

V7 – Biological PPI Networks

- graph bisection (-> communities)
- are biological networks really scale-free?
 - network growth
 - functional annotation in the network

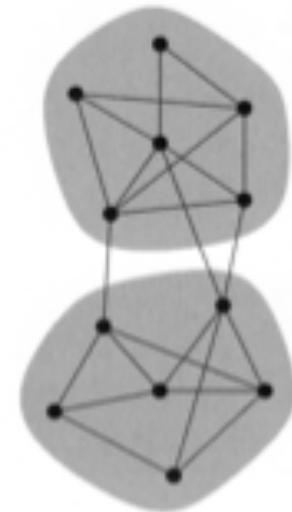
Mon, Nov 14, 2016

Modularity: an example of graph partitioning

The simplest graph partitioning problem is the division of a network into just 2 parts. This is called **graph bisection**.

If we can divide a network into 2 parts, we can also divide it further by dividing one or both of these parts ...

graph bisection problem: divide the vertices of a network into 2 non-overlapping groups of given sizes such that the **number of edges** running **between** vertices in **different groups is minimized.**



The number of edges between groups is called the **cut size**.

In principle, one could simply look through all possible divisions of the network into 2 parts and choose the one with smallest cut size.

Algorithms for graph partitioning

But this exhaustive search is prohibitively expensive!

Given a network of n vertices. There are $\frac{n!}{n_1!n_2!}$ different ways of dividing it into 2 groups of n_1 and n_2 vertices.

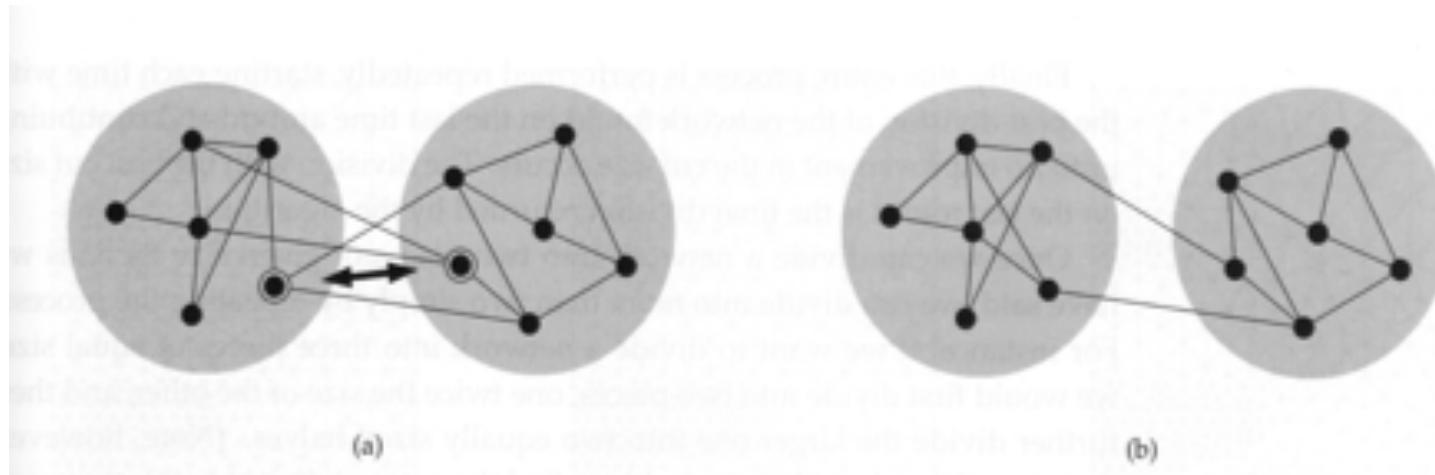
The amount of time to look through all these divisions will go up roughly exponentially with the size of the system.

Only values of up to $n = 30$ are feasible with today's computers.

In computer science, either an algorithm can be clever and run quickly, but will fail to provide the optimal answer in some (or perhaps in many) cases, or it will always find the optimal answer, but takes an impractical length of time to do so.

The Kernighan-Lin algorithm

This algorithm proposed by Brian Kernighan and Shen Lin in 1970 is one of the simplest and best known **heuristic** algorithms for the graph bisection problem. (Kernighan is also one of the developers of the C language).



- (a) The algorithm starts with any division of the vertices of a network into two groups (shaded) and then searches for pairs of vertices, such as the pair highlighted here, whose **interchange** would **reduce** the **cut size** between the groups.
- (b) The same network after interchange of the 2 vertices.

The Kernighan-Lin algorithm

- (1) Divide the vertices of a given network into 2 groups (e.g. randomly).
- (2) For each pair (i,j) of vertices, where i belongs to the first group and j to the second group, calculate **how much** the **cut size** between the groups would **change** if i and j were interchanged between the groups.
- (3) Find the pair that reduces the cut size by the largest amount and swap the vertices.
If no pair reduces it, find the pair that increases it by the smallest amount.

Repeat this process, but with the important restriction that **each vertex** in the network can **only** be **moved once**.

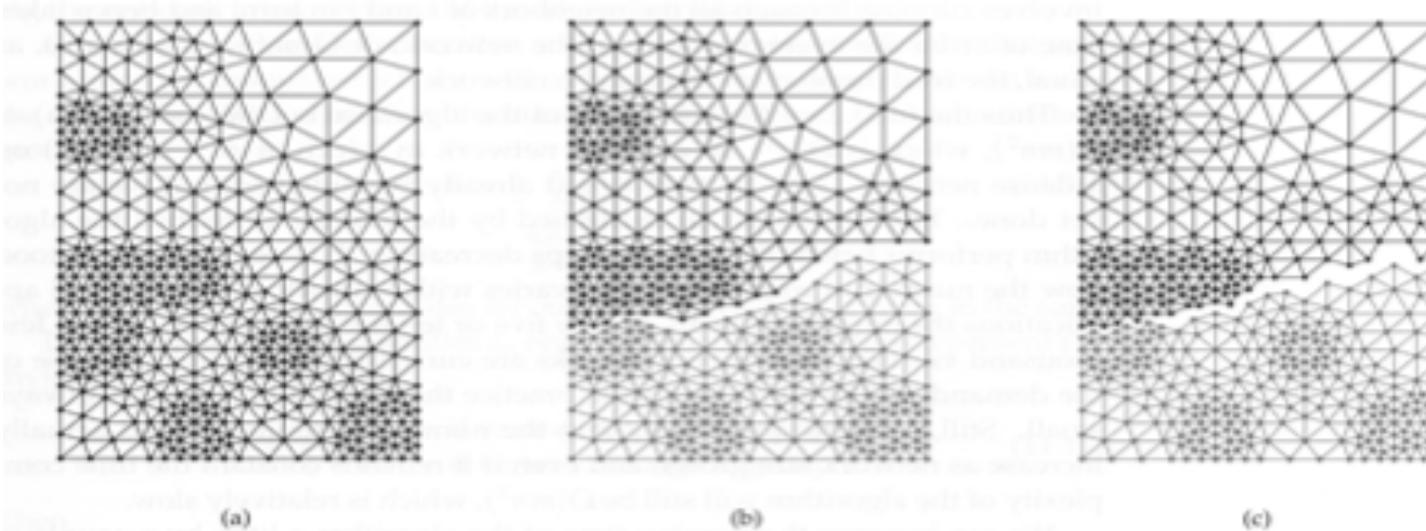
Stop when there is no pair of vertices left that can be swapped.

The Kernighan-Lin algorithm (II)

- (3) Go back through every state that the network passed through during the swapping procedure and choose among them the state in which the cut size takes its smallest value.
- (4) Perform this entire process repeatedly, starting each time with the best division of the network found in the last round.
- (5) Stop when no improvement on the cut size occurs.

Note that if the initial assignment of vertices to groups is done randomly, the Kernighan-Lin algorithm may give (slightly) different answers when it is run twice on the same network.

The Kernighan-Lin algorithm (II)



- (a) A mesh network of 547 vertices of the kind commonly used in finite element analysis.
- (b) The best division found by the Kernighan-Lin algorithm when the task is to split the network into 2 groups of almost equal size.
This division involves cutting 40 edges in this mesh network and gives parts of 273 and 274 vertices.
- (c) The best division found by spectral partitioning (alternative method).

Runtime of the Kernighan-Lin algorithm

The number of swaps performed during one round of the algorithm is equal to the smaller of the sizes of the two groups $\in [0, n / 2]$.

→ in the worst case, there are $O(n)$ swaps.

For each swap, we have to examine all pairs of vertices in different groups to determine how the cut size would be affected if the pair was swapped.

At most (if both groups have the same size),
there are $n / 2 \times n / 2 = n^2 / 4$ such pairs, which is $O(n^2)$.

Runtime of the Kernighan-Lin algorithm (ii)

When a vertex i moves from one group to the other group, any edges connecting it to vertices in its current group become edges between groups after the swap.

Let us suppose that there are k_i^{same} such edges.

Similarly, any edges that i has to vertices in the other group, (say k_i^{other} ones) become within-group edges after the swap.

There is one exception. If i is being swapped with vertex j and they are connected by an edge, then the edge is still between the groups after the swap

→ the change in the cut size due to the movement of i is $-(k_i^{\text{other}} - k_i^{\text{same}} - A_{ij})$

A similar expression applies for vertex j .

→ the total change in cut size due to the swap is

$$-(k_i^{\text{other}} - k_i^{\text{same}} + k_j^{\text{other}} - k_j^{\text{same}} - 2A_{ij})$$

Runtime of the Kernighan-Lin algorithm (iii)

For a network stored in adjacency list form, the evaluation of this expression involves running through all the neighbors of i and j in turn, and hence takes time on the order of the average degree in the network, or $O(m/n)$ with m edges in the network.

→ the total running time is $O(n \times n^2 \times m/n) = O(mn^2)$.

For a sparse network with $m \propto n$, this is $O(n^3)$.

For a dense network (with $m \rightarrow \frac{n(n-1)}{2}$), this is $O(n^4)$.

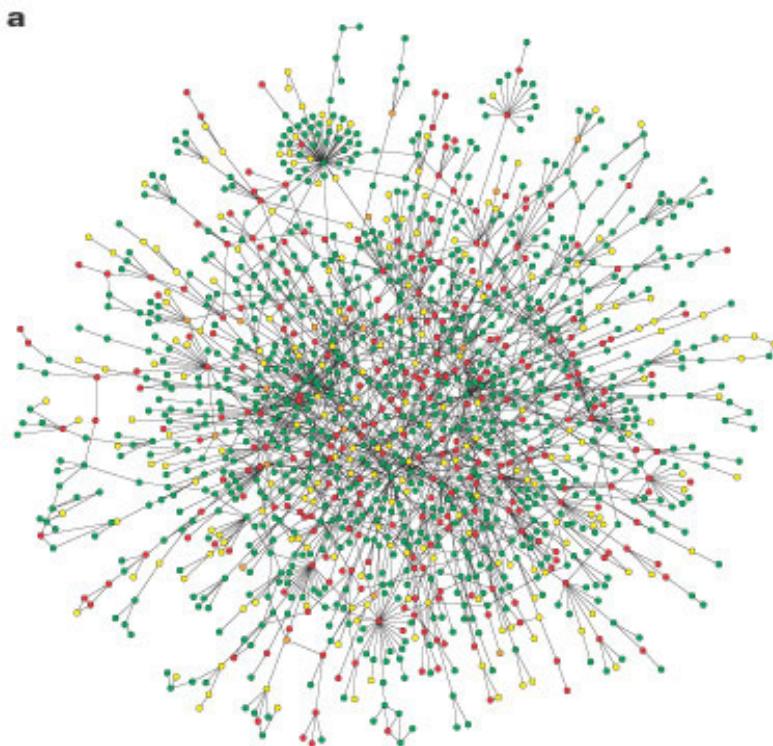
This time still needs to be **multiplied** by the **number of rounds** the algorithm is run before the cut size stops decreasing.

For networks up to a few 1000 of vertices, this number may be between 5 and 10.

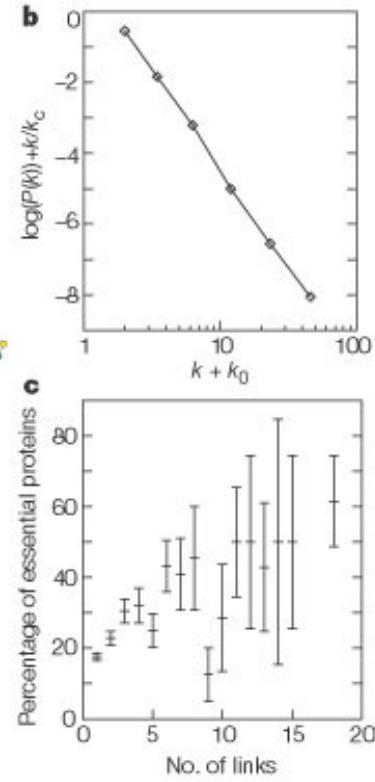
Lethality and centrality in protein networks

The most highly connected proteins in the cell are the most important for its survival.

Jeong, Mason, Barabási, Oltvai, *Nature* 411 (2001) 41



largest cluster of the yeast proteome (at 2001)



→ "PPI networks
apparently are
scale-free..."

"**Are**" they scale-free
or
"Do they **look** like"
scale-free???

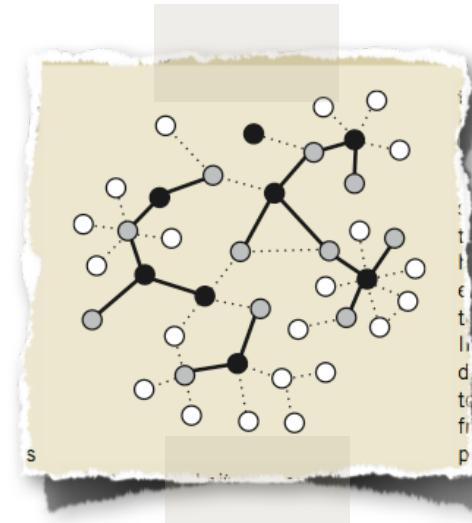
Effect of sampling on topology predictions of protein-protein interaction networks

Jing-Dong J Han¹⁻³, Denis Dupuy^{1,3}, Nicolas Bertin¹, Michael E Cusick¹ & Marc Vidal¹

Nature Biotech **23** (2005) 839

Generate networks of various types,
sample sparsely from them
→ determine degree distribution

- Random (ER / Erdös-Renyi) → $P(k) = \text{Poisson}$
- Exponential (EX) → $P(k) \sim \exp[-k]$
- scale-free / power-law (PL) → $P(k) \sim k^{-\gamma}$
- $P(k) = \text{truncated normal distribution (TN)}$



Partial Sampling

Estimated for yeast: 6000 proteins, 30000 interactions

Table 1 Topological properties of interactome maps

Data set	Ito <i>et al.</i> (yeast)	Uetz <i>et al.</i> (yeast)	Ito-Uetz combined	Li <i>et al.</i> (worm)	Giot <i>et al.</i> (fly)	Minimum value	Maximum value
Total number of nodes	797	1,005	1,417	1,415	4,651	797	4,651
Nodes in main component	417 (52%)	473 (47%)	970 (68%)	1,260 (89%)	3,039 (65%)	47%	89%
Total number of interactions	806	948	1,520	2,135	4,787	806	4,787
Interactions in main component	544	558	1,229	2,038	3,715	544	3,715
R-square	0.843	0.954	0.899	0.885	0.91	0.843	0.954
γ	-1.82	-2.42	-1.91	-1.59	-2.75	-2.75	-1.59
$\langle k \rangle$	1.96	1.84	2.15	2.98	2.04	1.84	2.98
Average clustering coefficient	0.2	0.11	0.09	0.09	0.06	0.06	0.2
Number of network components	143	177	160	70	591	70	591
Average component size	5.6	5.7	8.9	20.2	7.9	5.6	20.2
Characteristic path length	6.14	7.48	6.55	4.91	9.43	4.91	9.43
Number of baits	455	512	827	502	2,820	455	2,820

The linear regression R-square measures the linearity between $\log(n(k))$ and $\log(k)$ i.e. the fit to a power-law distribution. γ is the exponent of the power law distribution formula that best fits the observed distribution. $\langle k \rangle$ is the average number of interactions per protein observed in the network. For the Ito, Li and Giot data sets only the high confidence interactions were considered (core).

Y2H experiments **detected** only 3...9% of the complete interactome!

R square

Given: a data set with n values y_1, \dots, y_n and
a set of fitted / predicted / modelled) values f_1, \dots, f_n e.g. from linear regression.

We call their difference **residuals** $e_i = y_i - f_i$

and the mean value $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$

The total sum of squares (proportional to the variance of the data) is:

$$SS_{\text{tot}} = \sum_i (y_i - \bar{y})^2,$$

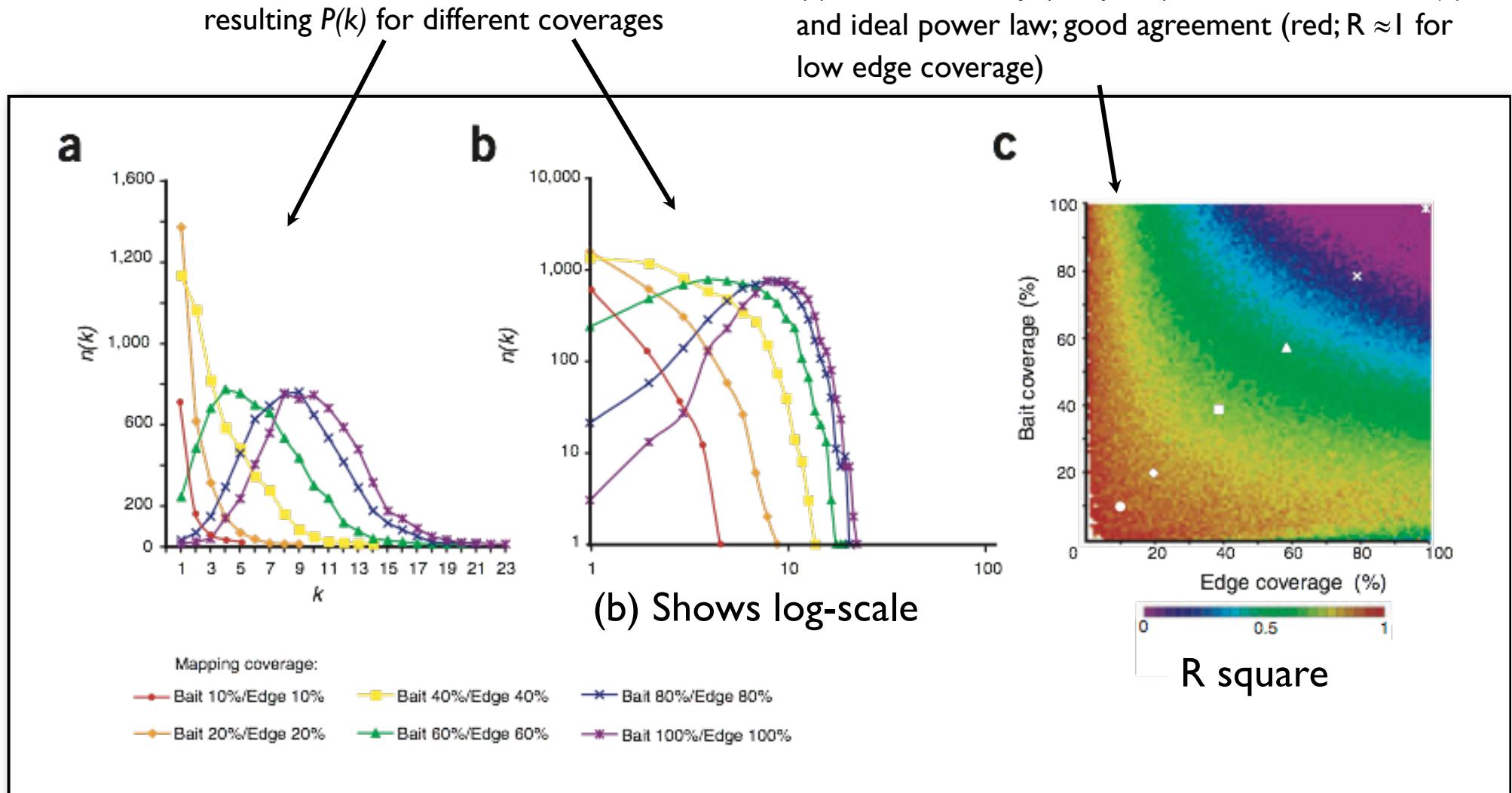
The sum of squares of residuals is:

$$SS_{\text{res}} = \sum_i (y_i - f_i)^2 = \sum_i e_i^2$$

The **coefficient of determination**, R^2 or r^2 is often defined as:

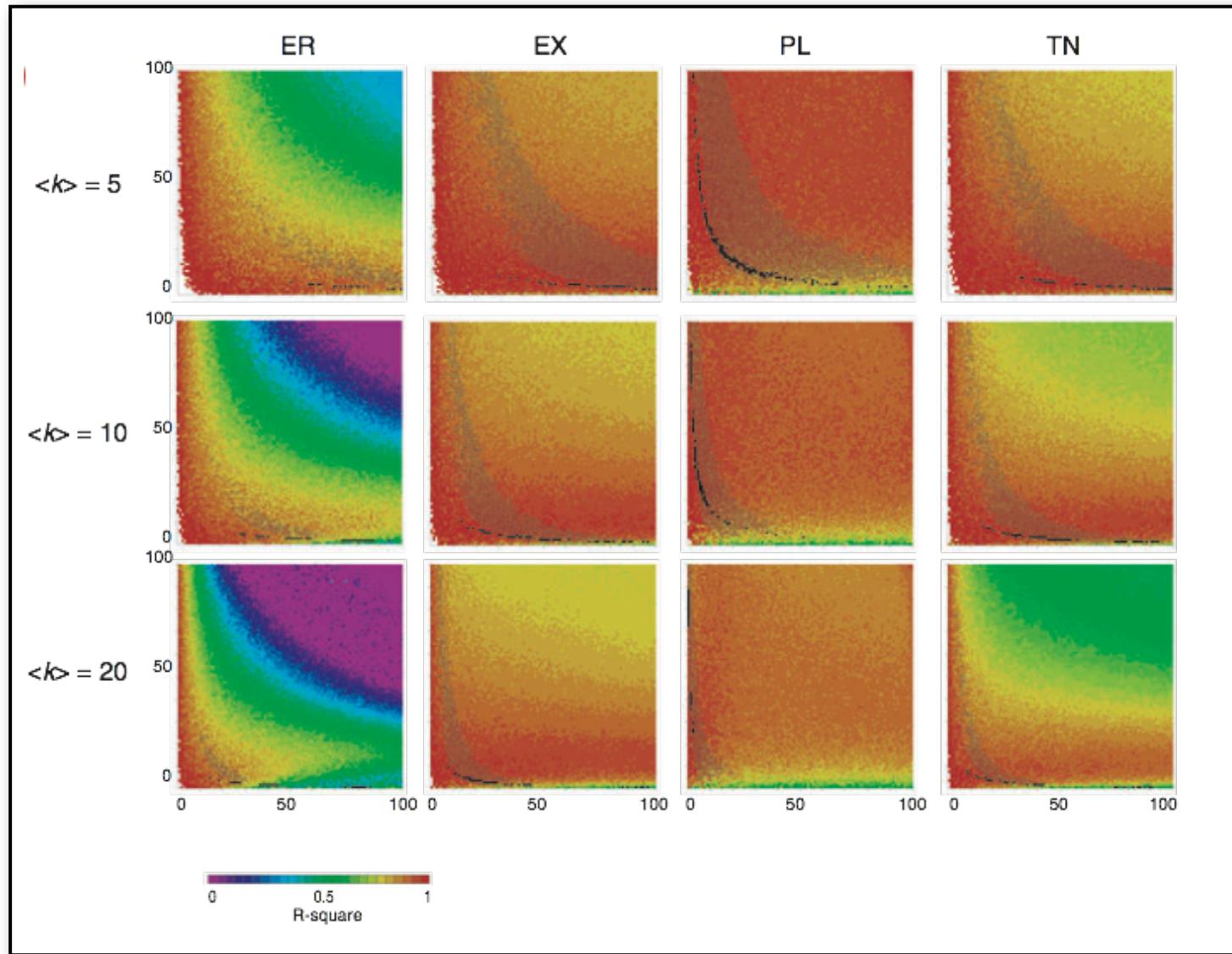
$$R^2 \equiv 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}.$$

Sparsely Sampled random (ER) Network

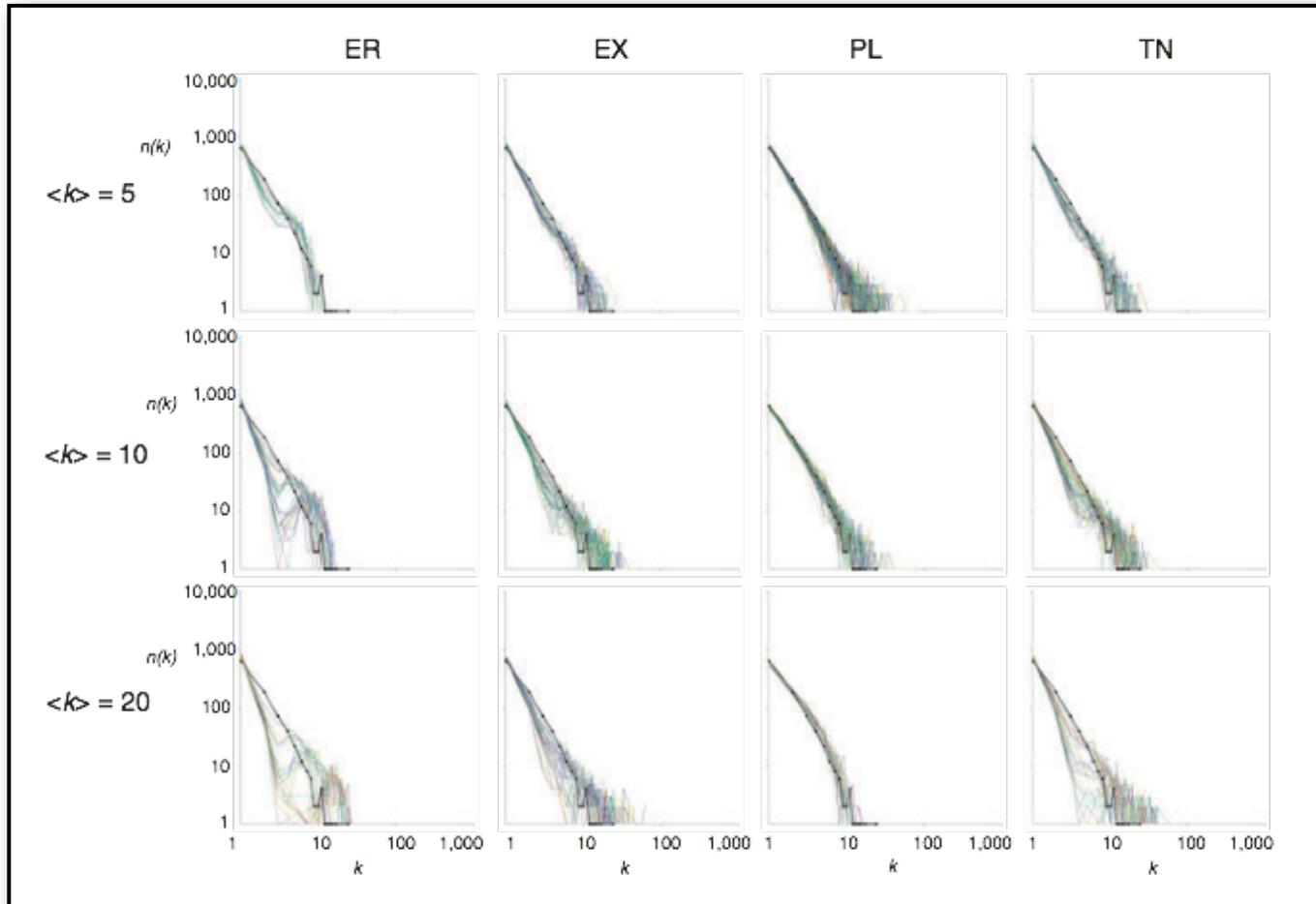


→ for **sparse** sampling (10-20%), even an ER network
"looks" **scale-free** (when only $P(k)$ is considered)

Anything Goes – different topologies



Compare to Uetz et al. data



Uetz et al. data
(solid line) is
compared to
sampled
networks of
similar size.

Sampling density affects observed degree distribution
→ true underlying network cannot be identified from available data

Network Growth Mechanisms

Given: an observed PPI network → how did it grow (evolve)?

Inferring network mechanisms: The *Drosophila melanogaster* protein interaction network

Manuel Middendorf[†], Ety Ziv[‡], and Chris H. Wiggins^{§¶||}

[†]Department of Physics, [‡]College of Physicians and Surgeons, [§]Department of Applied Physics and Applied Mathematics, and ^{¶||}Center for Computational Biology and Bioinformatics, Columbia University, New York, NY 10027

Communicated by Barry H. Honig, Columbia University, New York, NY, December 20, 2004 (received for review September 7, 2004)

PNAS 102 (2005) 3192

Look at **network motifs** (local connectivity):
compare motif distributions from various network prototypes to fly network

Idea: each growth **mechanism** leads to a typical motif **distribution**,
even if global measures are comparable

The Fly Network

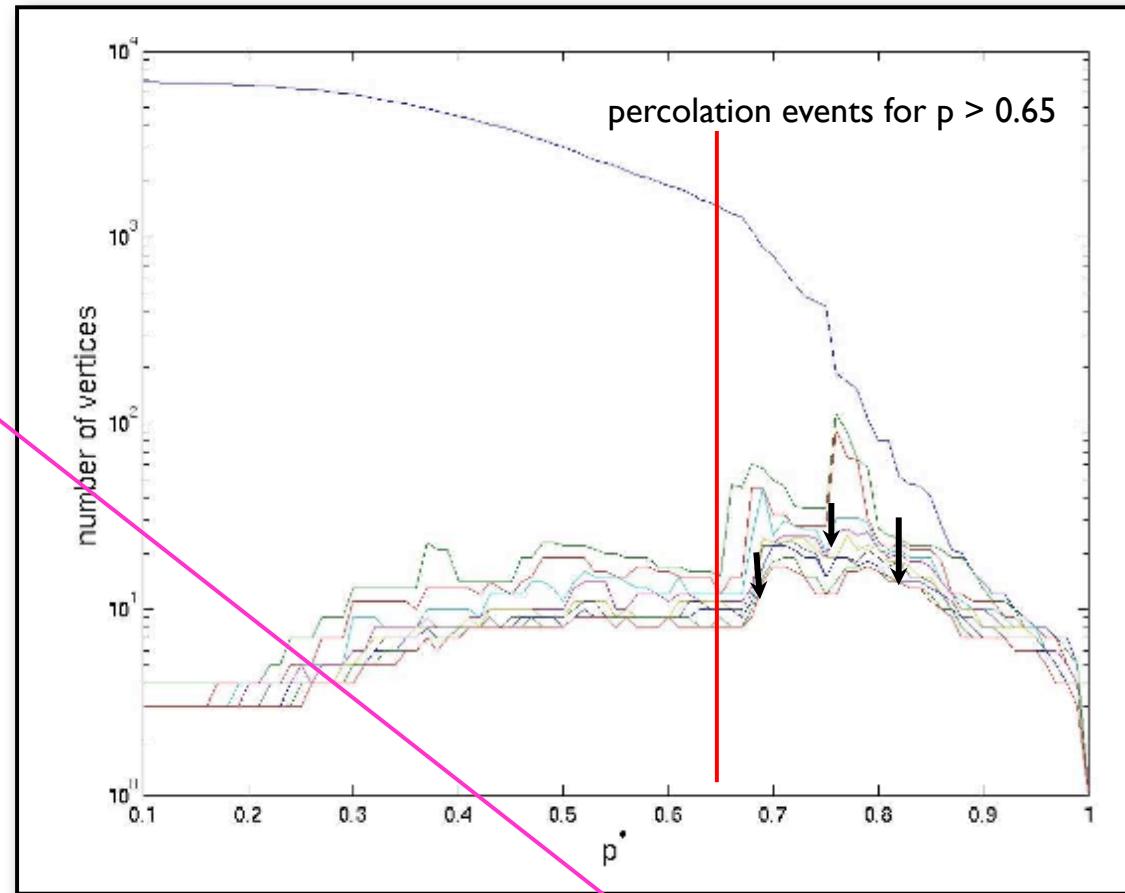
Y2H PPI network for *D. melanogaster* from Giot et al. [Science **302** (2003) 1727]

Giot et al. assigned a confidence score [0, 1] for every observed interaction.

- use only data with $p > 0.65$ (0.5) because ...
- remove self-interactions and isolated nodes

High confidence network with 3359 (4625) nodes and 2795 (4683) edges.

Use prototype networks of same size for training.



Size of largest components. At $p = 0.65$, there is one large component with 1433 nodes and the other 703 components contain at most 15 nodes.

Network subgraphs -> motives

All non-isomorphic subgraphs that can be generated with a walk of length 8



Growth Mechanisms

Generate 1000 networks, each, of the following 7 types
(same size as fly network, undefined parameters were scanned)

DMC Duplication-mutation, preserving complementarity

DMR Duplication with random mutations

RDS Random static networks

RDG Random growing network

LPA Linear preferential attachment network (Albert-Barabasi)

AGV Aging vertices network

SMW Small world network

Growth Type 1: DMC

"Duplication – mutation with preserved complementarity"

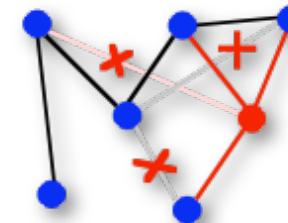
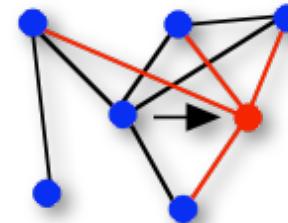
Evolutionary idea: gene **duplication**, followed by a partial **loss** of function of one of the copies, making the other copy essential

Algorithm:

Start from two connected nodes

- duplicate existing node with all interactions
- for all neighbors: delete with probability q_{del} either link from original node **or** from copy

Repeat these steps many (e.g. $N - 2$) times



Growth Type 2: DMR

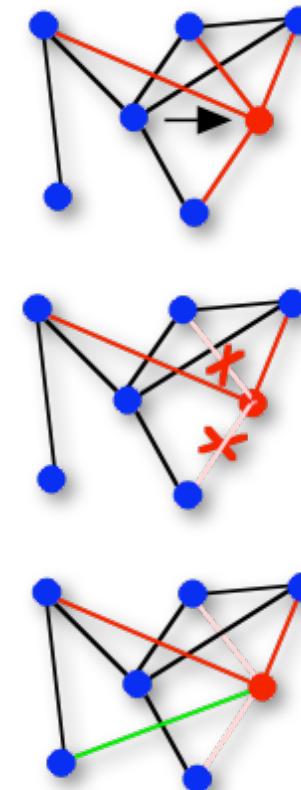
"Duplication with random mutations"

Gene duplication, but no correlation between original and copy
(original unaffected by copy)

Algorithm:

Start growth from five-vertex cycle,
repeat $N - 5$ times:

- duplicate existing node with all interactions
- for all neighbors: delete with probability q_{del}
link from copy
- add **new links** to non-neighbors with
probability q_{new}/n



Growth Types 3–5: RDS, RDG, and LPA

RDS = static random network

Start from N nodes, add L links randomly

RDG = growing random network

Start from small random network, add nodes,
then edges between all existing nodes

LPA = linear preferential attachment

Add new nodes similar to Barabási-Albert algorithm,
but with preference according to $(k_i + \alpha)$, $\alpha = 0\dots 5$
(BA for $\alpha = 0$)

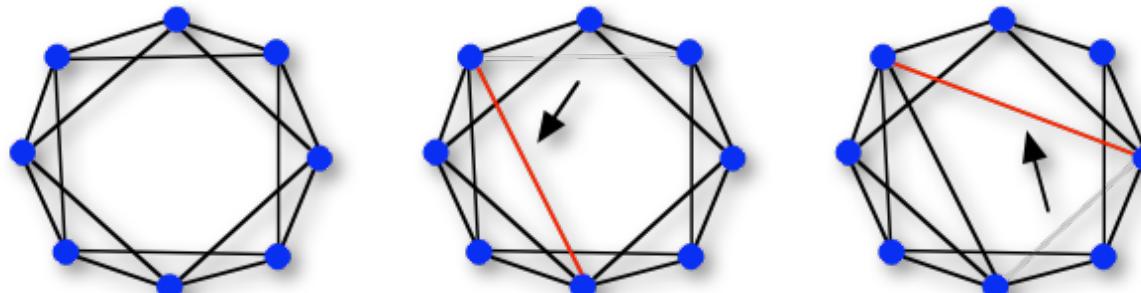
Growth Types 6-7: AGV and SMW

AGV = aging vertices network

Like growing random network,
but preference decreases with age of the node
→ citation network: more recent publications are cited more likely

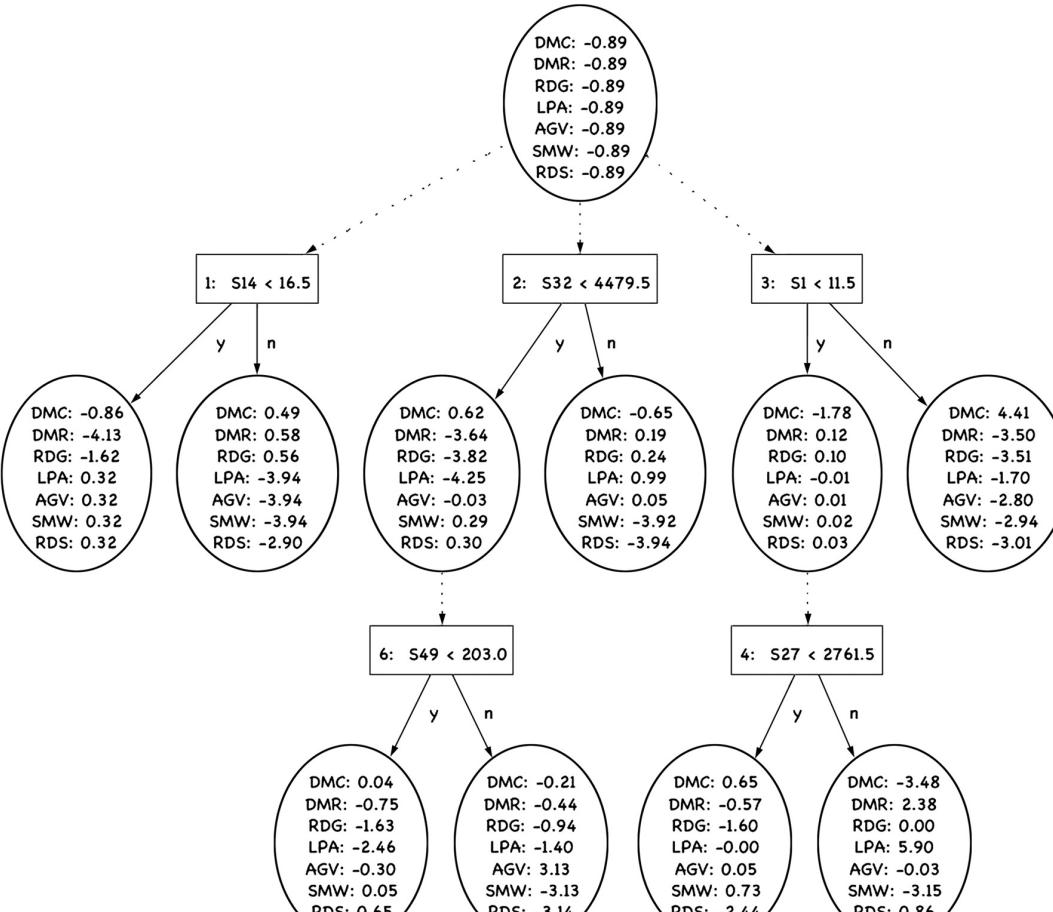
SMW = small world networks, see Watts, Strogatz, *Nature* **363**, 202 (1998)

Randomly rewire regular ring lattice



Alternating Decision Tree Classifier

Trained with the motif counts from 1000 networks of each of the 7 types
→ prototypes are well separated and can be reliably classified

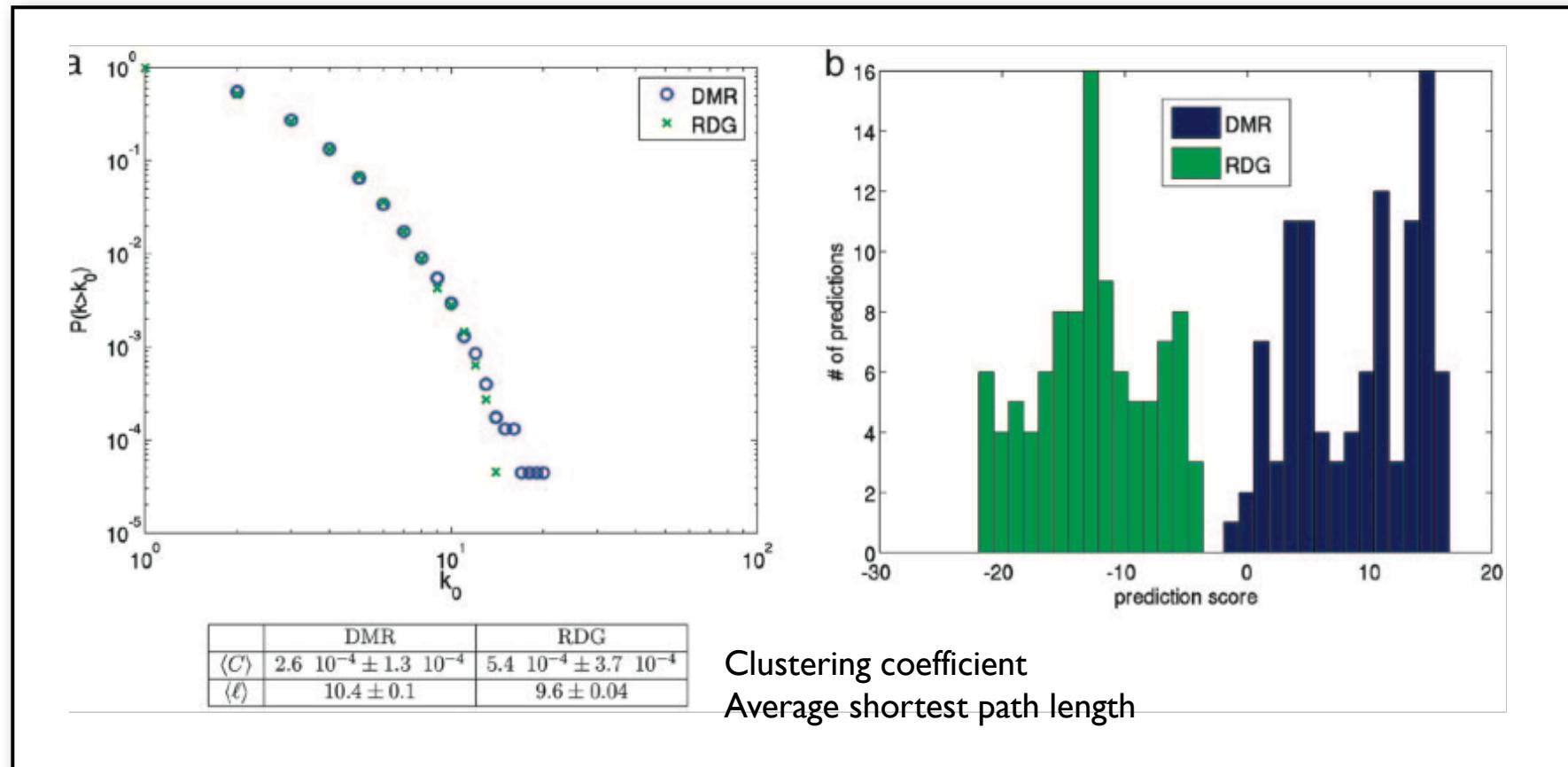


Decision nodes count
occurrence of subgraphs

Prediction accuracy for networks
similar to fly network with $p = 0.5$:

Truth	Prediction						
	DMR	DMC	AGV	LPA	SMW	RDS	RDG
DMR	99.3	0.0	0.0	0.0	0.0	0.1	0.6
DMC	0.0	99.7	0.0	0.0	0.3	0.0	0.0
AGV	0.0	0.1	84.7	13.5	1.2	0.5	0.0
LPA	0.0	0.0	10.3	89.6	0.0	0.0	0.1
SMW	0.0	0.0	0.6	0.0	99.0	0.4	0.0
RDS	0.0	0.0	0.2	0.0	0.8	99.0	0.0
RDG	0.9	0.0	0.0	0.1	0.0	0.0	99.0

Are the generated networks different?



Example: DMR vs. RDG: Similar global parameters $\langle C \rangle$ and $\langle l \rangle$ (left),
but different counts of the network motifs (right)

-> networks can (only) be perfectly separated by motif-based classifier

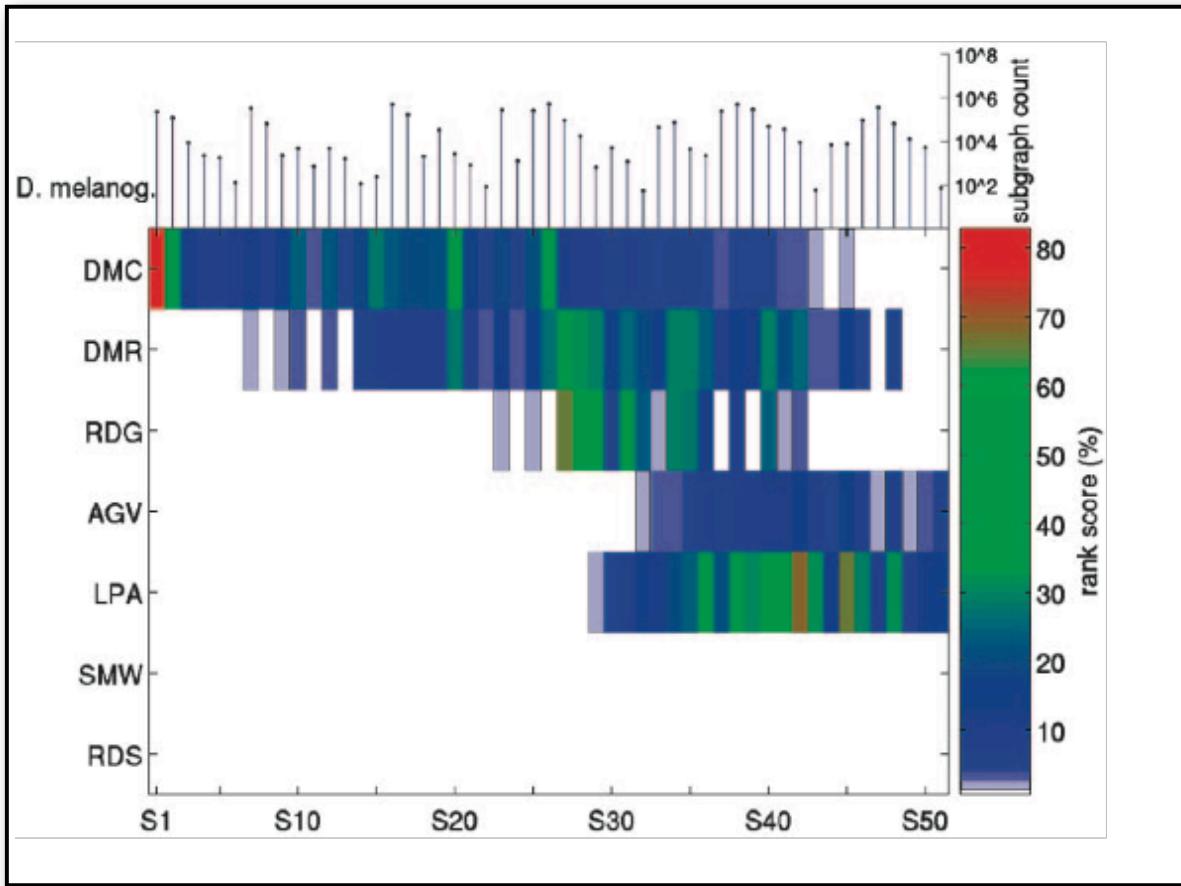
How Did the Fly Evolve?

Rank	Eight-step subgraphs ($p^* = 0.65$)		Subgraphs with up to seven edges ($p^* = 0.65$)		Eight-step subgraphs ($p^* = 0.5$)	
	Class	Score	Class	Score	Class	Score
1	DMC	8.2 ± 1.0	DMC	8.6 ± 1.1	DMC	0.8 ± 2.9
2	DMR	-6.8 ± 0.9	DMR	-6.1 ± 1.7	DMR	-2.1 ± 2.0
3	RDG	-9.5 ± 2.3	RDG	-9.3 ± 1.6	AGV	-3.1 ± 2.2
4	AGV	-10.6 ± 4.2	AGV	-11.5 ± 4.1	LPA	-10.1 ± 3.1
5	LPA	-16.5 ± 3.4	LPA	-14.3 ± 3.2	SMW	-20.6 ± 1.9
6	SMW	-18.9 ± 0.7	SMW	-18.3 ± 1.9	RDS	-22.3 ± 1.7
7	RDS	-19.1 ± 2.3	RDS	-19.9 ± 1.5	RDG	-22.5 ± 4.7

Drosophila is consistently (independently of the cut-off in subgraph size) classified as a DMC network, with an especially strong prediction for a confidence threshold of $p^* = 0.65$.

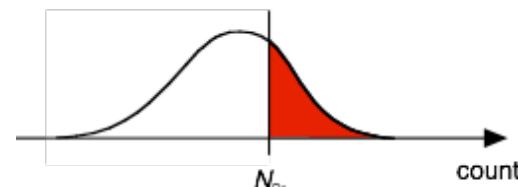
- Best overlap with DMC (Duplication-mutation, preserved complementarity)
- Scale-free (LPA) or random networks (RDS/RDG) are very unlikely

Motif Count Frequencies



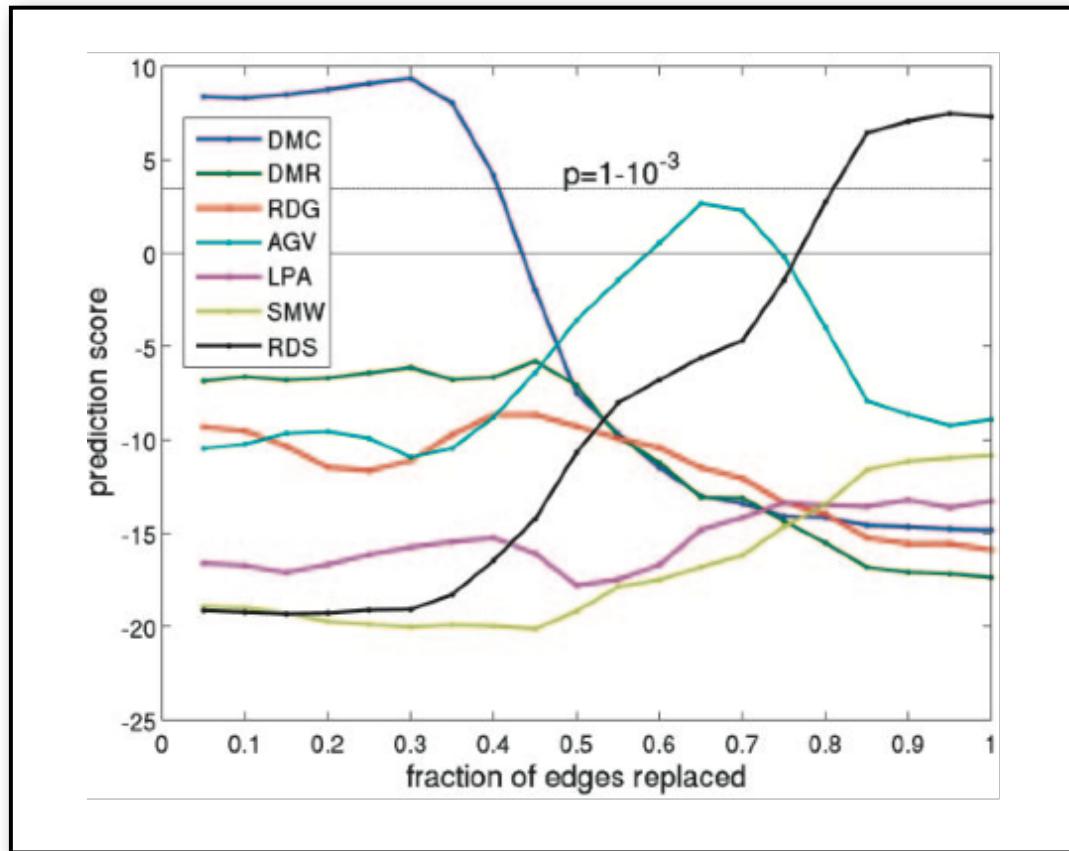
-> DMC and DMR networks contain most subgraphs in similar amount as fly network (top).

rank score: fraction of test networks with a higher count than Drosophila
(50% = same count as fly on avg.)



Experimental Errors?

Randomly replace edges in **fly** network and **classify** again:



→ Classification **unchanged** for $\leq 30\%$ incorrect edges,
at higher values RDS takes over (as to be expected)

Summary (I)

Sampling matters!

→ "Scale-free" $P(k)$ is obtained by sparse sampling from many network types

Test different **hypotheses** for

- **global** features

- depends on unknown parameters and sampling
 - no clear statement possible

- **local** features (motifs)

- are better preserved
 - DMC best among tested prototypes

What Does a Protein Do?



The Comprehensive Enzyme Information System



TU
Braunschweig
Dept. of
Bioinformatics

[EC] Explorer [SEARCH][BROWSE]

- ❑ 1 Oxidoreductases (4042 organisms)
- ❑ 2 Transferases (3198 organisms)
 - ❑ 2.1 Transferring one-carbon groups (615 organisms)
 - ❑ 2.1.1 Methyltransferases (514 organisms)
 - ❑ 2.1.2 Hydroxymethyl-, formyl- and related transferases (82 organisms)
 - ❑ 2.1.3 Carboxy- and carbamoyltransferases (105 organisms)
 - ❑ 2.1.4 Amidotransferases (32 organisms)
 - 2.1.4.1 glycine amidinotransferase (17 organisms)
 - 2.1.4.2 scyllo-o-inosamine-4-phosphate amidinotransferase (15 organisms)
 - ❑ 2.2 Transferring aldehyde or ketonic groups (91 organisms)
 - ❑ 2.3 Acyltransferases (930 organisms)
 - ❑ 2.4 Glycosyltransferases (925 organisms)
 - ❑ 2.5 Transferring alkyl or aryl groups, other than methyl groups (547 organisms)
 - ❑ 2.6 Transferring nitrogenous groups (377 organisms)
 - ❑ 2.7 Transferring phosphorus-containing groups (1343 organisms)
 - ❑ 2.8 Transferring sulfur-containing groups (276 organisms)
 - ❑ 2.9 Transferring selenium-containing groups (6 organisms)
 - ❑ 3 Hydrolases (4453 organisms)
 - ❑ 4 Lyases (2145 organisms)
 - ❑ 5 Isomerases (849 organisms)
 - ❑ 6 Ligases (686 organisms)

Enzyme Classification scheme
(from <http://www.brenda-enzymes.org/>)

What about Un-Classified Proteins?

BIOINFORMATICS

Vol. 21 Suppl. 1 2005, pages i302–i310
doi:10.1093/bioinformatics/bti1054



Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps

Elena Nabieva^{1,2}, Kam Jim², Amit Agarwal¹, Bernard Chazelle¹
and Mona Singh^{1,2,*}

¹Computer Science Department and ²Lewis-Sigler Institute for Integrative Genomics,
Princeton University, Princeton, NJ 08544, USA

Received on January 15, 2005; accepted on March 27, 2005

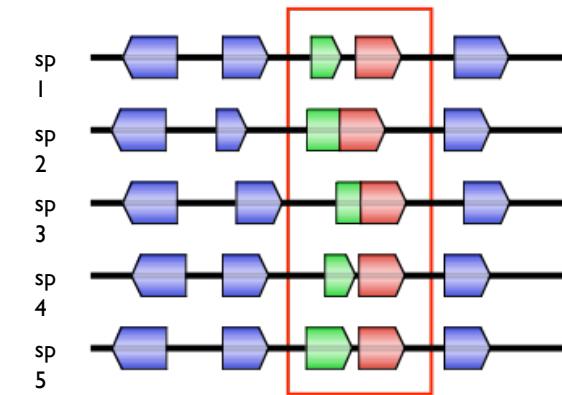
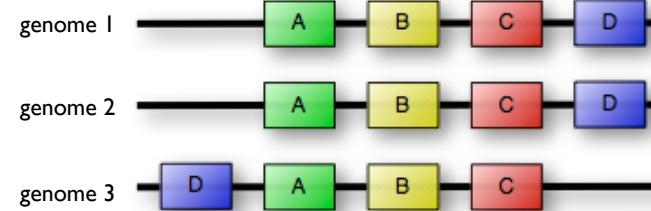
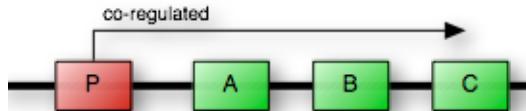
Many unclassified proteins:

- estimate: ~1/3 of the yeast proteome not annotated functionally
- BioGRID: 4495 proteins in the largest cluster of the yeast physical interaction map.
 - only 2946 have a MIPS functional annotation

Partition the Graph

Large **PPI networks** can be built from (see V3, V4, V5):

- HT experiments (Y2H, TAP, synthetic lethality, coexpression, coregulation, ...)
 - predictions (gene profiling, gene neighborhood, phylogenetic profiles, ...)
- proteins that are functionally linked



Identify **unknown functions** from **clustering** of these networks by, e.g.:

- shared interactions (similar neighborhood)
- membership in a community
- similarity of shortest path vectors to all other proteins (= similar path into the rest of the network)

Protein Interactions

Nabieva et al used the *S. cerevisiae* dataset from GRID of 2005 (now BioGRID)
→ 4495 proteins and 12 531 physical interactions in the largest cluster

The screenshot shows the BioGRID website homepage. The header features a red and black pixelated background with the BioGRID logo in white. A search bar with a dropdown menu set to "Organism: Escherichia coli K12" and a "GO" button is on the right. Below the header is a navigation bar with links: home, help / support, contribute, downloads, mirrors, and about us. The main content area has two columns. The left column is titled "About BioGRID" and contains a detailed paragraph about the database's history, current size (over 198,000 interactions), and features like curation and integration with other databases. The right column is titled "BioGRID Links" and lists several external links in blue text.

About BioGRID

The Biological General Repository for Interaction Datasets (BioGRID) database (<http://www.thebiogrid.org>) was developed to house and distribute collections of protein and genetic interactions from major model organism species. BioGRID currently contains over 198 000 interactions from six different species, as derived from both high-throughput studies and conventional focused studies. Through comprehensive curation efforts, BioGRID now includes a virtually complete set of interactions reported to date in the primary literature for both the budding yeast *Saccharomyces cerevisiae* and the fission yeast *Schizosaccharomyces pombe*. A number of new features have been added to the BioGRID including an improved user interface to display interactions based on different attributes, a mirror site and a dedicated interaction management system to coordinate curation across different locations. The BioGRID provides interaction data with monthly updates to *Saccharomyces Genome Database*, Flybase and Entrez Gene. Source code for the BioGRID and the linked Osprey network visualization system is now freely available without restriction.

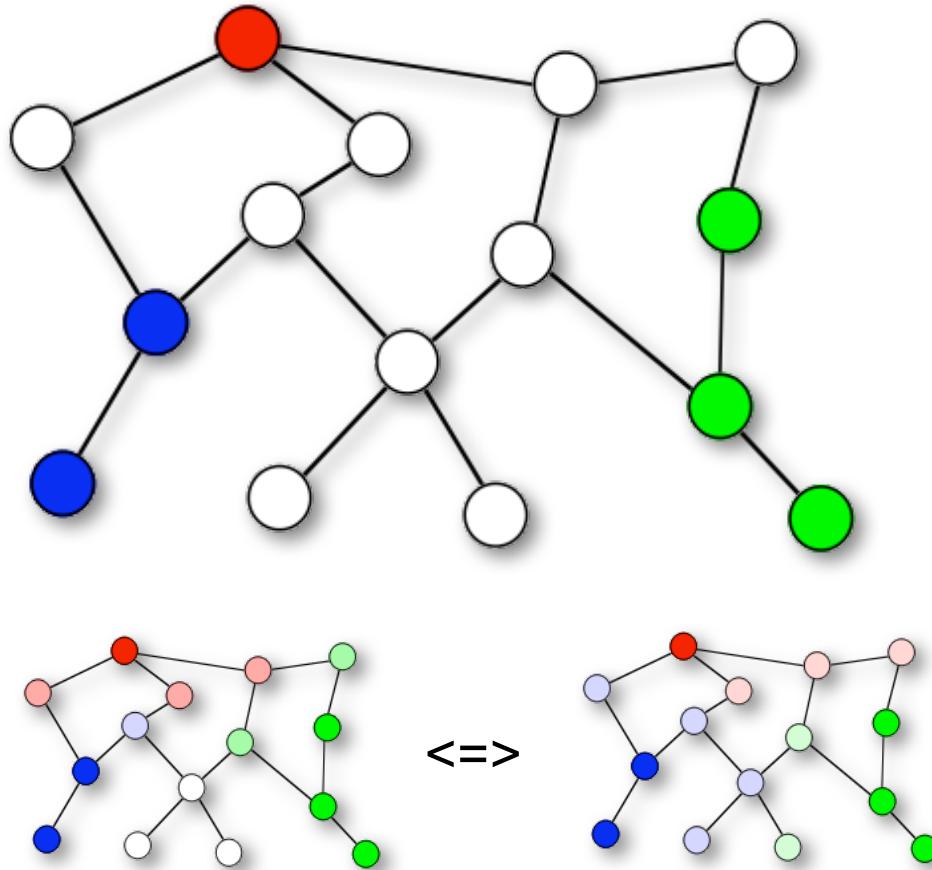
BioGRID Links

- Arabidopsis Information Resource
- BioPIXIE
- Biotechnology and Biological Sciences Research Council (BBSRC)
- Canadian Institutes of Health Research (CIHR)
- Cytoscape
- Database of Interacting Proteins
- Entrez-Gene
- Flybase
- Gene DB
- Gene Ontology
- Germ Online

<http://www.thebiogrid.org/about.php>

Function Annotation

Task: **predict** function (= functional annotation) for an unlabeled protein from the **available** annotations of other proteins in the network



Similar task:
How to **assign colors** to
the white nodes?

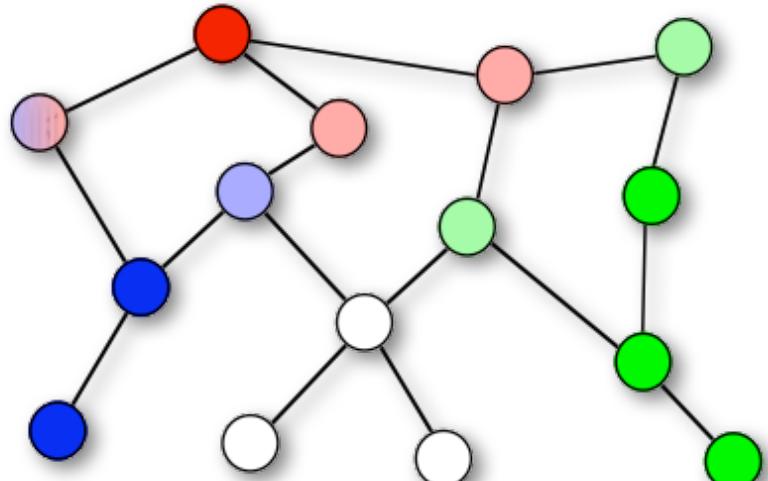
Use information on:

- distance to colored nodes
- local connectivity
- reliability of the links
- ...

Algorithm I: Majority

This concept was presented in
Schwikowski, Uetz, and Fields, "A network of protein–protein interactions in yeast"
Nat. Biotechnol. **18** (2000) 1257

Consider all direct neighbors and **sum up** how often a certain **annotation occurs**
→ score for an annotation = count among the direct neighbors
→ take the 3 most frequent functions



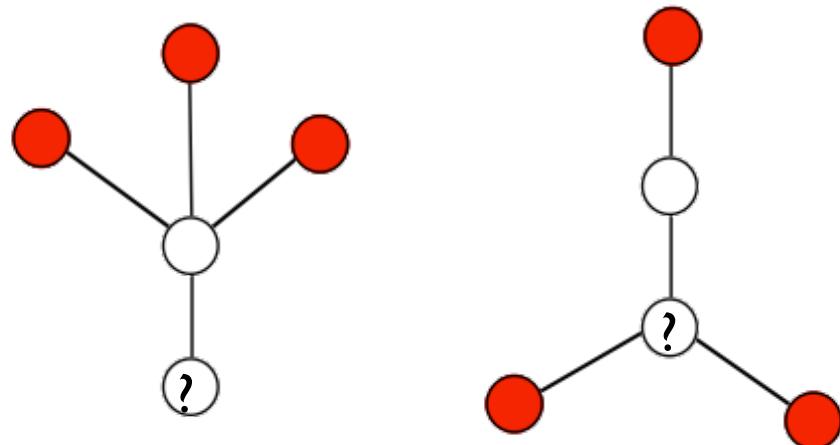
Majority makes only limited use
of the local connectivity
→ cannot assign function to
next-neighbors

For weighted graphs:
→ use weighted sum

Extended Majority: Neighborhood

This concept was presented in
Hishigaki, Nakai, Ono, Tanigami, and Takagi, "Assessment of prediction accuracy of protein function from protein–protein interaction data",
Yeast **18** (2001) 523

Look for **overrepresented** functions within a given **radius** of 1, 2, or 3 links
→ use as function score the value of a χ^2 –test



Neighborhood algorithm does not consider local network topology

Both examples (left) are treated **identically** with $r = 2$
although the right situation feels more certain (2 direct neighbors of ? are labeled)

Minimize Changes: GenMultiCut

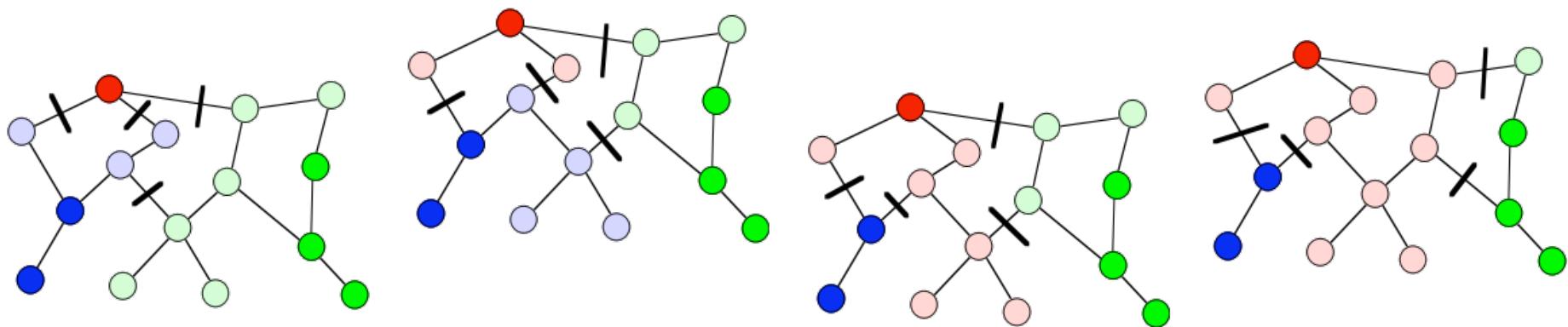
This concept was presented in

Karaoz, Murali, Letovsky, Zheng, Ding, Cantor, and Kasif, "Whole-genome annotation by using evidence integration in functional-linkage networks"

PNAS 101 (2004) 2888

"Annotate proteins so as to **minimize** the number of times that **different** functions are associated to **neighboring** (i.e. interacting) proteins"

→ generalization of the multiway k -cut problem for weighted edges,
can be stated as an integer linear program (ILP)



Multiple possible solutions → scores from **frequency** of annotations

Nabieva et al: FunctionalFlow

Extend the idea of "**guilty by association**"

- each annotated protein is considered as a source of "function"-flow
 - propagate/simulate for a few time steps
 - choose the annotation a with the highest accumulated flow

Each node u has a reservoir $R_t(u)$, each edge a capacity constraint (weight) $w_{u,v}$

Initially: $R_0^a(u) = \begin{cases} \infty, & \text{if } u \text{ is annotated with } a, \\ 0, & \text{otherwise.} \end{cases}$ and $g_0^a(u, v) = 0$

Then: **downhill flow** from node u to neighbor node v :

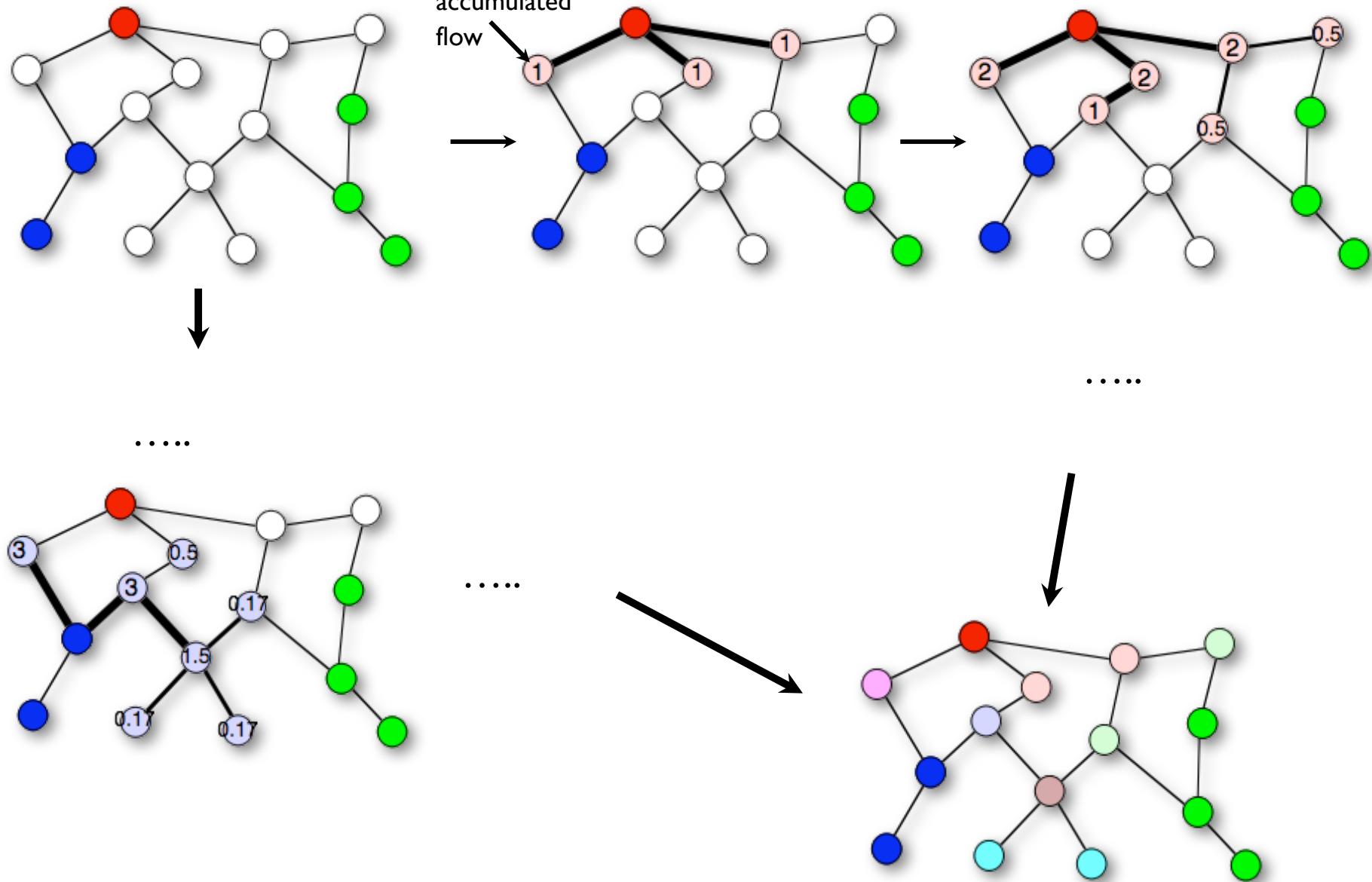
$$g_t^a(u, v) = \begin{cases} 0, & \text{if } R_{t-1}^a(u) < R_{t-1}^a(v) \\ \min\left(w_{u,v}, \frac{w_{u,v}}{\sum_{(u,y) \in E} w_{u,y}}\right), & \text{otherwise.} \end{cases}$$

Idea: Node v has already „more function“ than node $u \rightarrow$ no flow uphill

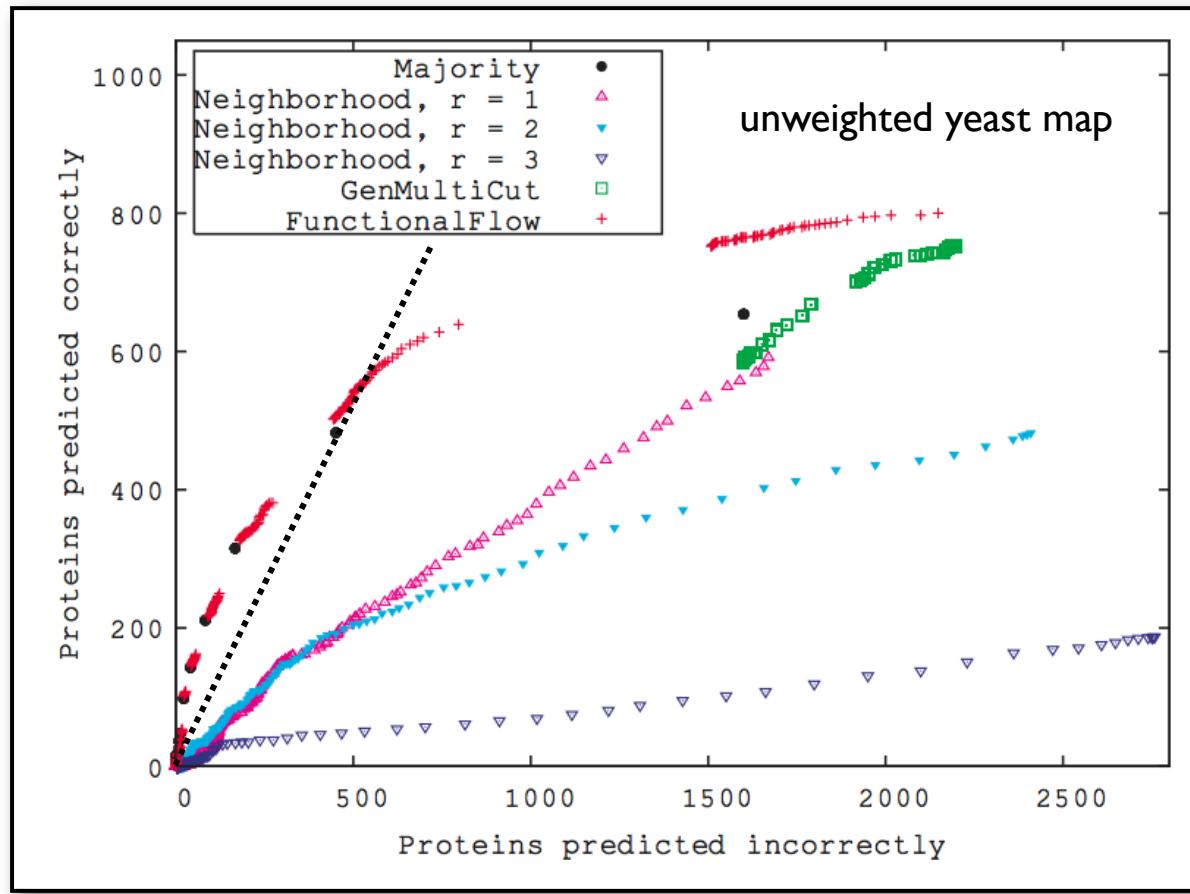
Score from accumulated in-flow:

$$f_a(u) = \sum_{t=1}^d \sum_{v:(u,v) \in E} g_t^a(v, u)$$

An Example



Comparison



For FunctionalFlow:
six propagation steps were
simulated; this is comparable
to the diameter of the yeast
network ≈ 12

Majority results are initially
very good, but has limited
coverage.

Results with neighborhood
get more imprecise for larger
radii r

Change **score threshold** for accepting annotations \rightarrow ratio **TP/FP**
 \rightarrow **FunctionalFlow** performs **best** in the high-confidence region
 \rightarrow but generates still many false predictions!!!

Going the Distance for Protein Function Prediction: A New Distance Metric for Protein Interaction Networks

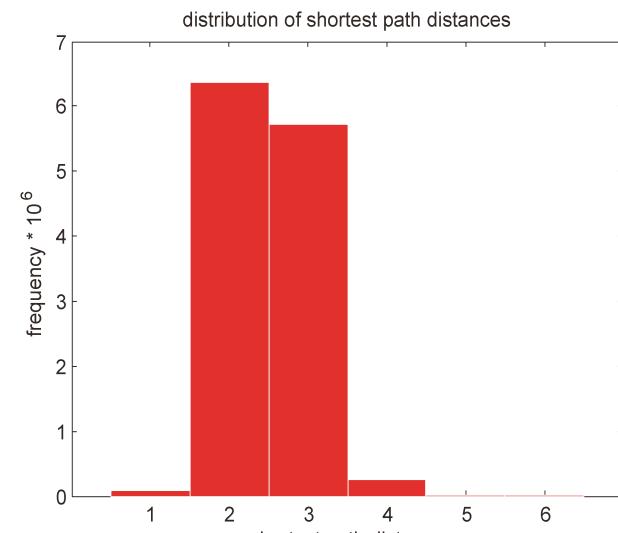
Citation: Cao M, Zhang H, Park J, Daniels NM, Crovella ME, et al. (2013) Going the Distance for Protein Function Prediction: A New Distance Metric for Protein Interaction Networks. PLoS ONE 8(10): e76339. doi:10.1371/journal.pone.0076339

Relying on the ordinary shortest-path distance metric in PPI networks is problematic because PPI networks are “small world” networks.
Most nodes are “close” to all other nodes.

→ any method that infers similarity based on proximity will find that a large fraction of the network is proximate to any typical node.

Largest connected component of *S. cerevisiae* PPI network (BioGRID) has 4990 nodes and 74,310 edges (physical interactions).

Right figure shows the histogram of shortest-path lengths in this network. Over 95% of all pairs of nodes are either 2 hops or 3 hops apart



(a)

What nodes mediate short contacts?

The 2-hop neighborhood of a typical node probably includes around half of **all nodes** in the graph.

One of the **reasons** that paths are typically short in biological networks like the PPI network is due to the **presence of hubs**.

But hub proteins often represent proteins with *different functional roles* than their neighbors.

Hub proteins likely also have multiple, distinct functions.

→ not all short paths provide equally strong evidence of similar function in PPI networks.

DSD Distance Metric

Given some fixed $k > 0$, we define $He^{\{k\}}(A, B)$ to be the expected number of times that a random walk starting at A and proceeding for k steps, will visit B . If there is no ambiguity about k , we can drop k .

$$He(v_i) = (He(v_i, v_1), He(v_i, v_2), \dots, He(v_i, v_n))$$

$He(v_i)$ is a „random walk distance vector“ of node v_i from all other nodes.

$$DSD(u, v) = \|He(u) - He(v)\|_1 \quad \text{where}$$

$\|He(u) - He(v)\|_1$ denotes the L_1 norm of the He vectors

Two nodes u and v have small DSD if they have similar distance from all other nodes.

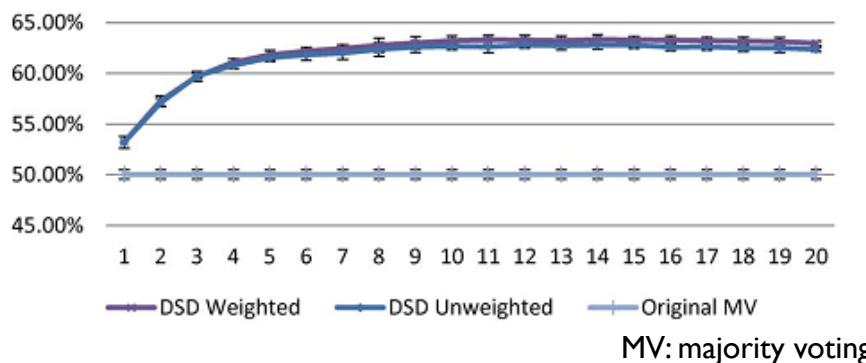
Explanation: The one-norm (also known as the L_1 -norm, ℓ_1 norm, or mean norm) of a vector \vec{v} is denoted $\|\vec{v}\|_1$ and is defined as the sum of the absolute values of its components:

$$\|\vec{v}\|_1 = \sum_{i=1}^n |v_i| \tag{1}$$

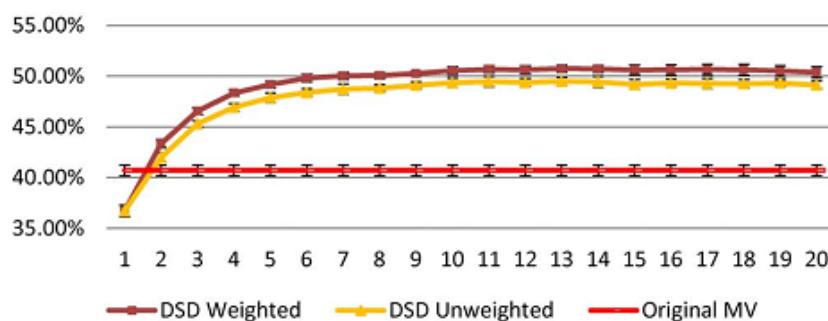
for example, given the vector $\vec{v} = (1, -4, 5)$, we calculate the one-norm:

DSD clearly improves functional predictions

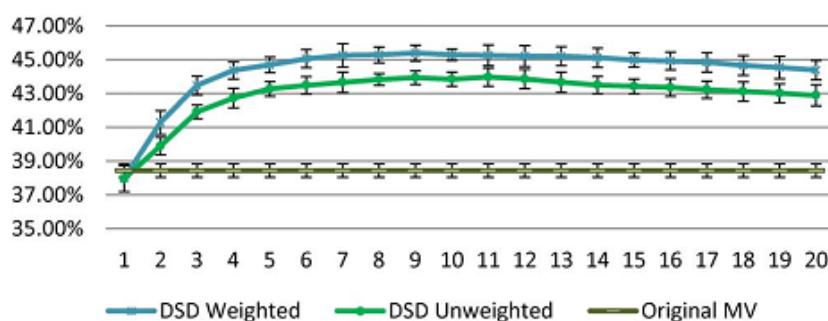
MIPS Top Level, Accuracy



MIPS Second Level, Accuracy



MIPS Third Level, Accuracy



F1 Score on GO term Prediction for *S. cerevisiae*

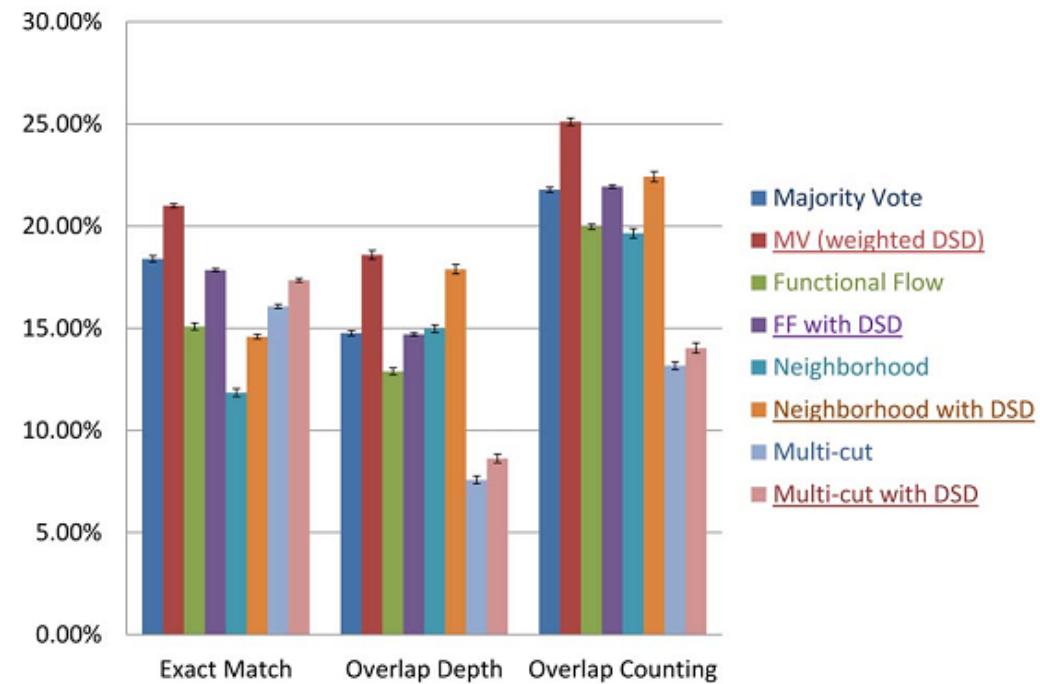


Figure 6. Improvement on F1 Score for DSD using three evaluation methods: exact match, overlap depth and overlap counting, on informative GO terms for the four algorithms for *S. cerevisiae* in 10 runs of 2-fold cross validation.

Summary

V8: wrap up protein interaction networks

Then next block of the lecture: gene-regulatory networks