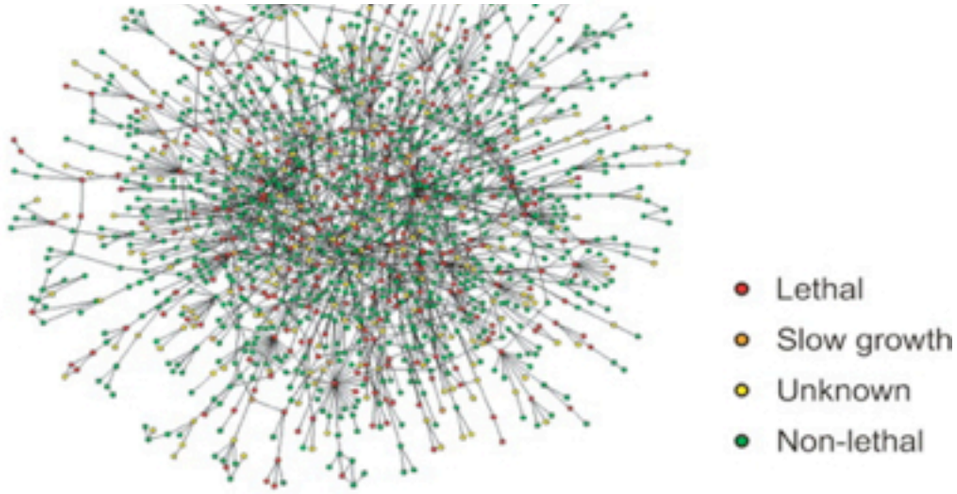
Biological Networks

02-251
Carl Kingsford

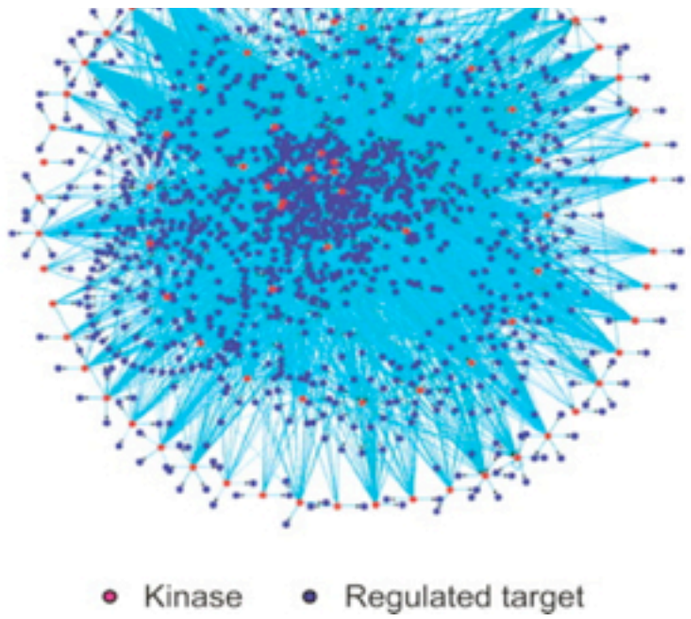
Types of Biological Networks

Yeast transcription network

Yeast Protein-Protein interaction network

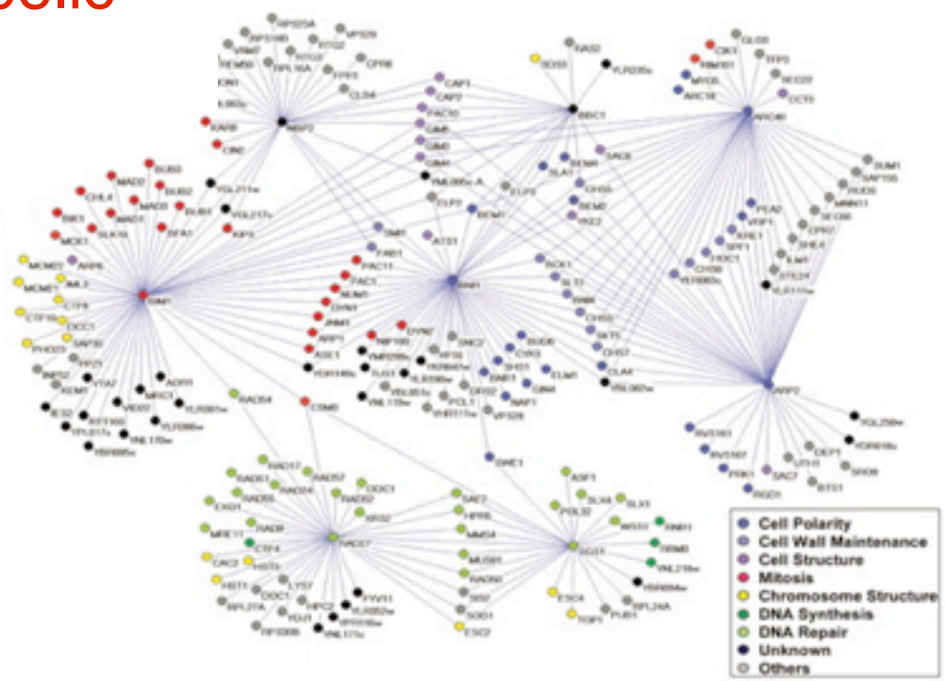


Yeast Phosphorylation network



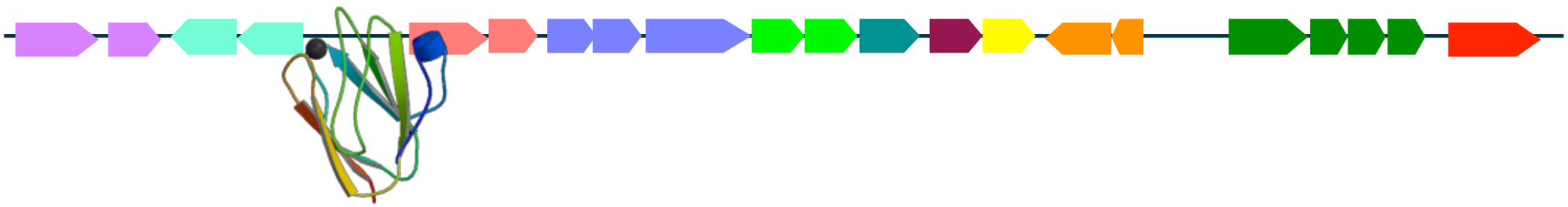
E. coli metabolic network

Yeast SSL network



Zhu et al, 2007.

Transcription network, aka regulatory network:



Transcription Factors =
proteins that bind to DNA
to activate or repress the
nearby, downstream genes.

regulates

gene

the regulated gene might
also be a transcription factor

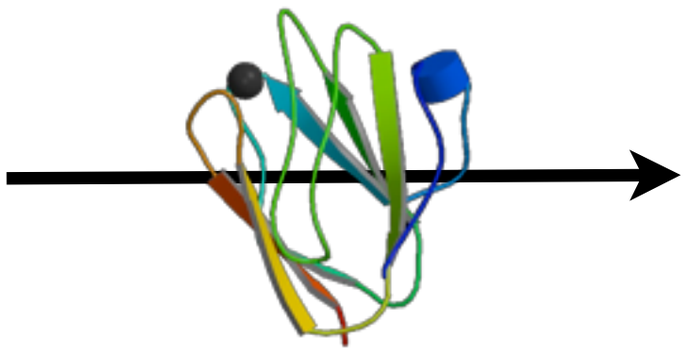
gene

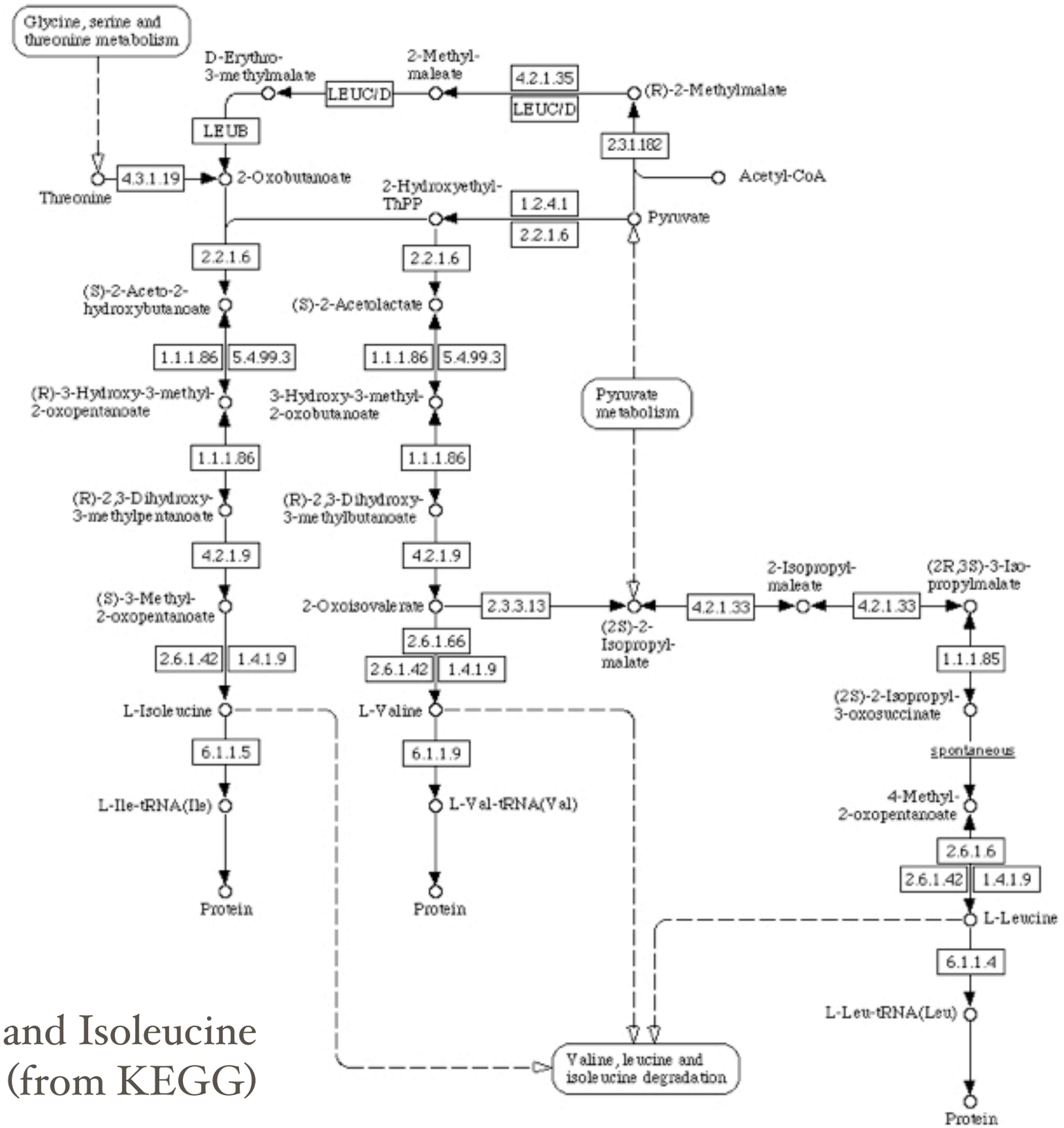
leads to a
directed graph

Metabolic network

Proteins are enzymes.

They label the *edges* and their substrates are the nodes.

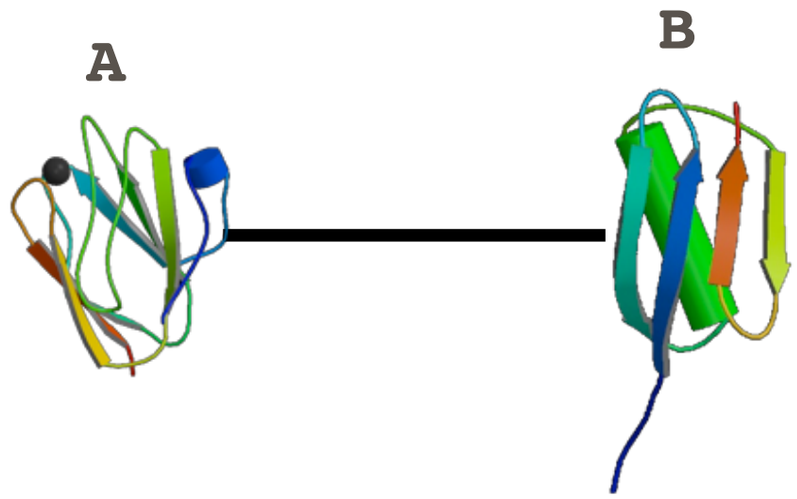




Valine, Leucine, and Isoleucine biosynthesis (from KEGG)

Protein-Protein Interaction Network

Proteins physically interact

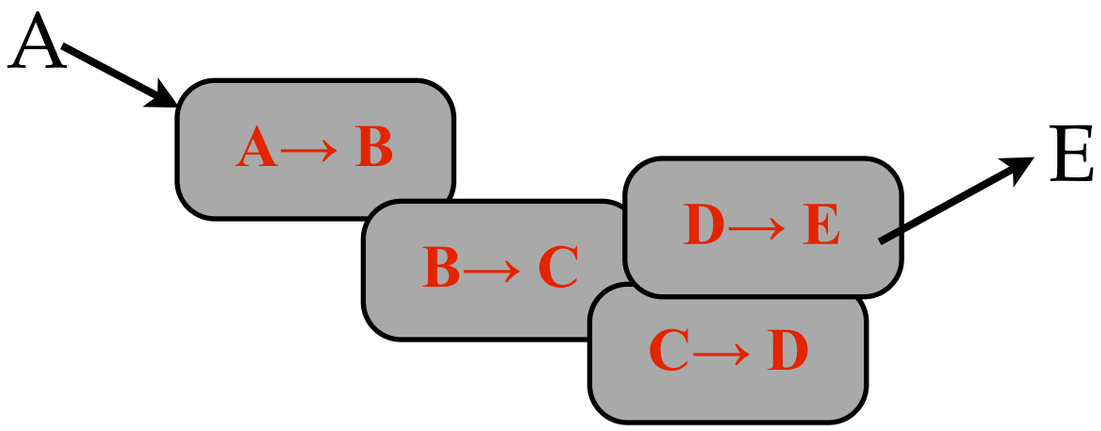


Edge {A,B}

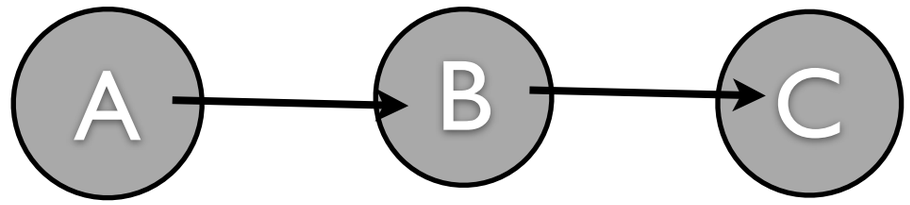
Assumption of binary interactions is imperfect.

Sometimes several proteins must bind simultaneously for there to be any “interaction”.

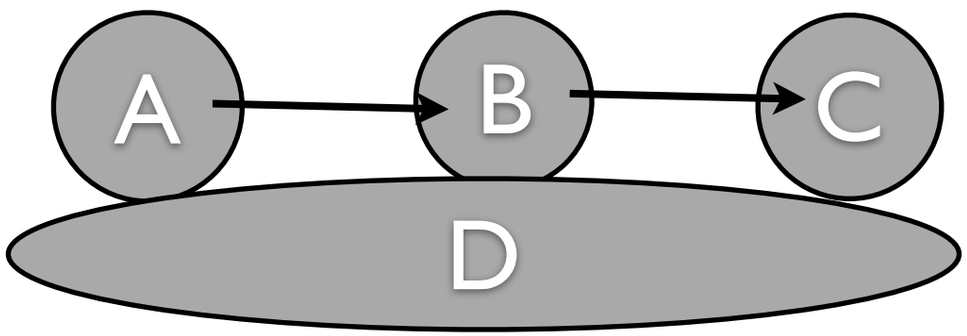
“Why” proteins interact:



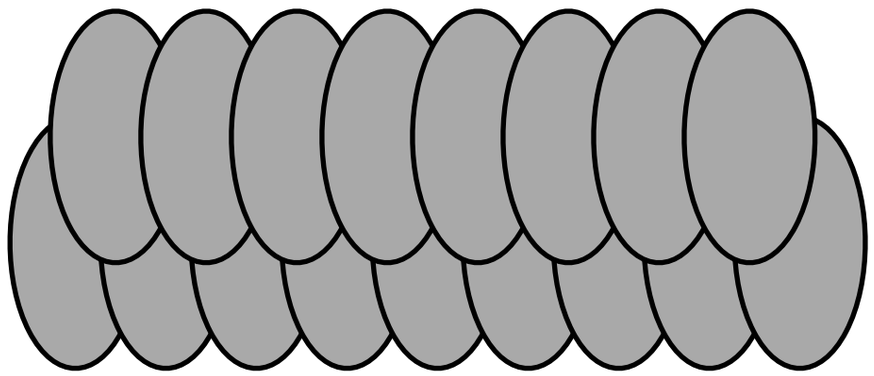
Bring chains of enzymes together



Signal Transduction



“Tethered” Signal Transduction



Form structures

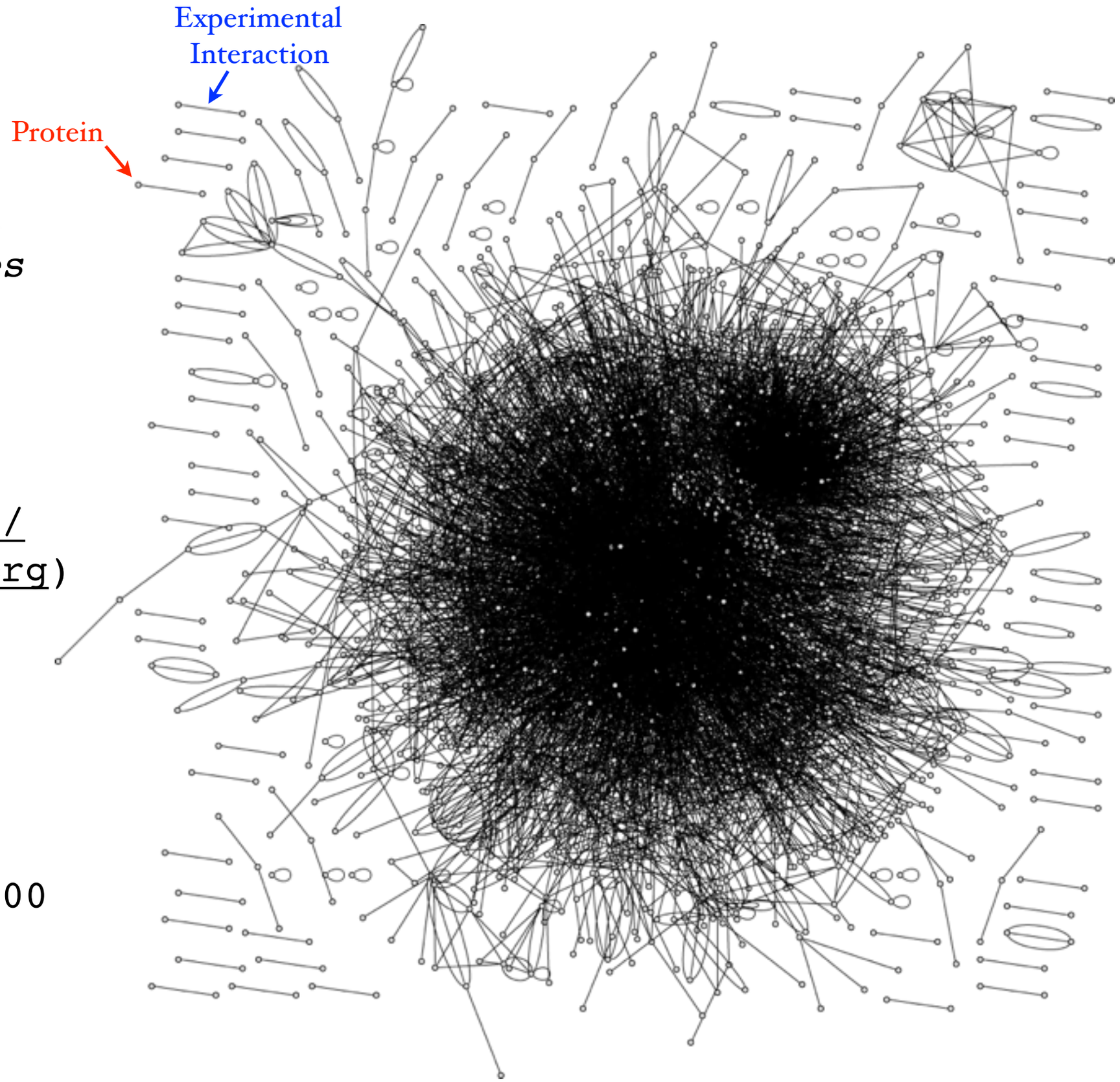
From “Analysis of Biological Networks” Junker and Schreiber, eds

A yeast (aka *Saccharomyces cerevisiae*) interaction network

GRID (http://thebiogrid.org)

8,742 edges

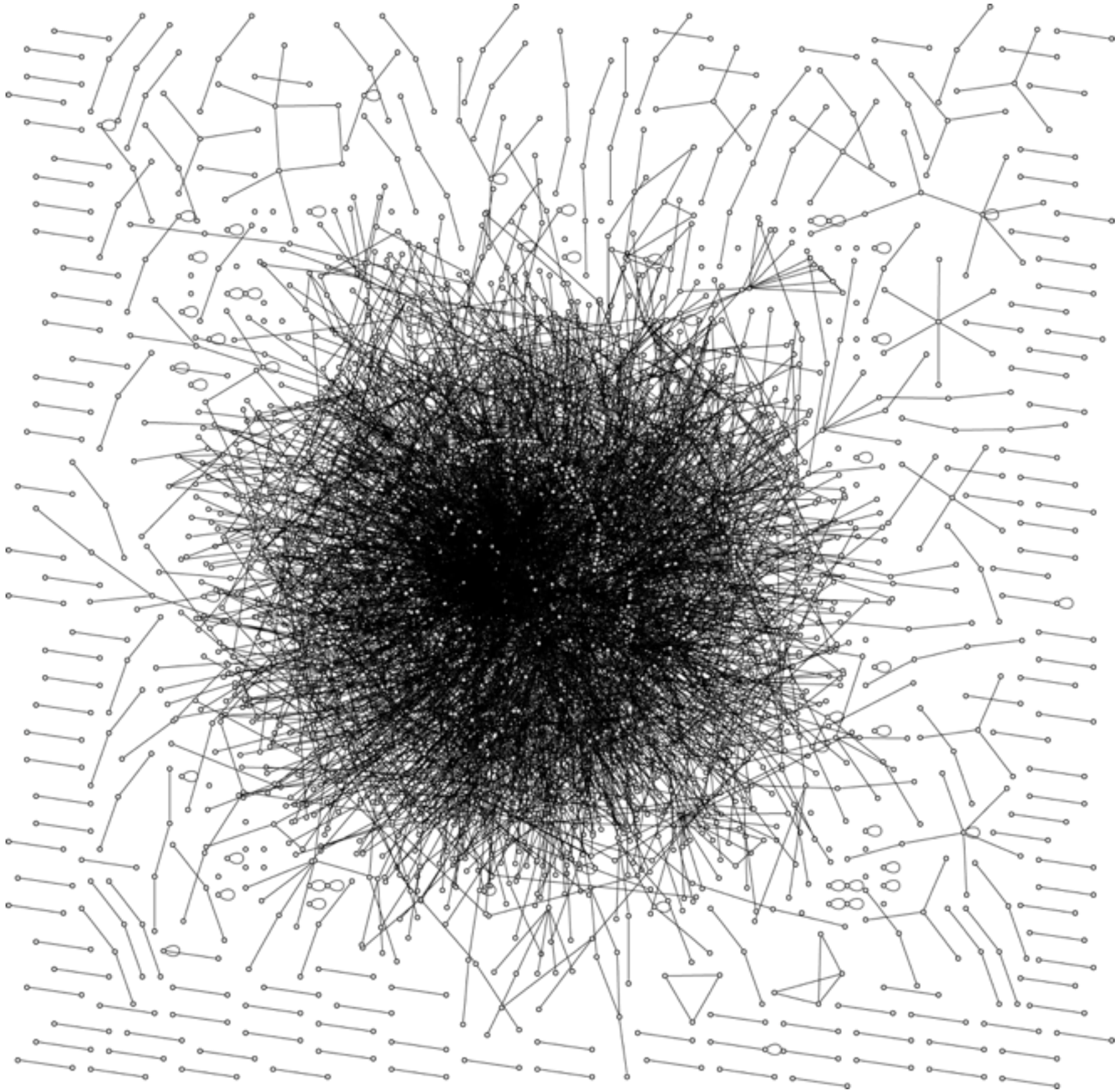
3113 nodes
(= proteins)
(out of ~6,000 genes)



A human interaction network

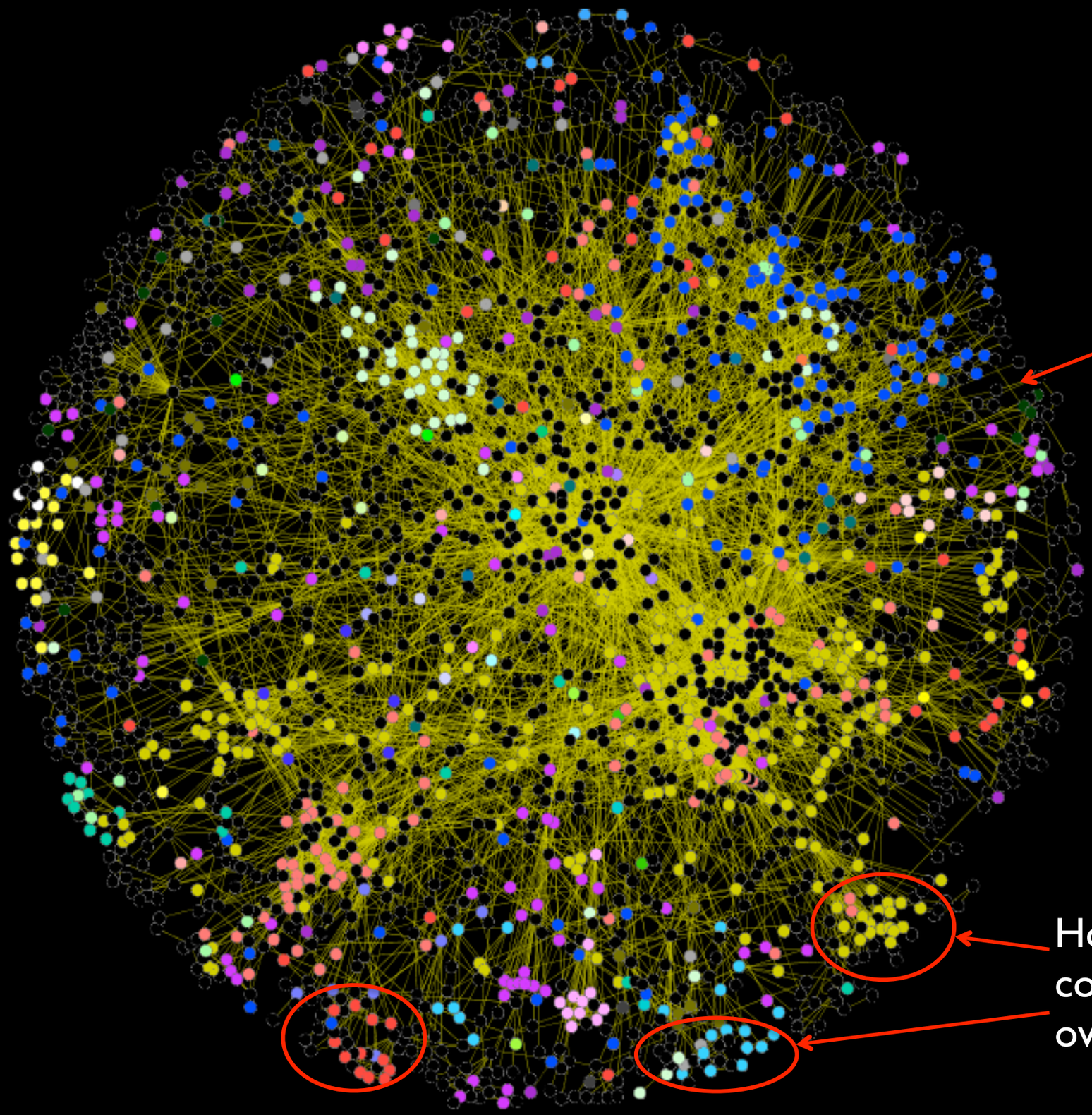
6,434 edges

4,083 nodes (out of ~22,000 genes)



How do we extract biological insight from these graphs?

<u>Yeast PPI network</u>:
2,599 proteins
8,275 interaction edges

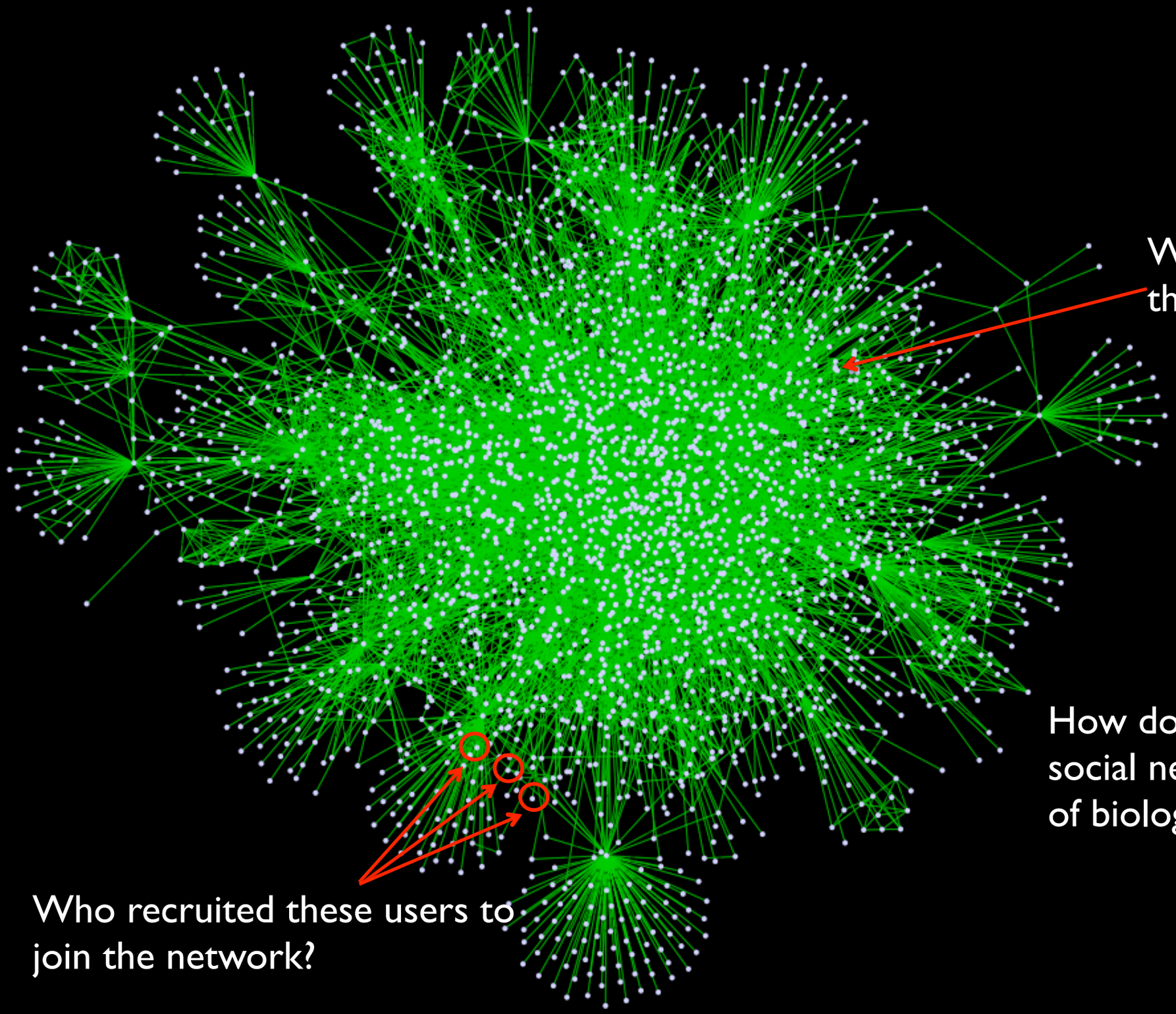


Did these proteins interact 10 million years ago?

Which model of evolution best characterizes the structure of this network?

How have these protein complexes reconfigured over evolutionary time?

<u>Last.fm social network</u>:
2,957 users
9,659 friendship edges



When did this user enter the network?

How do growth principles of social networks differ from those of biological networks?

Who recruited these users to join the network?

1. What role does each protein play in the cell?

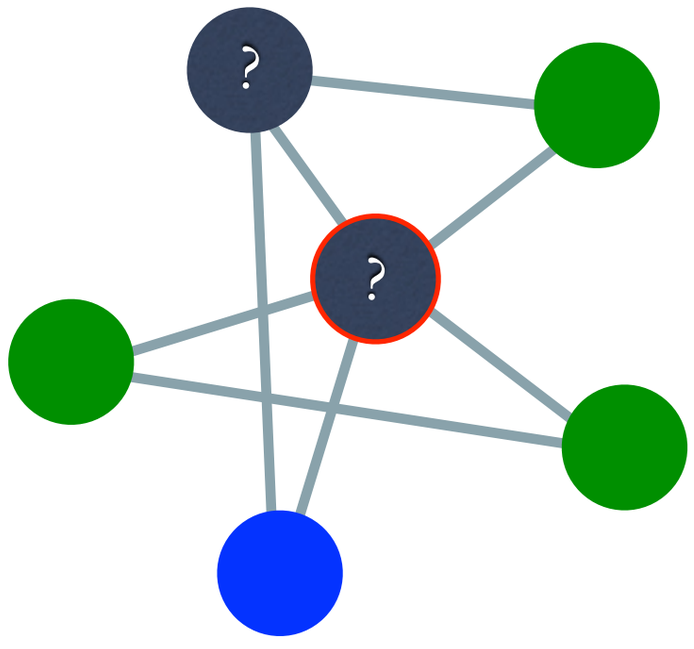
2. How do we uncover the true graph from noisy samples?

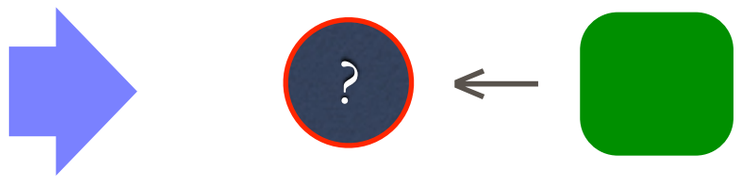
3. How do we compare interaction graphs across species?

4. What are the characteristics of interaction graphs?

1 Function Prediction

- Proteins with known function + network topology → function assignment for unknown proteins.
- Guilt by association
- Simple: Majority Rule:



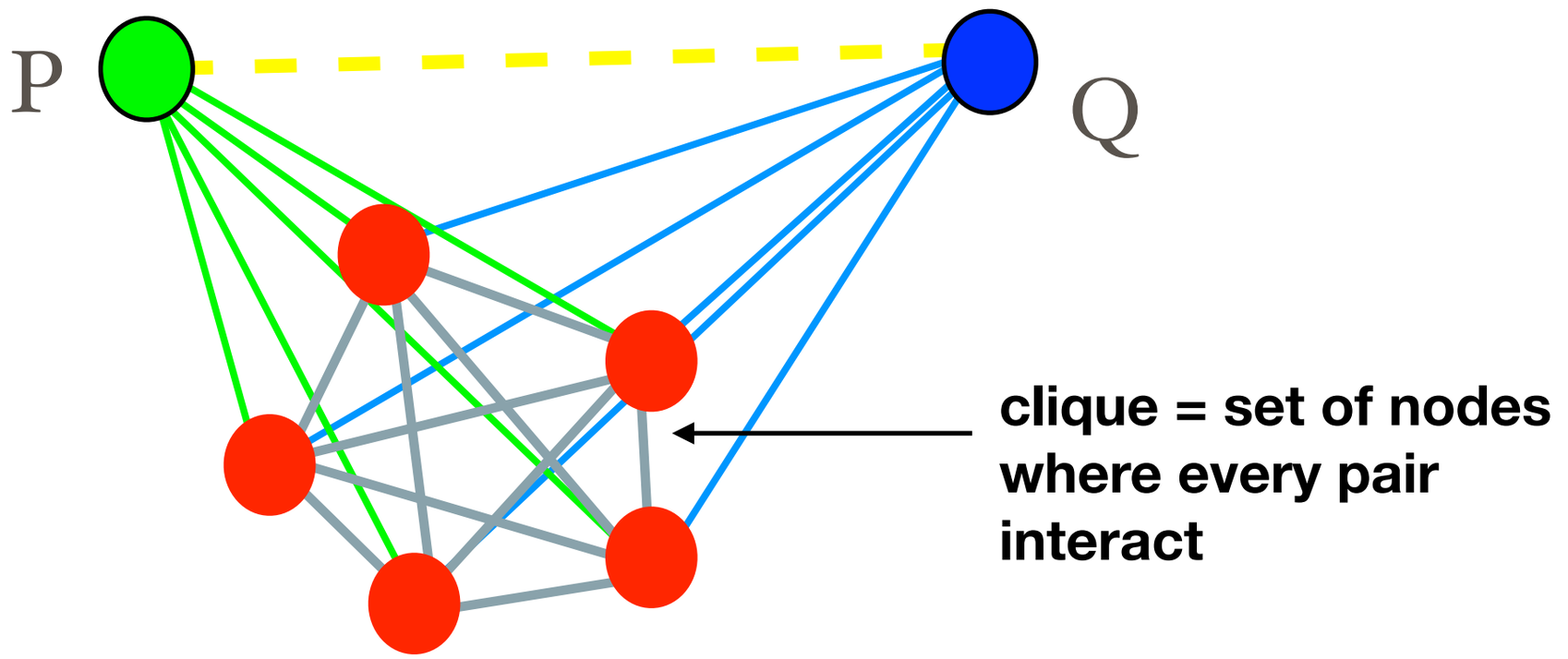


Doesn't take into account connections between neighbors
Or annotations at distance > one

2

Predicting Missing Edges

- Completing Defective Cliques (Yu, et al, 2006):



- P, Q both adjacent to all nodes in clique (there are two (n-1)-cliques that overlap by (n-2) nodes) ⇒ likely that P,Q should be adjacent.

3

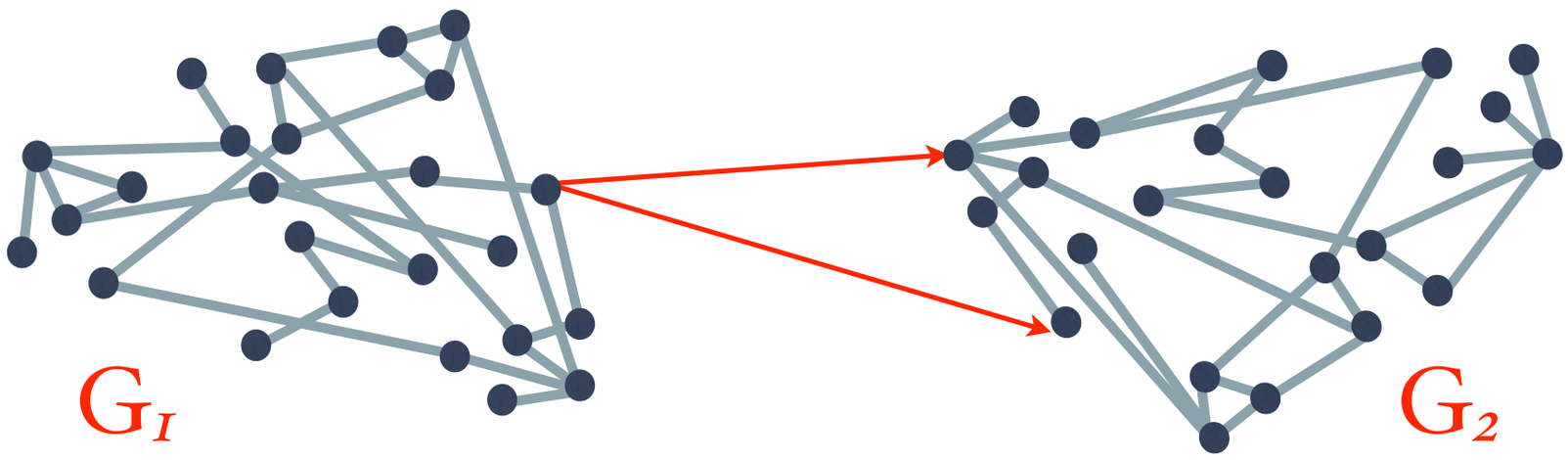
Aligning Networks

- Let $G_1 = (V_1, E_1)$, G_2, ... G_k be graphs, each giving noisy experimental estimations of interactions between proteins in organisms 1,.., k.

- If $G_i = (V_i, E_i)$, we also have a function:

$$\mathrm{sim}(u,v) : V_i \times V_j \rightarrow \mathrm{R}$$

that gives the sequence similarity between u and v.



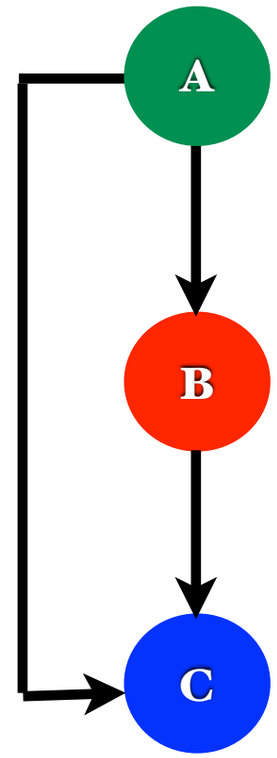
4 Over-represented Network Motifs

- Are there connection patterns that occur frequently?

- "Frequently" = more often than you'd expect in a random graph with the same degree distribution.

- Milo et al., 2002 found the feed forward motif over-represented in gene regulation networks.

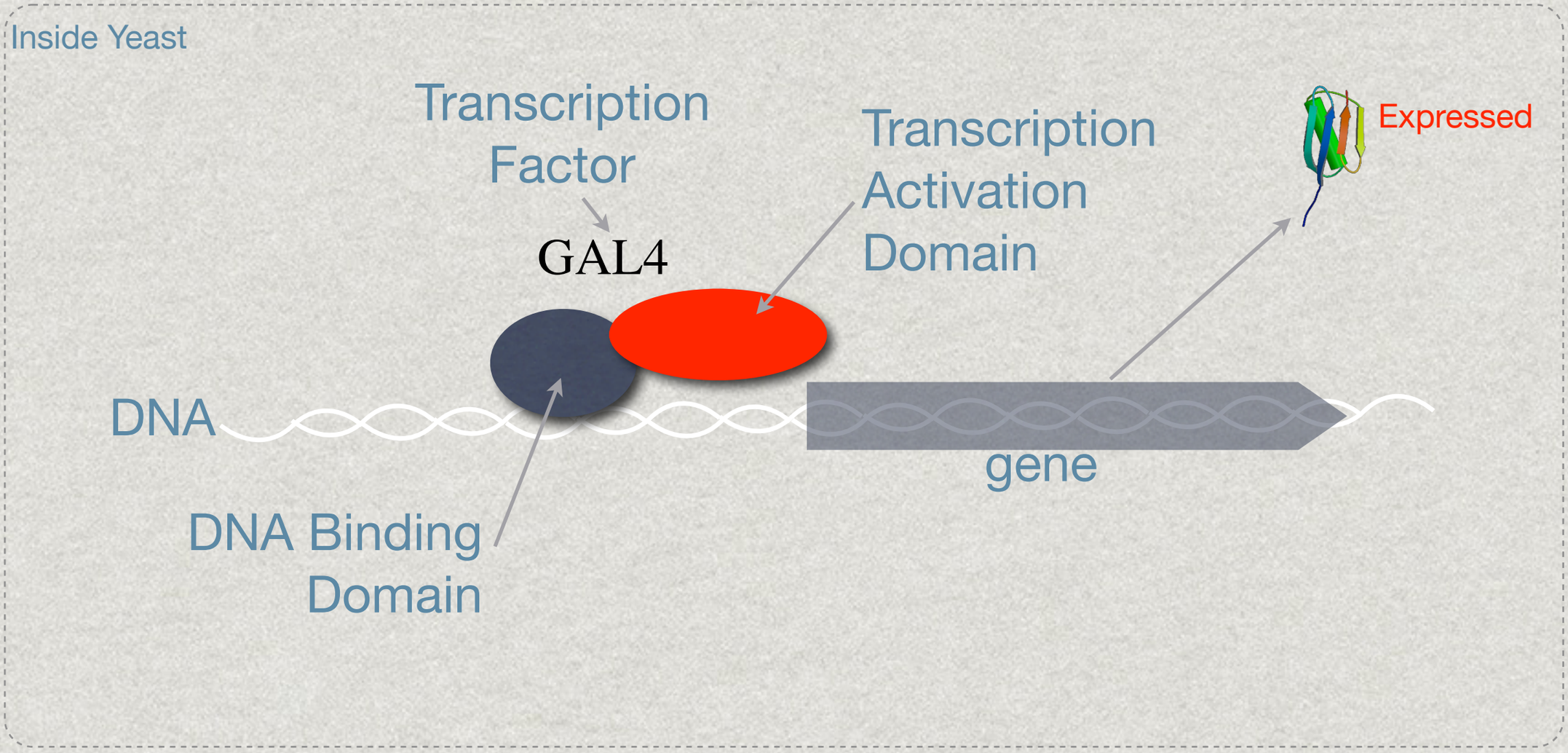
- Larger motifs are computationally difficult to find.



Experimental Techniques to Determine Protein Interactions

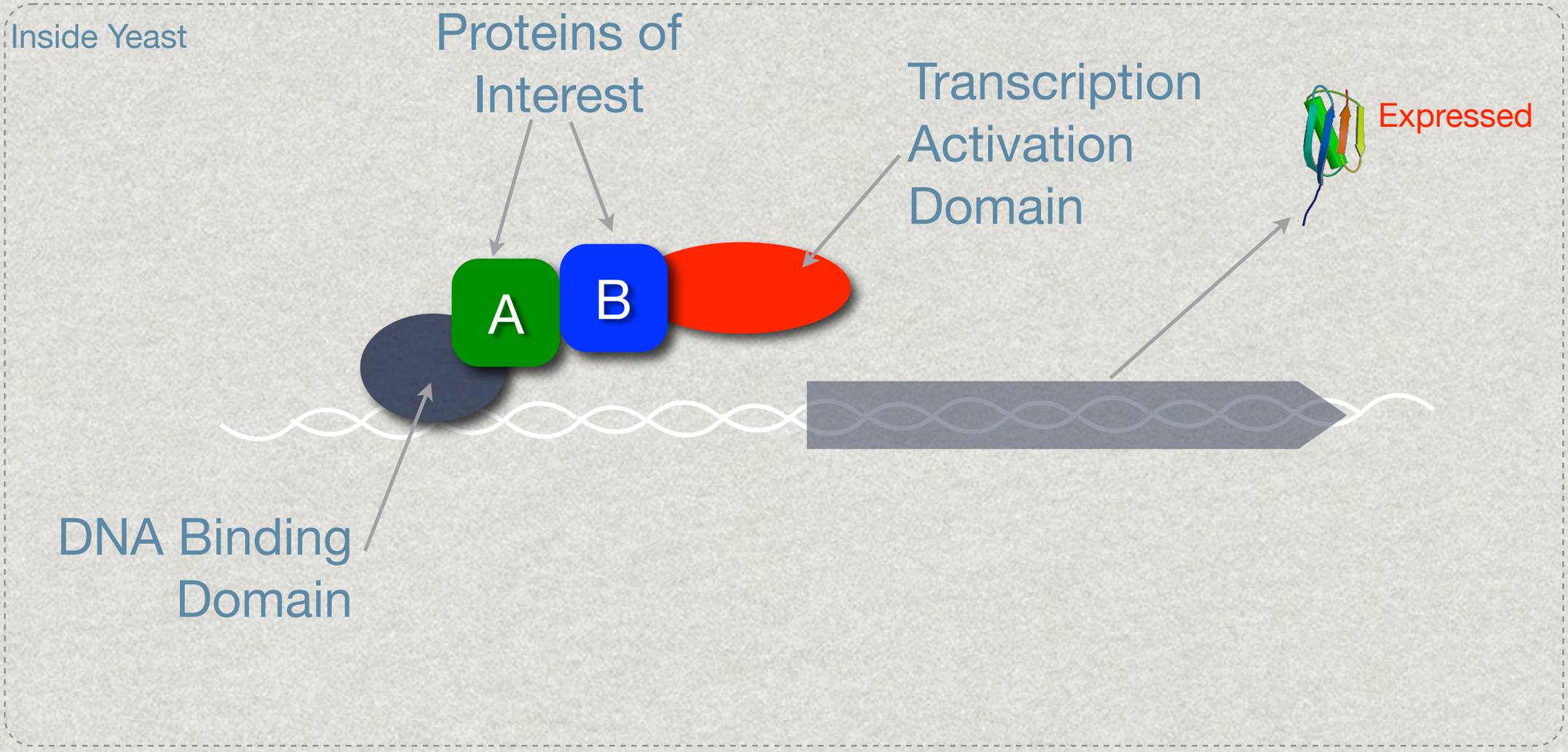
- **Slow, accurate, costly:**
  - X-ray crystallography
  - NMR
- **High throughput, but more error prone:**
  - Yeast Two-Hybrid
  - TAP-MS (tandem affinity purification / mass spec)

Yeast Two-Hybrid

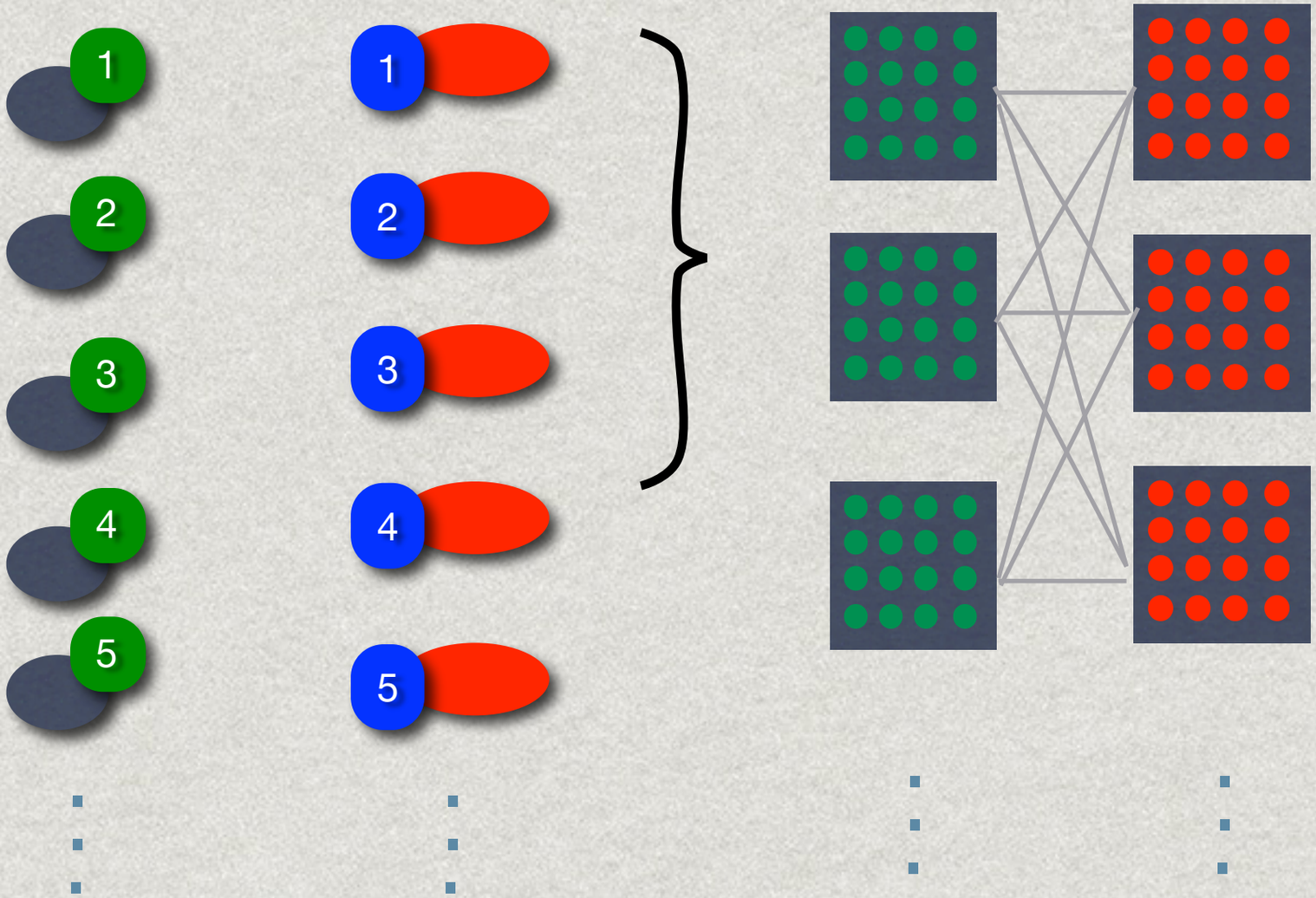


"Domain" = functional, evolutionary conserved unit of a protein

Yeast Two-Hybrid



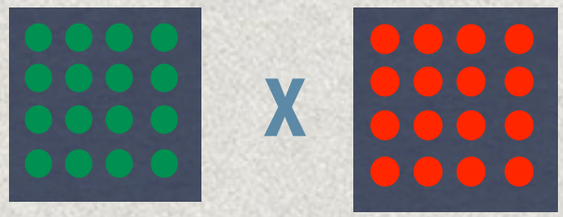
Scaling Up (Ito et al, 2001)



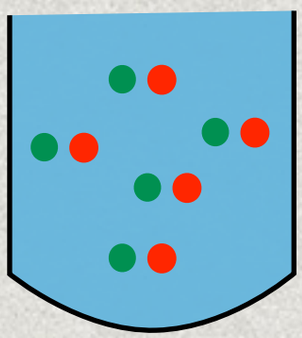
BAIT **PREY**

96-well plates
Each well contains a yeast strain with a different hybrid

~ 6,000 genes / 96
= 62 plates
= 3,844 crosses between plates

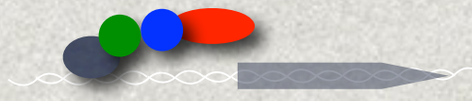


Mixed together
and allowed to mate

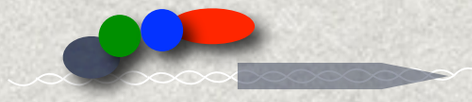


96 x 96 combinations
all mixed together

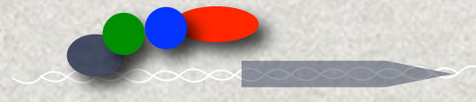
Gal4 actives 4 genes in the hybrids:



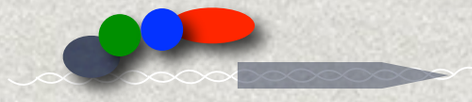
ADE2 => adenine



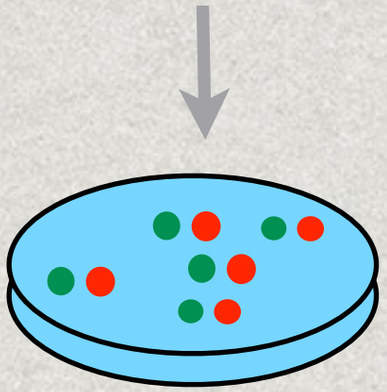
HIS3 => histidine



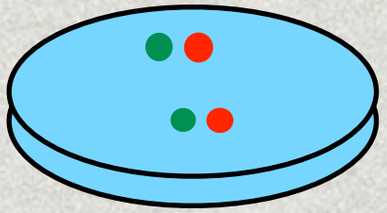
URA3 => uracil



MEL1



Kill off all strains
that don’t express
all 4 genes.



Sequence
remaining hybrids

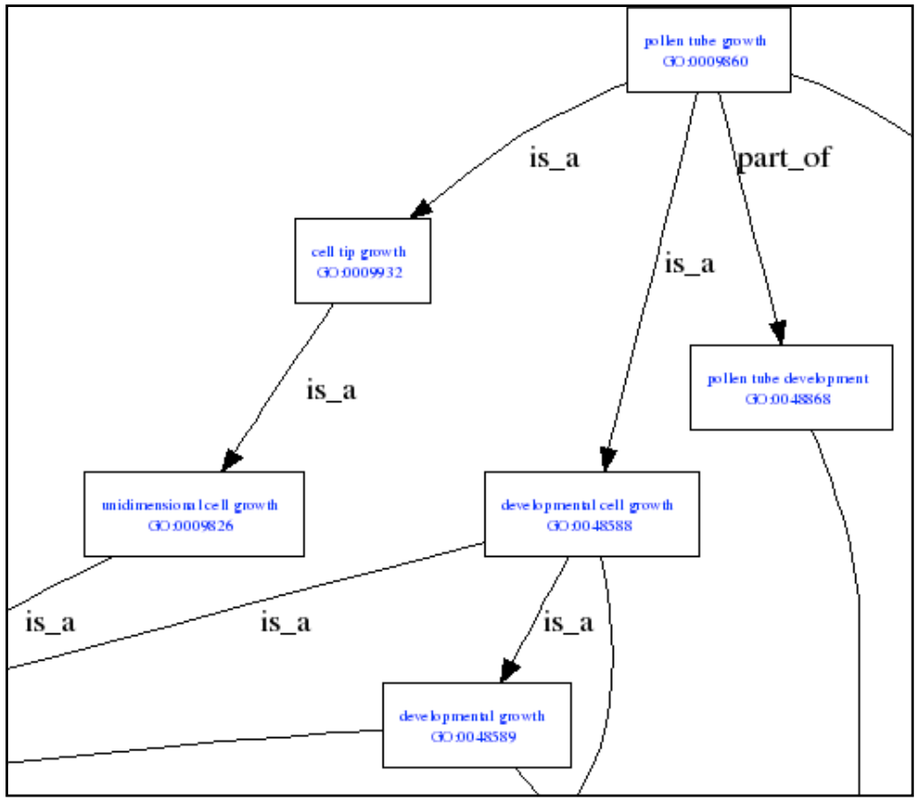
Function Prediction

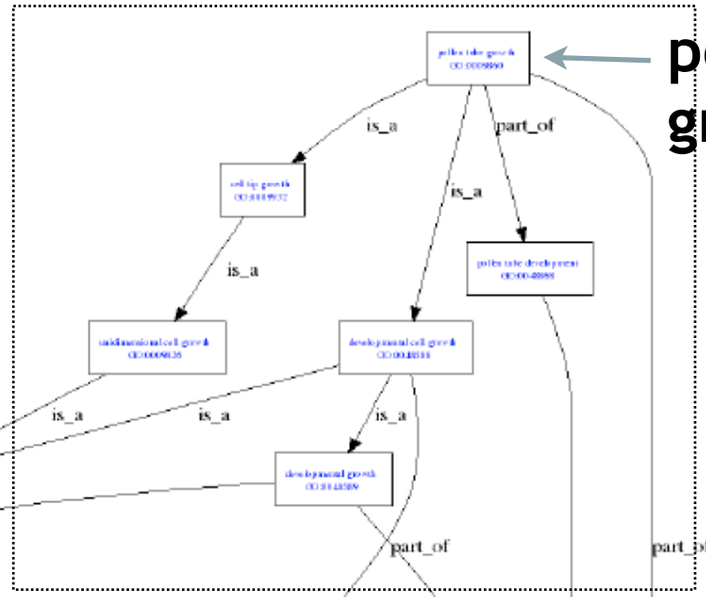
Predicting Protein Function from Networks

- Ultimately, we want to know how various processes in the cell work.
- A first step: figure out which proteins are involved in which biological role.

- What do we mean by a “biological role”?
  - Several different schemes:
    - Gene Ontology (largest, most widely used)
    - MIPS (good collection of known protein complexes)
    - KEGG (manually curated pathways)

Gene Ontology (GO)





pollen tube growth

Curated collection of biological functions

- Node = manually defined function
- **Directed, acyclic graph**
- Main edges are either “is a” or “part of”

Gene Ontology has 3 Sub-ontologies

- Cellular component: a part of the cell (a location, or organelle, or other structure)
- Biological process: a collection of steps that the cell carries out to achieve some purpose. E.g. cell division.
- Molecular function: a specific mechanism that a protein performs. E.g.
  - a kinase would have molecular function “phosphorylation”;
  - a transcription factor would have molecular function “DNA binding”
- Each protein may be *annotated* with several terms from each sub-ontology.

GO Edge Types

- **is_a**: like a C++ or Java subclass relationship.
  - A is_a B means A is a more specific version of B
  - E.g. “nuclear chromosome” **is_a** “chromosome”.
- **part_of**: A is some part of B
  - A piston is **part_of** an engine (but a piston is not an specific kind of engine)
- *Transitivity*:
  - If a protein is annotated with term A, it is implicitly annotated with **all** the ancestors of A (following every path to the root).
  - GO is explicitly designed so this is always true.

Predicting Protein Function from Networks

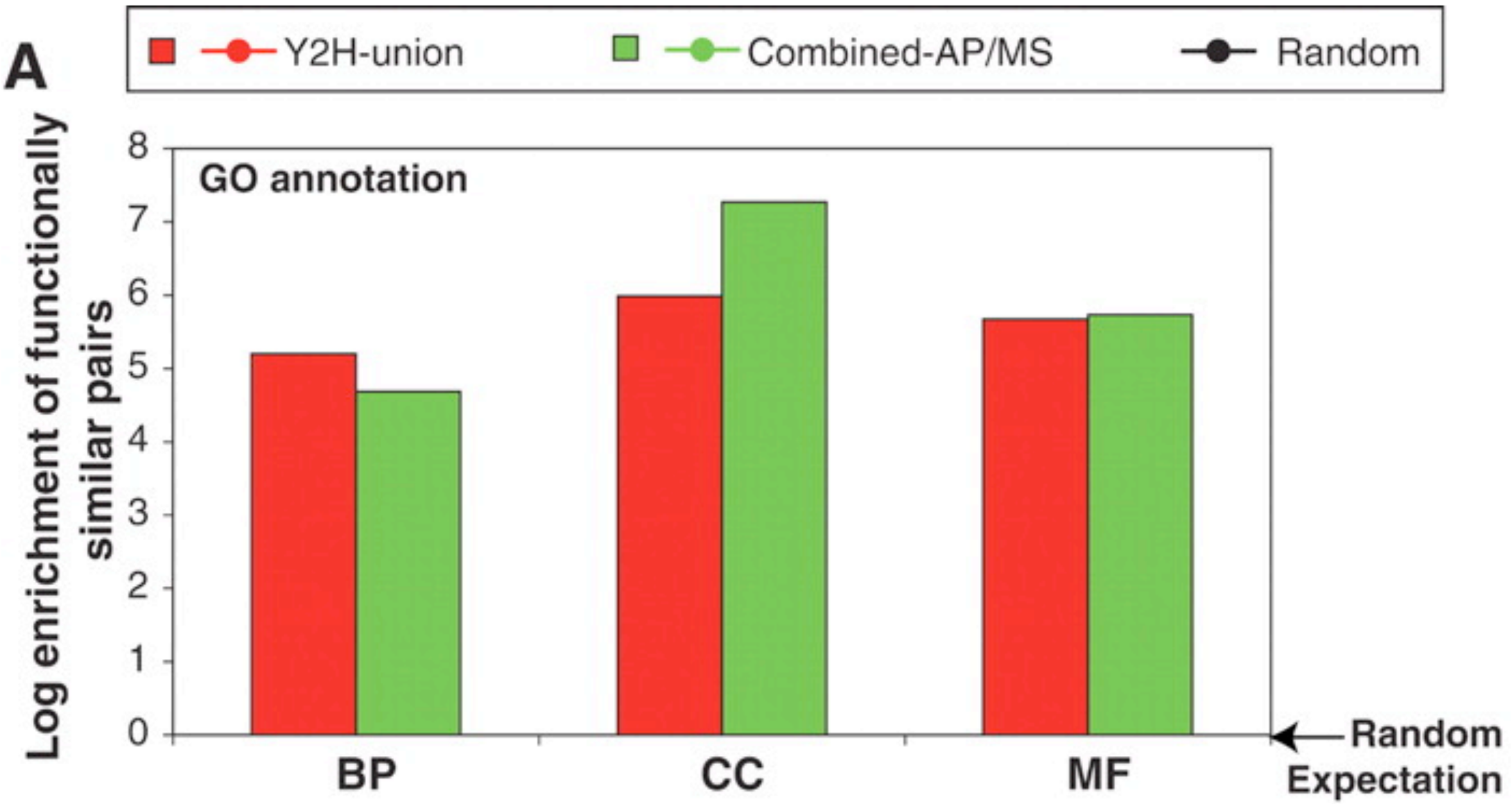
Machine learning problem:

Input: Network with some nodes (proteins) labeled with their (GO) function.
Goal: Label the unlabeled nodes (proteins) with their predicted function.

Some recent approaches: (see next slides)

- Majority Rule
- Neighborhood enrichment
- Minimum Multiway Cut
- “Functional Flow”

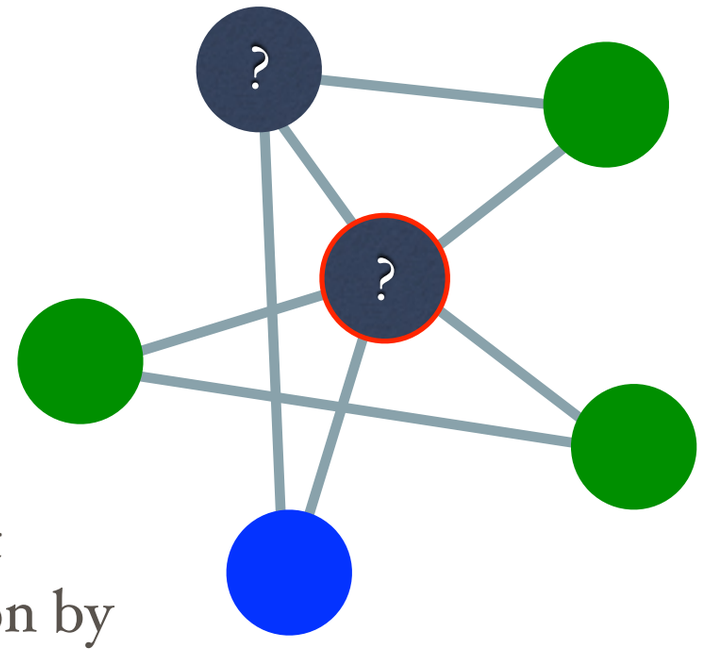
Neighboring Proteins More Likely to Share Function



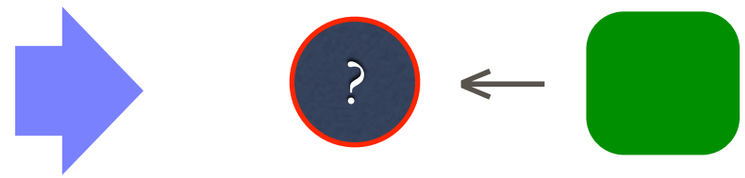
- (Yu et al., 2008)

Majority Rule

- Proteins with known function + network topology → function assignment for unknown proteins.
- Guilt by association
- Majority Rule:



Can weight contribution by edge weight.



Doesn't take into account connections between neighbors
Or annotations at distance > one

Neighborhood Approaches, e.g.:

- Let $N(u,r)$ be all the proteins within distance r to u.

$$f(u, r, a) = |\{u \in N(u, r) : u \text{ has function } a\}|$$

= # of proteins in neighborhood with function *a*

$$e(u, r, a) = |N(u, r)| \cdot \frac{|\{u \in V : u \text{ has function } a\}|}{|V|}$$

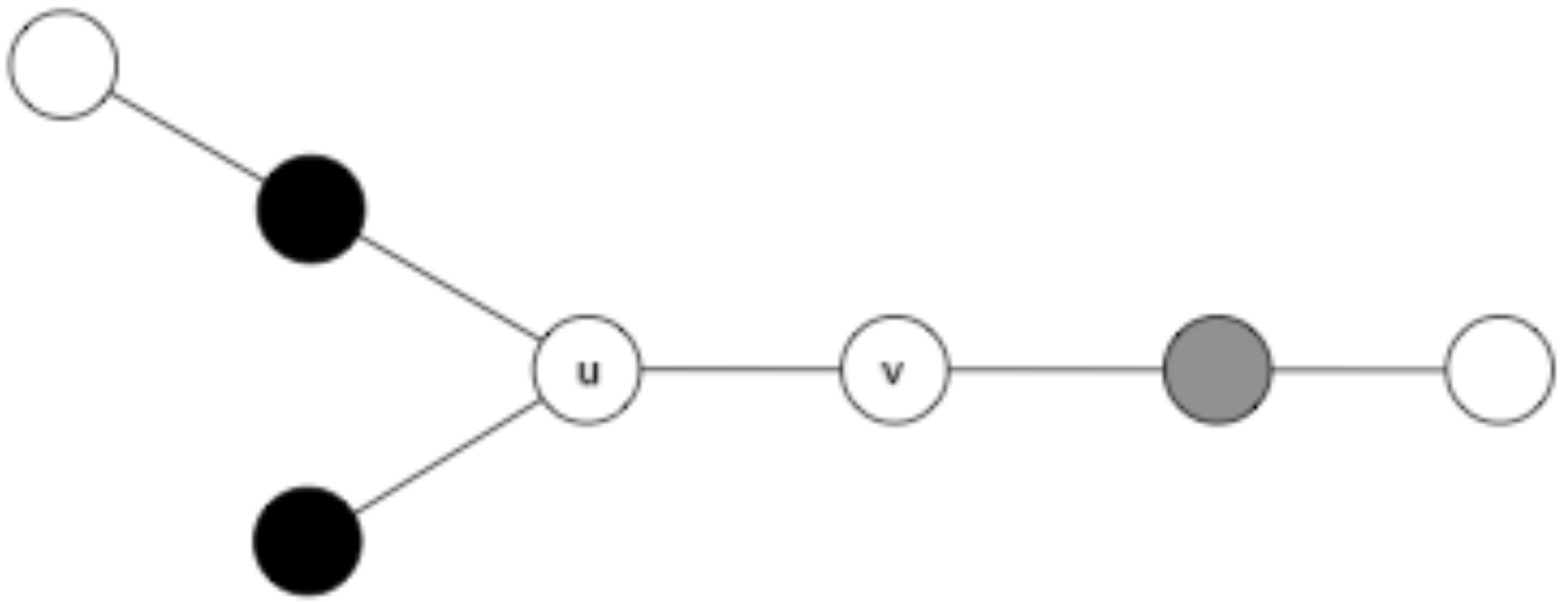
= Expected # of proteins in neighborhood with function *a*

$$\text{Score}(u, r, a) = \frac{(f(u, r, a) - e(u, r, a))^2}{e(u, r, a)}$$

- Protein u is assigned function argmax_a Score(u,r,a)

Problems with neighborhood

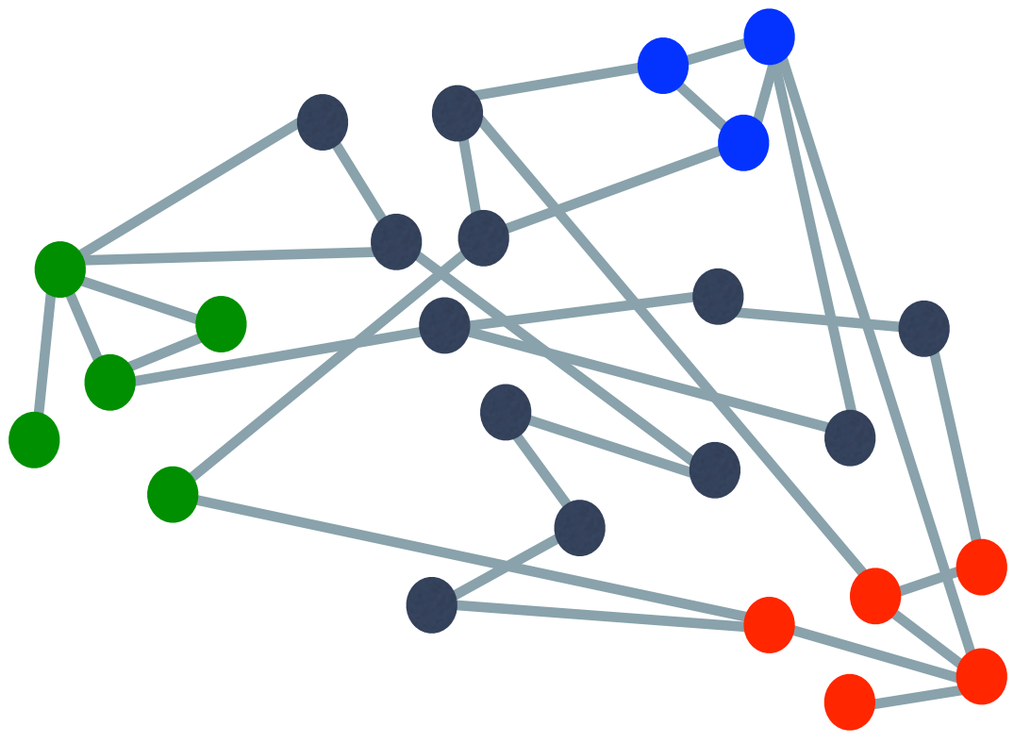
- Neighborhood with radius 2 gives the same scores for both black and gray functions to nodes u and v:



(Nabieva, Singh, 2008)

Minimum Multiway *k*-Cut: Partition the nodes so that each of *k* (sets of) terminal nodes is in a different partition & the number of edges cut is minimized.

- Proposed by Vazquez et al (2003) and Karaoz (2004) for function annotation.
- One “terminal node set” for each function, containing proteins known to have that function.
- NP-hard → an approach based on integer programming



Integer Programming

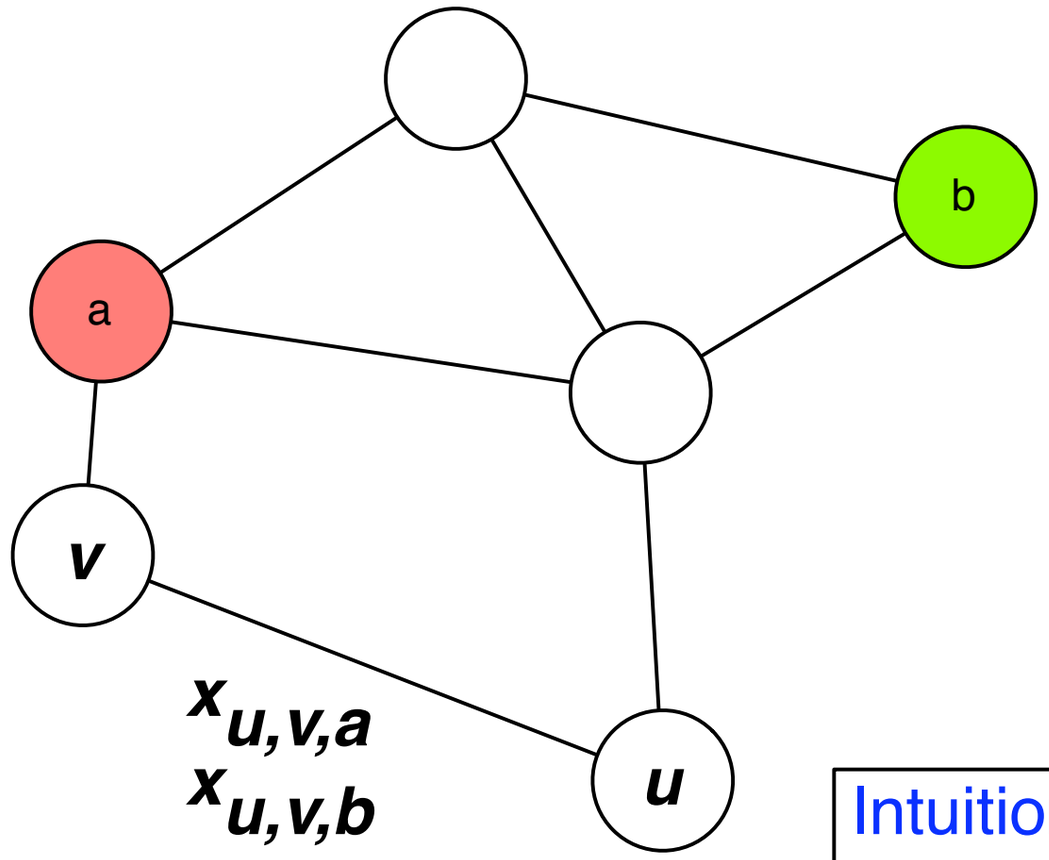
- General optimization framework:
  - Describe system by set of variables

IP :=
- Minimize a linear function.
- Subject to linear constraints (= or ≥).
- While requiring the variables to be {0, 1}.

- Computationally hard, but many advanced solver packages:
  - CPLEX, COIN-OR, ABACUS, FortMP, LINGO, …

Integer Programming (IP) Formulation for Multiway Cut

Introduce 0/1 variables associated with each node and edge:



Intuition: $x_{u,v,a}$ is 1 if both u and v are assigned to annotation **a**; 0 otherwise

Intuition: $x_{u,a}$ is 1 if node u is assigned to annotation **a**; 0 otherwise

IP for Min Multiway Cut

$$\text{maximize} \sum_{\{u,v\} \in E, a} x_{u,v,a}$$

Maximize # of "monochromatic edges" Equivalent to minimizing the number of cut edges.

Subject to:

$$x_{u,x} \text{ and } x_{u,v,a} \in \{0,1\}$$

$$\sum_a x_{u,a} = 1$$

Each node gets exactly 1 annotation

$$x_{u,v,a} \leq x_{u,a}$$

$$x_{u,v,a} \leq x_{v,a}$$

Can set $x_{u,v,a}$ to 1 iff both its endpoints are 1

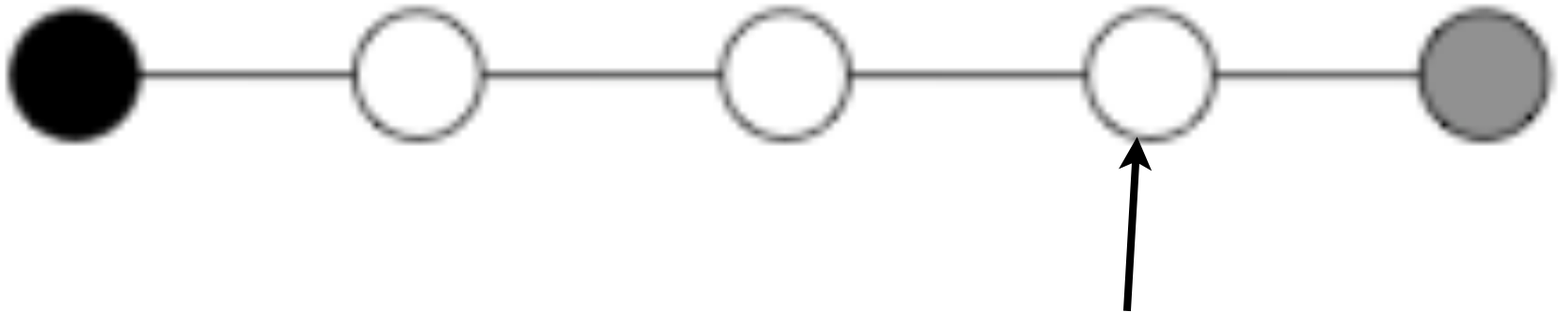
$$x_{u,a} = 1 \text{ if } a \in \text{annot}(u)$$

$$x_{u,a} = 0 \text{ if } a \notin \text{annot}(u) \neq \emptyset$$

Fix variables for nodes with known annotations.

Problem with Simple Cut Approaches

- Every cut is equally likely:

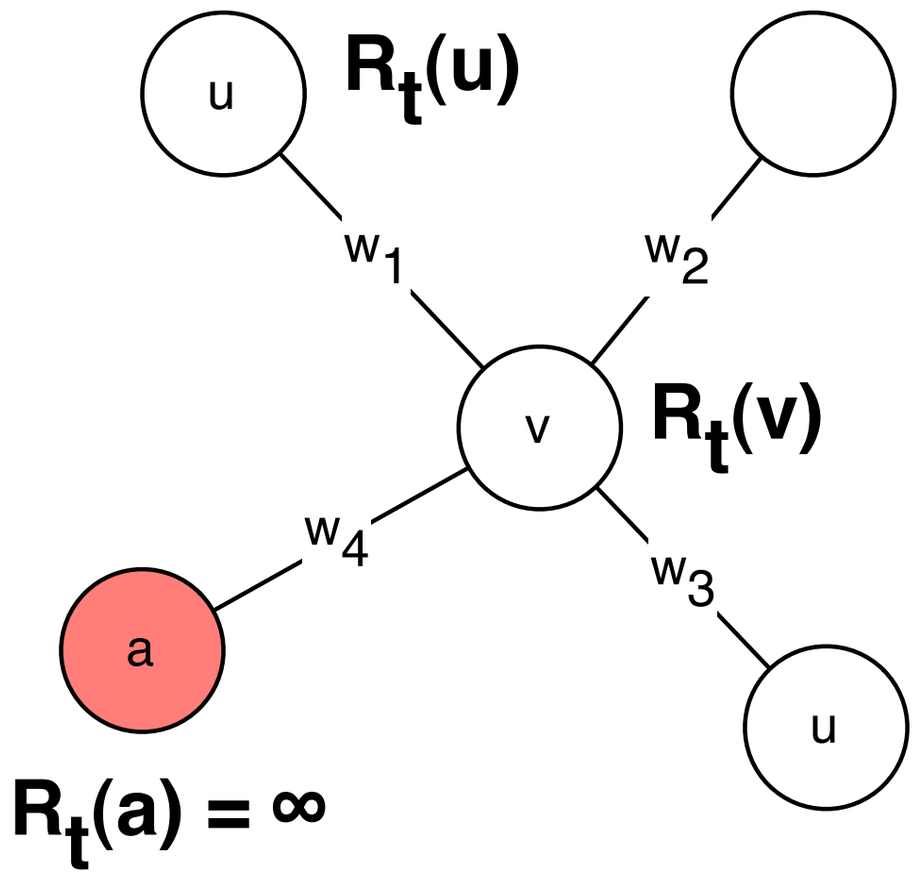


but this node is
more likely to be
grey than black

(Nabieva, Singh, 2008)

Functional Flow (Nabieva et al.)

Each node *u* has a "reservoir" at each time step t.

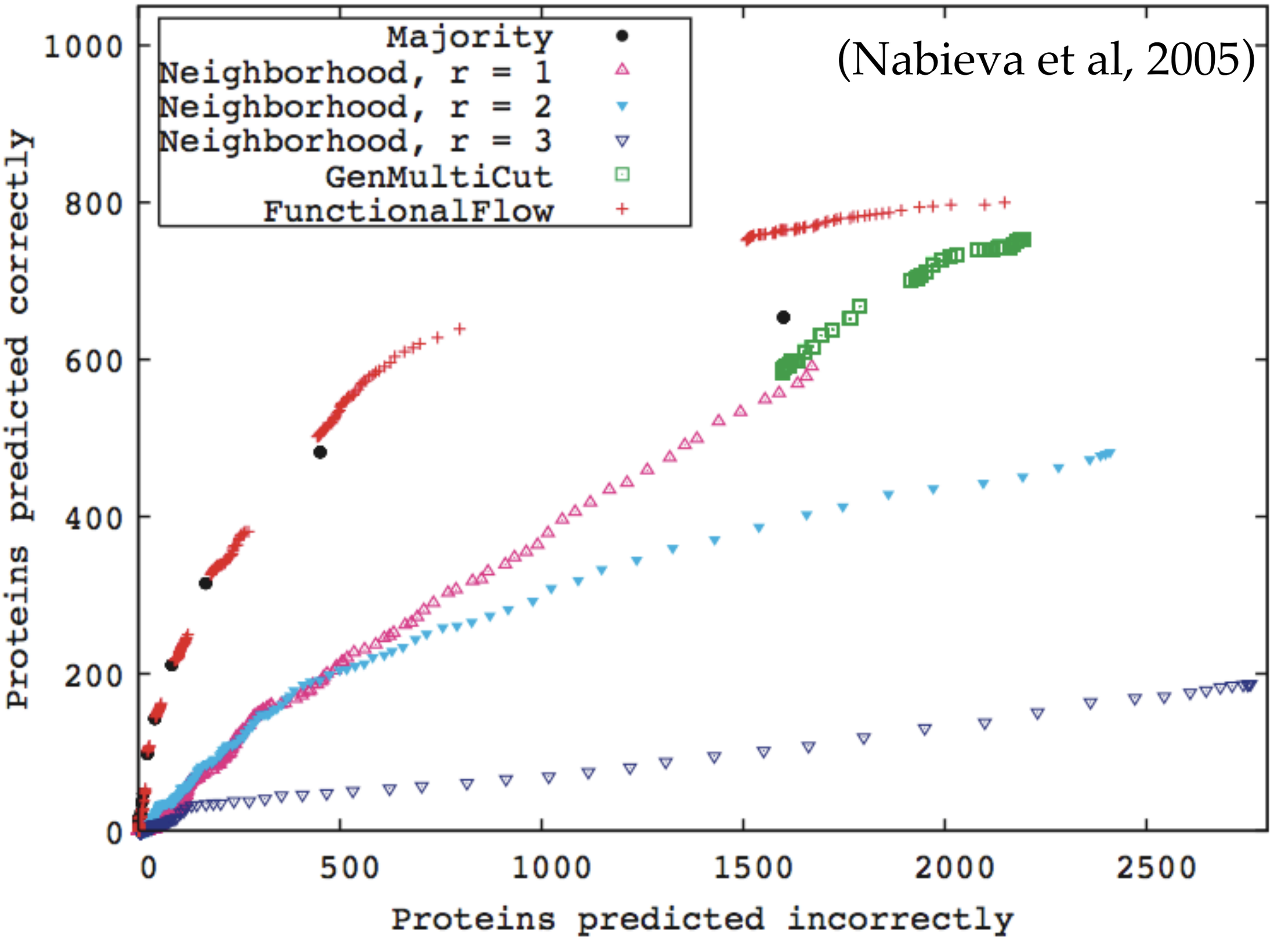


At every time step, water flows "downhill" from the more filled reservoir to the more empty reservoir, up to the capacity of the edge.

If there isn't enough water to fill the downhill pipes, it is distributed proportionally to the capacity of the edge.

Every function f is considered separately.
Score(u,f) is the total water that passed through u when considering f.
Predicted function for u is the function with the highest score.

Performance of These Predictions on Yeast



(Nabieva et al, 2005)

Summary

- Guilt-by-association = proteins near one another in the network are more likely to have the same function.

- Neighborhood 1 does better than larger neighborhoods Perhaps because the structure of the neighborhood is not taken into account.

- Integer programming NP-hard, but often practical.

- “Functional flow” is an embodiment of a general technique that has been applied in many problems: “information” being passed along the network.