Function Prediction

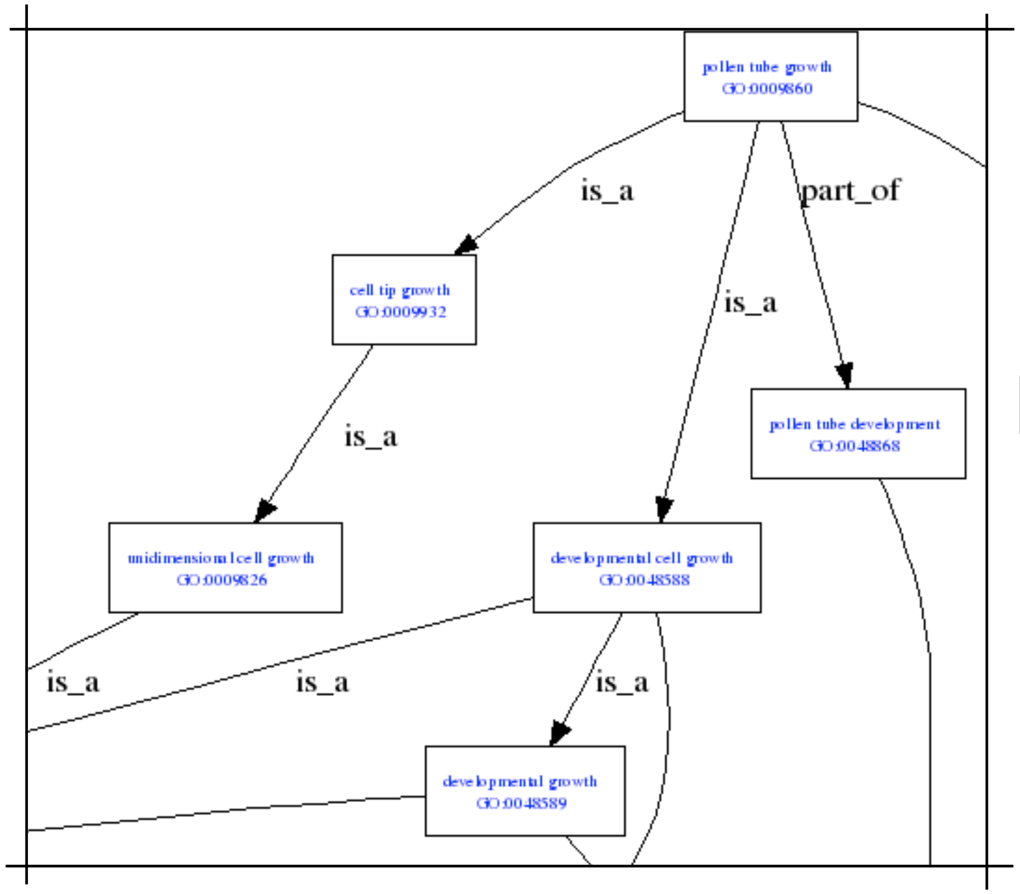
CMSC 858L

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.

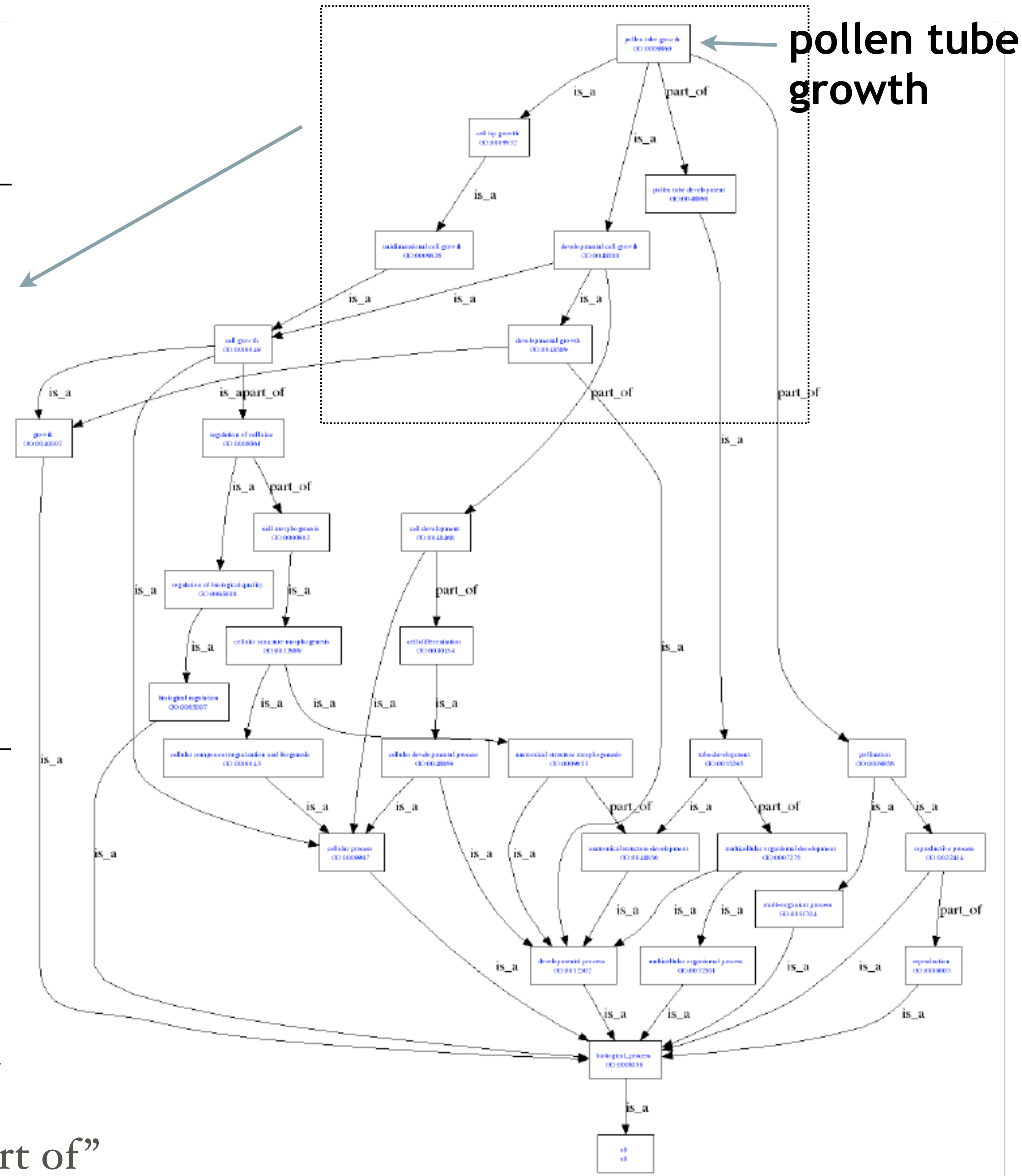
- What do we mean by a "biological role"?
  - Several different schemes:
    - Gene Ontology (largest, most widely used)
    - MIPS (good collection of known protein complexes)
    - KEGG (manually curated pathways)

Gene Ontology (GO)



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.

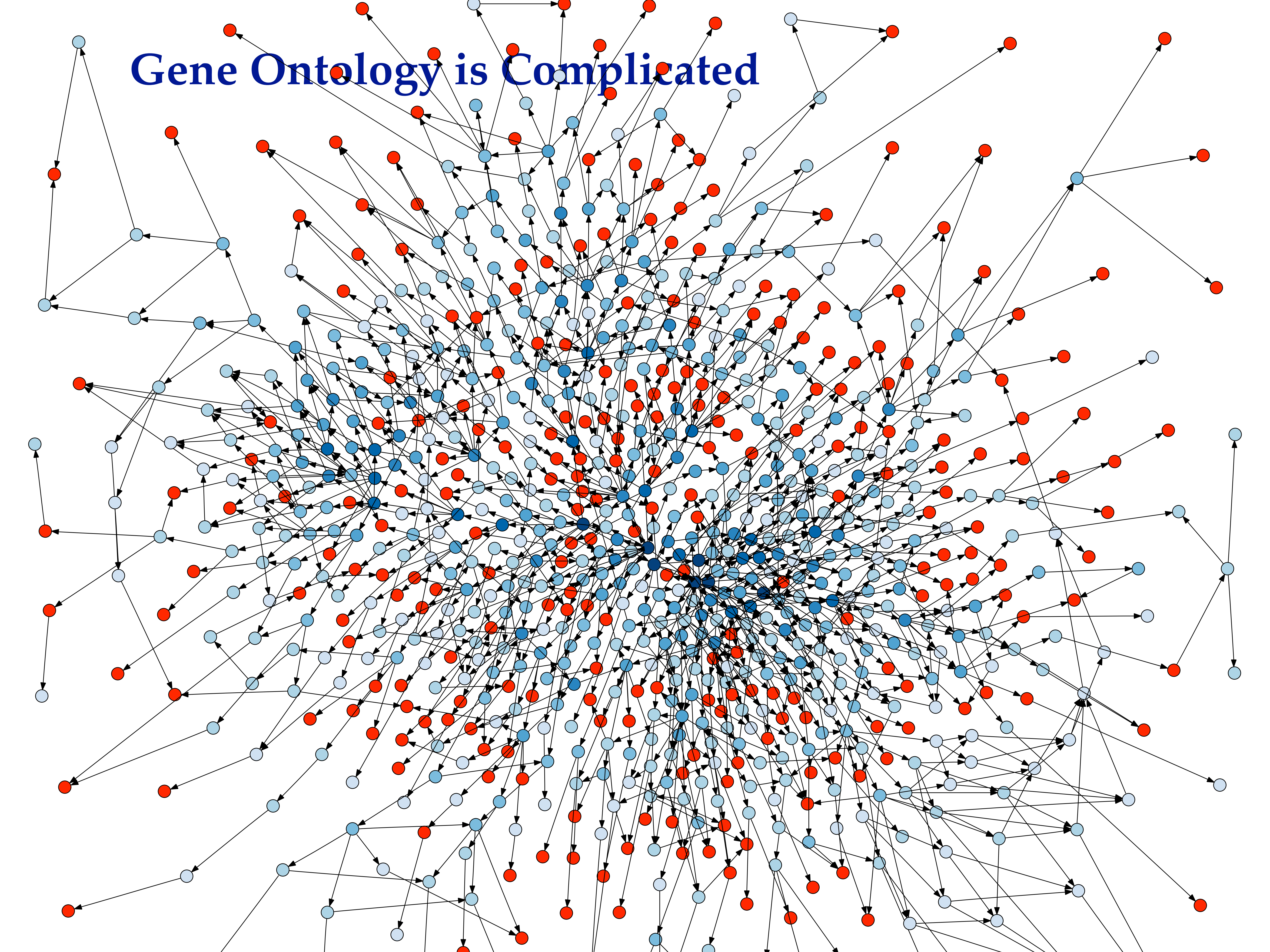
- Molecular function: a specific mechanism that a protein performs. E.g.
  - a kinase would have molecular function "phosphorylation";
  - a transcription factor would have molecular function "DNA binding"

- Each protein may be *annotated* with several terms from each sub-ontology.

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.

Gene Ontology is Complicated



KEGG is a tree of "pathways"

1. Metabolism

1.1 Carbohydrate Metabolism

- Glycolysis / Gluconeogenesis
- Citrate cycle (TCA cycle)
- Pentose phosphate pathway
- Pentose and glucuronate interconversions
- Fructose and mannose metabolism
- Galactose metabolism
- Ascorbate and aldarate metabolism
- Starch and sucrose metabolism
- Aminosugars metabolism
- Nucleotide sugars metabolism
- Pyruvate metabolism
- Glyoxylate and dicarboxylate metabolism
- Propanoate metabolism
- Butanoate metabolism
- C5-Branched dibasic acid metabolism
- Inositol metabolism
- Inositol phosphate metabolism

1.2 Energy Metabolism

- Oxidative phosphorylation
- Photosynthesis
- Photosynthesis - antenna proteins
- Carbon fixation in photosynthetic organisms
- Reductive carboxylate cycle (CO2 fixation)
- Methane metabolism
- Nitrogen metabolism
- Sulfur metabolism

1.3 Lipid Metabolism

- Fatty acid biosynthesis
- Fatty acid elongation in mitochondria
- Fatty acid metabolism
- Synthesis and degradation of ketone bodies
- Biosynthesis of steroids
- Bile acid biosynthesis
- C21-Steroid hormone metabolism
- Androgen and estrogen metabolism
- Glycerolipid metabolism
- Glycerophospholipid metabolism
- Ether lipid metabolism
- Sphingolipid metabolism
- Arachidonic acid metabolism
- Linoleic acid metabolism
- alpha-Linolenic acid metabolism
- Biosynthesis of unsaturated fatty acids

1.4 Nucleotide Metabolism

- Purine metabolism
- Pyrimidine metabolism

1.5 Amino Acid Metabolism

- Glutamate metabolism

1.5 Amino Acid Metabolism

- Glutamate metabolism

- Alanine and aspartate metabolism
- Glycine, serine and threonine metabolism
- Methionine metabolism
- Cysteine metabolism
- Valine, leucine and isoleucine degradation
- Valine, leucine and isoleucine biosynthesis
- Lysine biosynthesis
- Lysine degradation
- Arginine and proline metabolism
- Histidine metabolism
- Tyrosine metabolism
- Phenylalanine metabolism
- Tryptophan metabolism
- Phenylalanine, tyrosine and tryptophan biosynthesis
- Urea cycle and metabolism of amino groups

1.6 Metabolism of Other Amino Acids

- beta-Alanine metabolism
- Taurine and hypotaurine metabolism
- Aminophosphonate metabolism
- Selenoamino acid metabolism
- Cyanoamino acid metabolism
- D-Glutamine and D-glutamate metabolism
- D-Arginine and D-ornithine metabolism
- D-Alanine metabolism
- Glutathione metabolism

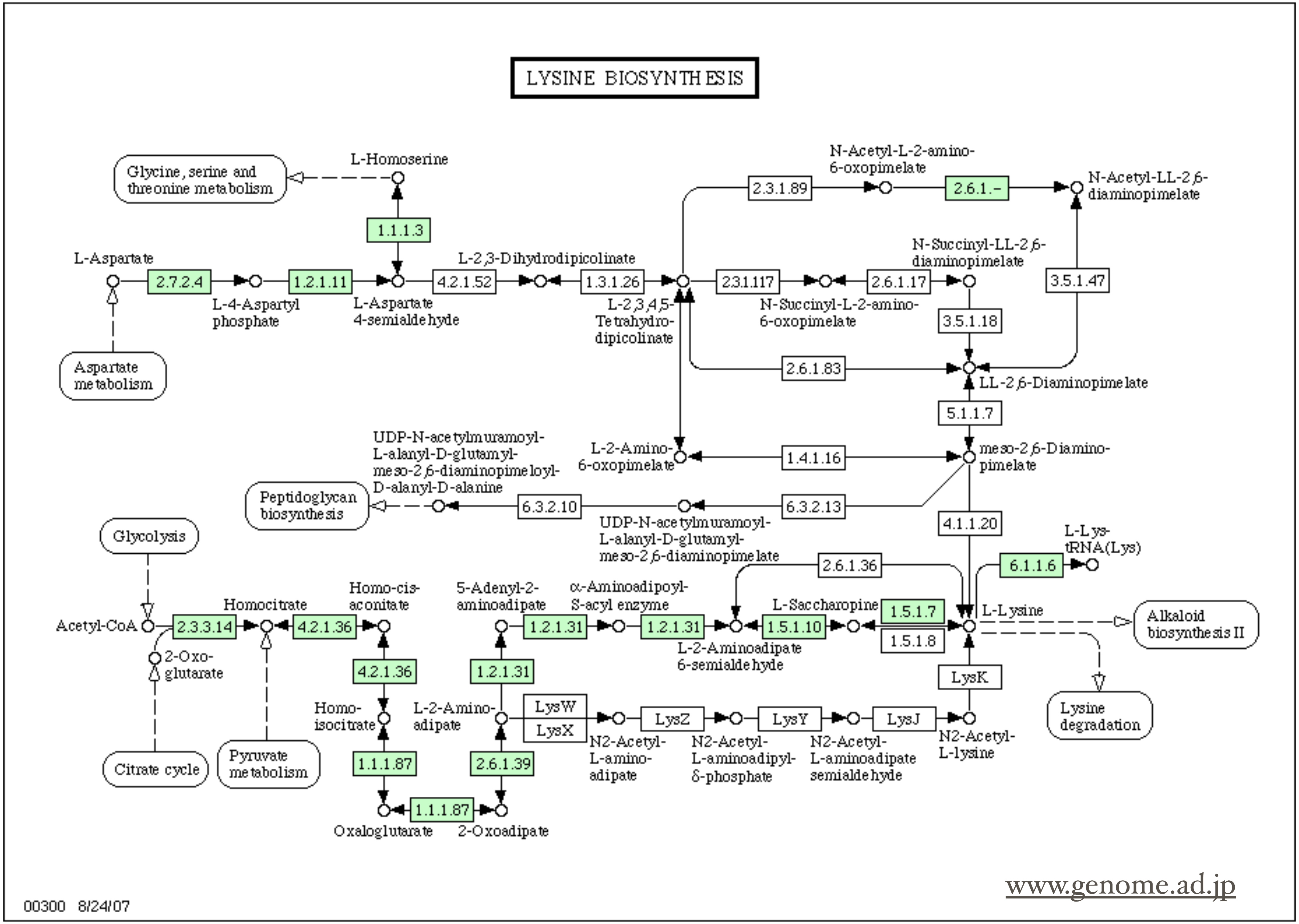
1.7 Glycan Biosynthesis and Metabolism

- N-Glycan biosynthesis
- High-mannose type N-glycan biosynthesis
- N-Glycan degradation
- O-Glycan biosynthesis
- Chondroitin sulfate biosynthesis
- Heparan sulfate biosynthesis
- Keratan sulfate biosynthesis
- Glycosaminoglycan degradation
- Lipopolysaccharide biosynthesis
- Peptidoglycan biosynthesis
- Glycosylphosphatidylinositol(GPI)-anchor biosynthesis
- Glycosphingolipid biosynthesis - lactoseries
- Glycosphingolipid biosynthesis - neo-lactoseries
- Glycosphingolipid biosynthesis - globoseries
- Glycosphingolipid biosynthesis - ganglioseries
- Glycan structures - biosynthesis 1
- Glycan structures - biosynthesis 2
- Glycan structures - degradation

1.8 Biosynthesis of Polyketides and Nonribosomal Peptides

- Type I polyketide structures
- Biosynthesis of 12-, 14- and 16-membered macrolides
- Biosynthesis of ansamycins
- Biosynthesis of type II polyketide backbone
- Biosynthesis of type II polyketide products

KEGG PATHWAY



MIPS has annotation terms organized in trees

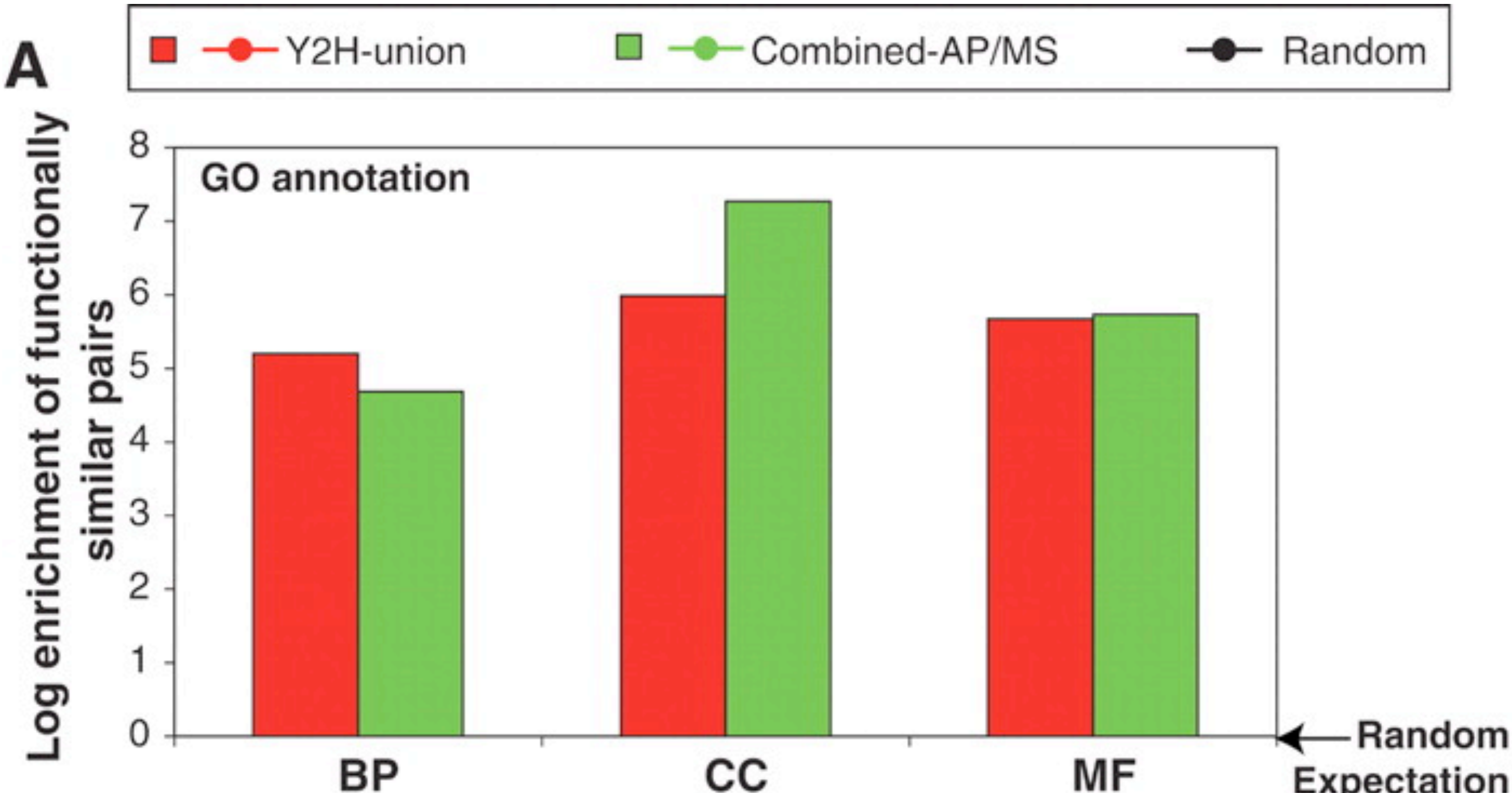
- Function Catalog (FunCat): a collection of functions and biological processes organized as a tree.
- Manually annotated protein complexes (e.g. at left)
  - also organized as a tree

Complex	Proteins
20 2-oxoglutarate dehydrogenase	3
40 Alpha-agglutinin anchor	2
60 Anaphase promoting complex (APC)	11
70 Anthranilate synthase	2
75 Arginase	1
80 Arginine-specific carbamoylphosphate synthase	2
90 Assembly complexes	7
100 Calcineurin B	3
110 cAMP-dependent protein kinase	4
120 Casein kinase	8
123 Catalase	2
125 Cell cycle checkpoint complexes	2
130 Chaperonine containing T-complex TRiC (TCP RING Complex)	8
132 CTP synthetase	1
133 Cyclin-CDK (Cyclin-dependent kinases) complexes	25
140 Cytoskeleton	73
143 D-arabinose dehydrogenase	1
145 delta3-cis-delta2-trans-enoyl-CoA isomerase	1
150 Endonuclease SceI, mitochondrial	1
160 Exocyst complex	7
170 Fatty acid synthetase, cytoplasmic	2
172 Fatty acid synthetase, mitochondrial	1
177 Gim complexes	5
180 Prenyltransferases	6
190 Glucan synthases	5
200 Glycine decarboxylase	4
210 H+-ATPase, plasma mebrane	4
220 H+-transporting ATPase, vacuolar	15
225 Hexokinase 2	1
230 Histone acetyltransferase complexes	19
240 Histone deacetylase complexes	5

Basic Methods for Predicting Function

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

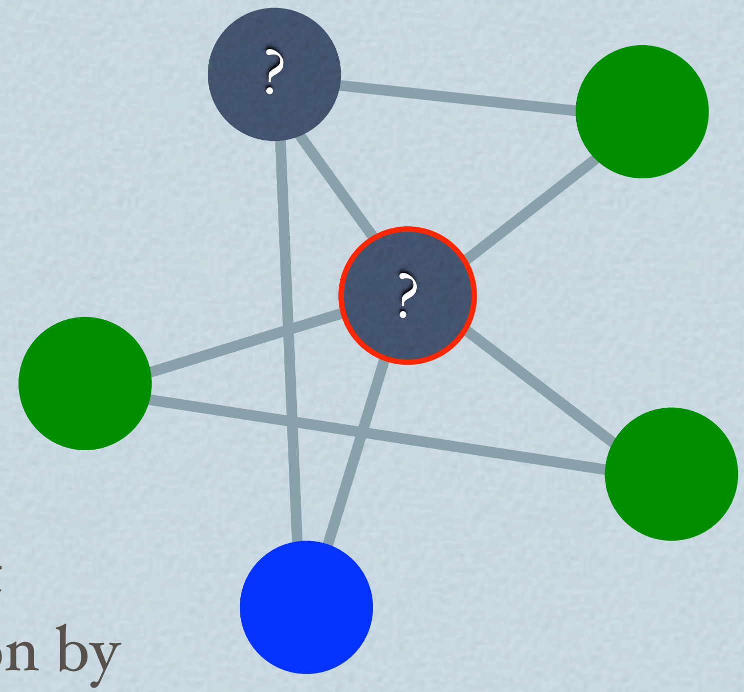
Neighboring Proteins More Likely to Share Function



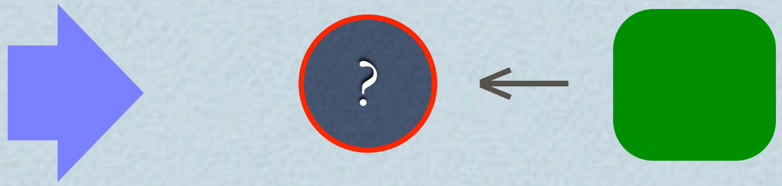
- (Yu et al., 2008)

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*

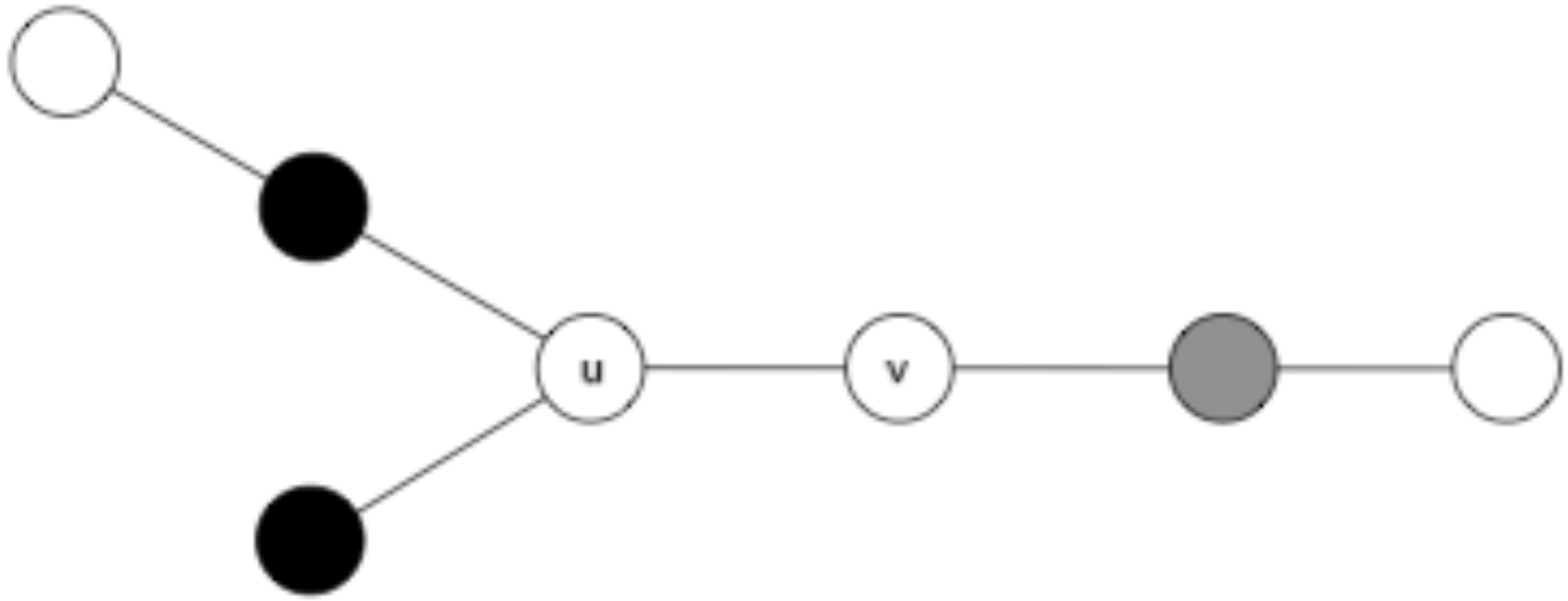
$$\text{Score}(u,r,a) = \frac{(f(u,r,a) - e(u,r,a))^2}{e(u,r,a)}$$

≈ χ_2 statistic measures how surprising it is to see the observed # of proteins annotated with *a* in the neighborhood

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

Problems with neighborhood

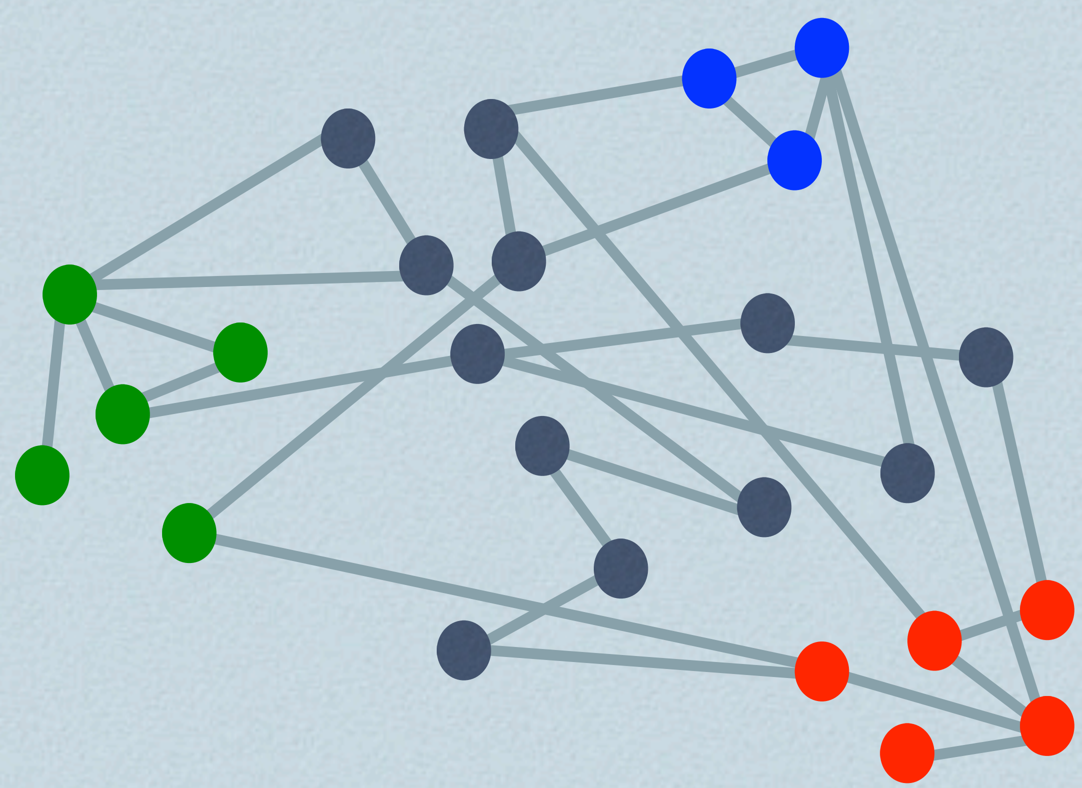
- Neighborhood with radius 2 gives the same scores for black and gray functions to nodes u and v:



(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: simulated annealing; integer programming



Integer Programming

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

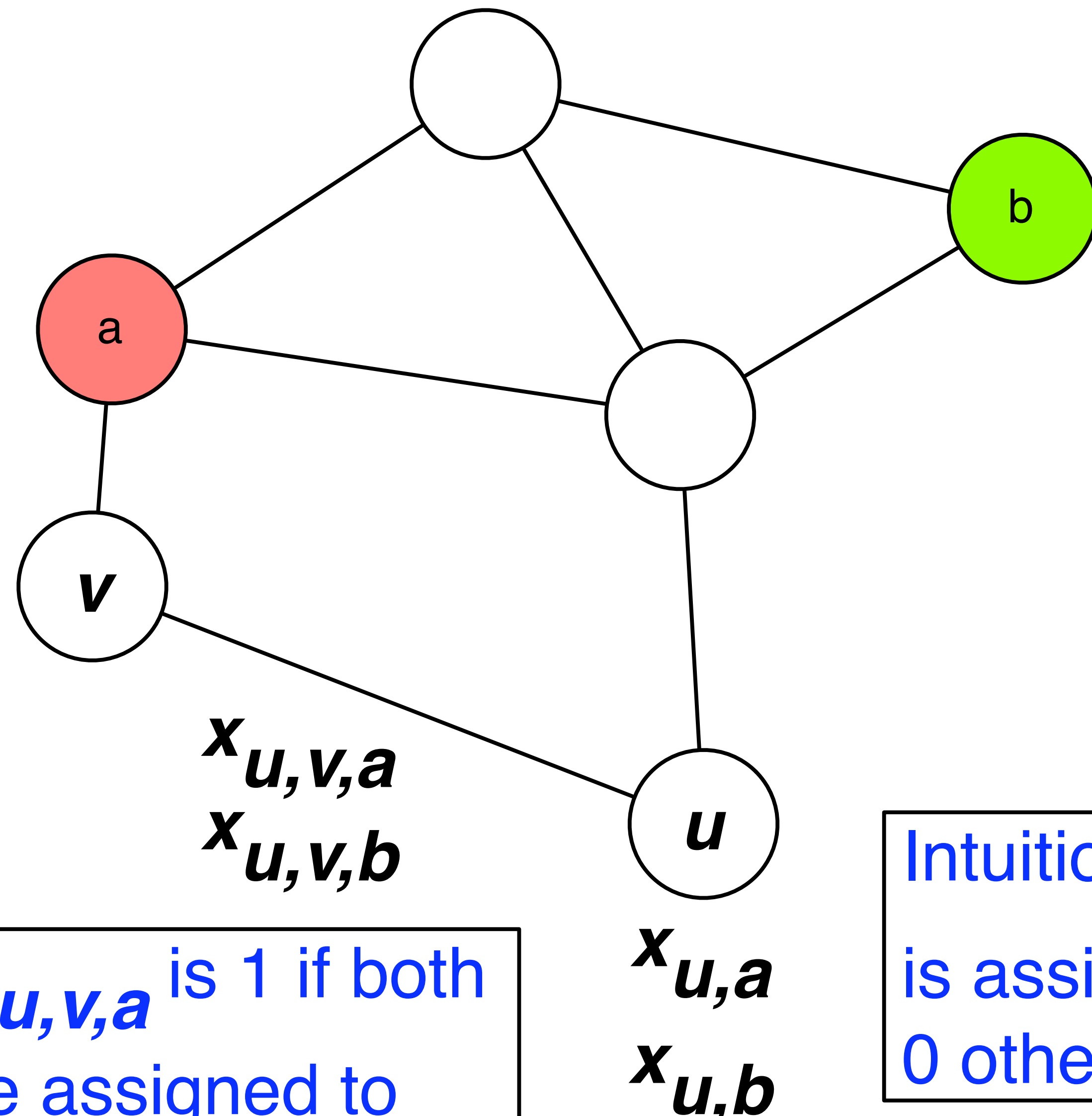
IP :=

- 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

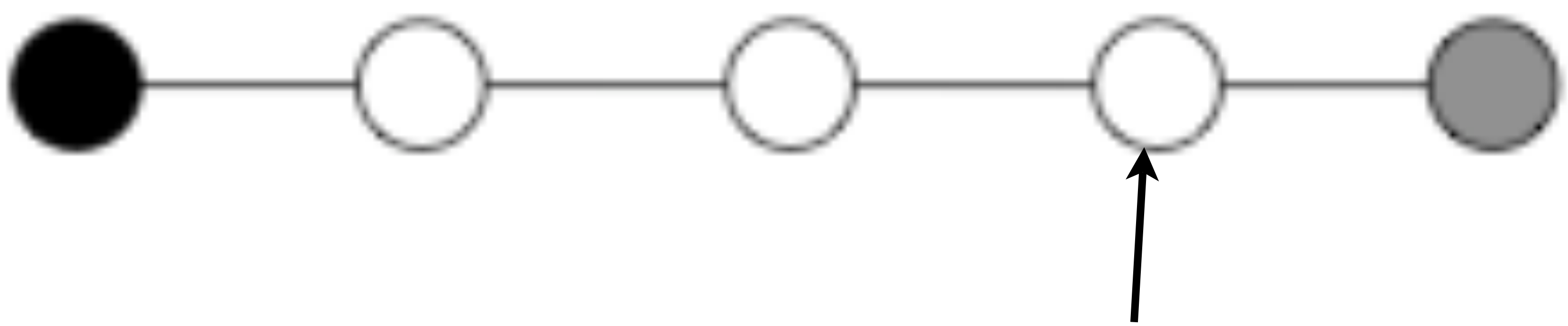
$$x_{u,a} = 1 \text{ if } a \in \text{annot}(u)$$

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

Fix variables for nodes with known annotations.

Problem with Simple Cut Approaches

- Every cut is equally likely:

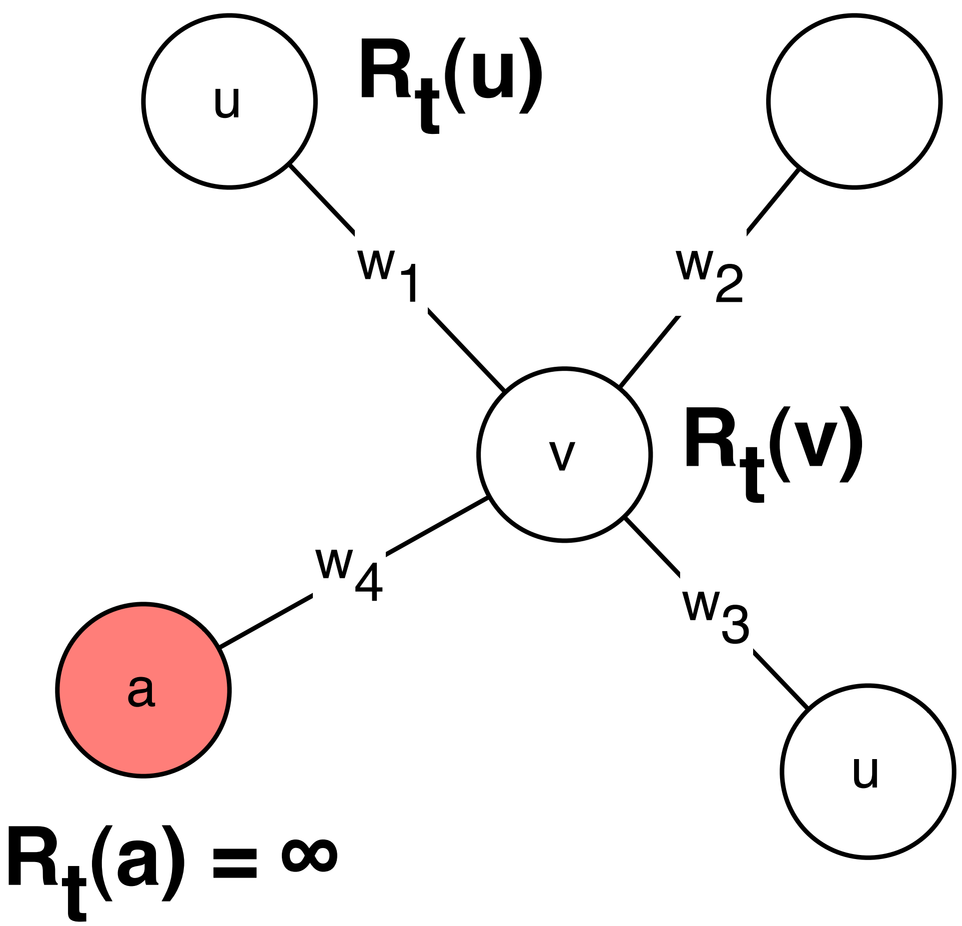


but this node is
more likely to be
grey than black

(Nabieva, Singh, 2008)

Functional Flow (Nabieva et al.)

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

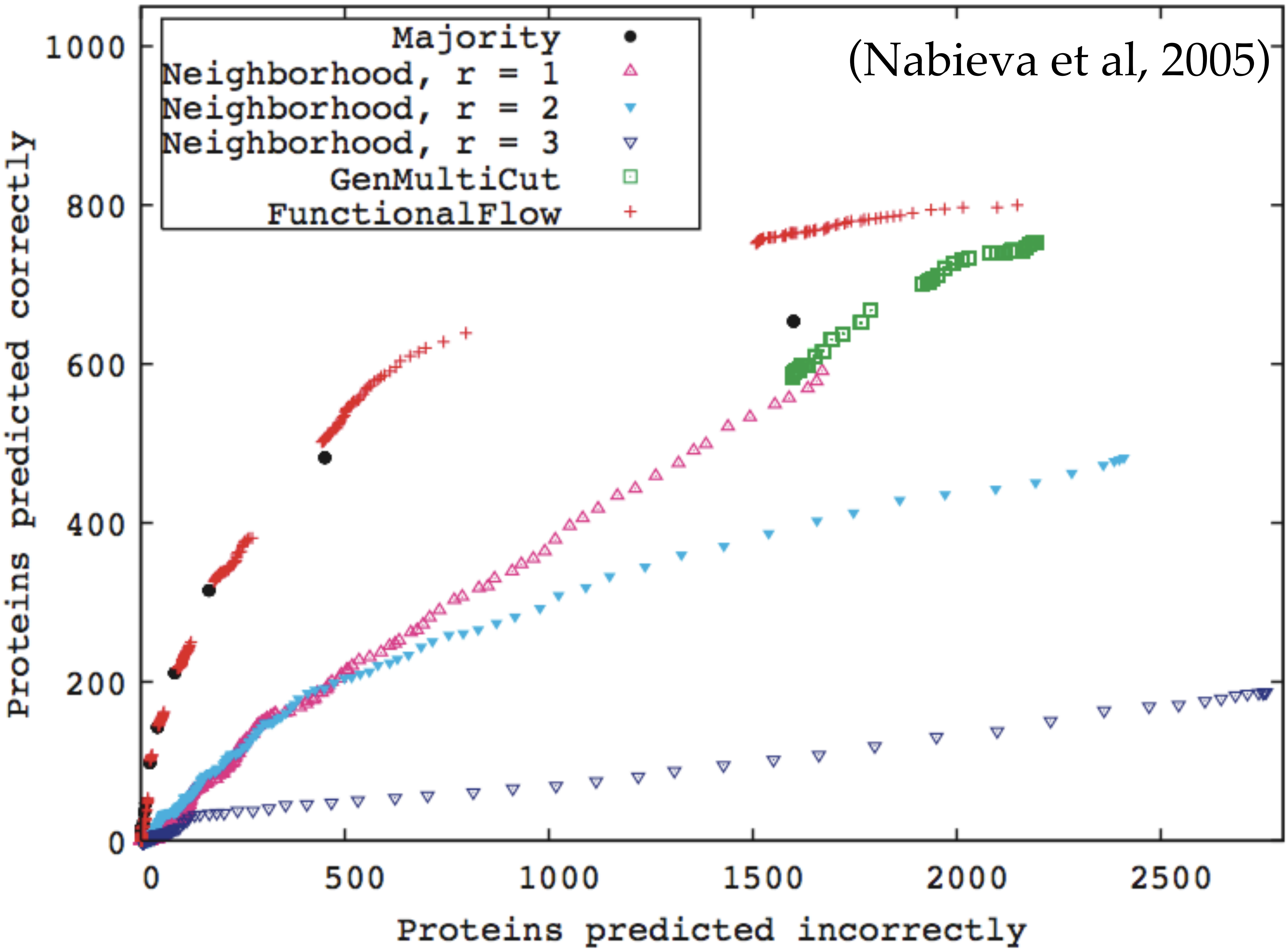


At every time step, water flows "downhill" from the more filled reservoir to the more empty reservoir, up to the capacity of the edge.

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



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. Can obtain multiple solutions in 2 ways:
  - Random perturbation of weights
  - Solving successive problems with additional constraints.
- “Functional flow” is an embodiment of a general technique: “information” being passed along the network.