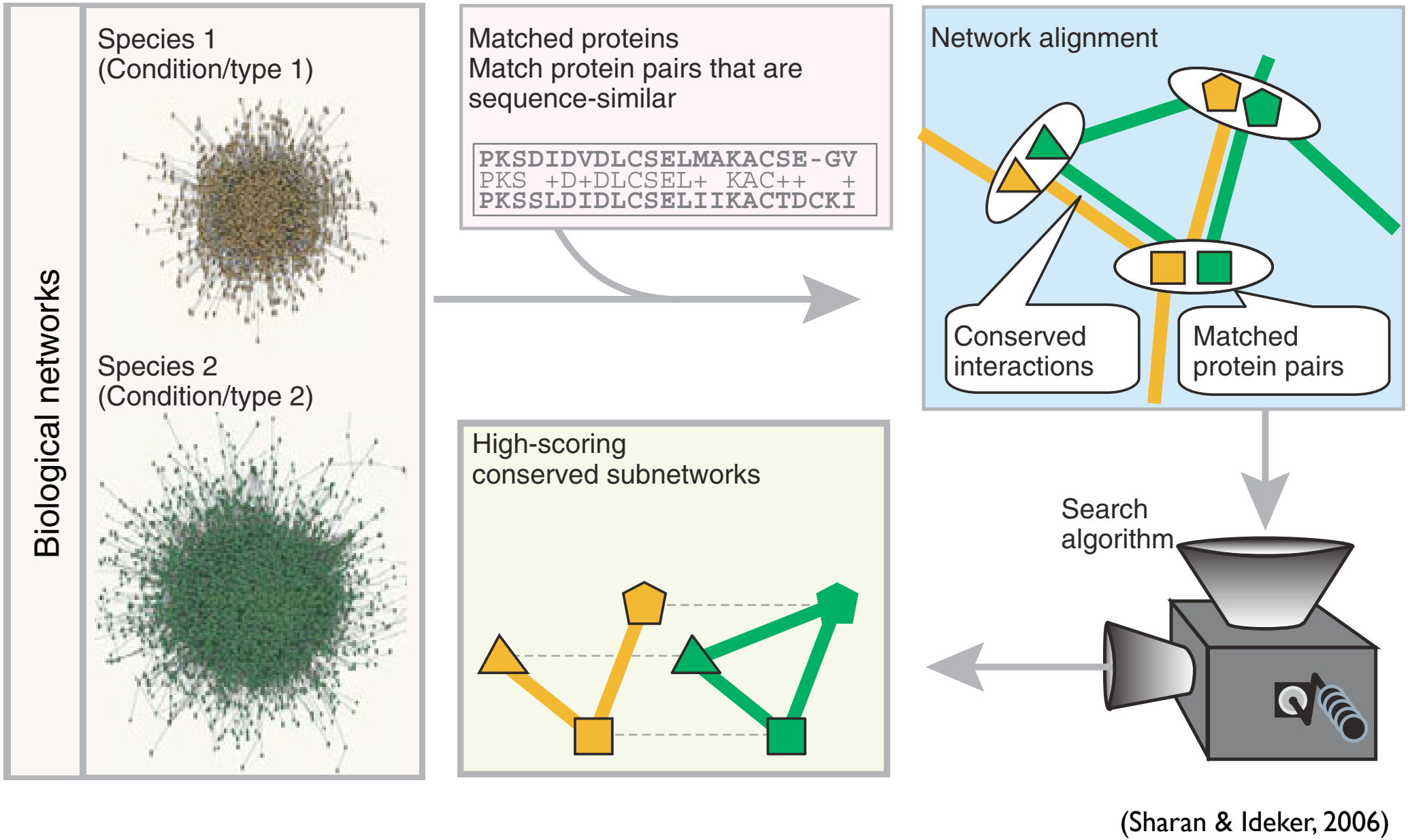
Network alignment graph



network alignment graph

- Analogous to pairwise sequence alignment

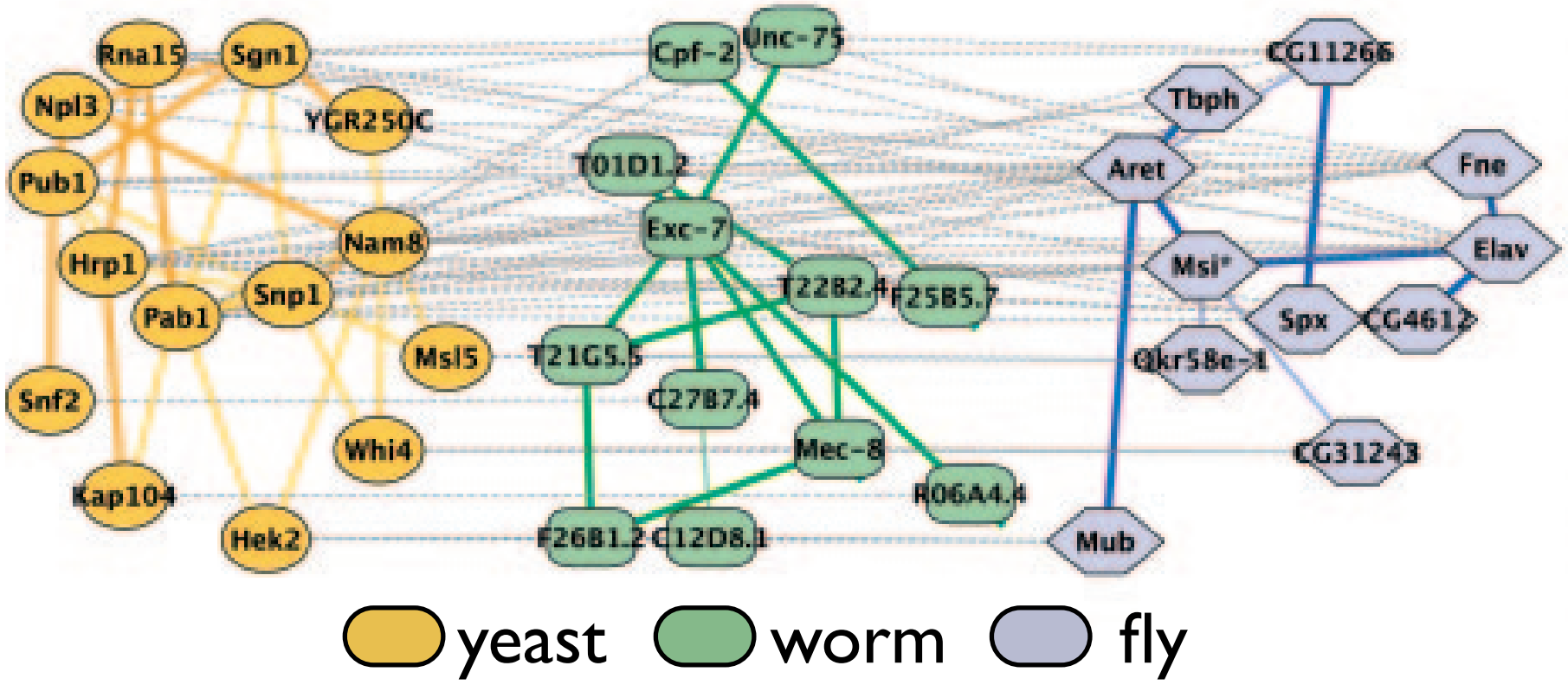
Conserved module detection



(Sharan & Ideker, 2006)

Real module example

Module for RNA metabolism (Sharan et al., 2005)



- Note: a protein may have more than one ortholog in another network

Basic alignment strategy

- Define scoring function on subnetworks
  - high score ⇒ conserved module
- Use BLAST to infer orthologous proteins
- Identify "seeds" around each protein: small conserved subnetworks centered around the protein
- Grow seeds by adding proteins that increase alignment score

Scoring functions via Subnetwork modeling

- We wish to calculate the likelihood of a certain subnetwork U under different models
  - Subnetwork model (M_s)
    - Connectivity of U given by target graph H, each edge in H appearing in U with probability β (large)
  - Null model (M_n)
    - Each edge appears with probability according to random graph distribution (but with degree distribution fixed)

(Sharan et al., 2005)

Noisy observations

- Typically weight edges in graph according to confidence in interaction (expressed as a probability)
- Let
  - T_{uv}: event that proteins u, v interact
  - F_{uv}: event that proteins u, v do not interact
  - O_{uv}: observations of possible interactions between proteins u and v

Subnetwork model probability

- Assume (for explanatory purposes) that subnetwork model is a clique:

$$
\begin{aligned}
Pr(O_U|M_s) &= \prod_{(u,v)\in U\times U} Pr(O_{uv}|M_s) \\
&= \prod_{(u,v)\in U\times U} [Pr(O_{uv}|T_{uv},M_s)Pr(T_{uv}|M_s) + Pr(O_{uv}|F_{uv},M_s)Pr(F_{uv}|M_s)] \\
&= \prod_{(u,v)\in U\times U} [\beta Pr(O_{uv}|T_{uv}) + (1-\beta)Pr(O_{uv}|F_{uv})]
\end{aligned}
$$

Null model probability

- Given values for p_{uv}: probability of edge (u,v) in random graph with same degrees

$$Pr(O_U|M_n) = \prod_{(u,v)\in U\times U} [p_{uv}Pr(O_{uv}|T_{uv}) + (1-p_{uv})Pr(O_{uv}|F_{uv})]$$

- How to get random graph if we don't know true degree distribution? Estimate them:

$$d_i = \sum_j Pr(T_{ij}|O_{ij})$$

$$Pr(T_{uv}|O_{uv}) = \frac{Pr(O_{uv}|T_{uv})Pr(T_{uv})}{Pr(O_{uv}|T_{uv})Pr(T_{uv}) + Pr(O_{uv}|F_{uv})(1-Pr(T_{uv}))}$$

Likelihood ratio

- Score subnetwork with (log) ratio of likelihoods under the two models

$$
\begin{aligned}
L(U) &= \log \frac{Pr(O_U|M_s)}{Pr(O_U|M_n)} \\
&= \sum_{(u,v)\in U\times U} \log \frac{\beta Pr(O_{uv}|T_{uv}) + (1-\beta)Pr(O_{uv}|F_{uv})}{p_{uv}Pr(O_{uv}|T_{uv}) + (1-p_{uv})Pr(O_{uv}|F_{uv})}
\end{aligned}
$$

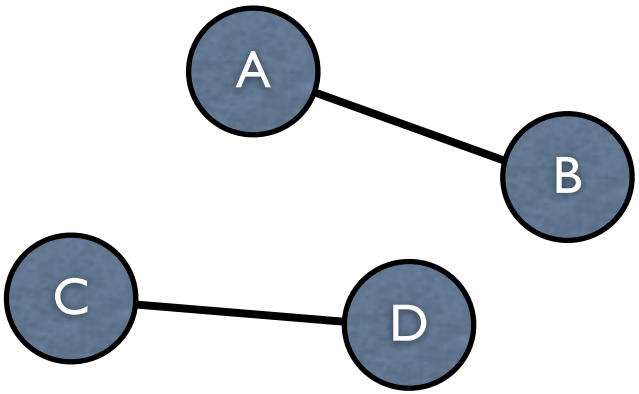
- Note the decomposition into sum of scores for each edge

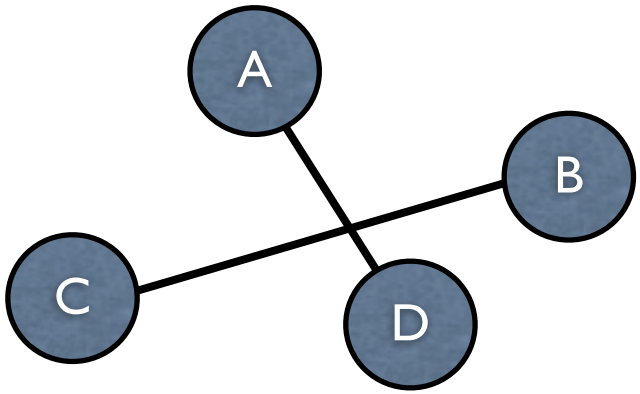
Seed construction

- Finding "heavy induced subgraphs" is NP-hard (Sharan et al., 2004)
- Heuristic:
  - Find high-scoring subgraph "seeds"
  - Grow seeds greedily
- Seed techniques: for each node v:
  - Find heavy subgraph of size 4 including v
  - Find highest-scoring length 4 path with v

Randomizing graphs

- For statistical tests, need to keep degree distribution the same
- Shuffle step:
  - Choose two edges (a,b), (c,d) in the current graph
  - Remove those edges
  - Add edges (a,d), (c,b)

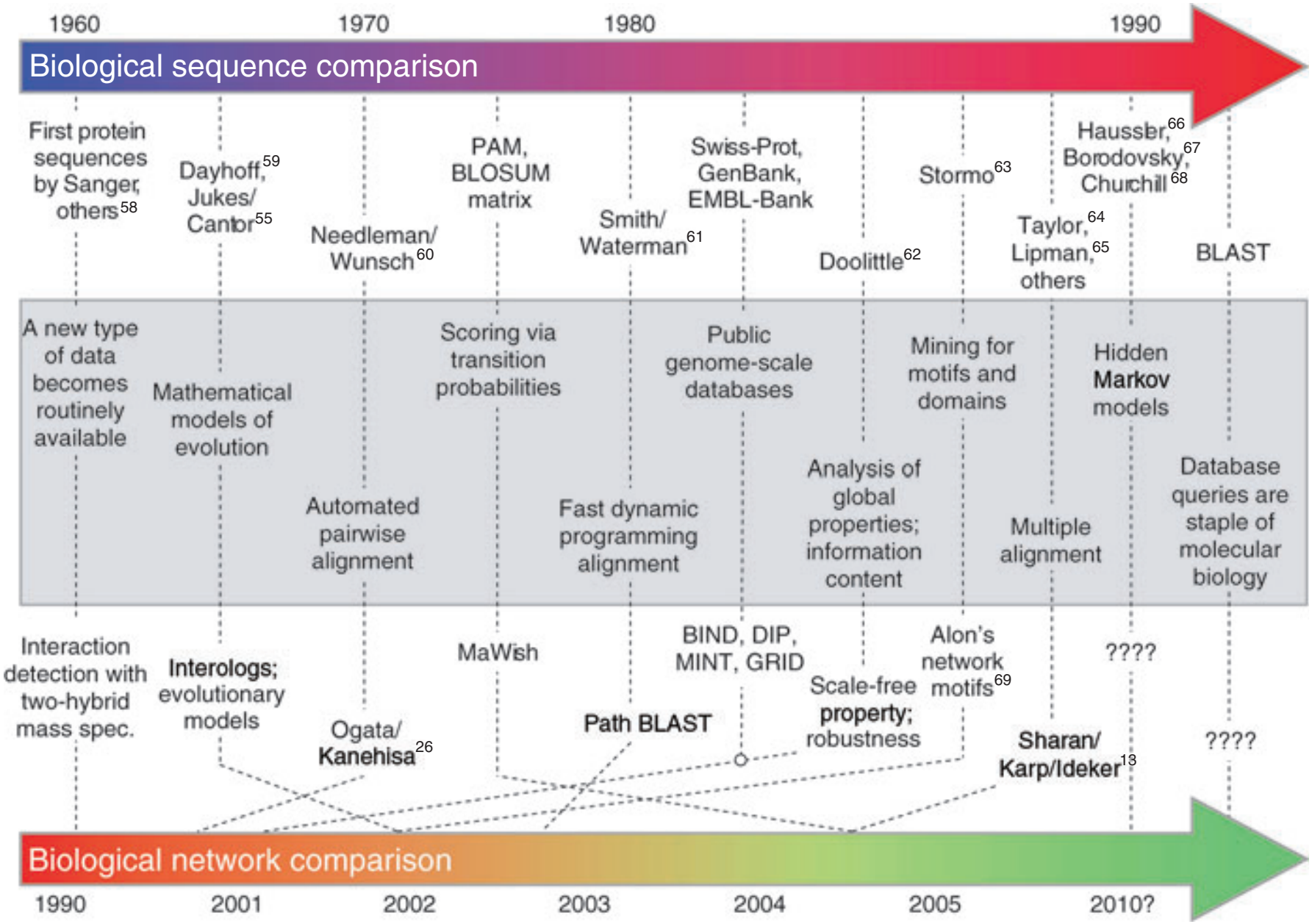




Predictions from alignments

- Conserved modules of proteins enriched for certain functions often indicate shared function of other proteins
  - Use to predict function of unannotated proteins
  - Sharan et al., 2005: annotated 4,645 proteins with estimated accuracy of 58-63%
- Predict missing interactions
  - Sharan et al., 2005: 2,609 predicted interactions in fly, 40 –52% accurate

Parallels to sequence analysis



(Sharan & Ideker, 2006)