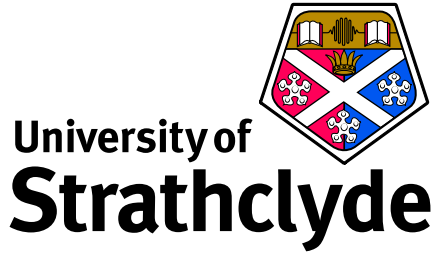
Spectral Algorithms for Biological Networks

Des Higham
Department of Mathematics
University of Strathclyde

djh@maths.strath.ac.uk



Collaborators

- **Julie Morrison**, formerly U. Strathclyde
- **David Gilbert**, U. Glasgow
- **Rainer Breitling**, U. Groningen

A lock-and-key model for protein-protein interactions, **Bioinformatics**, 2006

- **Nataša Pržulj**, U. C. Irvine

Modelling protein-protein interaction networks via a stickiness index, **J. Royal Society Interface**, 2006

- **Marija Rašajski**, U. C. Irvine & U. Belgrade

EPSRC-funded project

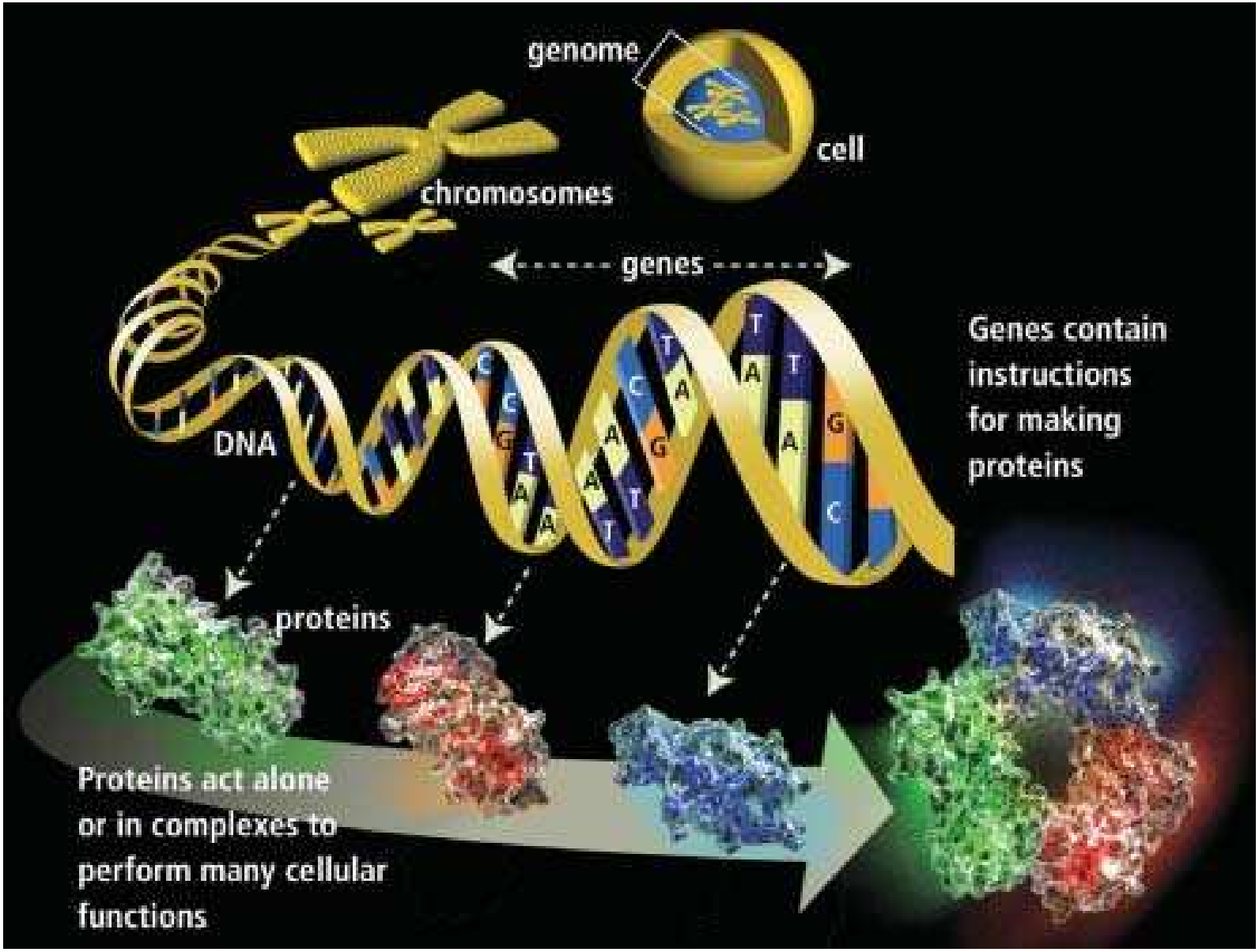
Theory and Tools for Complex Biological Networks (2007–2010)

- **Peter Grindrod**, University of Reading
- **Gabriela Kalna**, University of Strathclyde
- **Alastair Spence**, University of Bath
- **Zhivko Stoyanov**, University of Bath
- **Keith Vass**, Beatson Inst. for Cancer Research

Overview

- **Protein-protein interaction** (PPI) networks
- **Random graph** models
- **Geometric** model
- Algorithm for **testing geometric model**
- **Lock-and-key** model
- Algorithm for **discovering locks and keys**
- Results on **biological data**

Central Dogma of Molecular Biology

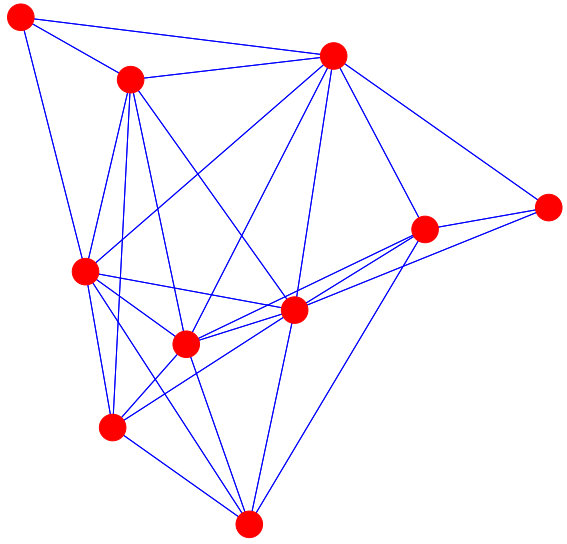


Yeast 2-Hybrid Protein-Protein Interaction Networks

Data:

- list of N **proteins** (nodes)
- list of **protein pairs** (edges)

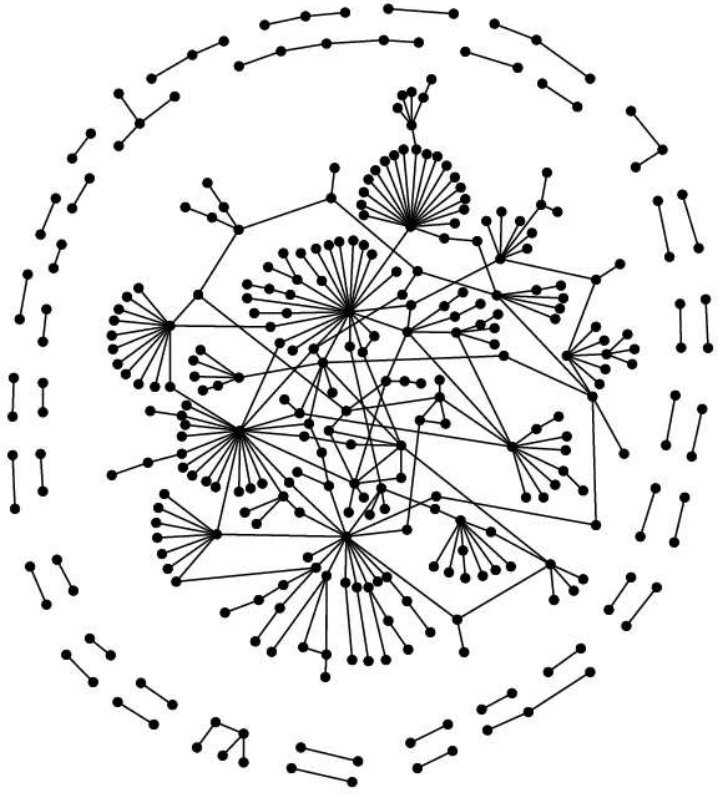
This is an **undirected, unweighted graph**



Also, a symmetric $N \times N$ matrix of 0's and 1's

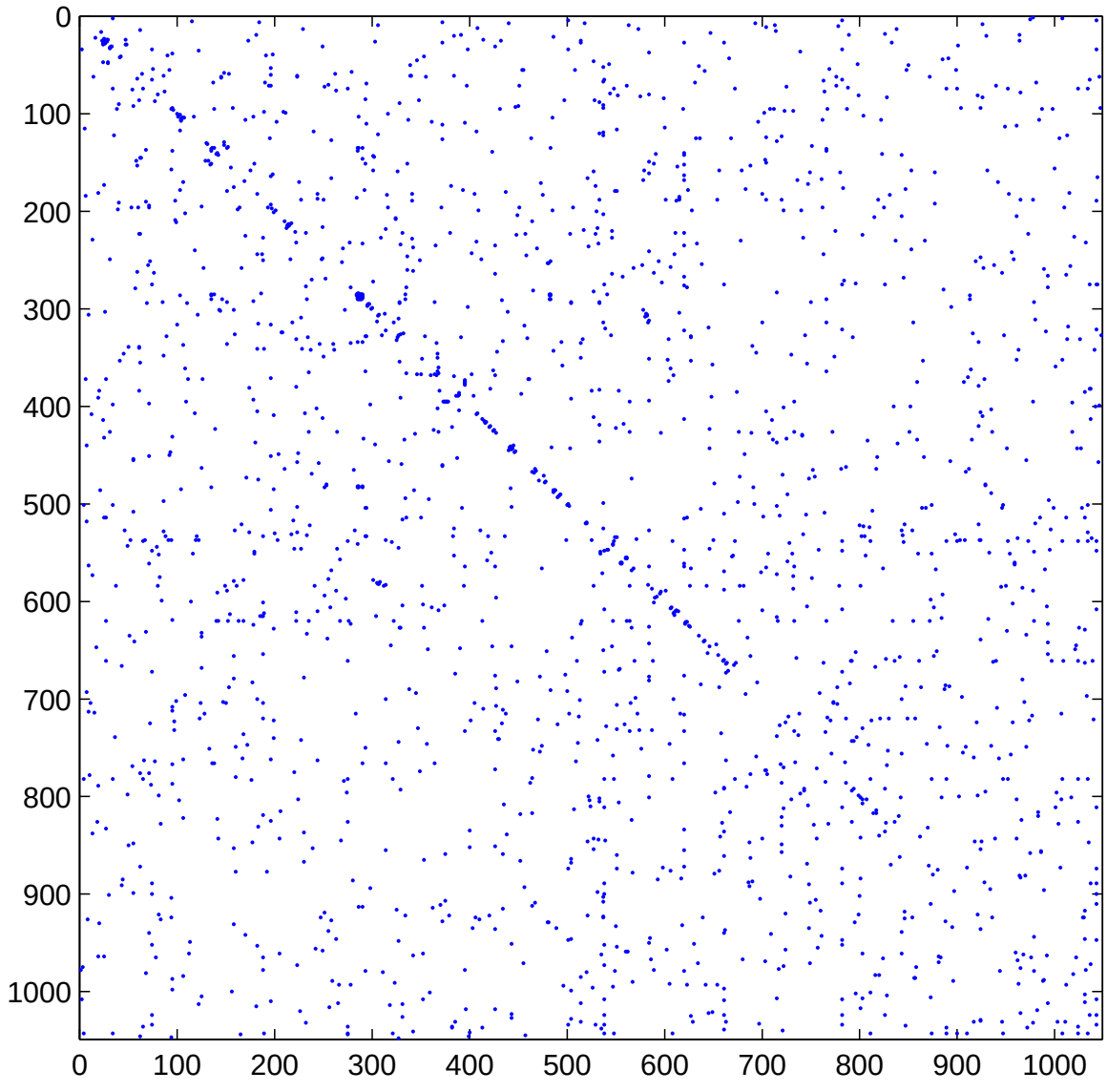
Yeast has $N \approx 3,000$

Uetz et al. 2000, Yeast PPI

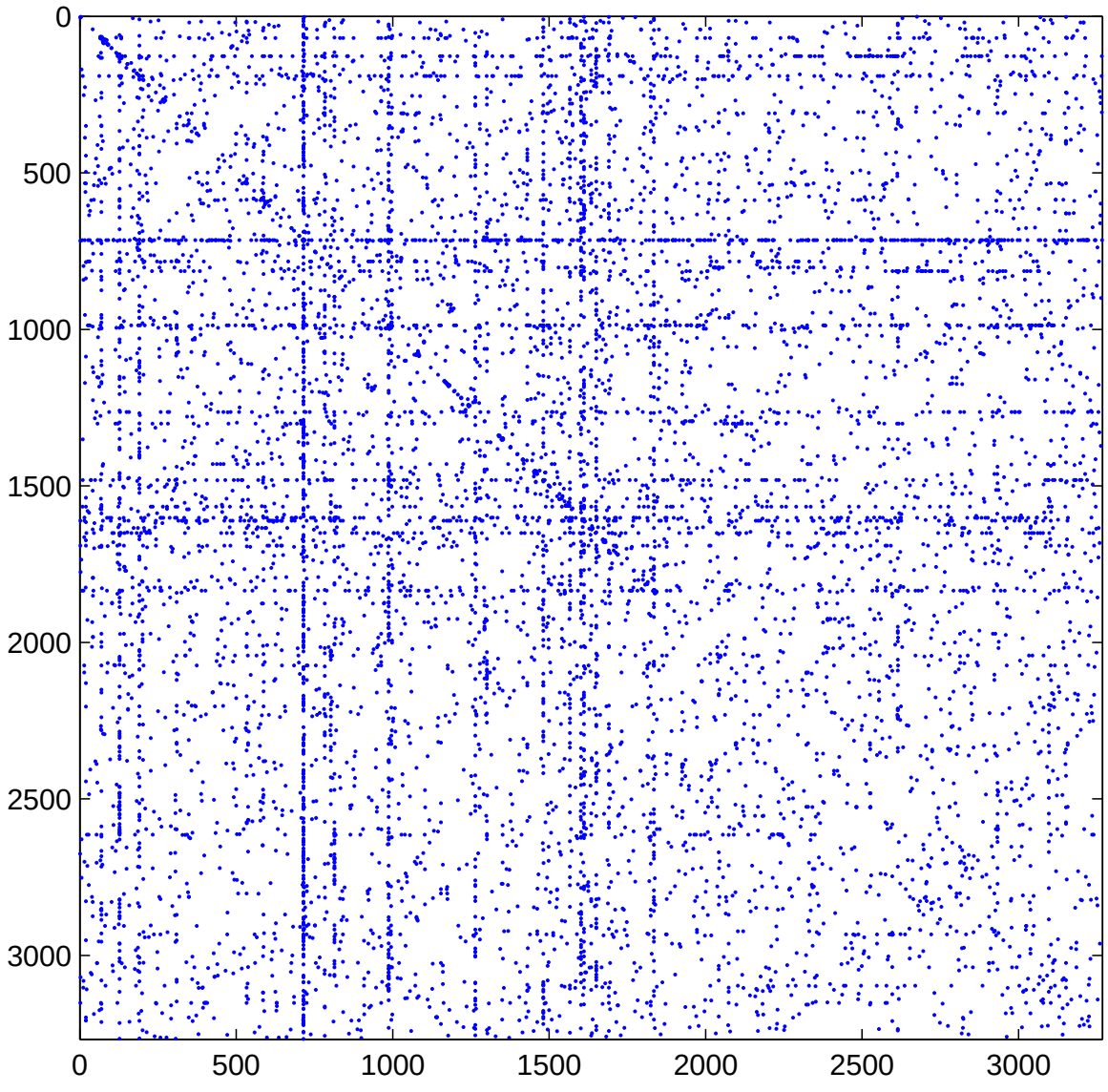


Specificity and stability in topology of protein networks, **S. Maslov & K. Sneppen**, **Science, 2002**

Adjacency Matrix: Uetz et al. 2000, Yeast PPI



Adjacency Matrix: Ito et al. 2001, Yeast PPI



Y2H Protein-Protein Interaction Networks

Noisy: 50–90% false positive, 50–90% false negative

Two types of false positive

- Technical: experimental limitations
- Biological: don't occur in vivo
  - not expressed at same time
  - not in same sub-cellular compartment, or same tissue

Interactions may also depend on the environment

How can we use this data . . . . . ?

Fine details...

Typical questions:

- Are there any other proteins like protein Y?
- What is the biological function of protein X?
- Which proteins act together?
- What happens if protein Z is removed?

Also: which are the false pos/negs ?

Big picture. . .

PPI networks are not regular
Describe them by a random graph model?

- capture many PPI networks with a **small number of parameters**:
  - **distinguish between different organisms**
  - **get evolutionary insights**
- generate **synthetic data sets** to test algorithms

Several random graph “models” have been proposed . . . . .

Comparing Networks

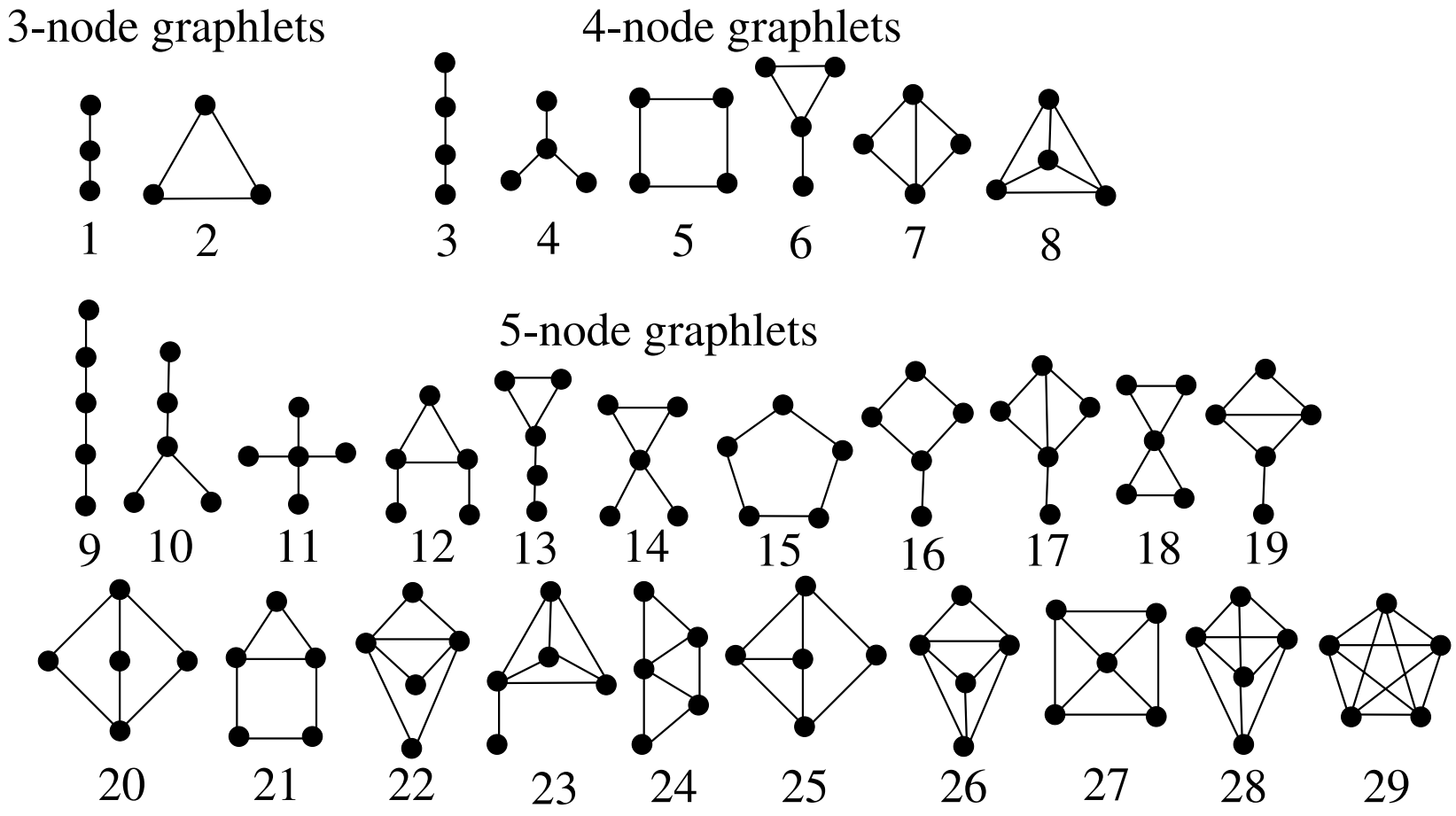
Global Measures

- Degree distribution
- Pathlength distribution
- Clustering coefficients

Local Measure

- graphlet frequencies . . .

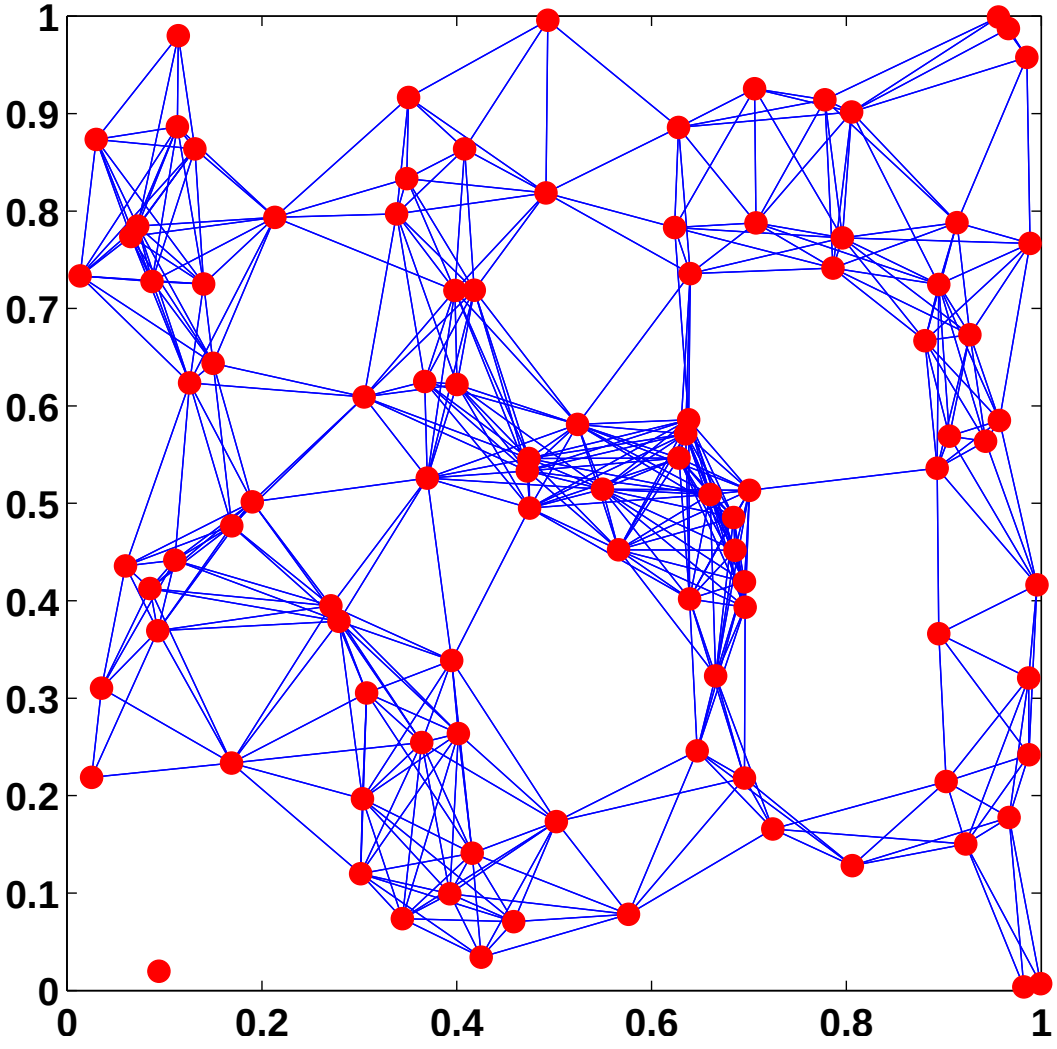
Graphlet Frequencies (Pržulj et al.)



Frequency of graphlet i $\quad (0 \leq i \leq 29)$

$$\frac{\text{number of graphlets of type } i}{\text{total number of graphlets}}$$

Geometric Model (Pržulj et al., 2004)



Geometric Random Graph

- randomly place N nodes in unit square
- connect nodes within distance ϵ

Able to match PPI properties (pathlengths, clustering coefficients, degree distributions, graphlet frequencies)

- *Modeling Interactome: Scale-Free or Geometric?*, N. Pržulj, D. Corneil and I. Jurisica, **Bioinformatics**, 2004
- *Analyzing Large Biological Networks . . .* , N. Pržulj, Ph.D. Thesis, **University of Toronto**, 2005

Question: Given a PPI network, can we map it on to a geometric random graph?

$\Rightarrow$ develop a **tool** for **reverse engineering** a GRG

Given nodes and edges, optimally place the nodes in $\mathbb{R}^2$ such that nodes within a distance ϵ are connected

Multi-Dimensional Scaling (MDS)

Problem:

Given all pairwise distances $\{d_{ij}\}_{i,j=1}^{N}$,

find vectors $\{\mathbf{x}^{[i]}\}_{i=1}^{N} \in \mathbb{R}^m$ such that

$$\| \mathbf{x}^{[i]} - \mathbf{x}^{[j]} \| = d_{ij}, \quad \forall i, j$$

i.e. go from **pairwise distance** to **location**

Notation

$$X = \begin{bmatrix} \mathbf{x}^{[1]} & \mathbf{x}^{[2]} & & \mathbf{x}^{[N]} \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \in \mathbb{R}^{m \times N}$$

MDS Theory

Define (sym. pos. def.) $A \in \mathbb{R}^{N \times N}$ by

```
A(i,j) = -0.5*( Dsq(i,j) - mean(Dsq(i,:)) ...
          - mean(Dsq(:,j)) + mean(mean(Dsq)) );
```

Then $X^T X = A \Rightarrow \| \mathbf{x}^{[i]} - \mathbf{x}^{[j]} \| = d_{ij}$

Symm. Real Schur Decomp. $A = U^T \Sigma U \quad \Rightarrow \quad$ use

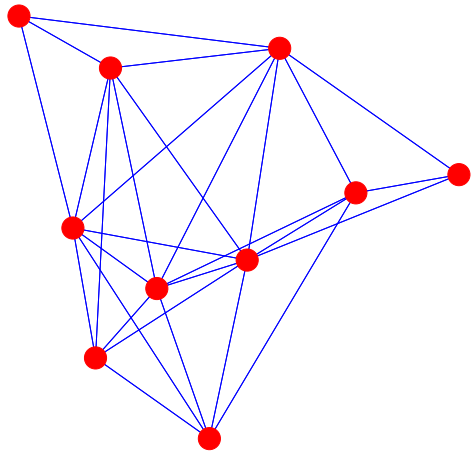
$$X = \Sigma^{\frac{1}{2}} U = \begin{bmatrix} \sqrt{\sigma_1}\mathbf{u}^{[1]} & \dots & \dots & \dots \\ \sqrt{\sigma_2}\mathbf{u}^{[2]} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \sqrt{\sigma_N}\mathbf{u}^{[N]} & \dots & \dots & \dots \end{bmatrix} \in \mathbb{R}^{N \times N}$$

To embed into, say, $\mathbb{R}^2$ "best" approximation is

$$X = \Sigma^{\frac{1}{2}} U = \begin{bmatrix} \sqrt{\sigma_1}\mathbf{u}^{[1]} & \dots & \dots & \dots \\ \sqrt{\sigma_2}\mathbf{u}^{[2]} & \dots & \dots & \dots \end{bmatrix}$$

MDS to reverse engineer a GRG?

PPI data is “0 or 1”, we don’t have Euclidean distances



Idea use **pathlength**

$$d_{ij}^2 = \text{pathlength from node } i \text{ to node } j$$

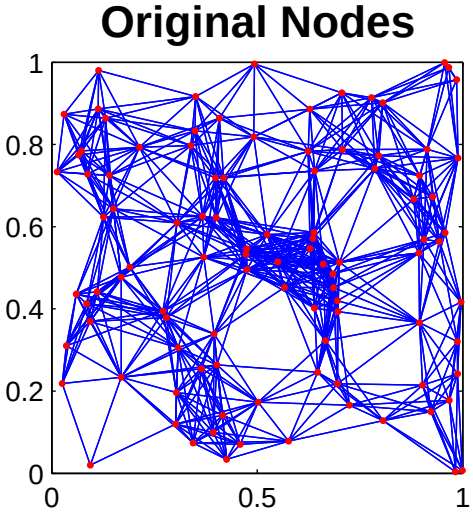
Compute pathlengths $1, 2, \ldots, K$, set the rest to $K_{\max}$, so

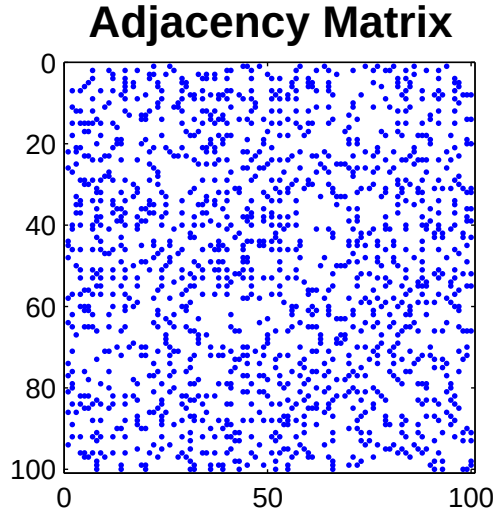
- distance matrix is sparse plus rank 1
- ∞’s avoided

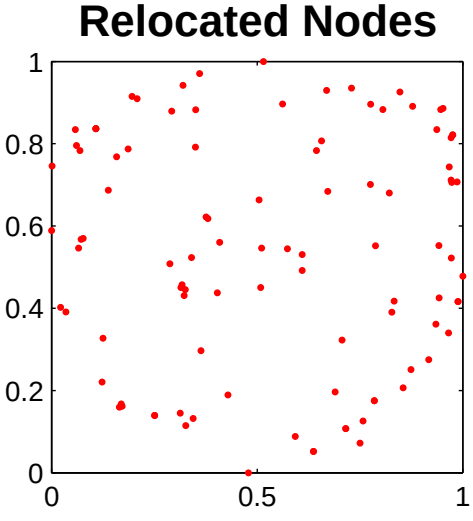
Now apply MDS to recover locations in $\mathbb{R}^2$

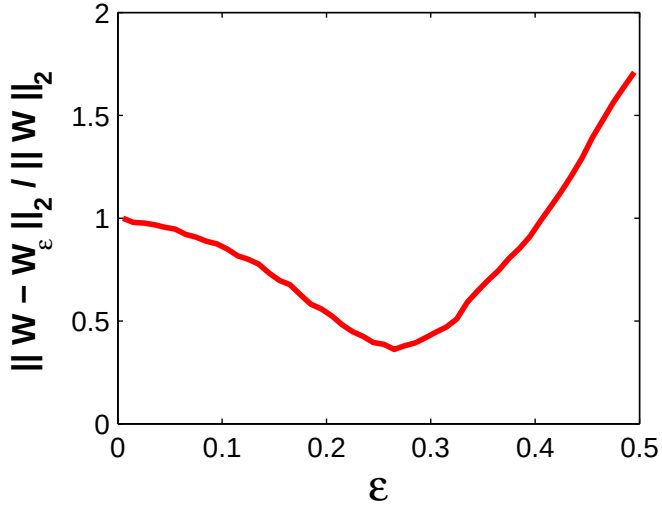
$N = 100$, $\epsilon = 0.25$ $(K = 4, K_{\max} = 5)$

Eigs of A: $38.2, 30.1, 10.7, 8.9, 6.1, 3.8, \ldots$

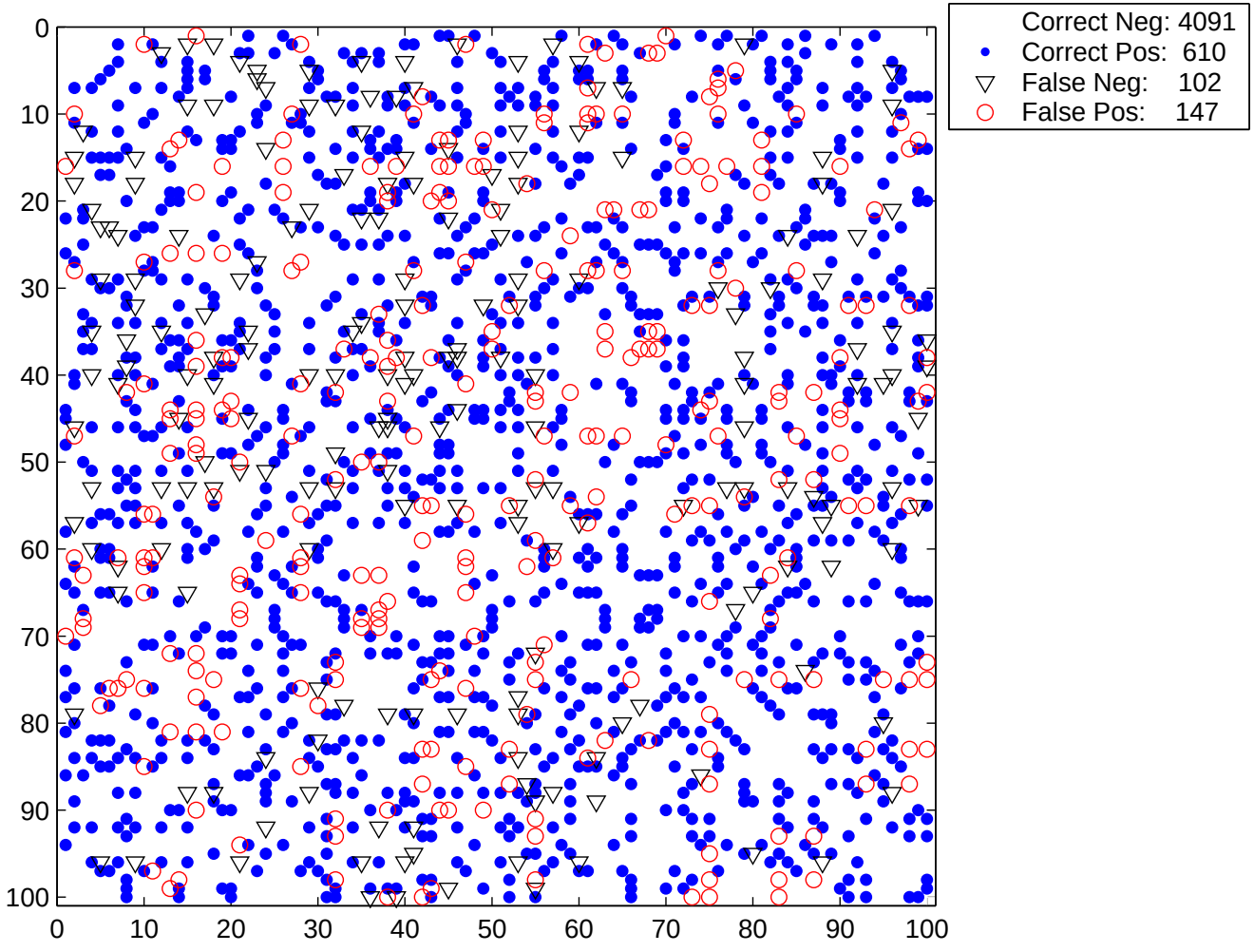






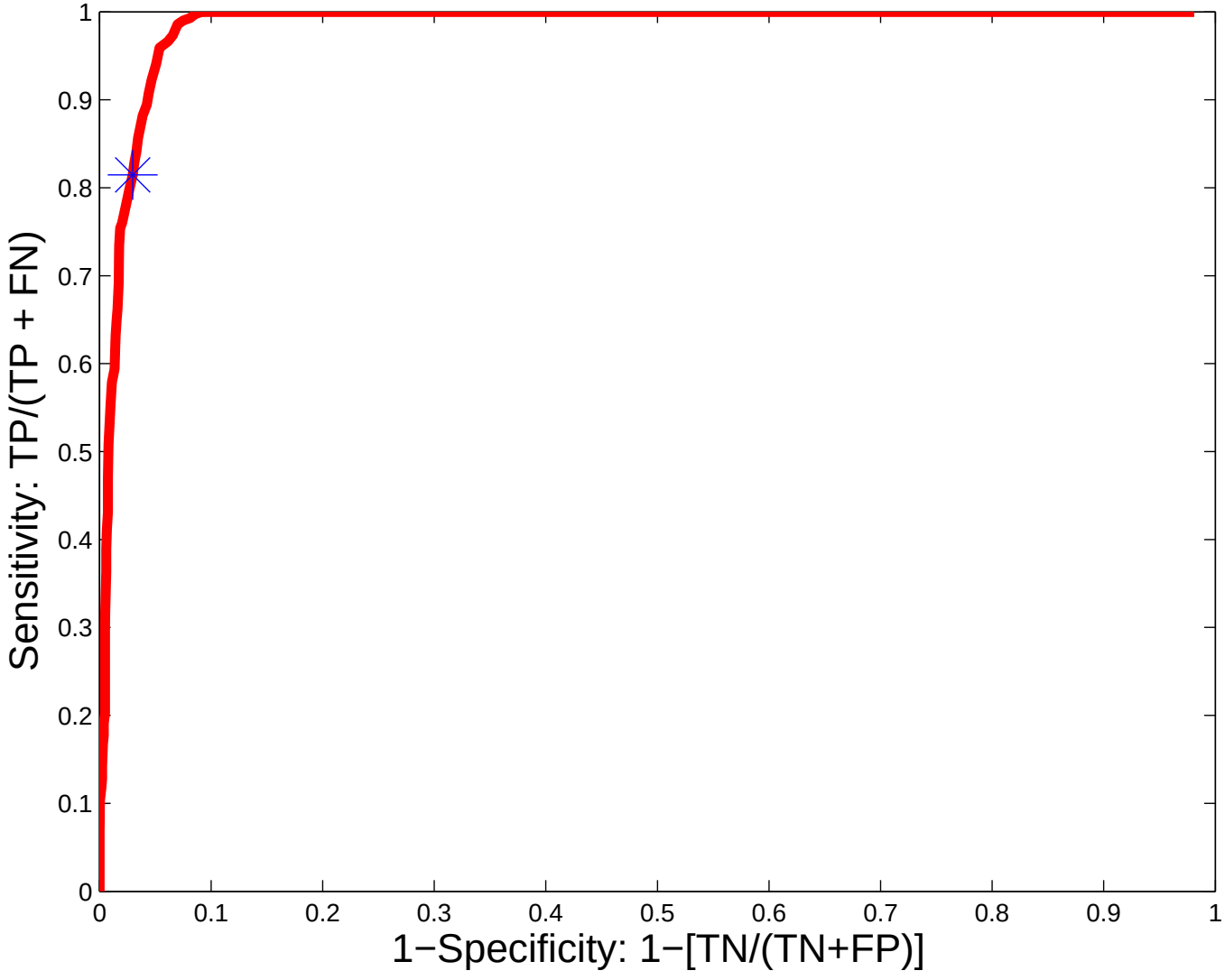


Same Example, “optimal” ϵ



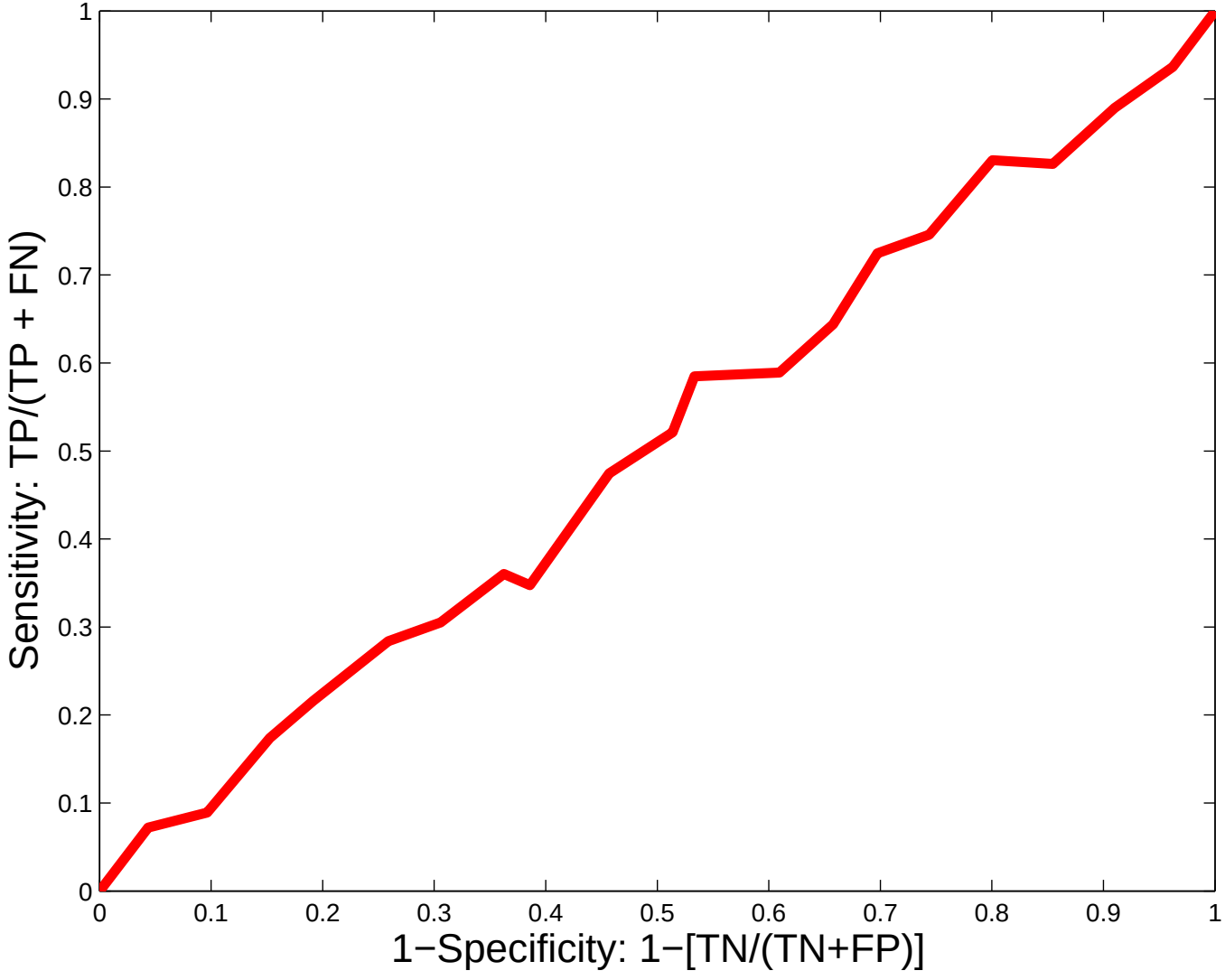
Same Example, ROC curve

Area under curve is 0.965



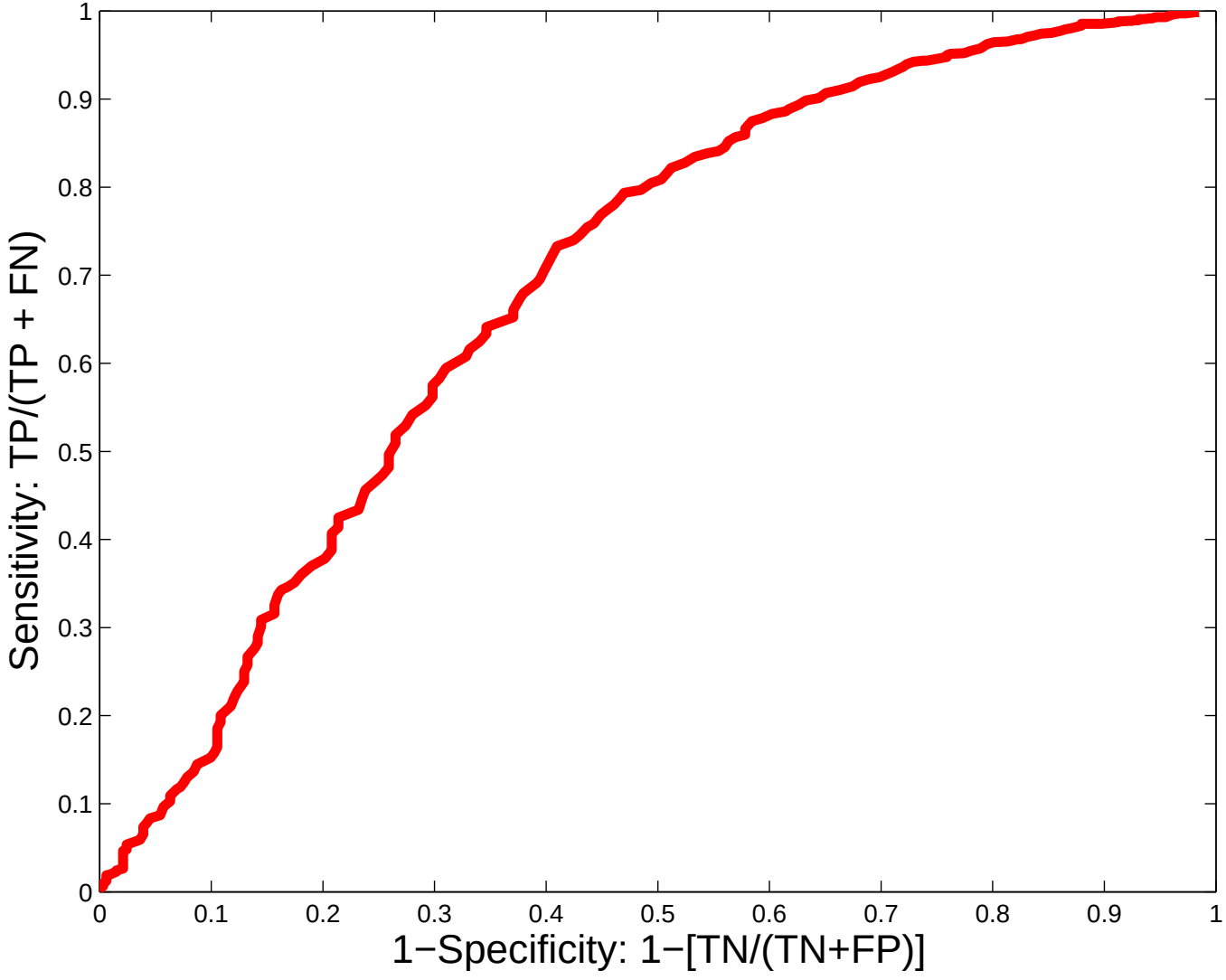
GRG data, coin flip to predict links

Area under curve is 0.48

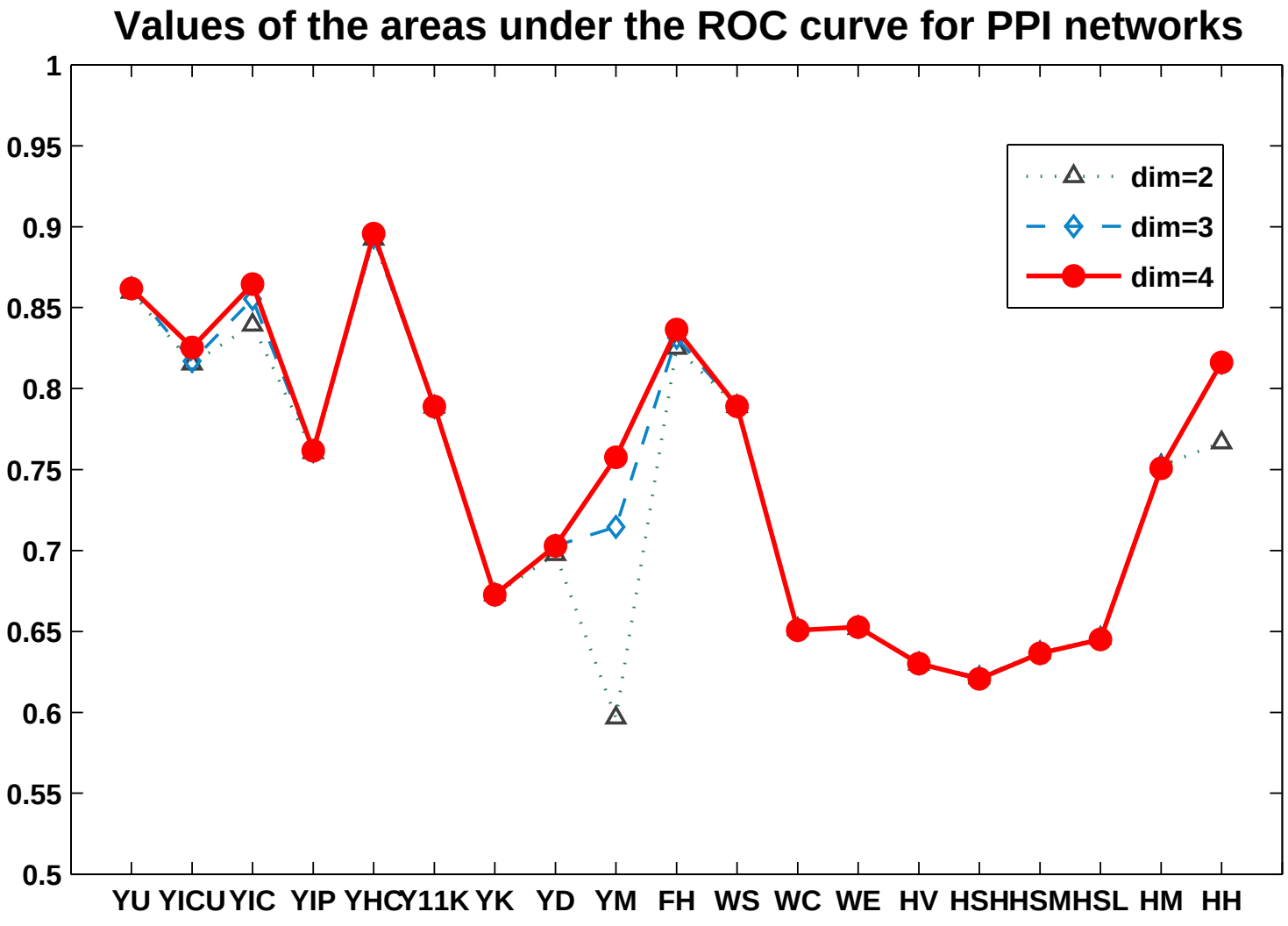


Erdõs–Rényi Random Graph with MDS algorithm

Area under curve is 0.67



Nineteen PPI networks



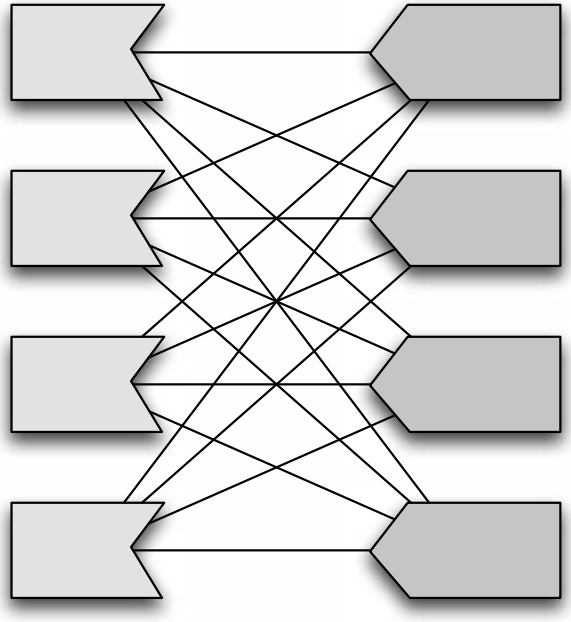
High Confidence Von Mering et al. gave 0.89

Lock-and-Key Model

On the structure of protein-protein interaction networks, A. Thomas, R. Cannings, N. Monk, C. Cannings, *Biochemical Society Transactions* 31, 2003

Idea: two proteins interact because they 'fit together'

$\Rightarrow$ complementary domains, i.e. locks and keys



Lock-and-Key Model

Thomas et al. model: m locks and m matching keys

- let each protein have each lock and key with independent probability p
- put an edge between two proteins $\Leftrightarrow$ they share at least one lock/key pair

Thomas et al. looked at **big picture** issue:

Does this model reproduce the almost scale free nature of PPI networks?

Our approach:

- introduce different modelling assumptions
- develop an algorithm for inferring locks and keys
- answer both big picture and fine detail questions

Our Assumptions

There exists a lock/key pair in the network such that any protein with this lock/key

- does not have the matching key/lock
- will only interact with a protein having the matching key/lock
- only has a fixed proportion $0 \leq \theta \leq 1$ of its lock/key matches recorded as interactions

$\Rightarrow$ the adjacency matrix has a pair of eigenvalues

$$\lambda = \pm\theta\sqrt{\text{locksum} \times \text{keysum}}$$

with eigenvectors

$$\sqrt{\text{keysum}}\ \mathbf{ind}^{[\text{lock}]} \pm \sqrt{\text{locksum}}\ \mathbf{ind}^{[\text{key}]}$$

Algorithm

- Calculate eigenvals/vecs
- Group into $\approx \pm\lambda$ pairs
- For each pair with eigvecs u_a and u_b
  - choose a threshold, K
  - $|u_a + u_b|_i \geq K$ means protein i has lock
  - $|u_a - u_b|_i \geq K$ means protein i has key

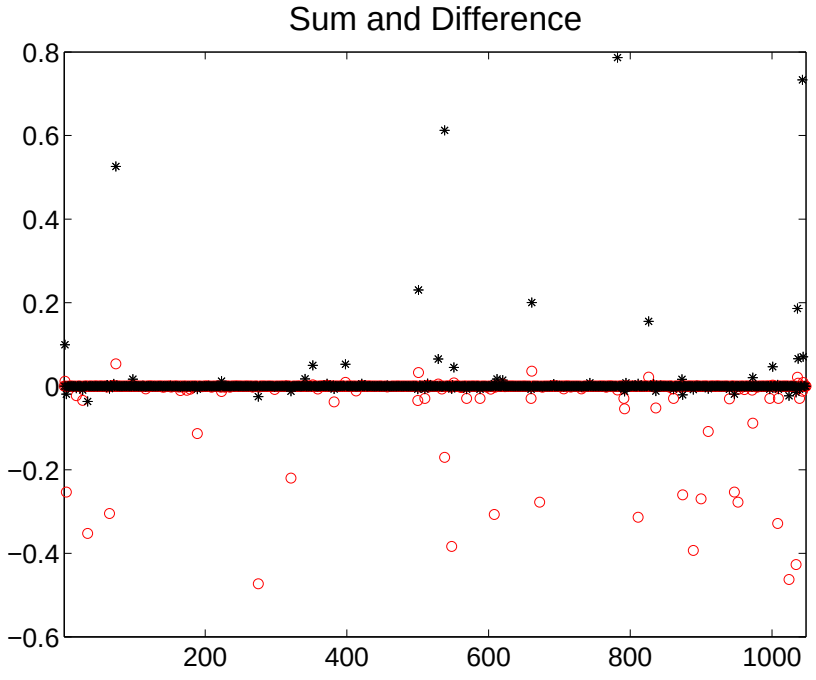
Successful at recovering locks and keys in synthetically generated networks (good **sensitivity** and **specificity**)

Spectral Properties: Uetz (2000) data

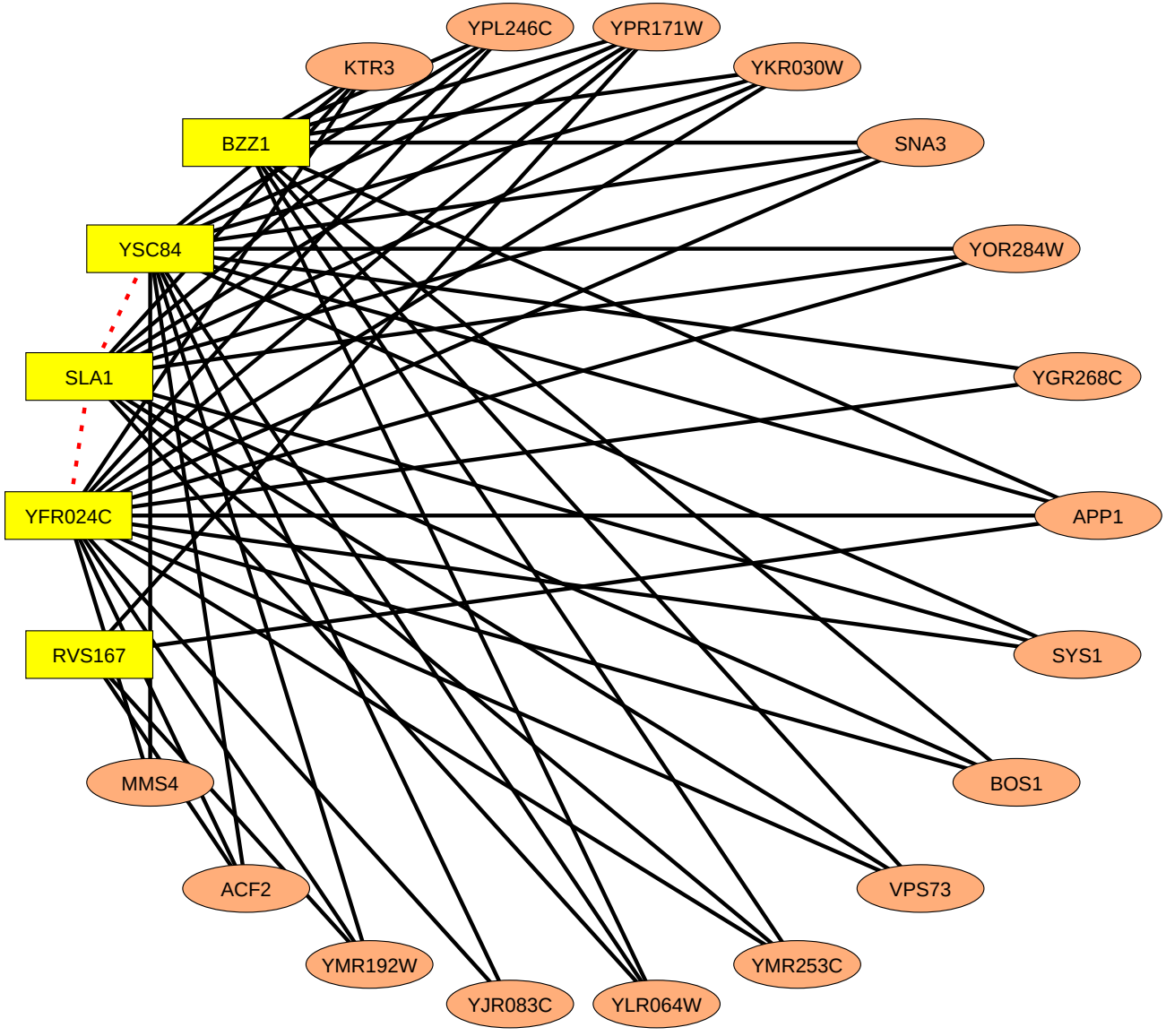
```
>> [U,D] = eigs(W,8,'BE');
>> diag(D)

ans =

   -6.4614
   -5.1460
   -4.1557
   -4.1270
    4.3778
    5.4309
    5.8397
    7.4096
```



Result for Uetz et al. (2000) yeast data



Further Investigation . . .

Other biological data shows that

all five proteins in one group possess the SH3 domain

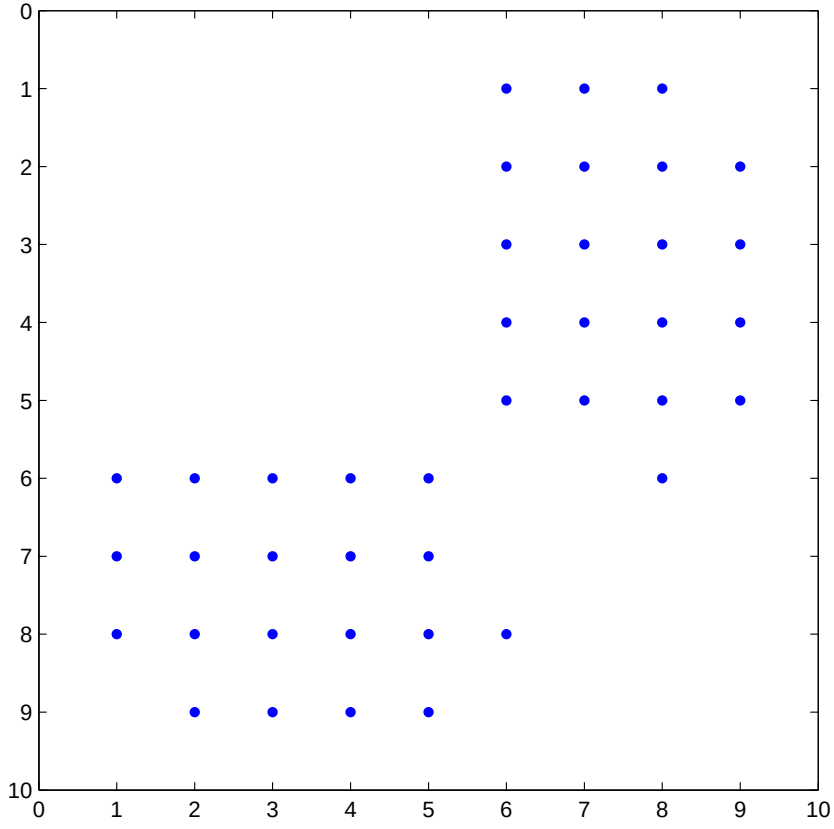
$\Rightarrow$ we have identified the key!

Recent experiments (Kessels & Qualmann 2004, Friesen et al. 2005) show that the SH3 domain is involved in trafficking of **vesicles**

All proteins in the other group are part of the actin cortical patch assembly mechansim of **vesicle** endocytosis (Drees et al. 2001)

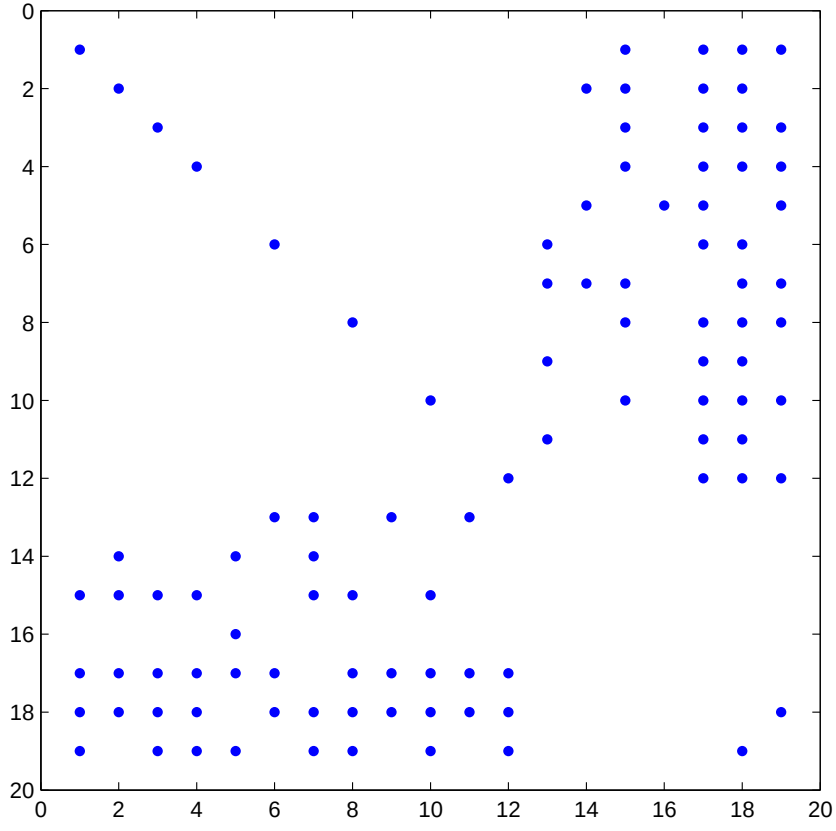
[*vesicle*: *small, enclosed compartment within a cell*]

Arabidopsis Thaliana (small flowering plant)



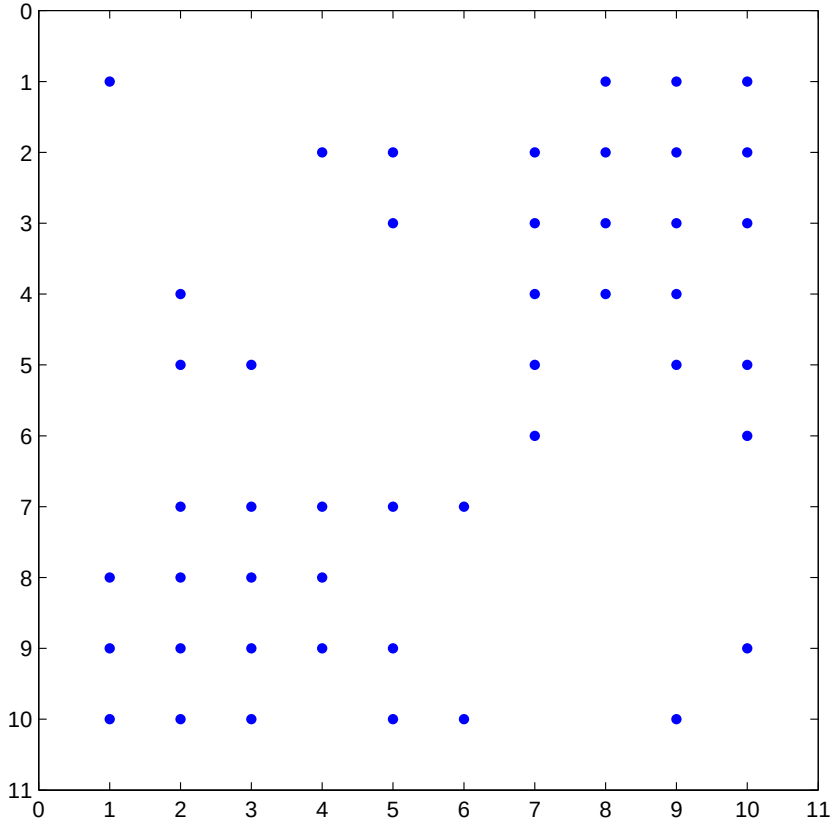
Homeobox Transcription Factor module?

Saccharomyces Cerevisiae (yeast)



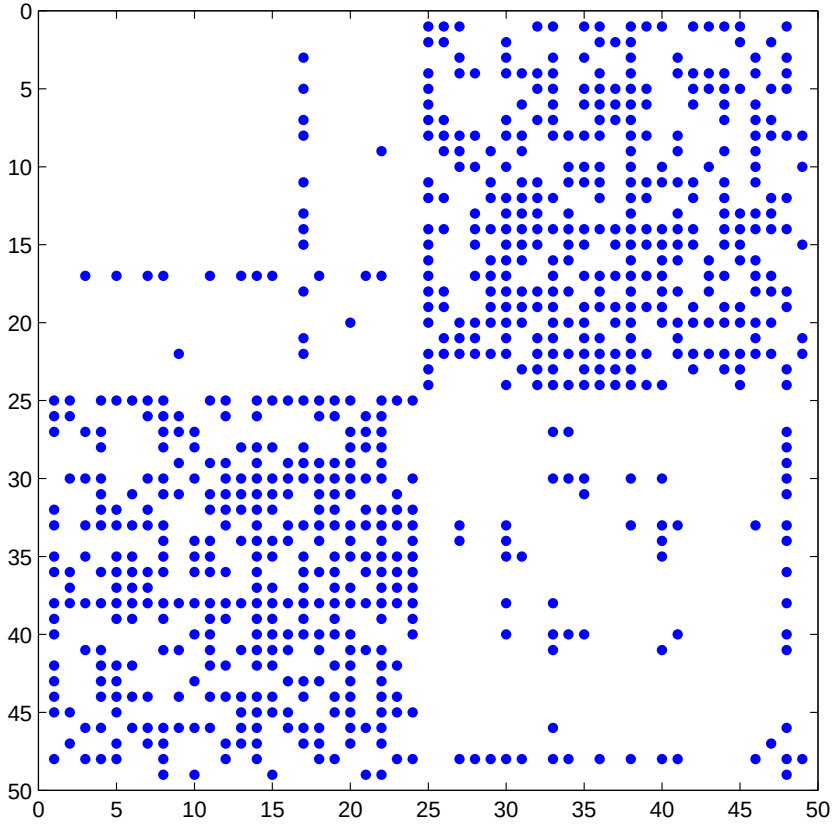
Protein Trafficking module?

Homo Sapiens (us!)



Smad Transcription Factor module?

Drosophila Melanogaster (fruit fly)



Cell Cycle Transcriptional Regulation module?

. . . plus many more . . .

Recap

Lock-and-key model: Extension of Thomas et al. (2003) model to

- make **testable predictions** about PPI network structure
- extract **important structural information** from (noisy) PPI data sets

Note: different to traditional **clustering**

Essentially clustering on paths of length two

Match local/global PPI properties? . . .

Stickiness Model

Back to the **big picture**

Can we produce a model that matches PPI network properties?

Inferring number and distribution of locks and keys in a real (noisy) network: very challenging

Idea summarize abundance/popularity of binding domains as a single number per protein: **stickiness index**

[Analogous idea of fitness in physics community]

Modelling Assumptions

Assumption 1

High degree implies many and/or popular binding domains: high stickiness

So **high degree** $\Rightarrow$ **high stickiness**

Assumption 2

A pair of proteins is more likely to interact (share complementary binding domains) if they both have high stickiness index

Take the **product of stickiness indices**

Hence, we suppose $\mathbb{P}\left(i \leftrightarrow j\right) = f\left(\deg_i\right) f\left(\deg_j\right)$

Match expected degree $\Rightarrow f\left(\deg_i\right) = \frac{\deg_i}{\sqrt{\sum_{k=1}^{N} \deg_k}}$

PseudoCode

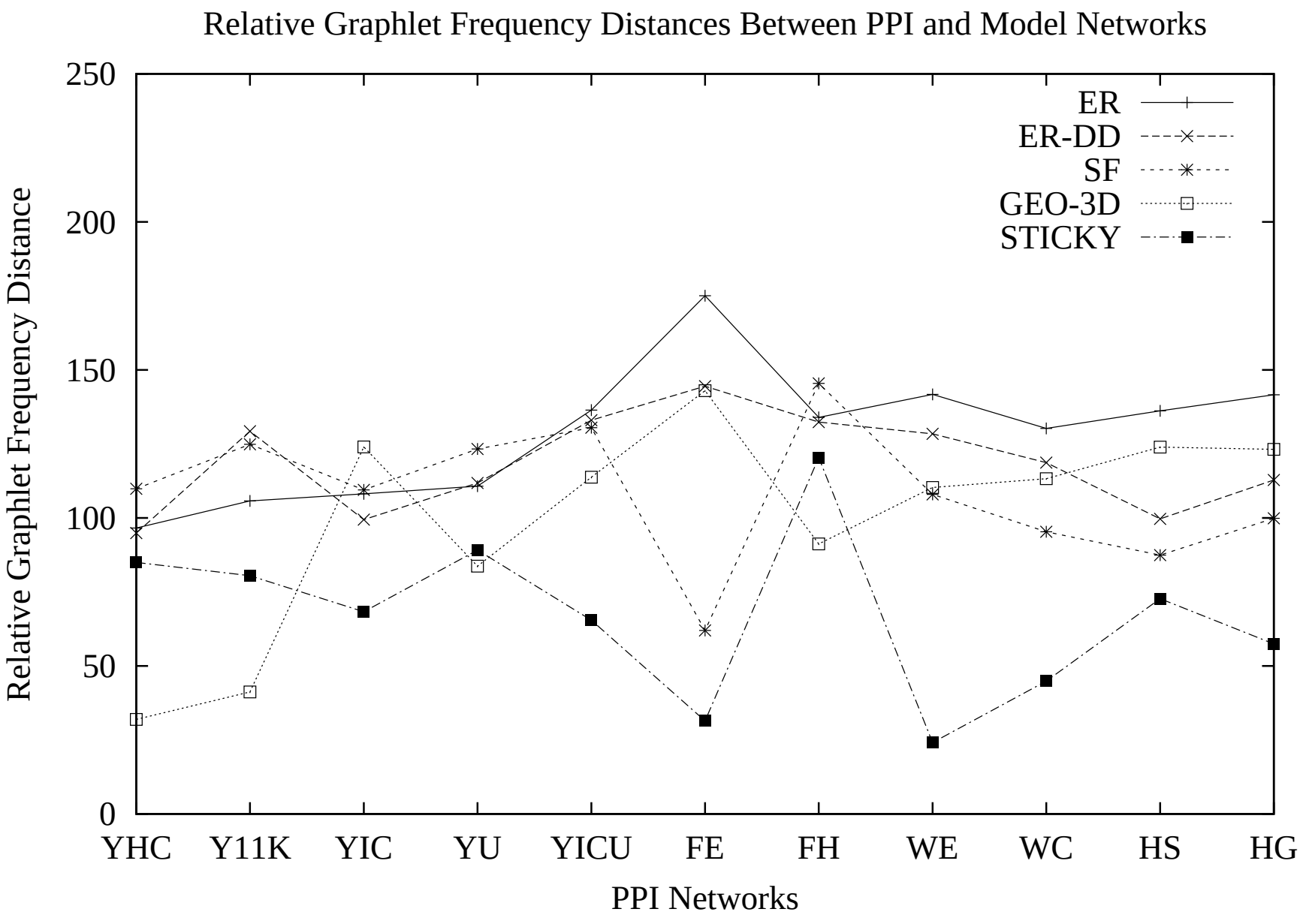
input $\{\deg_i\}_{i=1}^N$
output $\{w_{ij}\}_{i,j=1}^N$

for i = 1 to N

$$\theta_i = \deg_i / \sqrt{\sum_{j=1}^N \deg_j}$$

end
Initialize all $w_{ij} = 0$
for i = 1 to $N - 1$
for j = $i + 1$ to N
compute a uniform $(0, 1)$ sample, r
if $r \leq \theta_i \theta_j$
$w_{ij} = 1$ and $w_{ji} = 1$
end if
end for
end for

Graphlet Frequency Comparison



What's New?

- **Spectral algorithm** for discovering bi-partite subgraphs (locks and keys)
- Realistic results on **PPI** networks
- **Spectral algorithm** for **reverse engineering** a **geometric graph**
- Supports the claim that **PPI networks** have some geometric structure
- Simplified **stickiness** model gives excellent local and global fit to PPI data

with **Alan Taylor**:

CONTEST (CONTrolable TEST matrices) for MATLAB at

`http://www.maths.strath.ac.uk/research/groups/numerical_analysis/contest`