Detecting topological patterns in complex networks

Sergei Maslov

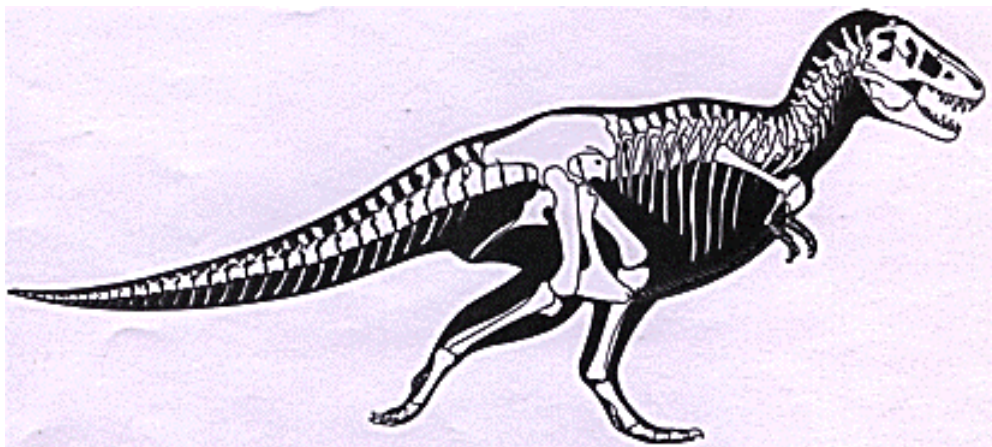
Brookhaven National Laboratory

Networks in complex systems

- Complex systems
  - Large number of components
  - They interact with each other
  - All components and/or interactions are different from each other (unlike in traditional physics)
  - 10^4 types of proteins in an organism, 10^6 routers, 10^9 web pages, 10^{11} neurons
- The simplest property of a complex system: who interacts with whom? can be visualized as a network
- Backbone for real dynamical processes

Why study the topology of complex networks?

- Lots of easily available data: that's where the state of the art information is (at least in biology)
- Large networks may contain information about basic design principles and/or evolutionary history of the complex system
- This is similar to paleontology: learning about an animal from its backbone



Plan of my lectures

- Part 1: Bio-molecular (protein) networks
  - What is a protein network?
  - Degree distributions in real and random networks
  - Beyond degree distributions: How it all is wired together? Correlations in degrees
  - How do protein networks evolve?
- Part 2: Interplay between an auxiliary diffusion process (random walk) and modules and communities on the Internet and WWW

Collaborators:

- Biology:
  - Kim Sneppen – U. of Copenhagen
  - Kasper Eriksen – U. of Lund
  - Koon-Kiu Yan – Stony Brook U.
  - Ilya Mazo – Ariadne Genomics
  - Iaroslav Ispolatov – Ariadne Genomics
- Internet and WWW:
  - Kasper Eriksen – U. of Lund
  - Kim Sneppen – U. of Copenhagen
  - Ingve Simonsen – NORDITA
  - Koon-Kiu Yan – Stony Brook U.
  - Huafeng Xie –City University of New York

Postdoc position

- Looking for a postdoc to work in my group at Brookhaven National Laboratory in New York starting Fall 2005
- Topic - large-scale properties of (mostly) bionetworks (partially supported by a NIH/NSF grant with Ariadne Genomics)
- E-mail CV and 3 letters of recommendation to: maslov@bnl.gov
- See www.cmth.bnl.gov/~maslov

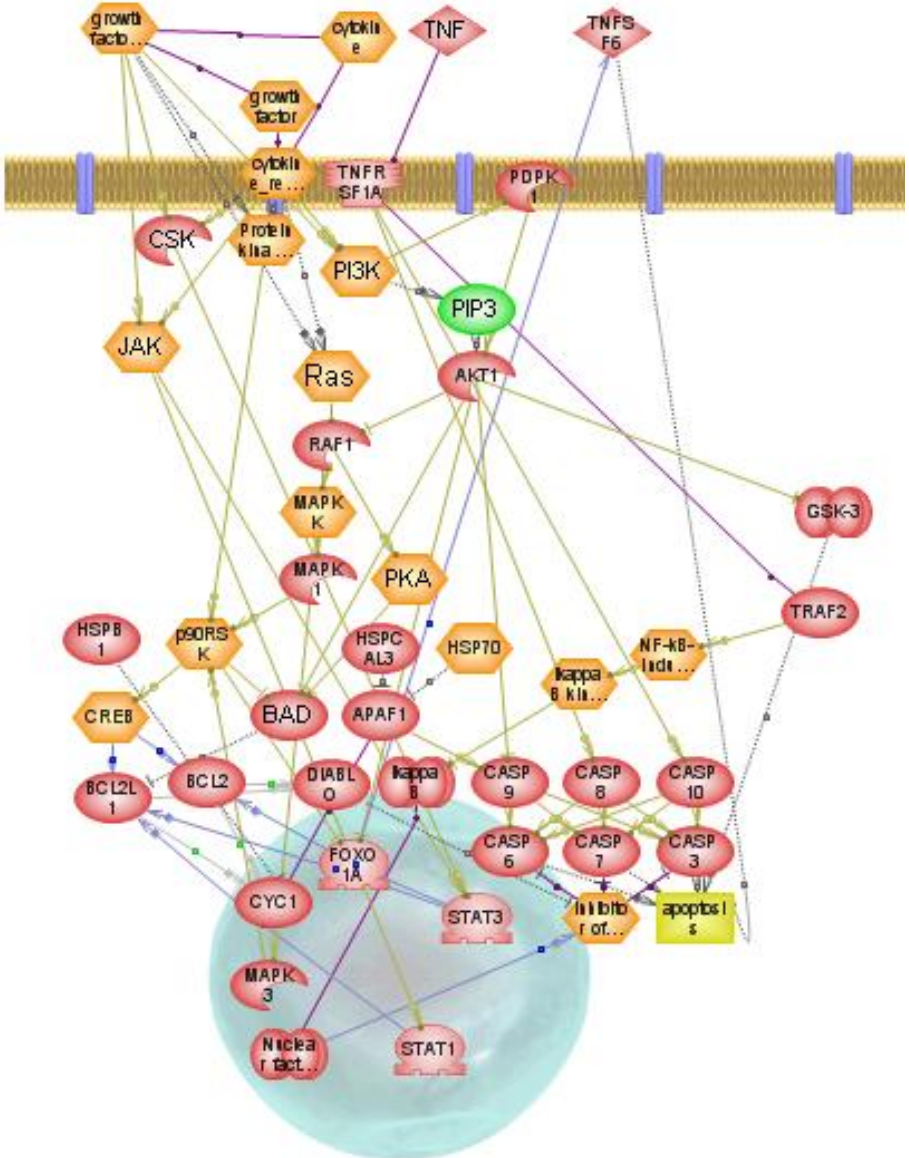
Protein networks

- Nodes – proteins
- Edges – interactions between proteins
  - Metabolic (all metabolic pathways of an organism)
  - Bindings (physical interactions)
  - Regulations (transcriptional regulation, protein modifications)
  - Disruptions in expression of genes (remove or overexpress one gene and see which other genes are affected)
  - Co-expression networks from microarray data (connect genes with similar expression patterns under many conditions)
  - Synthetic lethals (removal of A doesn't kill, removal of B doesn't kill but removal of both does)
  - Etc, etc, etc.

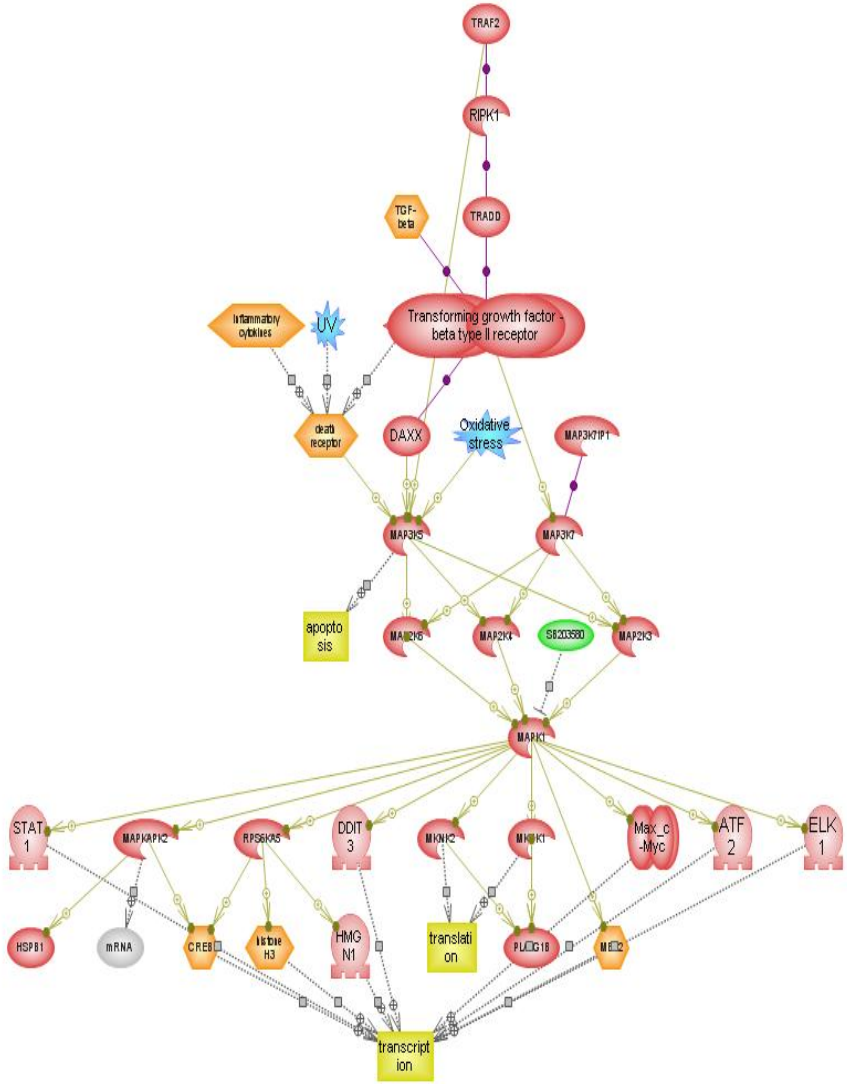
Sources of data on protein networks

- Genome-wide experiments
  - Binding – two-hybrid high throughput experiments
  - Transcriptional regulation – ChIP-on-chip, or ChIP-then-SAGE
  - Expression, disruption networks – microarrays
  - Lethality of genes (including synthetic lethals):
    - Gene knockout – yeast
    - RNAi –worm, fly
- Many small or intermediate-scale experiments
  - Public databases: DIP, BIND, YPD (no longer public), SGD, Flybase, Ecocyc, etc.

Images from ResNet3.0 by Ariadne Genomics

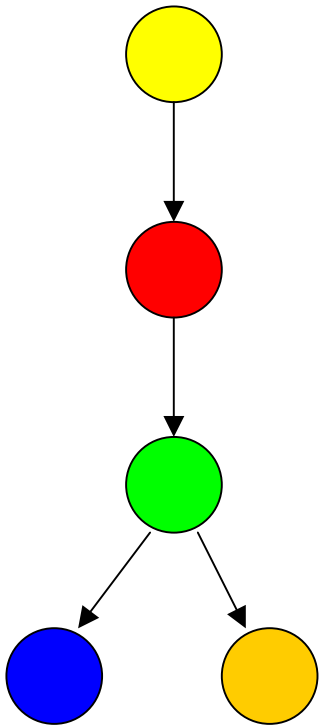


Inhibition of apoptosis

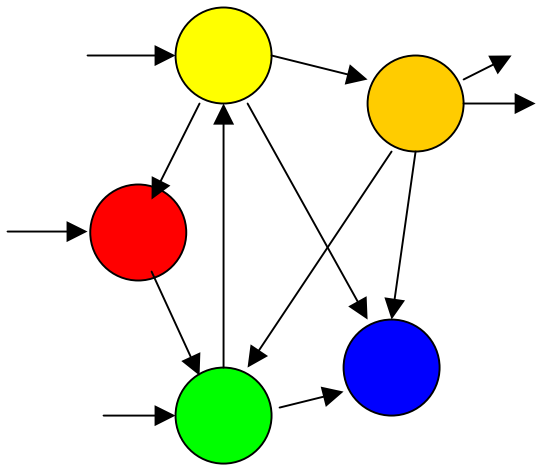


MAPK signaling

Pathway → network paradigm shift

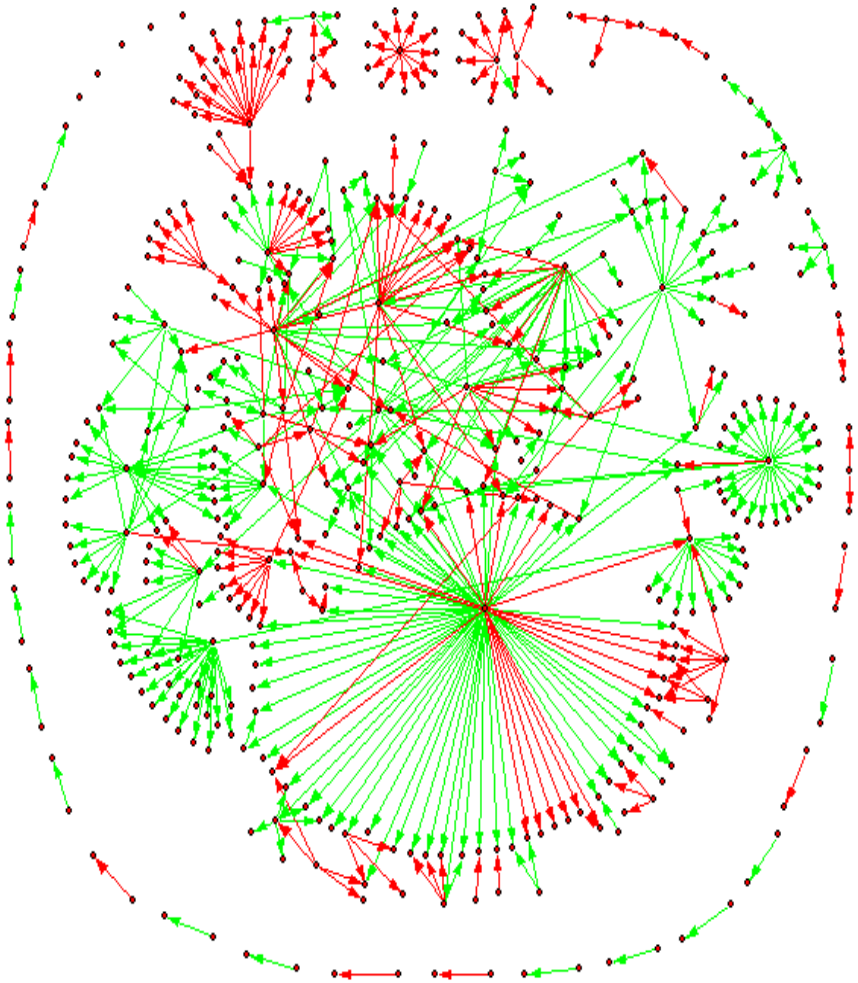


Pathway

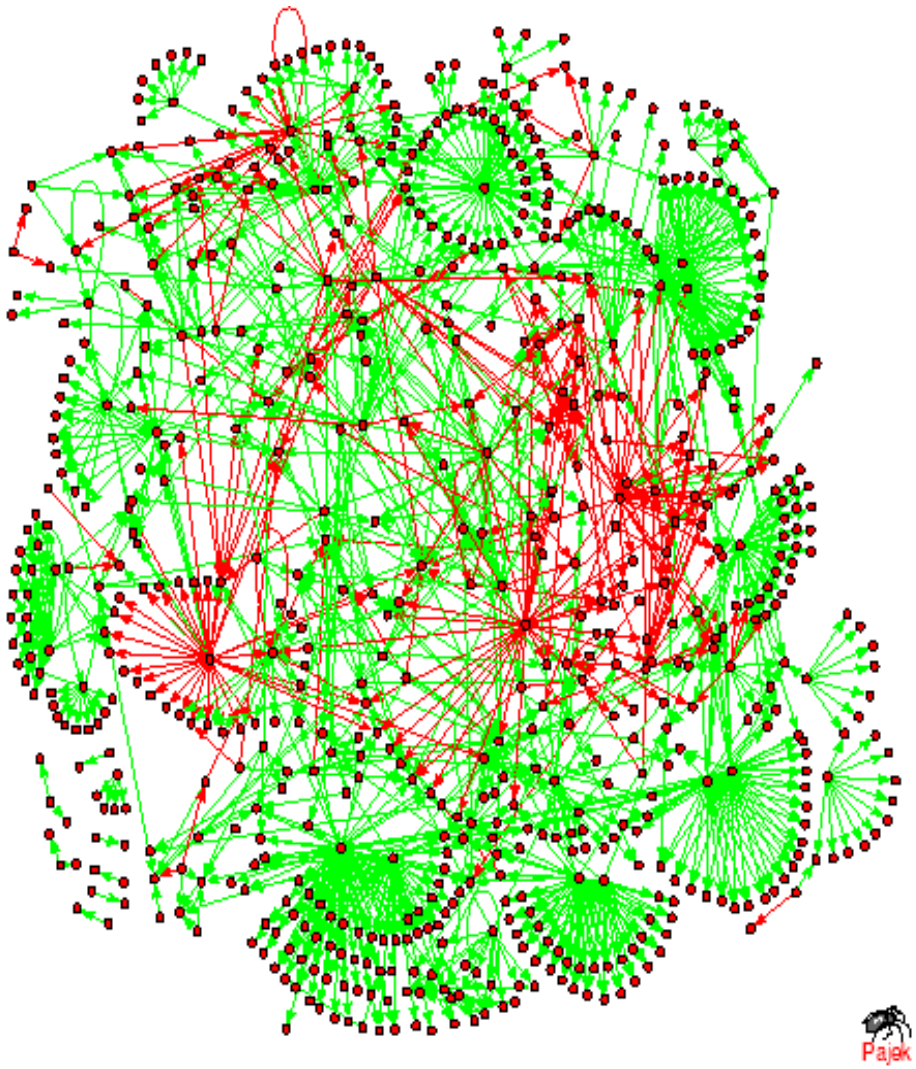


Network

Transcription regulatory networks

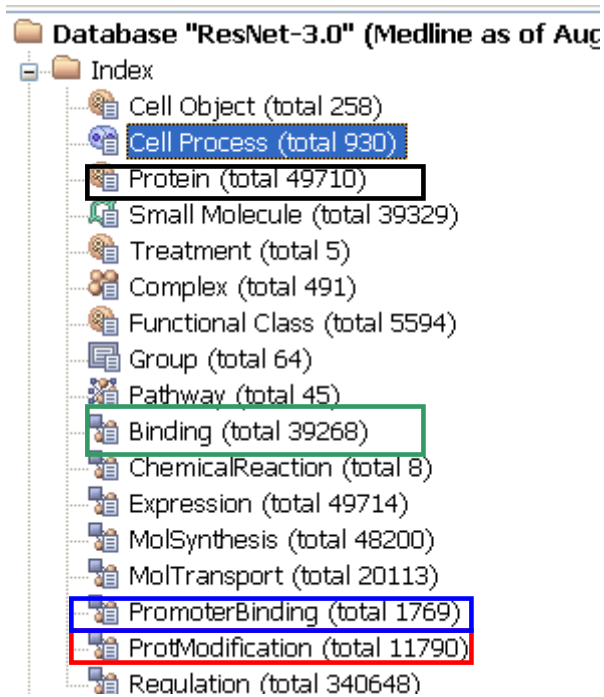


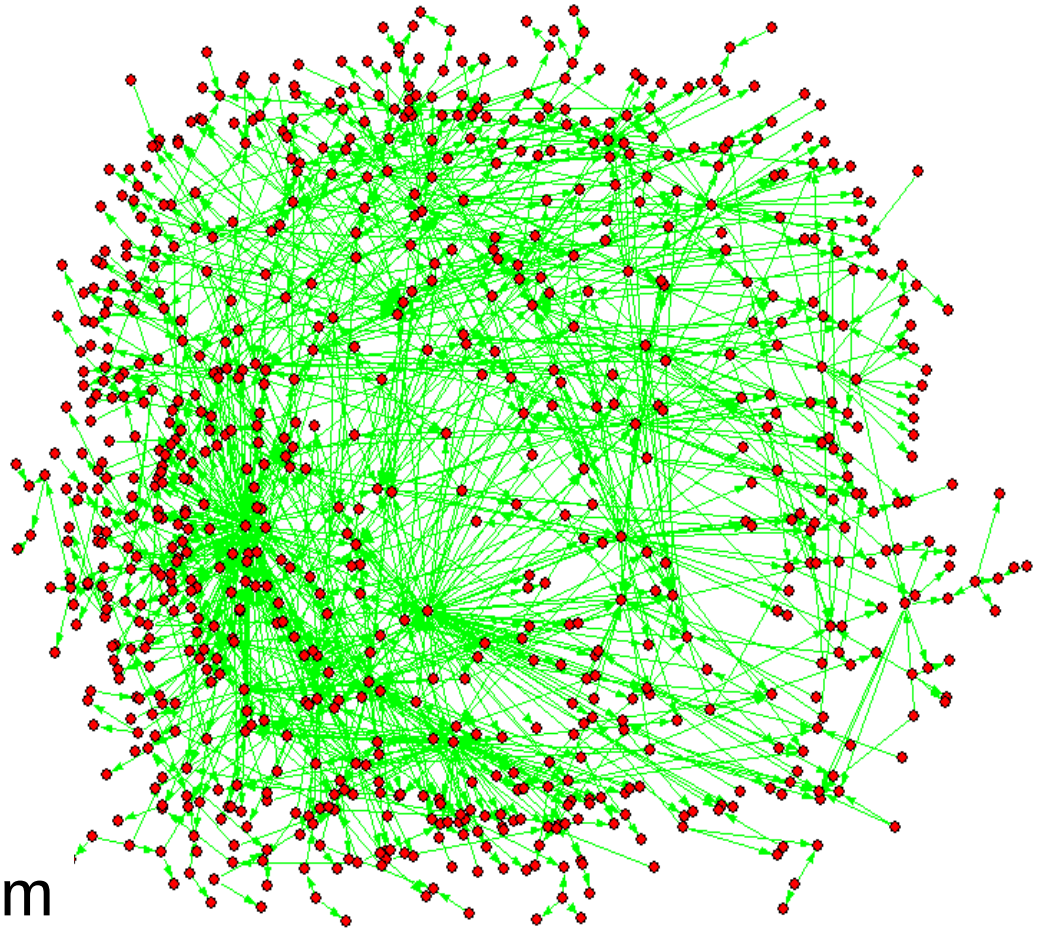
Bacterium: *E. coli*
3:2 ratio



Single-celled eukaryote:
S. cerevisiae; 3:1 ratio

Homo sapiens

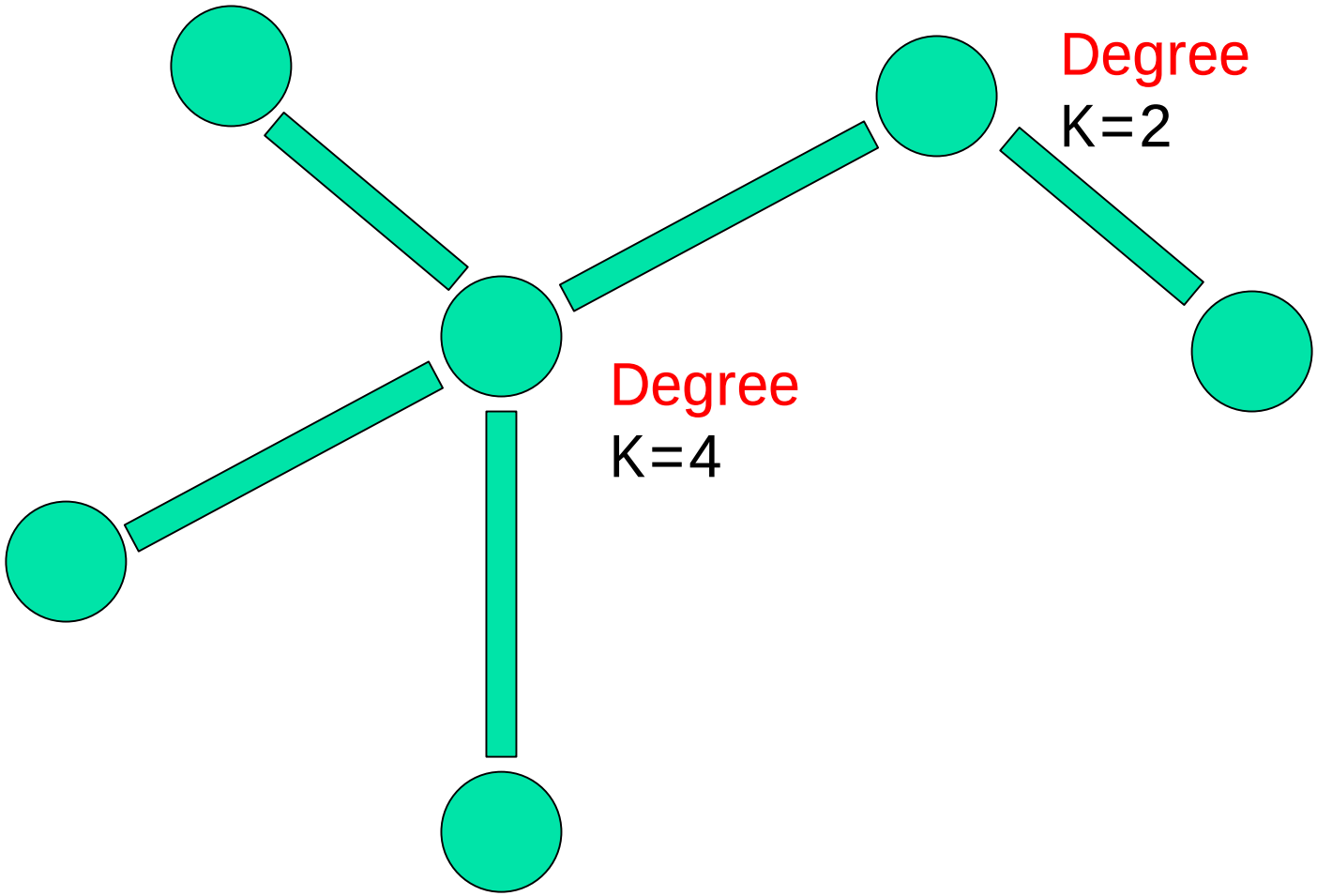




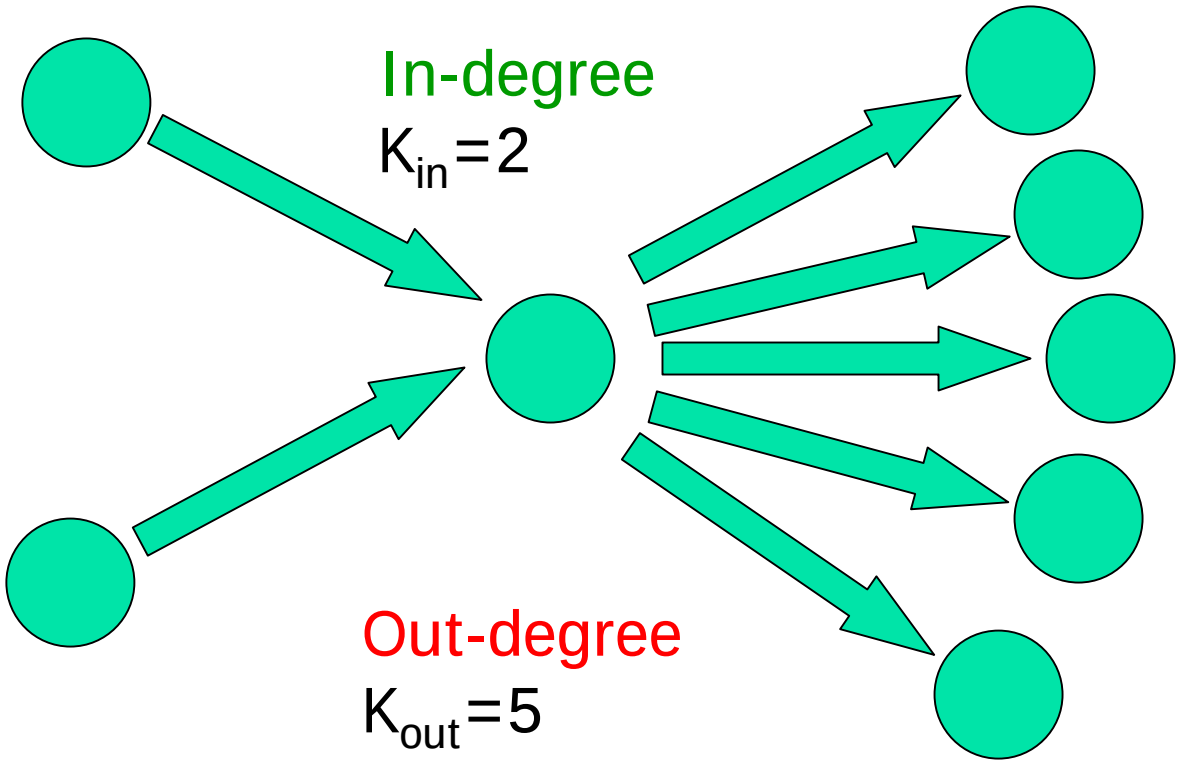
Total: 120,000 interacting protein pairs extracted from PubMed as of 8/2004

Data from Ariadne Genomics

Degree (or connectivity) of a node – the # of neighbors

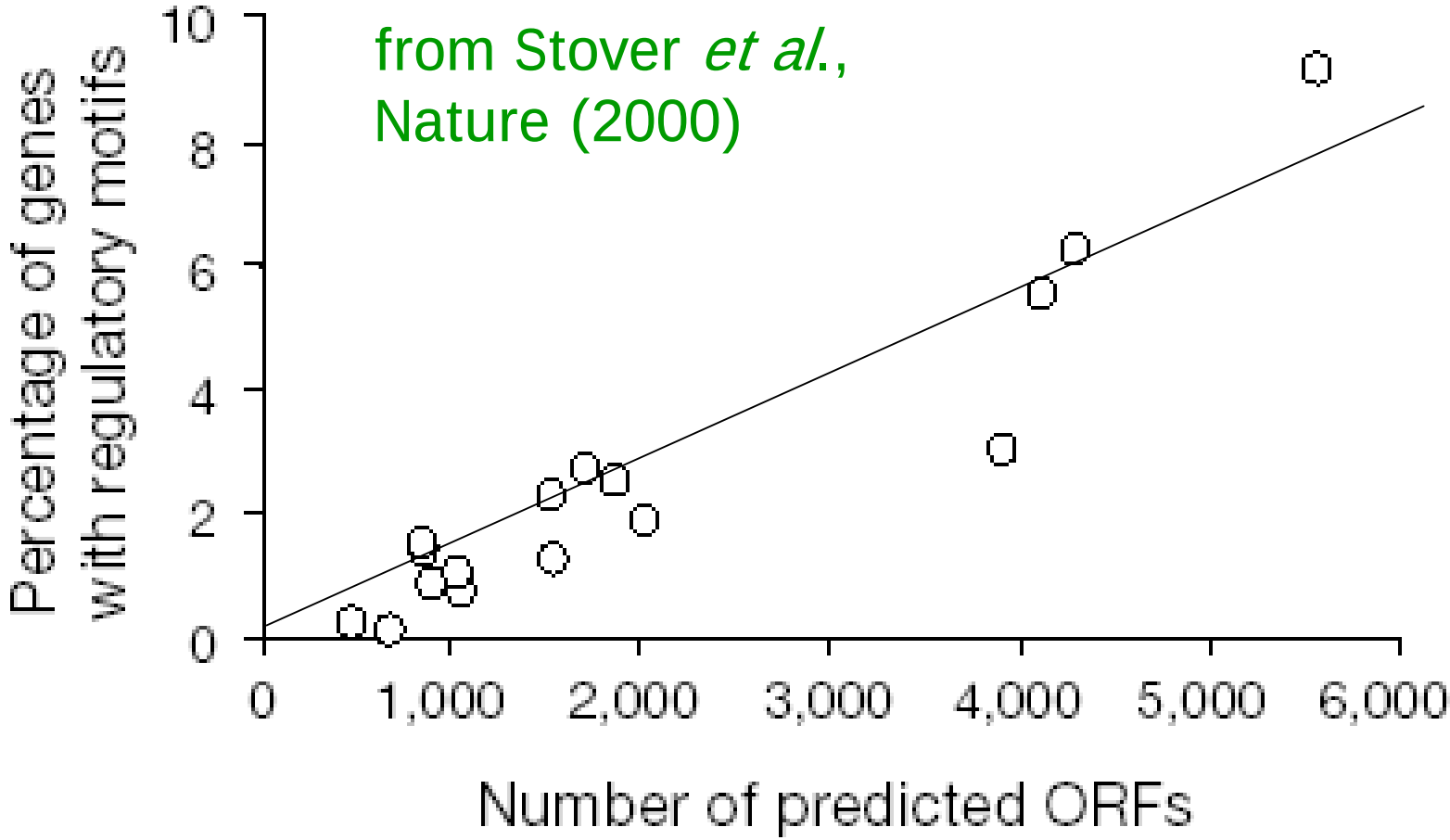


Directed networks have in- and out-degrees



How many transcriptional regulators are out there?

Fraction of transcriptional regulators in bacteria

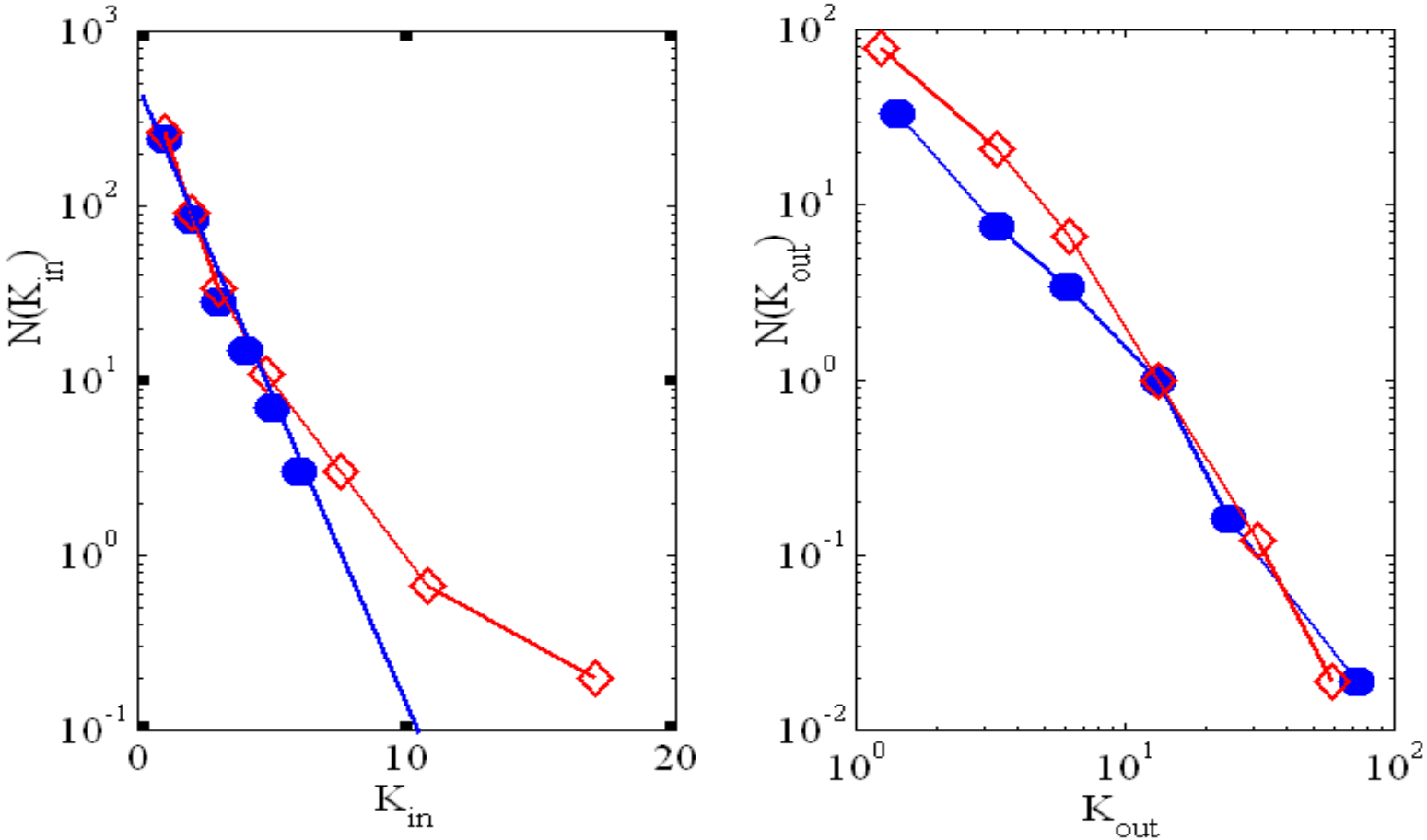


Complexity of regulation grows with complexity of organism

- $N_R \langle K_{out} \rangle = N \langle K_{in} \rangle =$ number of edges
- $N_R/N = \langle K_{in} \rangle / \langle K_{out} \rangle$ increases with N
- $\langle K_{in} \rangle$ grows with N
  - In bacteria $N_R \sim N^2$ (Stover, et al. 2000)
  - In eucaryots $N_R \sim N^{1.3}$ (van Nimwengen, 2002)
- Networks in more complex organisms are more interconnected then in simpler ones

Complexity is manifested in K_{in} distribution

E. coli vs H. sapiens



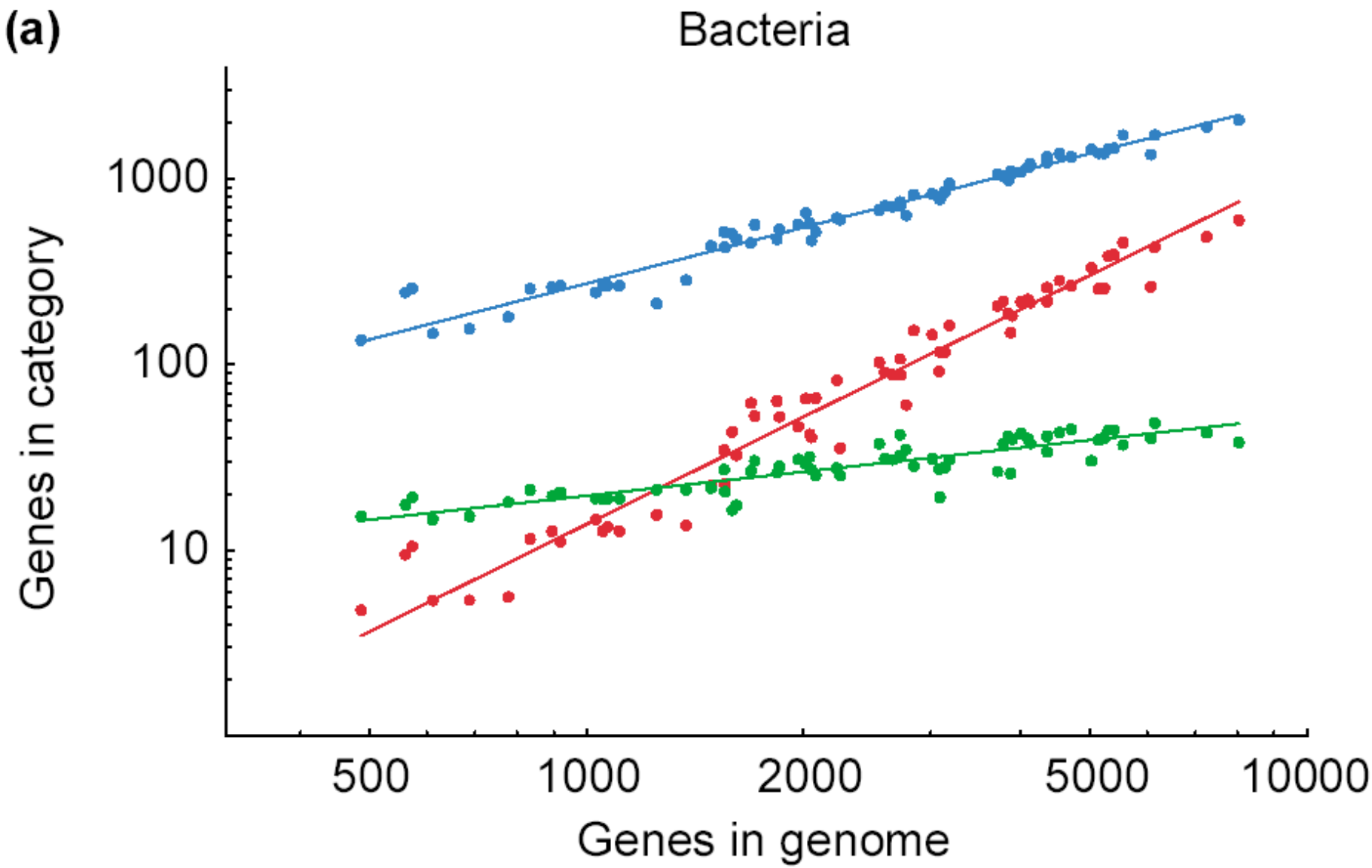


Figure from Erik van Nimwegen, TIG 2003

Category	Bacteria	Eukaryotes
Transcription regulation	1.87 ± 0.13	1.26 ± 0.10
Metabolism	1.01 ± 0.06	1.01 ± 0.08
Cell cycle	0.47 ± 0.08	0.79 ± 0.16
Signal transduction +	1.72 ± 0.18	1.48 ± 0.39
DNA repair	0.64 ± 0.08	0.83 ± 0.31
DNA replication	0.43 ± 0.08	0.72 ± 0.23
Protein biosynthesis	0.13 ± 0.02	0.41 ± 0.15
Protein degradation	0.97 ± 0.09	0.90 ± 0.11
Ion transport	1.42 ± 0.28	1.43 ± 0.20
Catabolism	0.88 ± 0.07	0.92 ± 0.08
Carbohydrate metabolism	1.01 ± 0.11	1.36 ± 0.36
Two-component systems +	2.07 ± 0.21	NA[b]
Cell communication +	1.81 ± 0.19	1.58 ± 0.34
Defense response +	NA[b]	3.35 ± 1.41

Table from Erik van Nimwegen, TIG 2003

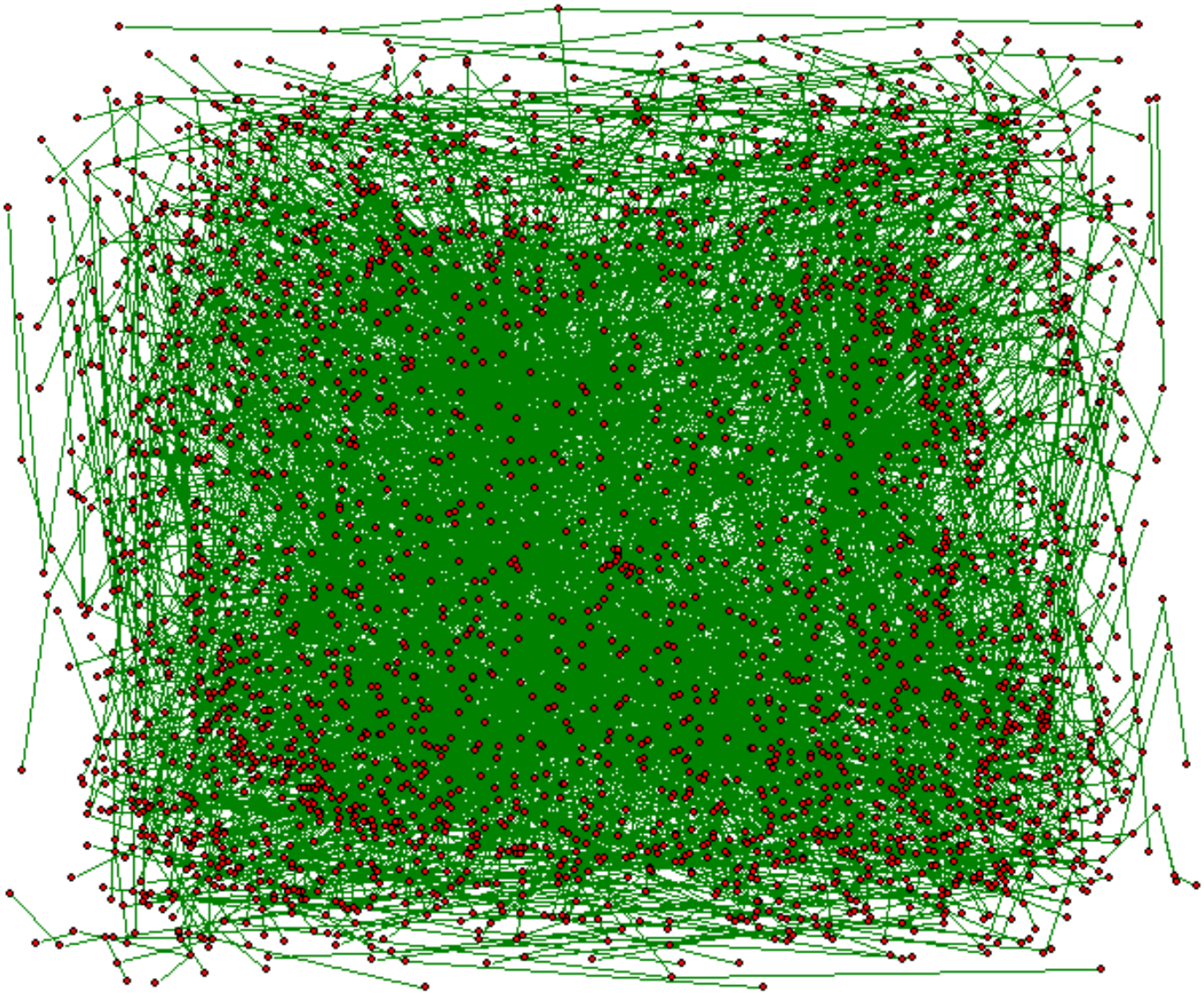
Protein-protein binding (physical interaction) networks

Two-hybrid experiment

- To test if A interacts with B create two hybrids A* (with Gal4p DNA-binding domain) and B* (with Gal4p activation domain)

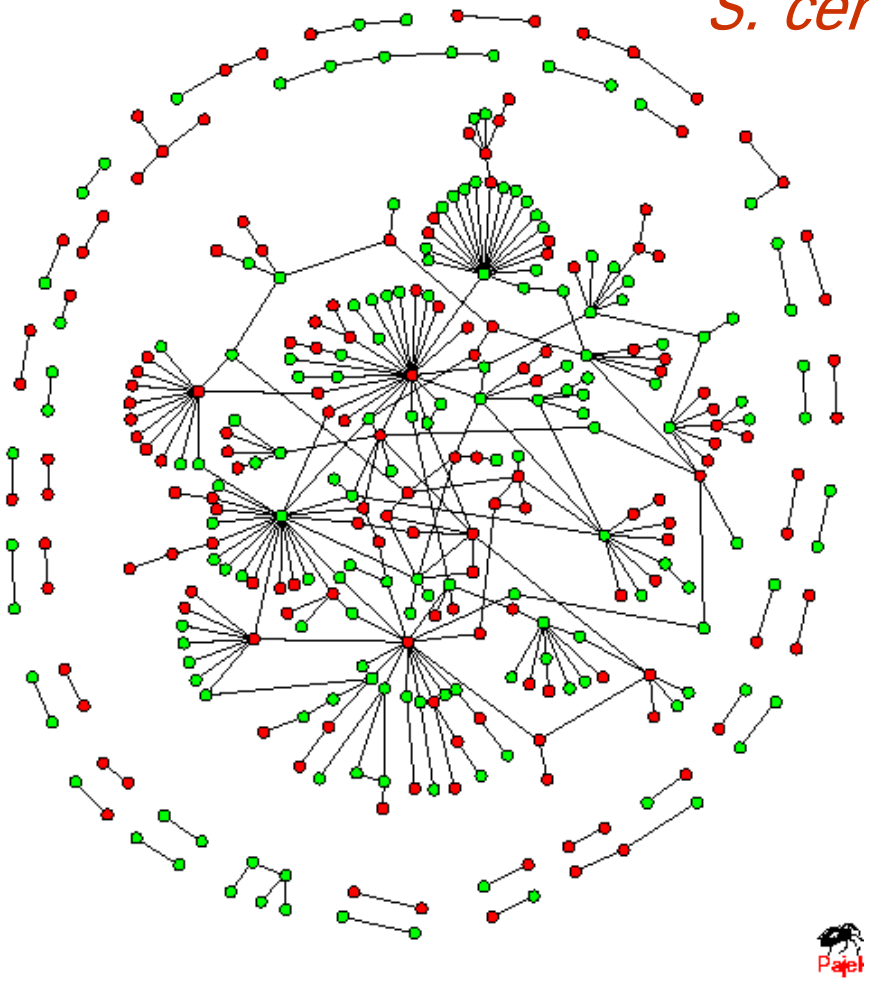
- High-throughput: all pairs among 6300 yeast proteins are tested: Yeast: Uetz, *et al.* Nature (2000), Ito, *et al.*, PNAS (2001); Fly: L. Giot et al. Science (2003)

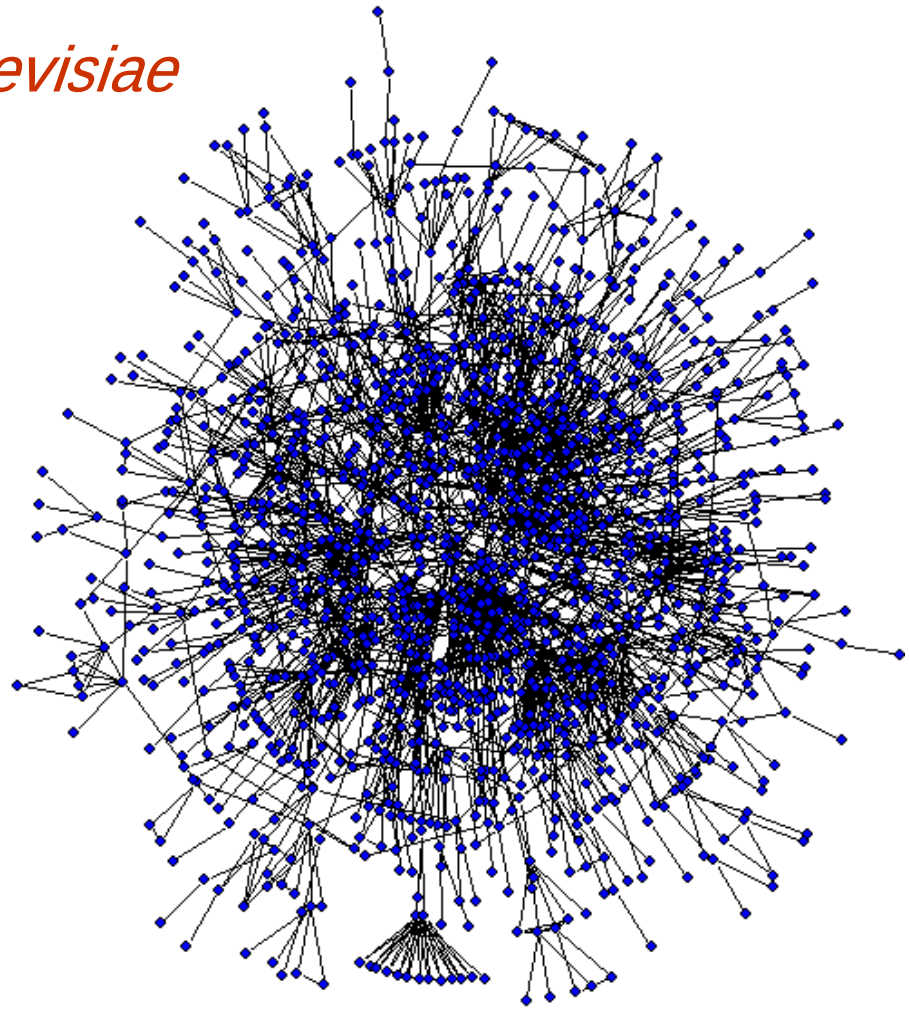
Protein binding network in yeast



Protein binding networks

S. cerevisiae





Two-hybrid nuclear

Database (DIP) core set

Degree distributions
in random and real networks

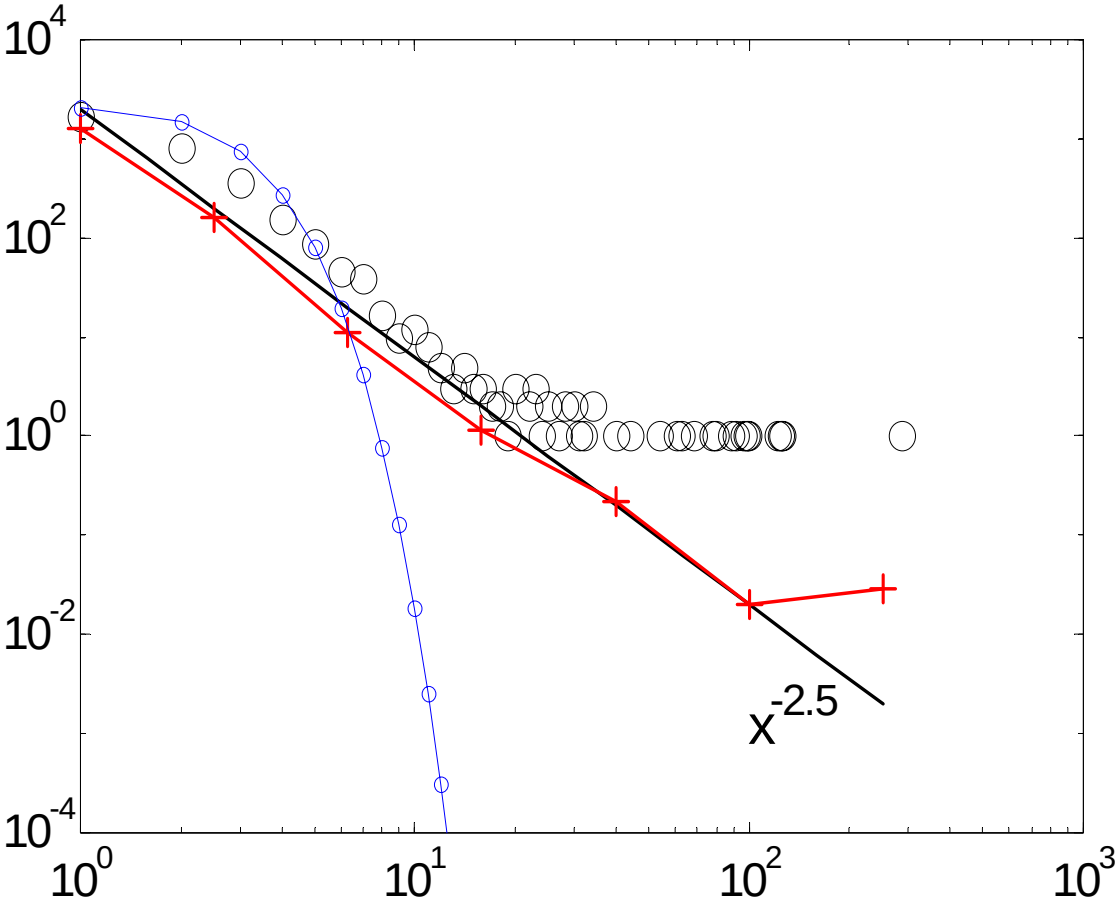
Degree distribution in a random network

- Randomly throw E edges among N nodes
- Solomonoff, Rapaport, Bull. Math. Biophysics (1951)
  Erdos-Renyi (1960)
- Degree distribution – Poisson
- K~λ±√λ with no hubs (fast decay of N(K))

$$N(K) = N\frac{\lambda^K}{K!}\exp(-\lambda)$$

$$\lambda = \langle K \rangle = 2E/N$$

Degree distribution in real protein networks



- Histogram N(K) is broad: most nodes have low degree ~ 1, few nodes – high degree ~100
- Can be approximately fitted with $N(K) \sim K^{-\gamma}$ functional form with $\gamma \sim = 2.5$

H. Jeong et al. (2001)

Network properties of self-binding proteins

AKA homodimers

I. Ispolatov, A. Yuryev, I. Mazo, SM
q-bio.GN/0501004.

There are just TOO MANY homodimers

	N_{dimer}	$\langle k \rangle$
yeast	179	6.6 ± 0.2
worm	89	3.3 ± 0.1
fly	160	5.9 ± 0.1
human	1045	5.7 ± 0.1

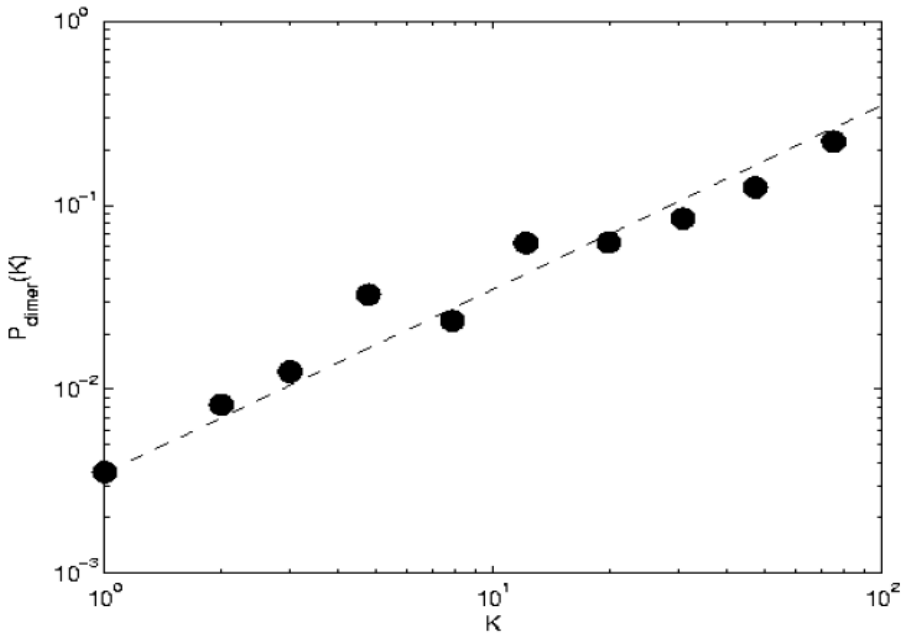
- Null-model
- $P_{self} \sim \langle k \rangle / N$
- $N_{dimer} = N \bullet P_{self} = \langle k \rangle$
- Not surprising as homodimers have many functional roles

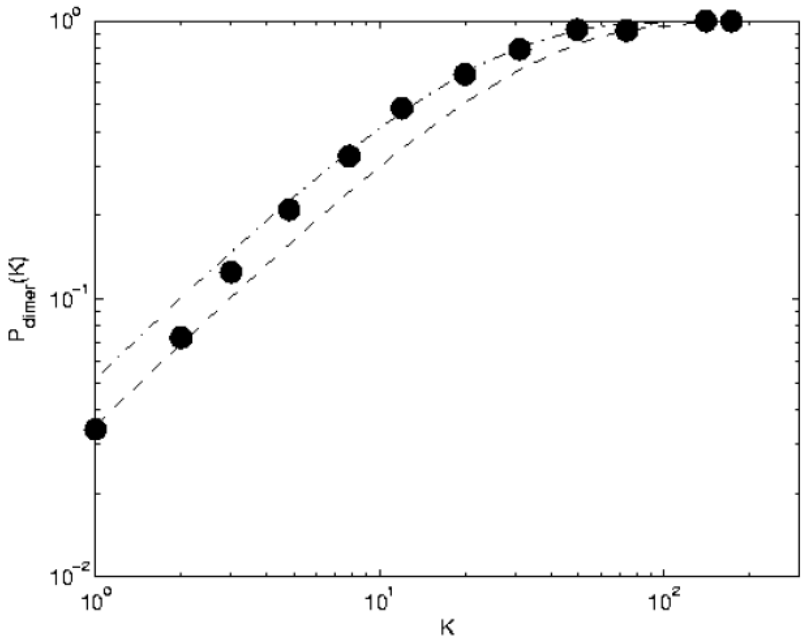
Network properties around homodimers

	$\langle k \rangle$	$\langle k \rangle_{\text{dimer}}$
yeast	6.6 ± 0.2	12.4 ± 1.2
worm	3.3 ± 0.1	13.1 ± 2.2
fly	5.9 ± 0.1	14.2 ± 1.2
human	5.7 ± 0.1	14.0 ± 0.6

Likelihood to self-interact vs. K

$$P_{dimer}(k) = 1 - (1 - p_{self})^k$$





Fly: two-hybrid data

P_{self}~0.003, P_{others}~0.0002

Human: database data

P_{self}~0.05, P_{others}~0.0002

What we think it means?

- In random networks $p_{dimer}(K) \sim K^2$ not $\sim K$ like our empirical observation
- K is proportional to the "stickiness" of the protein which in its turn scales with
  - the area of hydrophobic residues on the surface
  - # copies/cell
  - its popularity (in datasets taken from a database)
  - Etc.
- Real interacting pairs consists of an "active" and "passive" protein and binding probability scales only with the "stickiness" of the active protein
- "Stickiness" fully accounts for higher than average connectivity of homodimers

Propagation of signals and perturbations in networks

Naïve argument

- An average node has <K> first neighbors, <K><K-1> second neighbors, <K><K-1><K-1> third neighbors
- Total number of neighbors affected is 1+<K> + <K><K-1> +<K><K-1><K-1> +…
- If <K-1> ≥ 1 a perturbation starting at a single node MAY affect the whole network

Where is it wrong?

- Probability to arrive to a node with K neighbors is proportional to K!
- The right answer: $\langle K(K-1)\rangle / \langle K\rangle \geq 1$ a perturbation would spread
- Correlations between degrees of neighbors would affect the answer

How many clusters?

- If $\langle K(K-1)\rangle/\langle K\rangle \gg 1$ there is one "giant" cluster and few small ones. If $\langle K(K-1)\rangle/\langle K\rangle \sim 1$ small clusters have a broad (power-law) distribution
- Perturbation which affects neighbors with probability p propagates if $p\langle K(K-1)\rangle/\langle K\rangle \geq 1$
- For scale-free networks $P(K)\sim K^{-\gamma}$ with $\gamma<3$, $\langle K^2\rangle=\infty \Rightarrow$ perturbation always spreads in a large enough network

Problem 2: For a given p and γ how large should be the network for the perturbation to spread

Amplification ratios

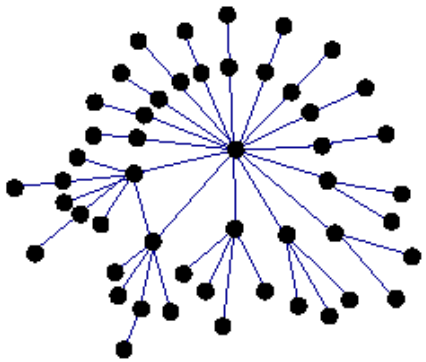
$$\mathcal{A}^{(undir)} = \frac{\langle K(K-1)\rangle}{\langle K\rangle} \qquad \mathcal{A}^{(dir)} = \frac{\langle K_{in}K_{out}\rangle}{\langle K_{in}\rangle}$$

Problem 3: derive the above formula for directed networks

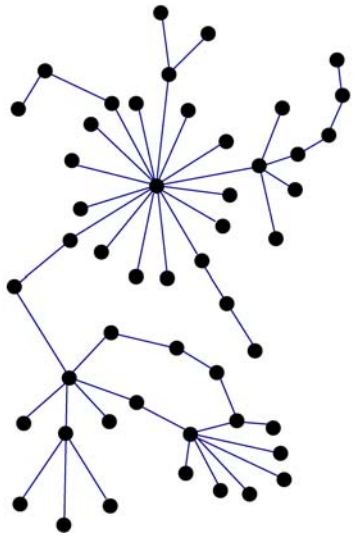
- $A^{(dir)}$: 1.08 - E. Coli, 0.58 - Yeast
- $A^{(undir)}$: 10.5 - E. Coli, 13.4 – Yeast
- $A^{(PPI)}$: ? - E. Coli, 26.3 - Yeast

Beyond degree distributions: How is it all wired together?

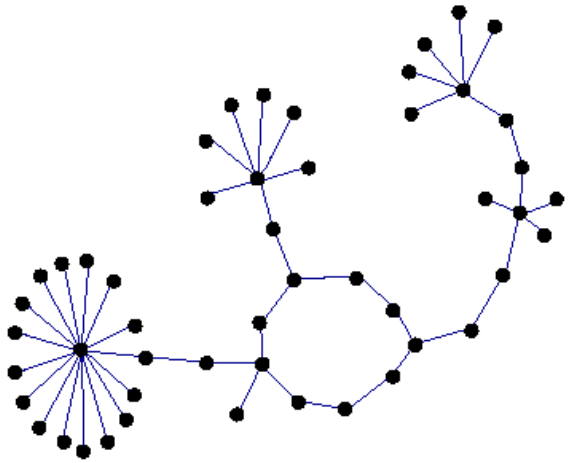
Central vs peripheral network architecture



central
(hierarchical)



random

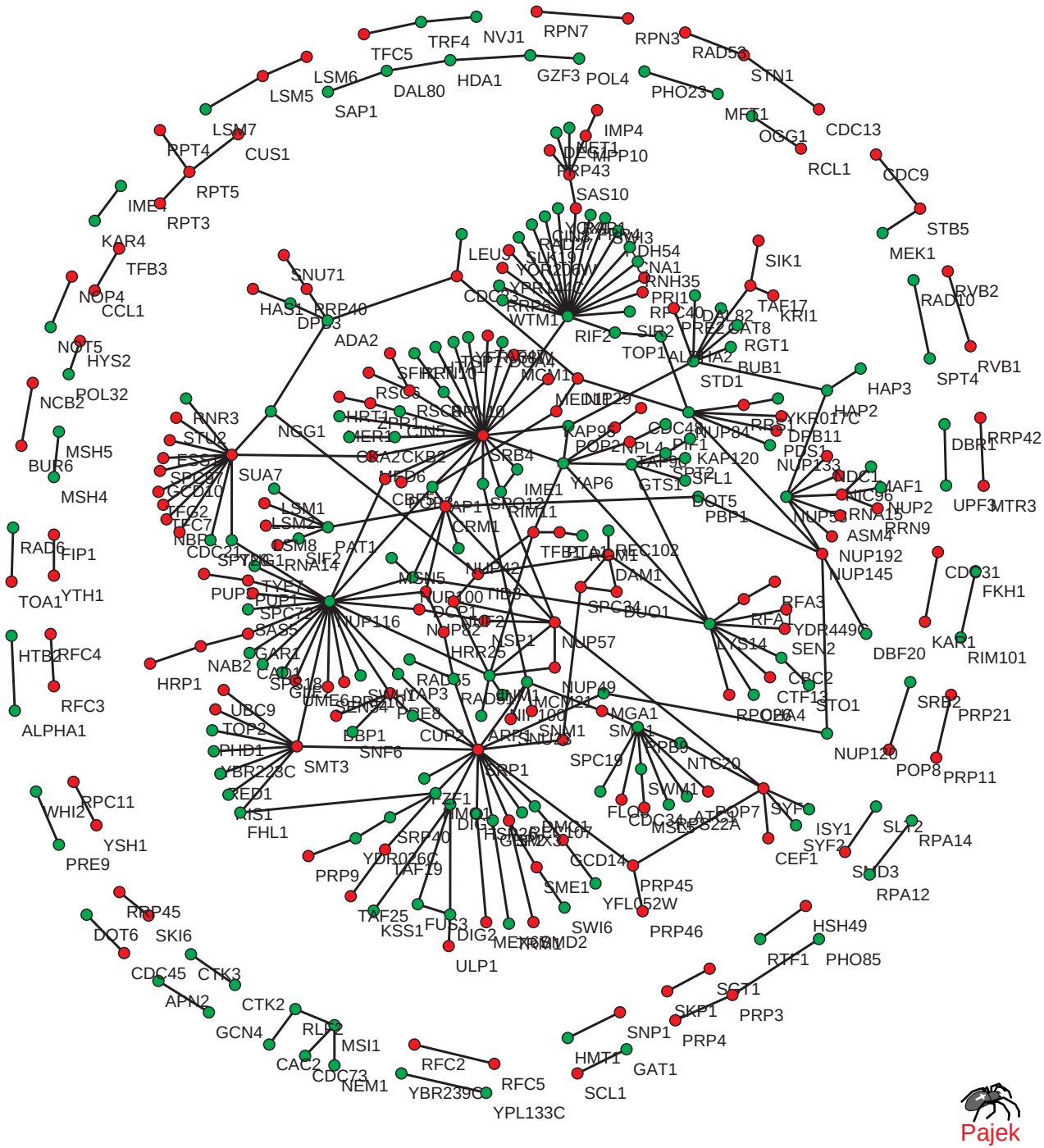


peripheral
(anti-hierarchical)

From A. Trusina, P. Minnhagen, SM, K. Sneppen, *Phys. Rev. Lett. (2004*)

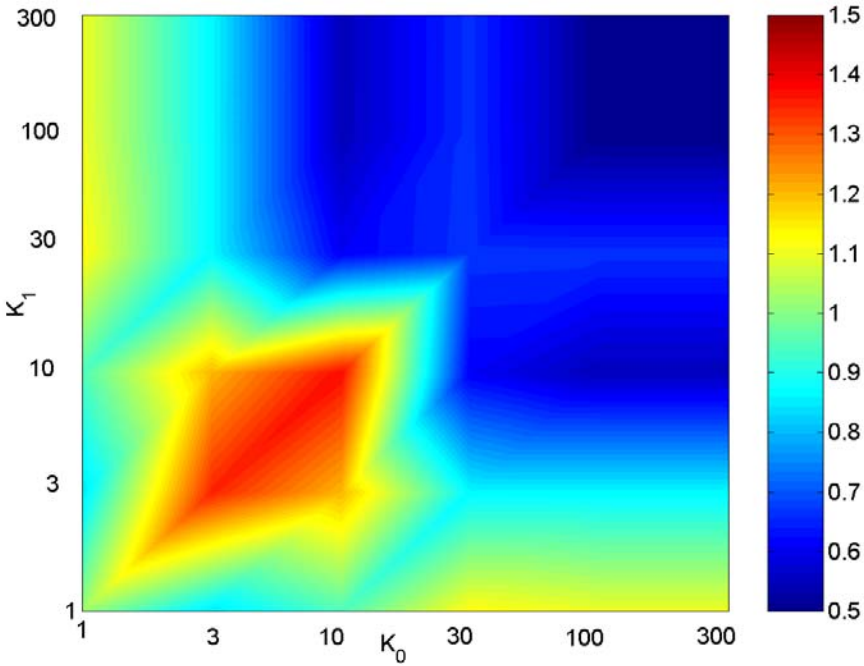
Correlation profile

- Count $N(k_0,k_1)$ – the number of links between nodes with connectivities k_0 and k_1
- Compare it to $N_r(k_0,k_1)$ – the same property in a random network
- Qualitative features are very noise-tolerant with respect to both false positives and false negatives

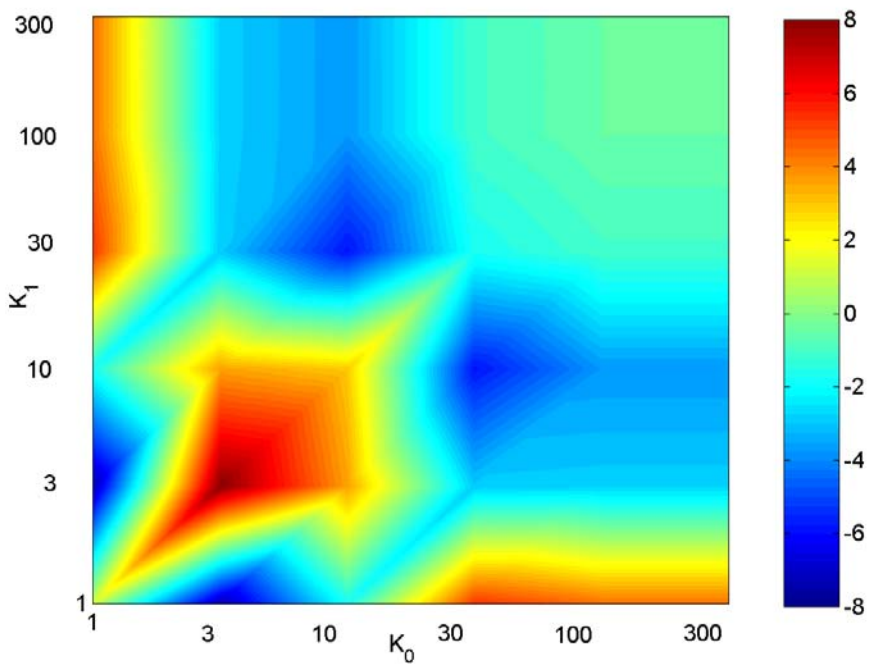

RPN7
NVJ1
TRF4
TFC5
RPN3
RAD53
STN1
GZF3
POL4
HDA1
LSM6
DAL80
LSM5
SAP1
PHO23
MET1
OGG1
CDC13
LSM7
CUS1
RPT4
IMP4
MPP10
RRP43
RCL1
CDC9
RPT5
SAS10
IME4
RPT3
STB5
KAR4
MEK1
TFB3
SIK1
SNU71
RVB2
NOP4
CCL1
HAS1
RRP40
DPB3
TAF17
KRI1
RAD10
WTM1
ADA2
RIF2
TOP1
RGT1
NOT5
HYS2
BUB1
STD1
SPT4
RVB1
HAP3
NCB2
POL32
HAP2
RNR3
NGG1
DBR1
RRP42
STU2
MSH5
BUR6
SUA7
NUP133
MSH4
IME1
YAP6
GTS1
DOT5
MAF1
UPF3
MTR3
NUP2
RRN9
ASM4
RAD6
FIP1
PAT1
NUP192
NUP145
CDC31
FKH1
TOA1
YTH1
DAM1
RFA3
RFA1
YDR449C
SEN2
DBF20
KAR1
RIM101
HTB2
RFC4
NAB2
HRP1
CBC2
CTF13
STO1
SRB2
PRP21
RFC3
ALPHA1
UBC9
TOP2
PHD1
YBR228C
SMT3
SNF6
NUP120
POP8
PRP11
RPC11
WHI2
RED1
HIS1
FHL1
YSH1
PRE9
SYF1
ISY1
SLU2
RPA14
SYF2
CEF1
SLD3
RPA12
PRP45
YFL052W
PRP46
HSH49
RTF1
PHO85
RRP45
DOT6
SKI6
CDC45
CTK3
APN2
CTK2
GCN4
MSI1
CAC2
CDC73
NEM1
RFC2
YBR239C
RFC5
YPL133C
SCL1
GAT1
HMT1
SNP1
PRP4
SKP1
SGT1
PRP3
ULP1
SME1
GCD14
PRP9
YDR026C
TAF19
SRP40
TAF25
KSS1
FUS3
DIG2
SWI6
SWM1
NTC20
SPC19
MGA1
NUP49
MCM21
NUP57
NSP1
HRR25
RAD55
NUP116
SAS5
GAR1
SPC42
DCP1
NUP82
BUO1
SPC34
LYS14
CDC48
NUP84
KAP120
POP2
NPL4
PIF1
SRB4
RSC6
RSC8
MCM1
MEP1
NIP29
YKR017C
DPB11
PDS1
NDC1
NIC96
RNA15
ASM4
NUP192
NUP145
CDC31
FKH1
KAR1
RIM101
DBF20
SRB2
PRP21
POP8
PRP11
NUP120
STO1
CTF13
CBC2
SEN2
YDR449C
RFA3
RFA1
DAM1
RIM11
CRM1
SRO12
YAP1
CIN5
ZPR1
HRT1
MER1
CKA2
CKB2
SUA7
LSM1
LSM8
SIF2
RNA14
TYE7
PUP1
SPC72
SAS5
GAR1
NAB2
HRP1
UBC9
TOP2
PHD1
YBR228C
SMT3
SNF6
BBP1
CUP2
ARF1
SNC1
SNM1
SPC19
NTC20
SWM1
FLC1
CDC34
MSL5
RPB9
SYF1
ISY1
SLU2
RPA14
SYF2
CEF1
SLD3
RPA12
PRP45
YFL052W
PRP46
HSH49
RTF1
PHO85
SGT1
PRP3
SKP1
PRP4
SNP1
HMT1
GAT1
SCL1

Correlation profile of the protein interaction network

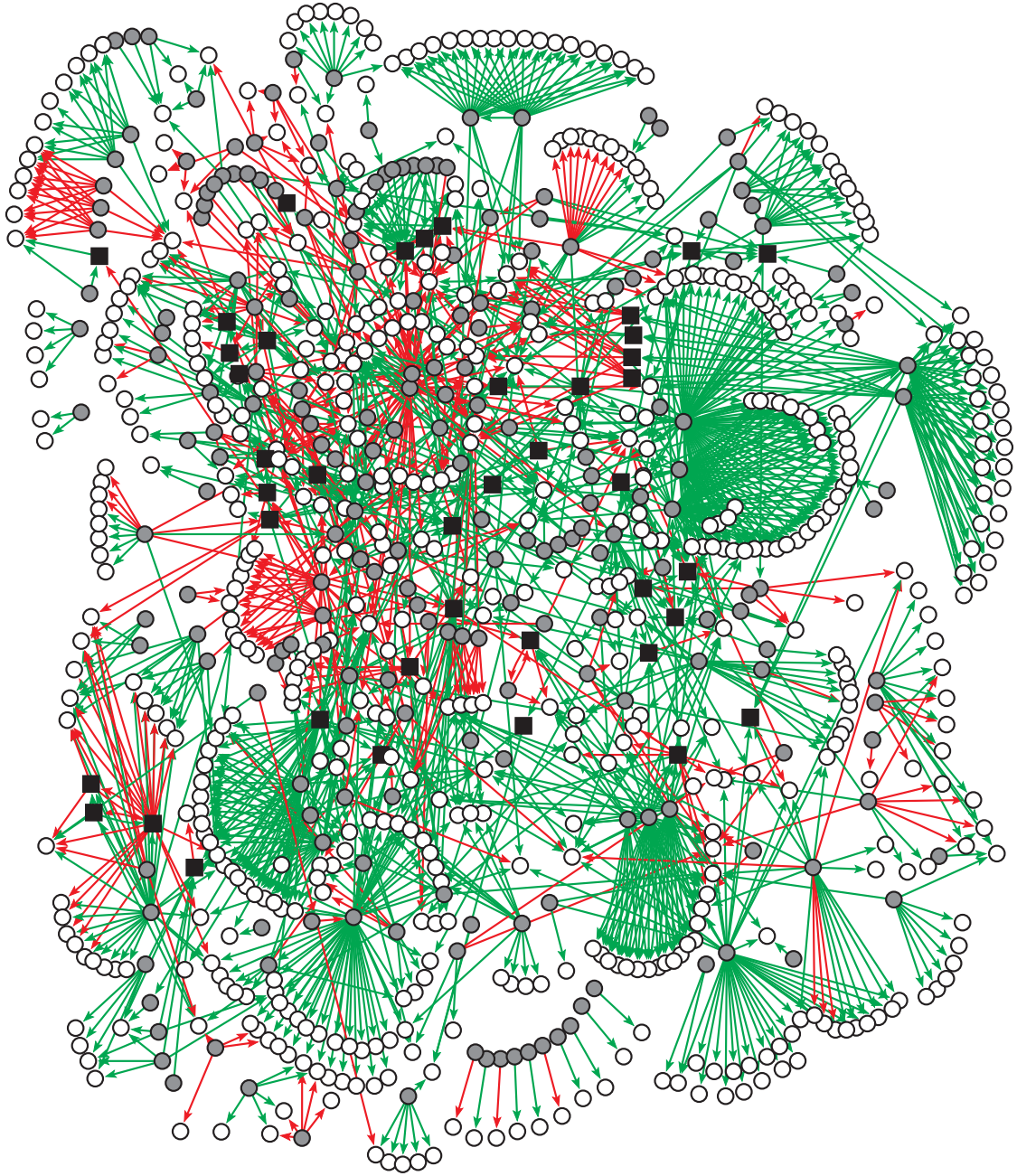
$R(k_0,k_1)=N(k_0,k_1)/N_r(k_0,k_1)$



$Z(k_0,k_1) =(N(k_0,k_1)-N_r(k_0,k_1))/\Delta N_r(k_0,k_1)$



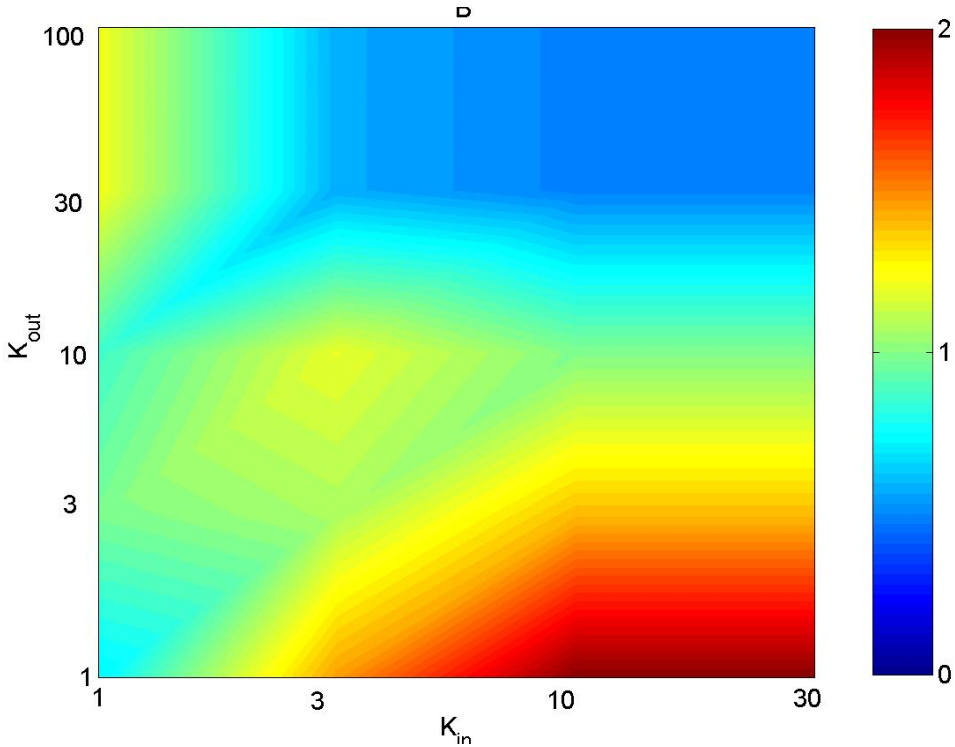
Similar profile is seen in the yeast regulatory network



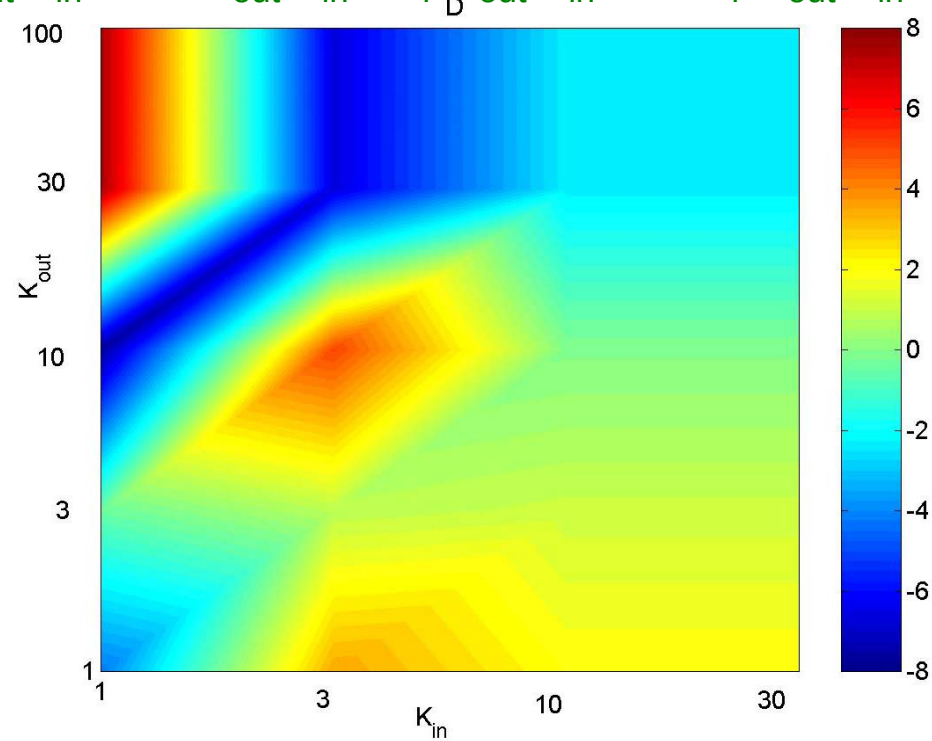
Pajek

Correlation profile of the yeast regulatory network

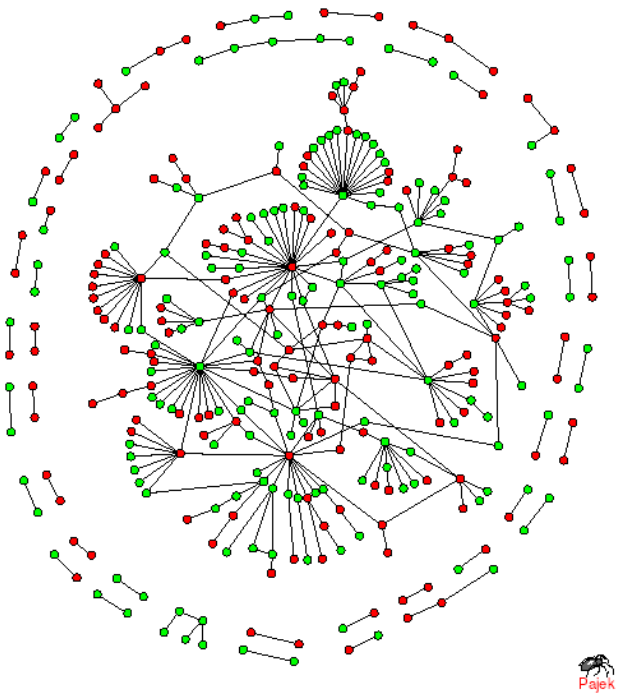
$R(k_{out}, k_{in})=N(k_{out}, k_{in})/N_r(k_{out},k_{in})$

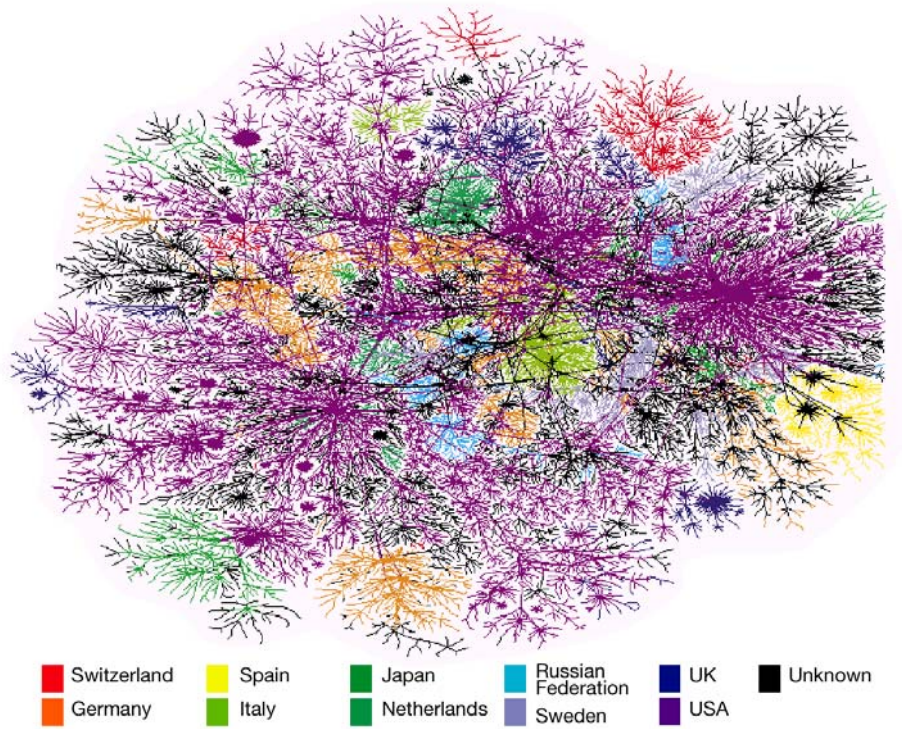


$Z(k_{out},k_{in})=(N(k_{out},k_{in})-N_r(k_{out},k_{in}))/\Delta N_r(k_{out},k_{in})$



Some scale-free networks may appear similar

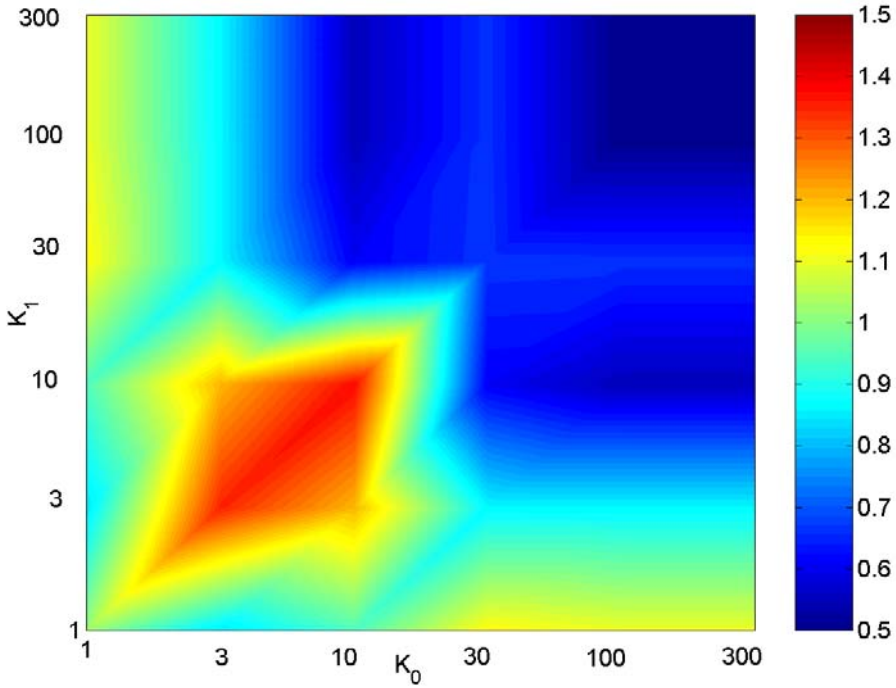




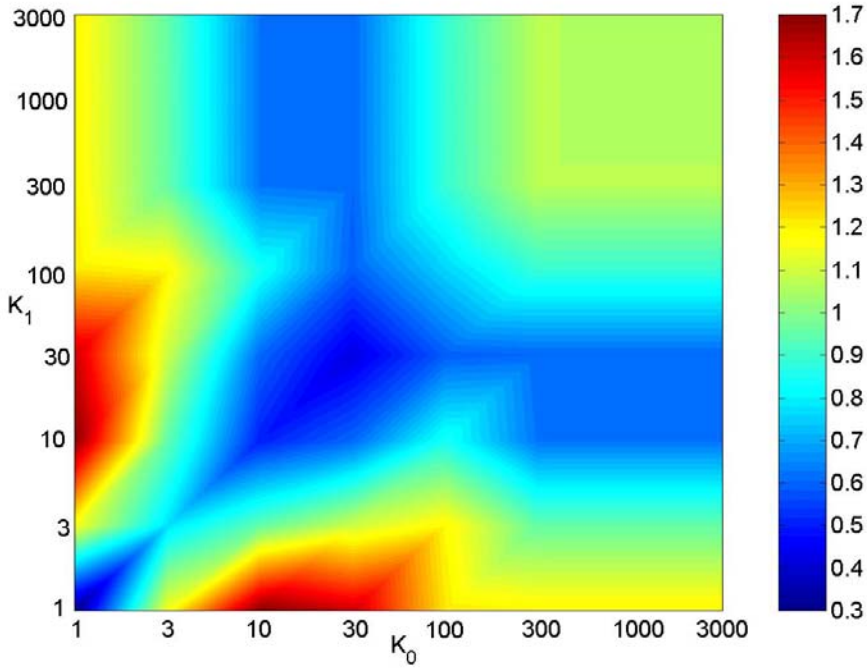
In both networks the degree distribution is scale-free $P(k) \sim k^{-\gamma}$ with $\gamma \sim 2.2\text{-}2.5$

But: correlation profiles give them unique identities

Protein interactions

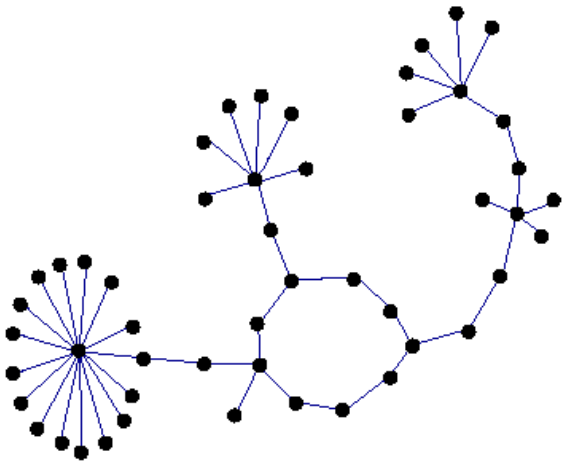


Internet

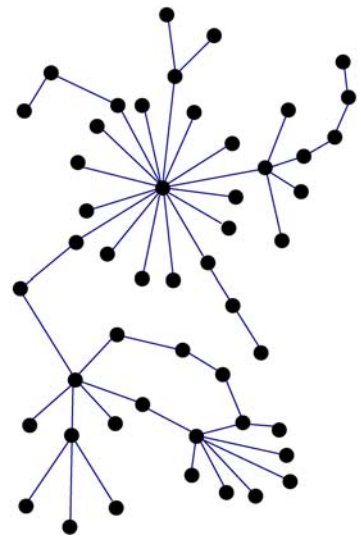


How to construct a proper random network?

Randomization of a network

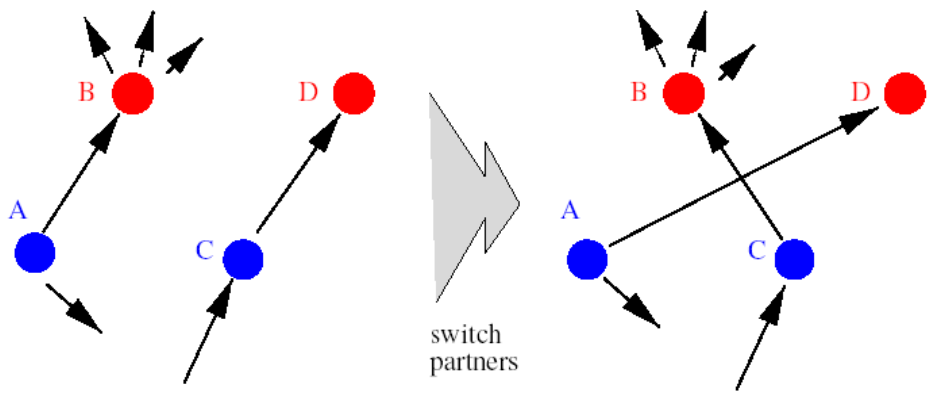


given complex network



random

Basic rewiring algorithm



- Randomly select and rewire two edges
- Repeat many times

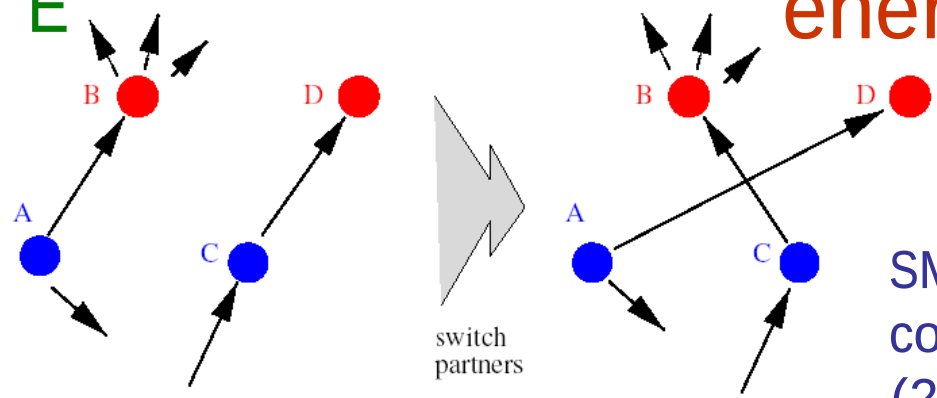
• R. Kannan, P. Tetali, and S. Vempala, *Random Structures and Algorithms* (1999)

• SM, K. Sneppen, *Science* (2002)

Metropolis rewiring algorithm

"energy" E

"energy" E+ΔE



SM, K. Sneppen: cond-mat preprint (2002), Physica A (2004)

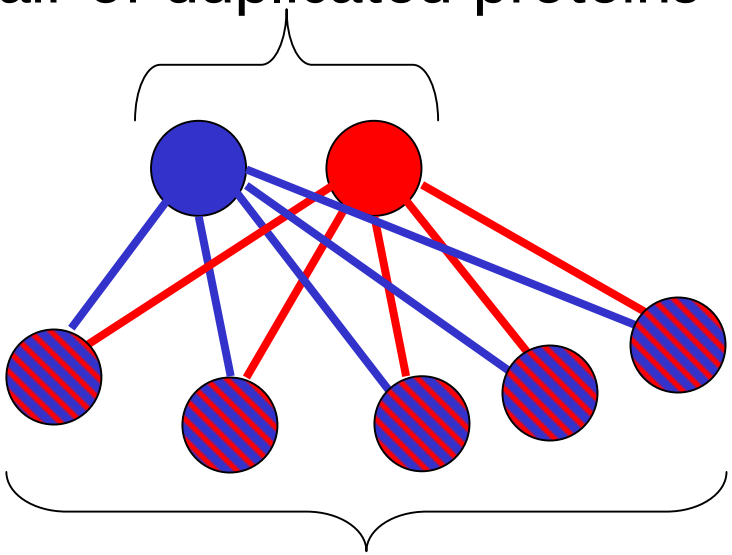
- Randomly select two edges
- Calculate change ΔE in "energy function" $E=(N_{actual}-N_{desired})^2/N_{desired}$
- Rewire with probability $p=\exp(-\Delta E/T)$

How do protein networks evolve?

Gene duplication

Right after duplication

Pair of duplicated proteins

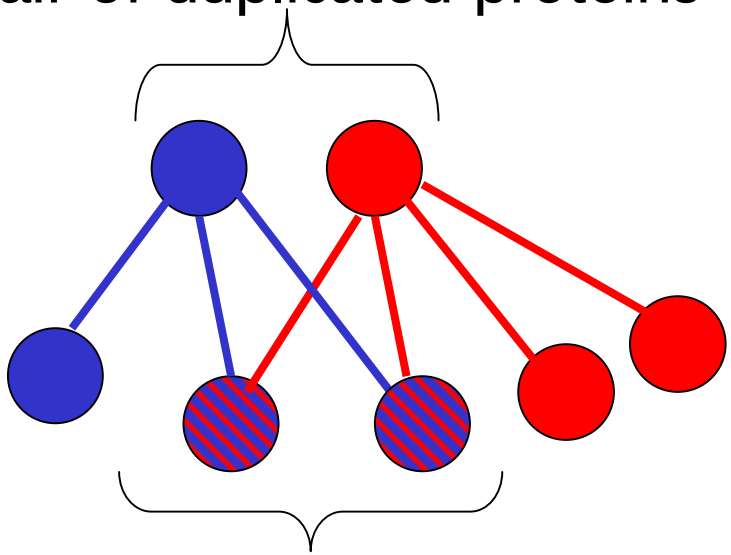


5 Shared interactions

Overlap=5/5=1.0

After some time

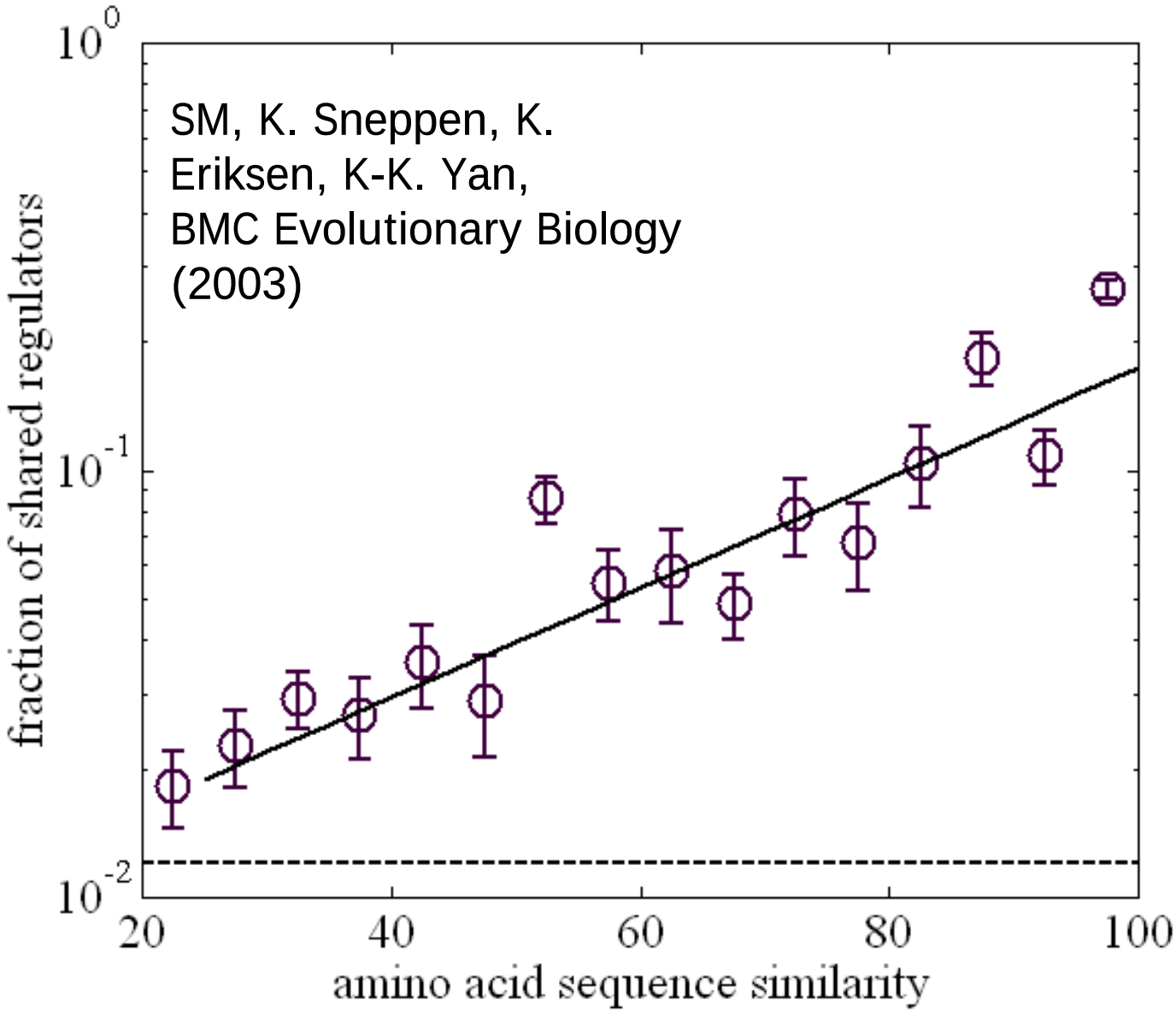
Pair of duplicated proteins

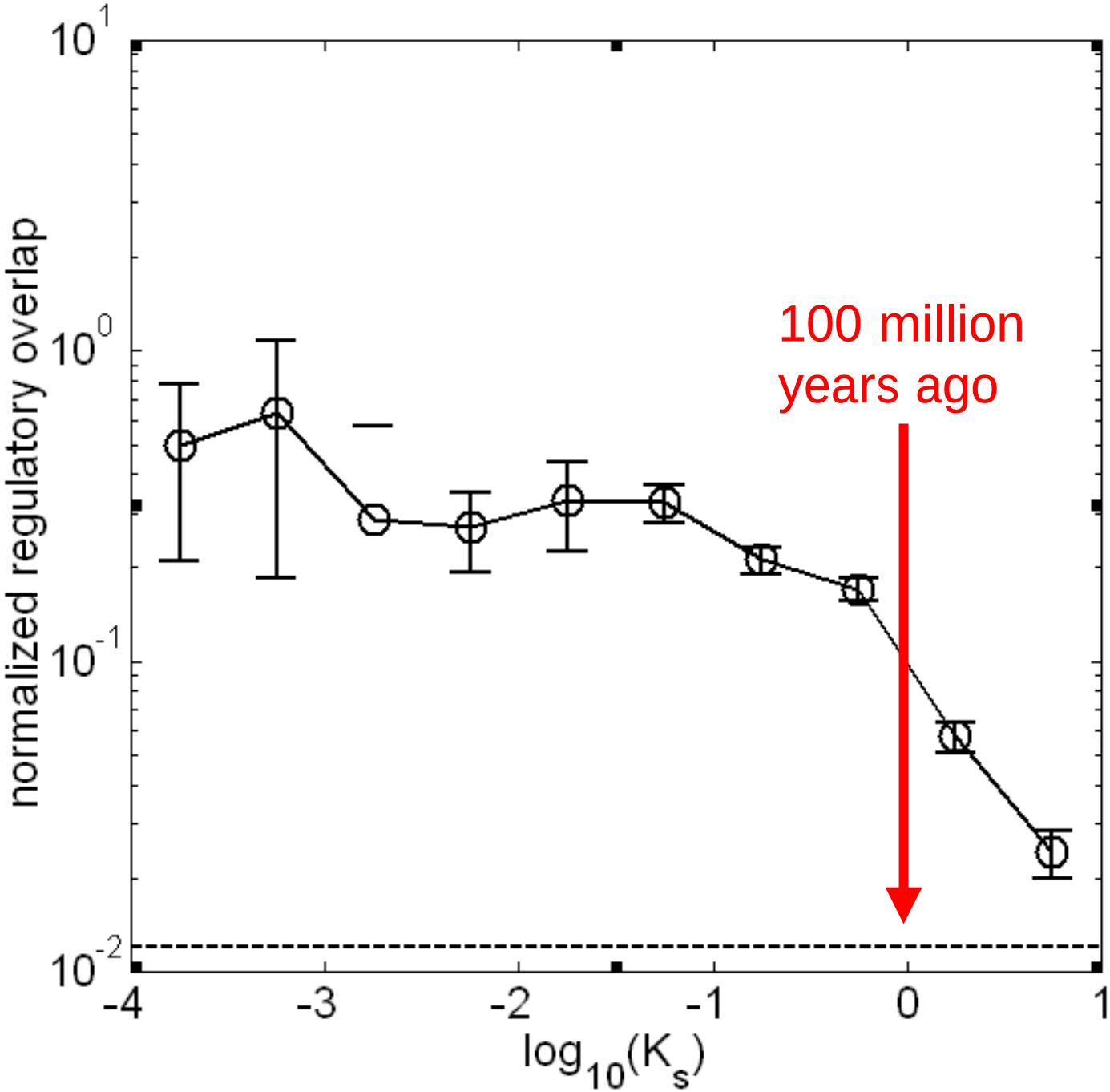


2 Shared interactions

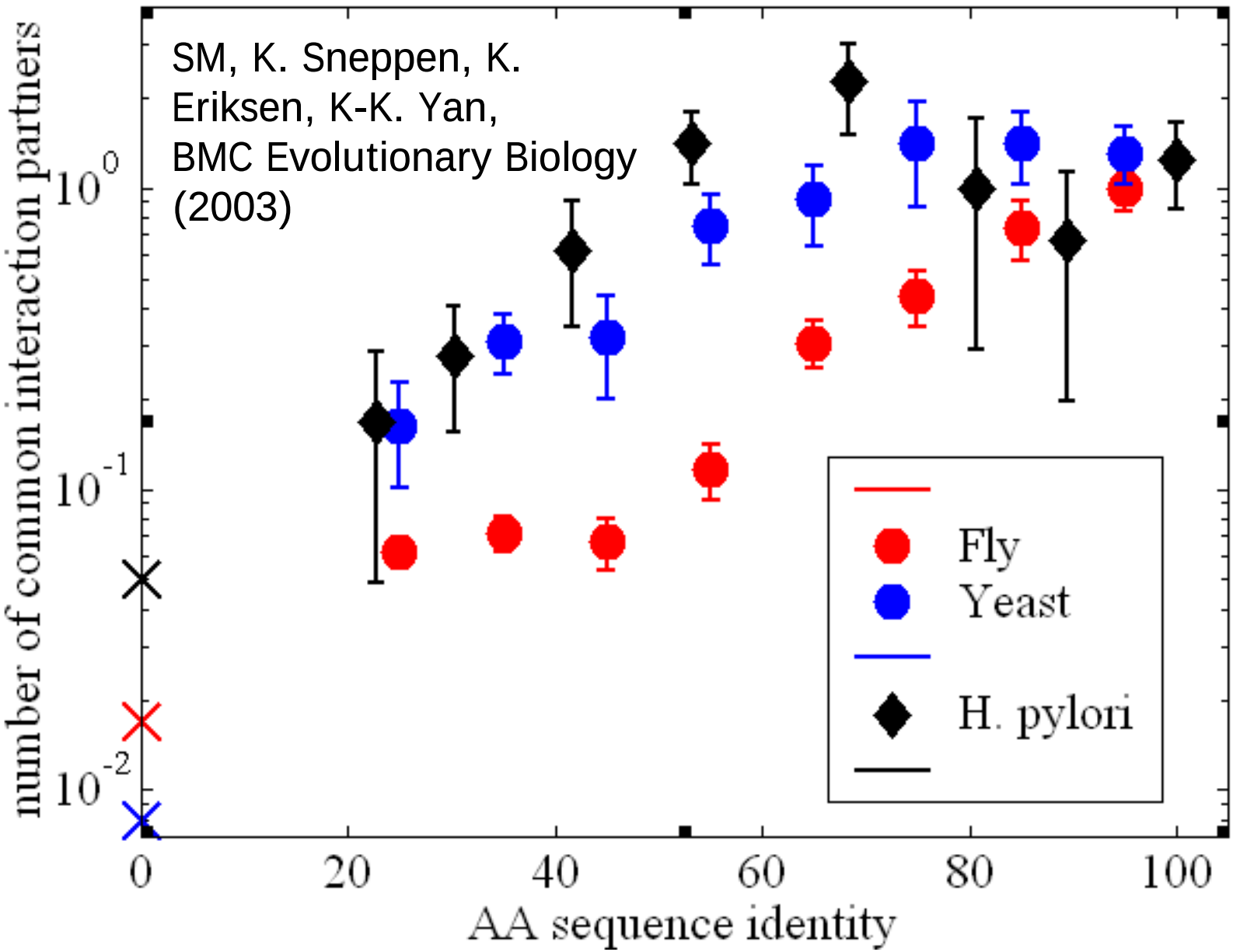
Overlap=2/5=0.4

Yeast regulatory network




100 million
years ago
normalized regulatory overlap
10^{1}
10^{0}
10^{-1}
10^{-2}
-4
-3
-2
-1
0
1
$\log_{10}(K_s)$

Protein interaction networks

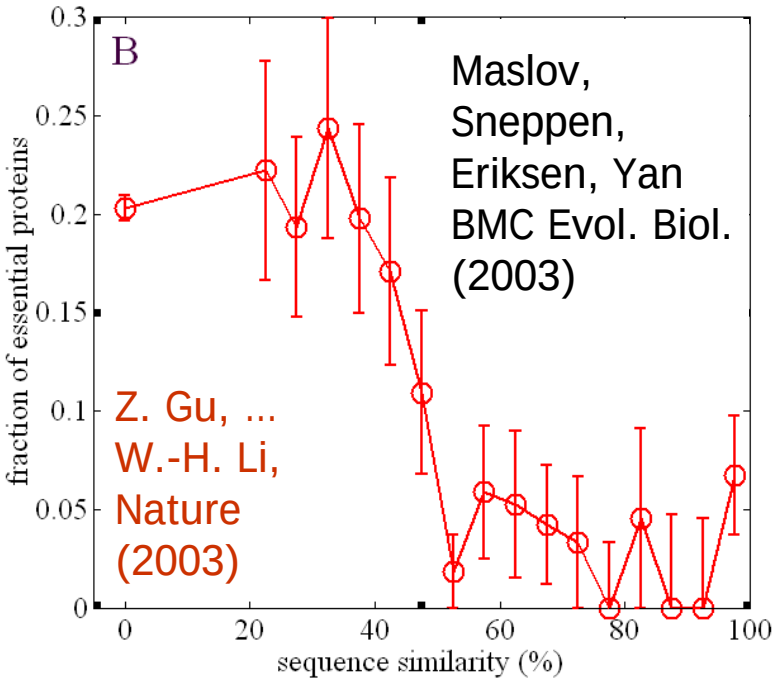


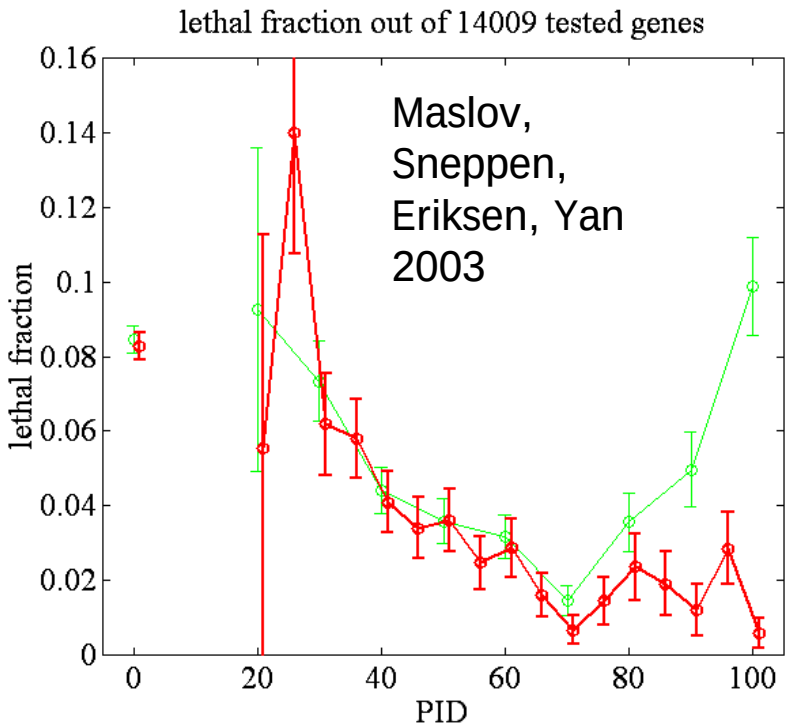
Why so many genes could be deleted without any consequences?

Possible sources of robustness to gene deletions

- Backup via the network (e.g. metabolic network could have several pathways for the production of the necessary metabolite)
- Not all genes are needed under a given condition (say rich growth medium)
- Affects fitness but not enough to kill
- Protection by closely related homologs in the genome

Protective effect of duplicates





Yeast

Worm

Summary

- There are many kinds of protein networks
- Networks in more complex organisms are more interconnected
- Most have hubs – highly connected proteins
- Hubs often avoid each other (networks are anti-hierarchical)
- There are many self-interacting proteins. Probability to self-interact linearly scales with the degree K.
- Networks evolve by gene duplications
- Robustness is often (but not always) provided by gene duplicates

Part 2

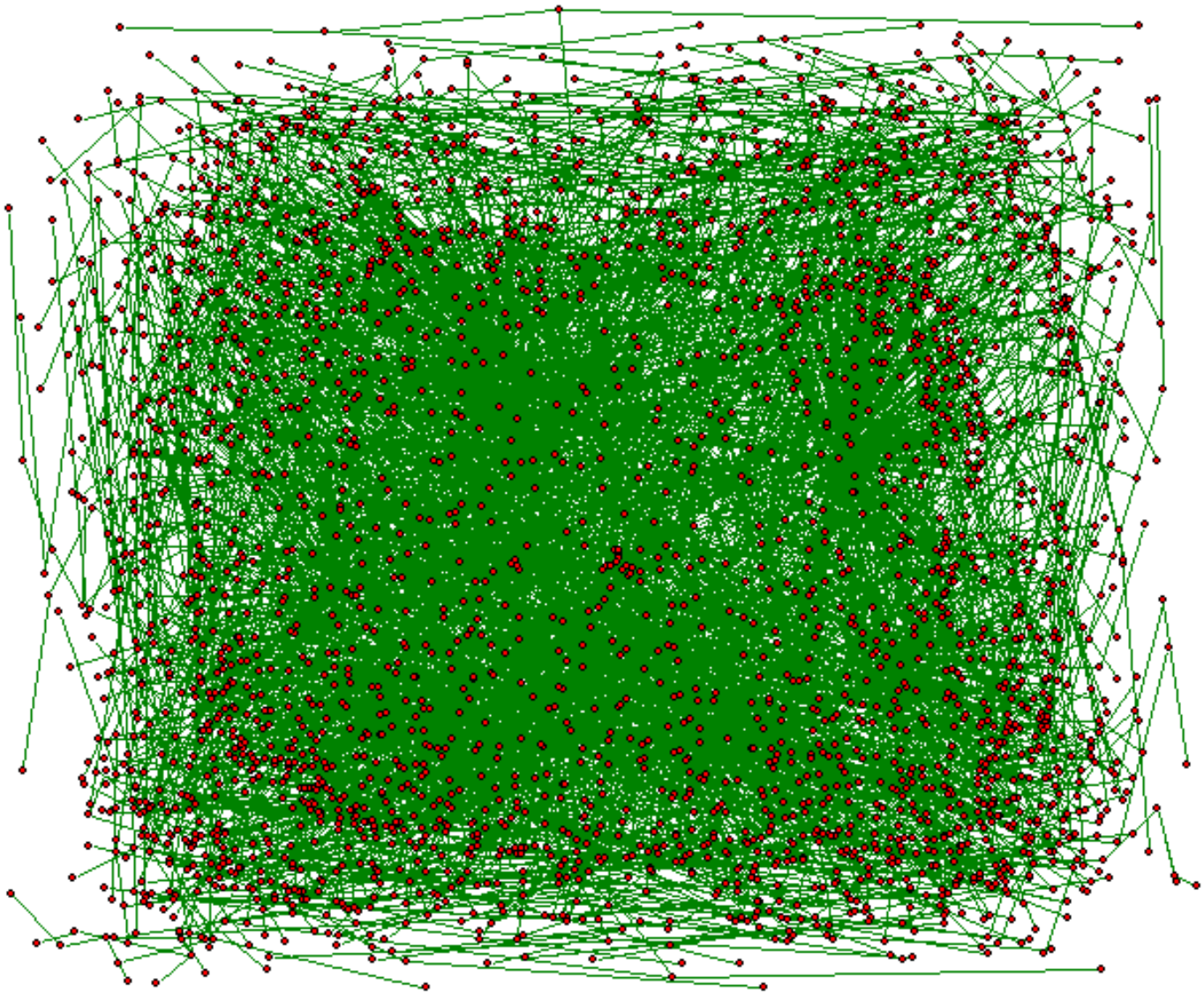
Modules in networks and how to detect them using the Random walks/diffusion

K. Eriksen, I. Simonsen, SM, K. Sneppen, PRL (2003)

What is a module?

- Nodes in a given module (or community group or functional unit) tend to connect with other nodes in the same module
  - Biology: proteins of the same function or sub-cellular localization
  - WWW – websites on a common topic
  - Internet – geography or organization (e.g. military)

Do you see any modules here?



Random walkers on a network

- Study the behavior of many VIRTUAL random walkers on a network
- At each time step each random walker steps on a randomly selected neighbor
- They equilibrate to a steady state $n_i \sim k_i$ (solid state physics: n_i = const)
- Slow modes allow to detect modules and extreme edges

Matrix formalism

$$n_i(t+1) = \sum_j \hat{T}_{ij} n_j(t)$$

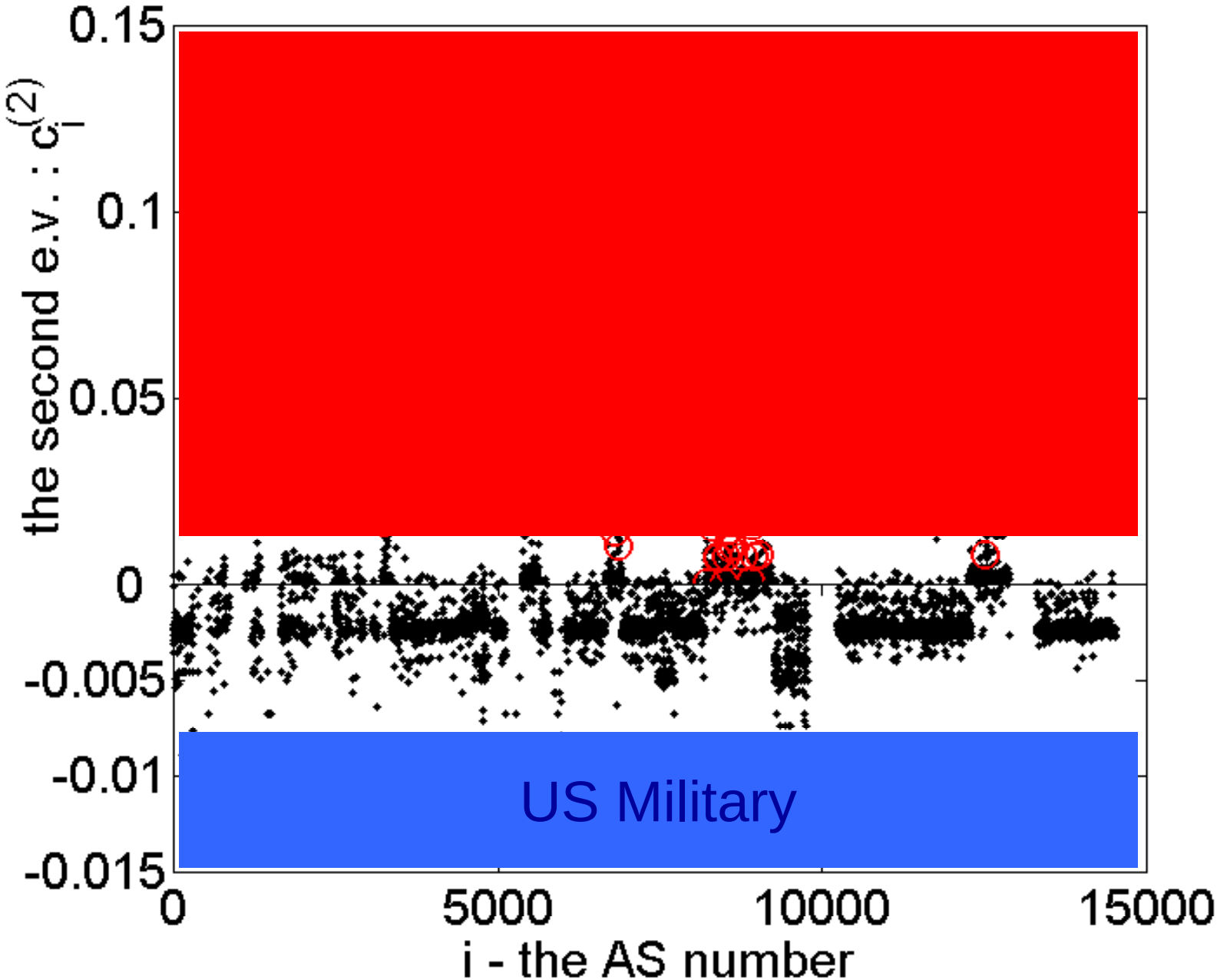
$$\hat{T}_{ij} = \begin{cases} 1/K_j & \text{if j} \leftrightarrow \text{i} \\ 0 & \text{otherwise} \end{cases}$$

Eigenvectors of the transfer matrix T_{ij}

$$\lambda^{(\alpha)} v_i^{(\alpha)} = \sum_j \hat{T}_{ij} v_j^{(\alpha)}$$

$$n_i(t) = \left(\lambda^{(\alpha)}\right)^t v_i^{(\alpha)}$$

$$-1 \le \lambda^{(\alpha)} \le 1$$


0.15
0.1
0.05
0
-0.005
-0.01
-0.015
the second e.v. : $c_i^{(2)}$
0
5000
10000
15000
i - the AS number
US Military

2 0.9626 RU RU RU RU CA RU RU
?? ?? US US US US ??
(US Department of Defence)

3 0.9561 ?? FR FR FR ?? FR ??

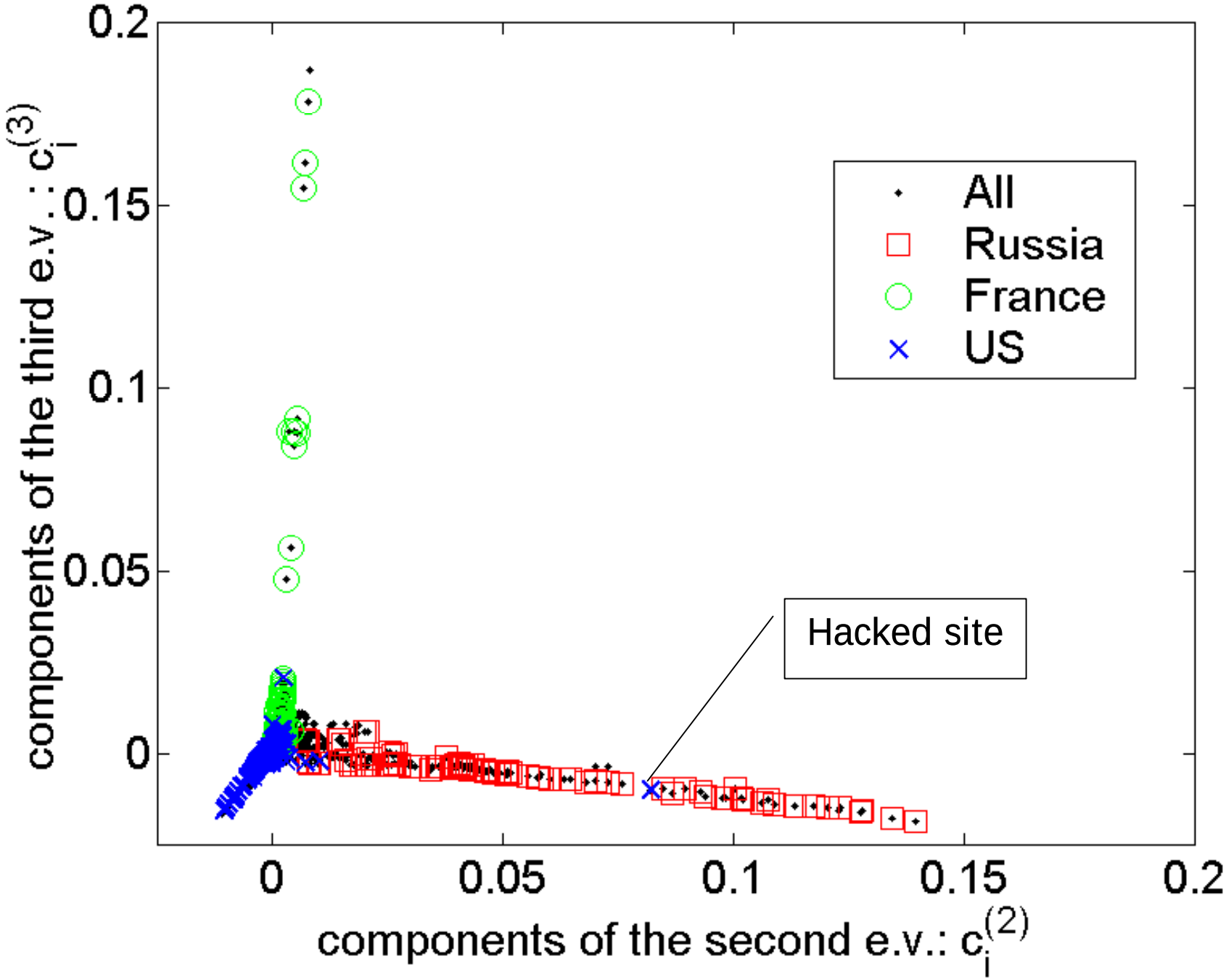
RU RU RU ?? ?? RU ??

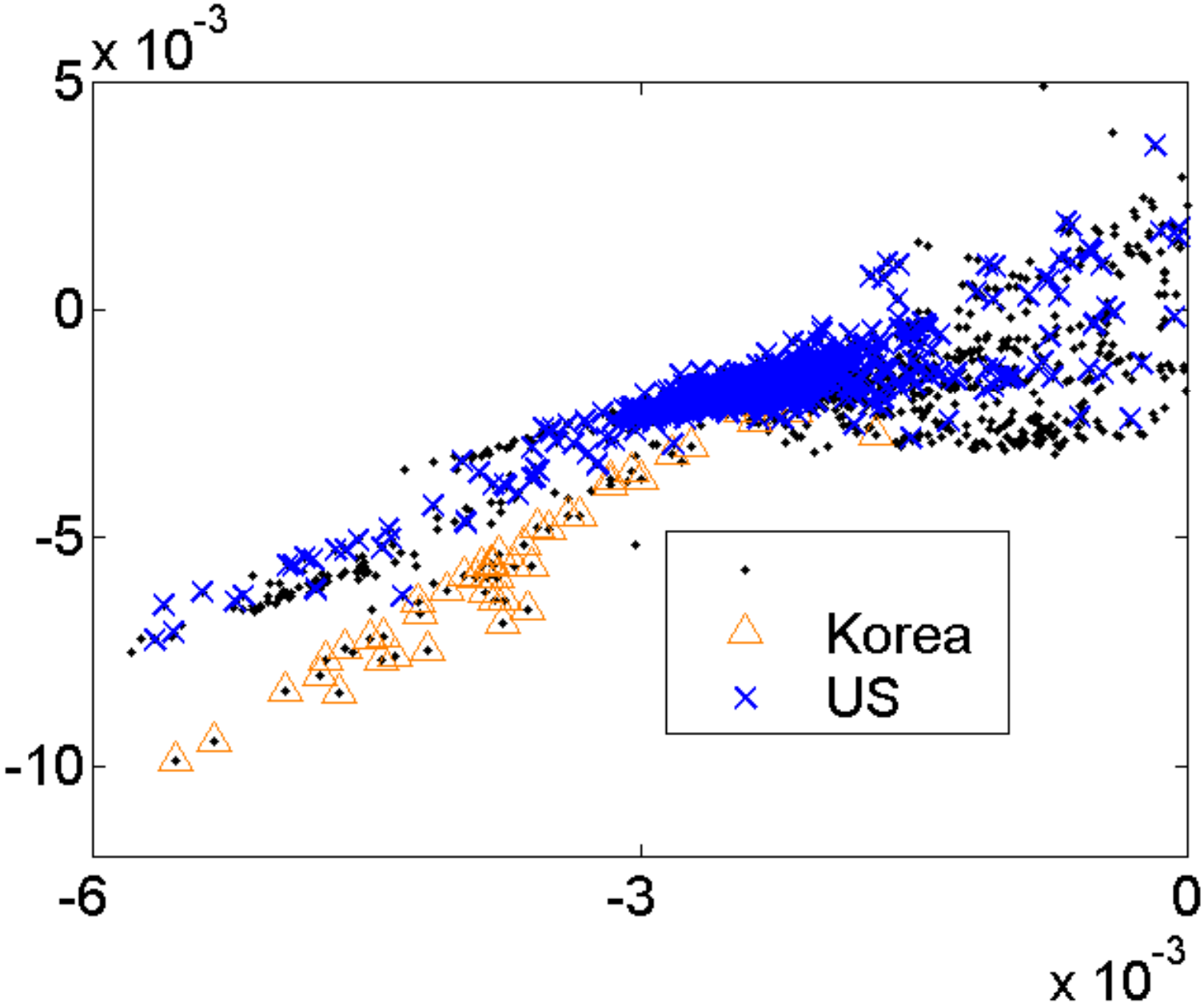
4 0.9523 US ?? US ?? ?? ?? ?? (US Navy)

NZ NZ NZ NZ NZ NZ NZ

5. 0.9474 KR KR KR KR KR ?? KR

UA UA UA UA UA UA UA


0.2
0.15
0.1
0.05
0
0
0.05
0.1
0.15
0.2
components of the third e.v.: $c_i^{(3)}$
components of the second e.v.: $c_i^{(2)}$
All
Russia
France
US
Hacked site


x 10^-3
5
0
-5
-10
-6
-3
0
x 10^-3
Korea
US

How Modules in the WWW influence Google ranking

H. Xie, K.-K. Yan, SM, cond-mat/0409087

How Google works?

- Too many hits to a typical query: need to rank
- One could rank the importance of webpages by K_{in}, but:
  - Too democratic: It doesn't take into account importance of webpages sending links
  - it's easy to trick and artificially boost the rank
- Google's solution: simulate the behavior of many "random surfers" and then count the number of hits for every webpage
  - Popular pages send more surfers your way → the Google weight is proportional to K_{in} weighted by popularity
  - Surfer get bored following links → use α=0.15 for random jumps without any links
  - Last rule also solves the problem that some pages have no out-links

Mathematics of the Google

- Google solves a self-consistent Eq.:

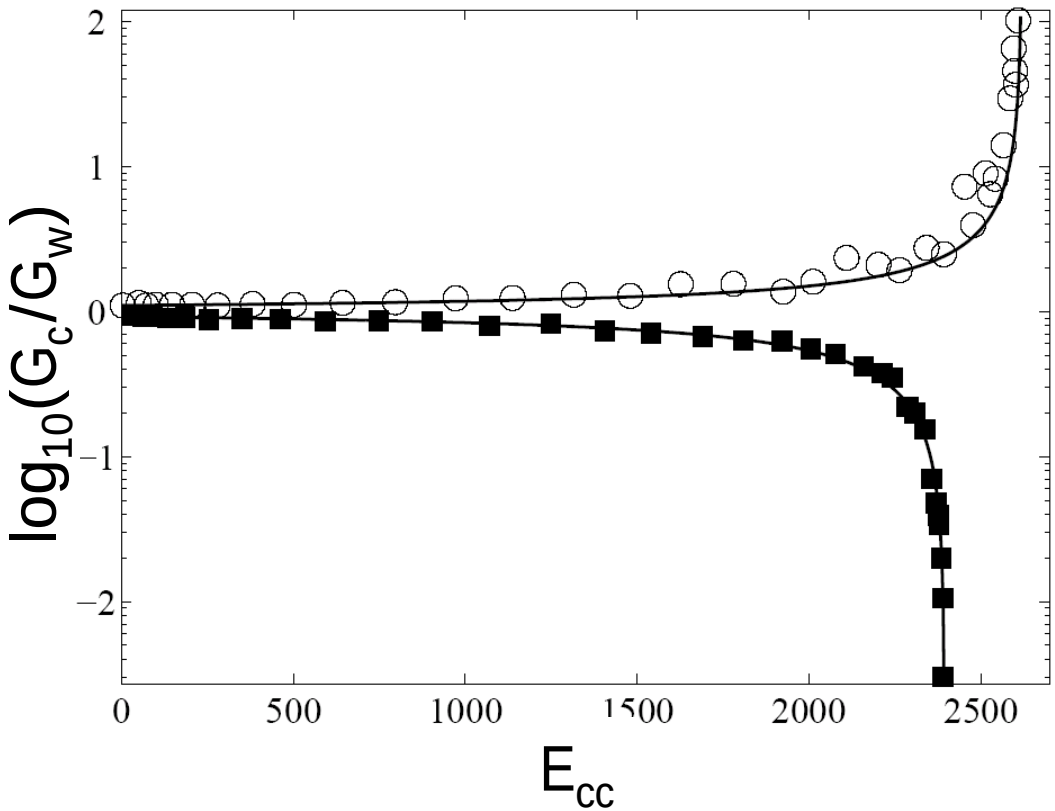
  $G_i \sim \sum_j T_{ij} G_j$ - find principal eigenvector
- $T_{ij} = A_{ij}/K_{out}(j)$
- Model with random jumps:

  $G_i \sim (1-\alpha)\sum_j T_{ij} G_j + \alpha \sum_j G_j$

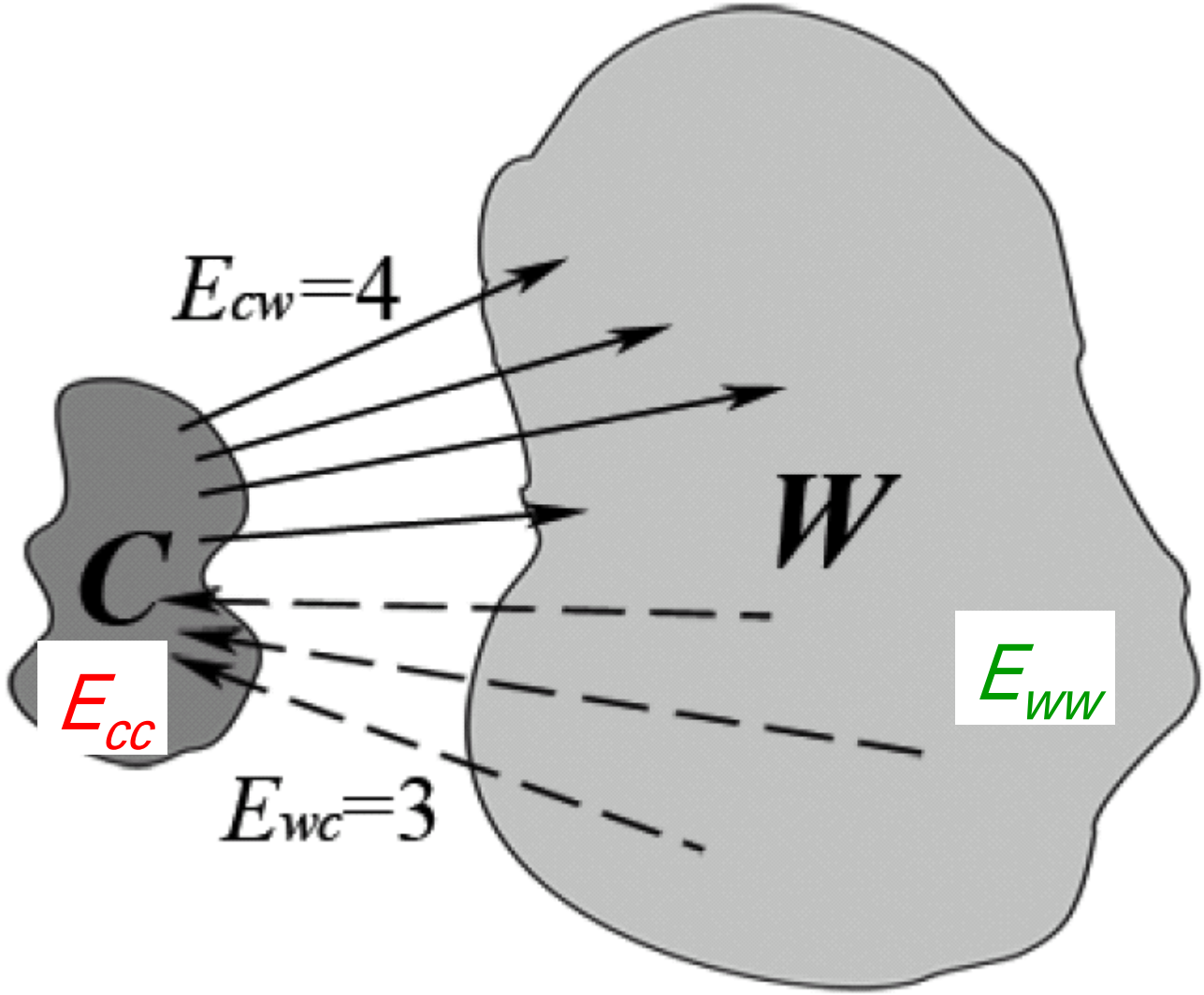
How do WWW communities (modules) influence the G_i?

- Naïve argument: communities tend to "trap" random surfers → they should increase the Google ranking of nodes in the community

Test of a naïve argument



- Naïve argument is completely wrong!


E_{cw}=4
C
W
E_{cc}
E_{ww}
E_{wc}=3

- G_c – average Google rank of pages in the community; G_w – in the outside world
- $E_{cw}\ G_c/\langle K_{out}\rangle_c$ – current from C to W
- must be equal to:
  $E_{wc}\ G_w/\langle K_{out}\rangle_w$ – current from W to C

$$\frac{G_c}{G_w} = \frac{E_{wc}}{E_{cw}} \cdot \frac{\langle K_{out}\rangle_c}{\langle K_{out}\rangle_w}$$

- G_c/G_w depends on the ratio between E_{cw} and E_{wc} – the number of edges (hyperlinks) between the community and the world

$$E_{cc} + E_{wc} = N_c \langle K_{in} \rangle_c$$

$$E_{cc} + E_{cw} = N_c \langle K_{out} \rangle_c ;$$

$$\boxed{\frac{G_c}{G_w} = \left(\frac{\langle K_{in} \rangle_c N_c - E_{cc}}{\langle K_{out} \rangle_c N_c - E_{cc}} \right) \cdot \frac{\langle K_{out} \rangle_c}{\langle K_{out} \rangle_w}}$$

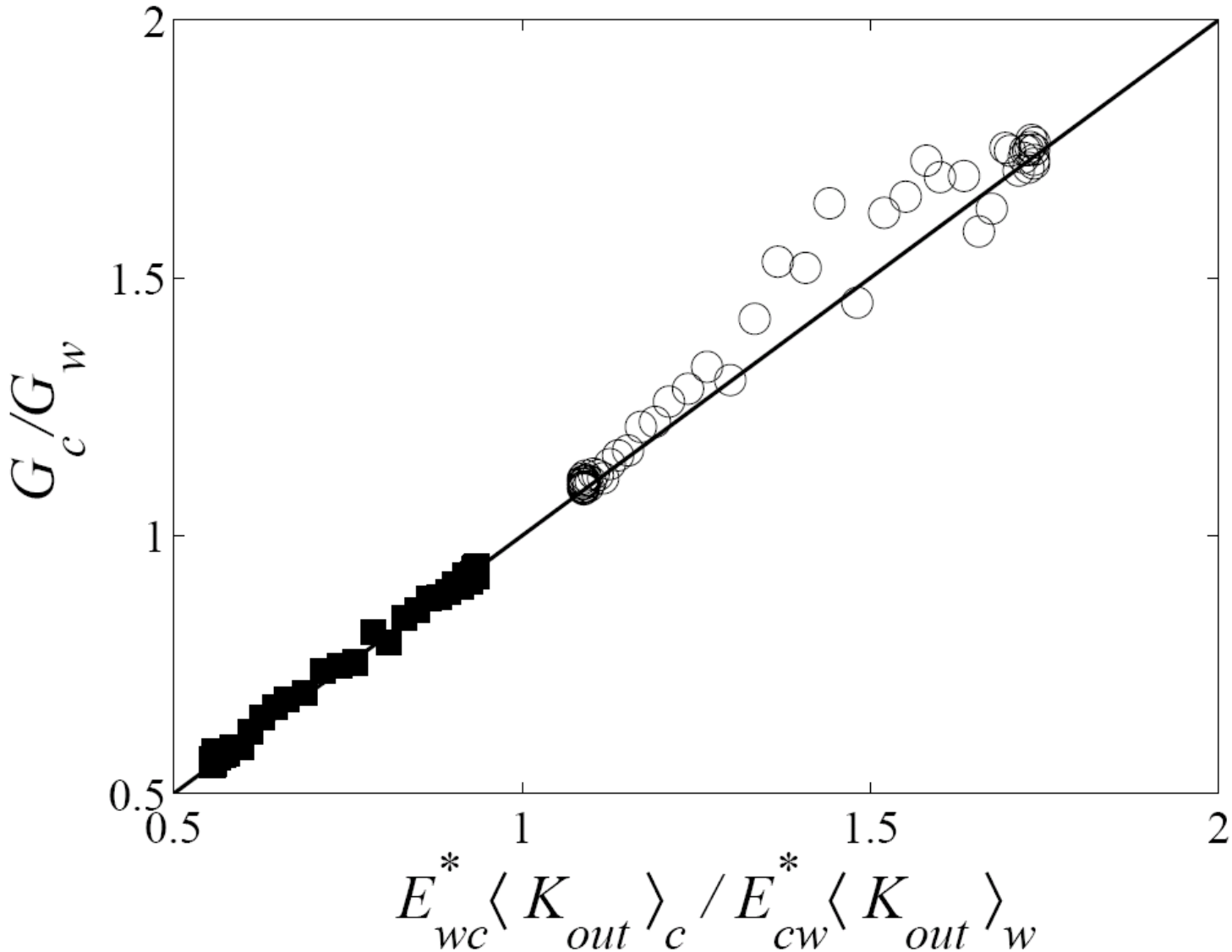
$$J_{wc} = (1-\alpha)G_w E_{wc}/\langle K_{out}\rangle_w + \alpha G_w N_c$$

$$J_{cw} = (1-\alpha)G_c E_{cw}/\langle K_{out}\rangle_c + \alpha G_c N_c$$

$$E^*_{cw} = E_{cw}(1-\alpha) + N_c\langle K_{out}\rangle_c \alpha$$

$$E^*_{wc} = E_{wc}(1-\alpha) + N_c\langle K_{out}\rangle_w \alpha$$

$$\boxed{\frac{G_c}{G_w} = \frac{E^*_{wc}}{E^*_{cw}} \cdot \frac{\langle K_{out}\rangle_c}{\langle K_{out}\rangle_w}}$$


2
1.5
1
0.5
0.5
1
1.5
2
G_c / G_w
$E^*_{wc} \langle K_{out} \rangle_c / E^*_{cw} \langle K_{out} \rangle_w$

Networks with artificial communities

- To test let's generate a scale-free network with an artificial community of N_c pre-selected nodes
- Use metropolis algorithm with H=-(# of intra-community nodes) and some inverse temperature β
- Detailed balance: $$E_{cw}E_{wc} = E_{cc}E_{ww}e^{-\beta}$$

Collaborators:

- Kim Sneppen – Nordita + NBI
- Kasper Eriksen –Lund U.
- Ingve Simonsen – U. of Trondheim

- Huafen Xie – City University of NY
- Koon-Kiu Yan - Stony Brook U.

Summary

- Diffusion process and modules (communities) in a network influence each other
- In the “hardware” part of the Internet – the network of routers (Autonomous systems) -- diffusion allows one to detect modules and extreme edges
- In the “software” part – WWW communities affect Google ranking in a non-trivial way

THE END