Announcements:

Project milestones graded

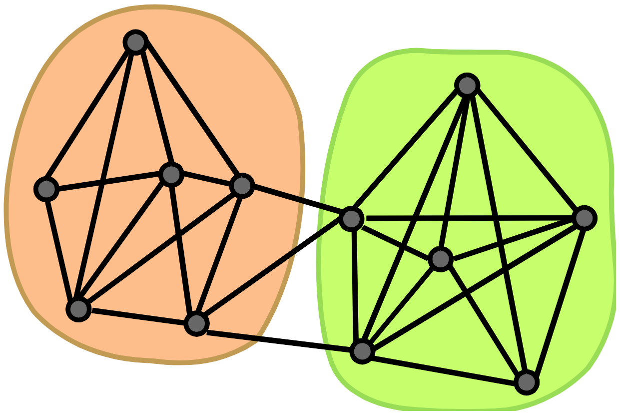
Keep up the good work!

Community Detection: Overlapping Communities

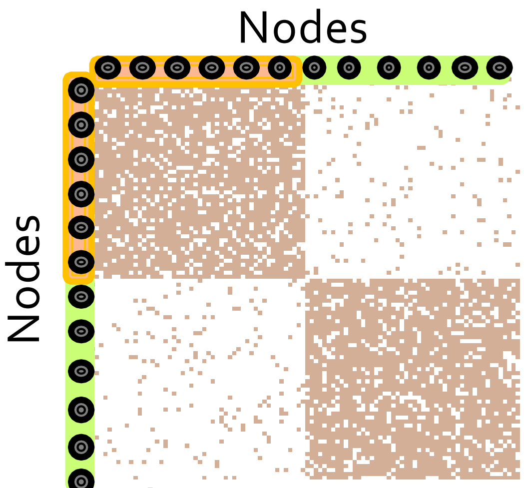
CS224W: Social and Information Network Analysis
Jure Leskovec, Stanford University

http://cs224w.stanford.edu

Non-overlapping Communities



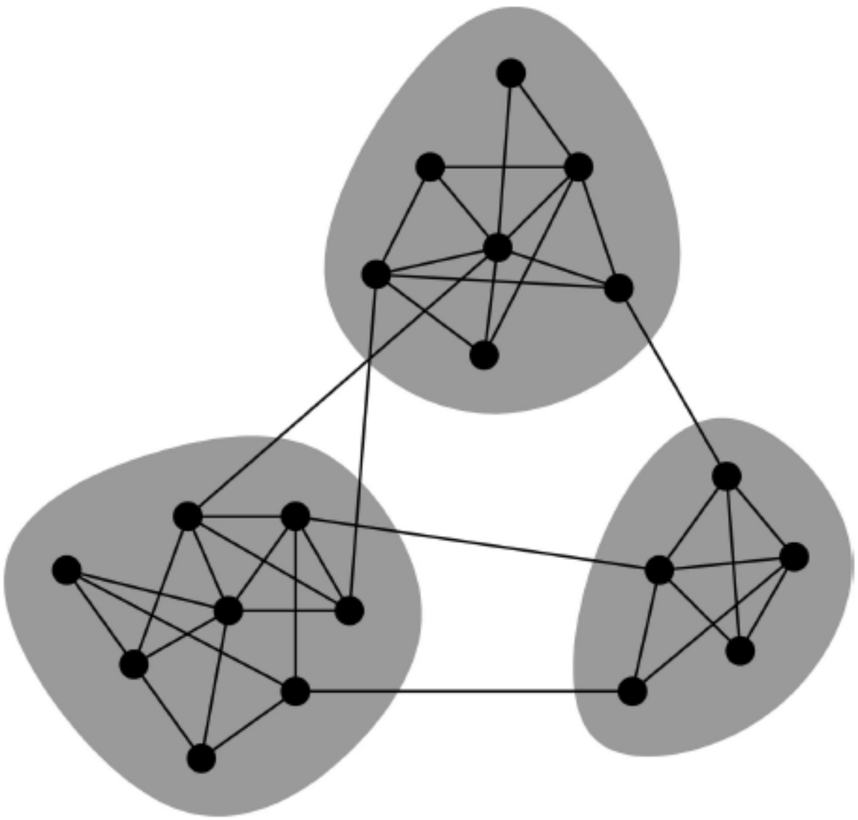
Network

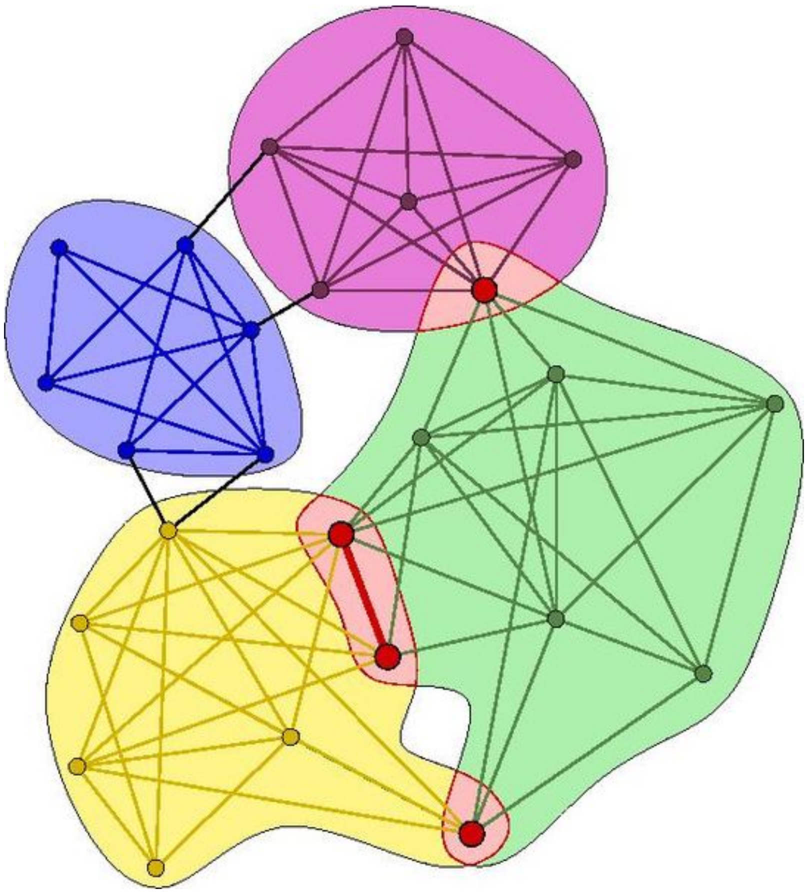


Adjacency matrix

Overlapping Communities

- **Non-overlapping vs. overlapping communities**

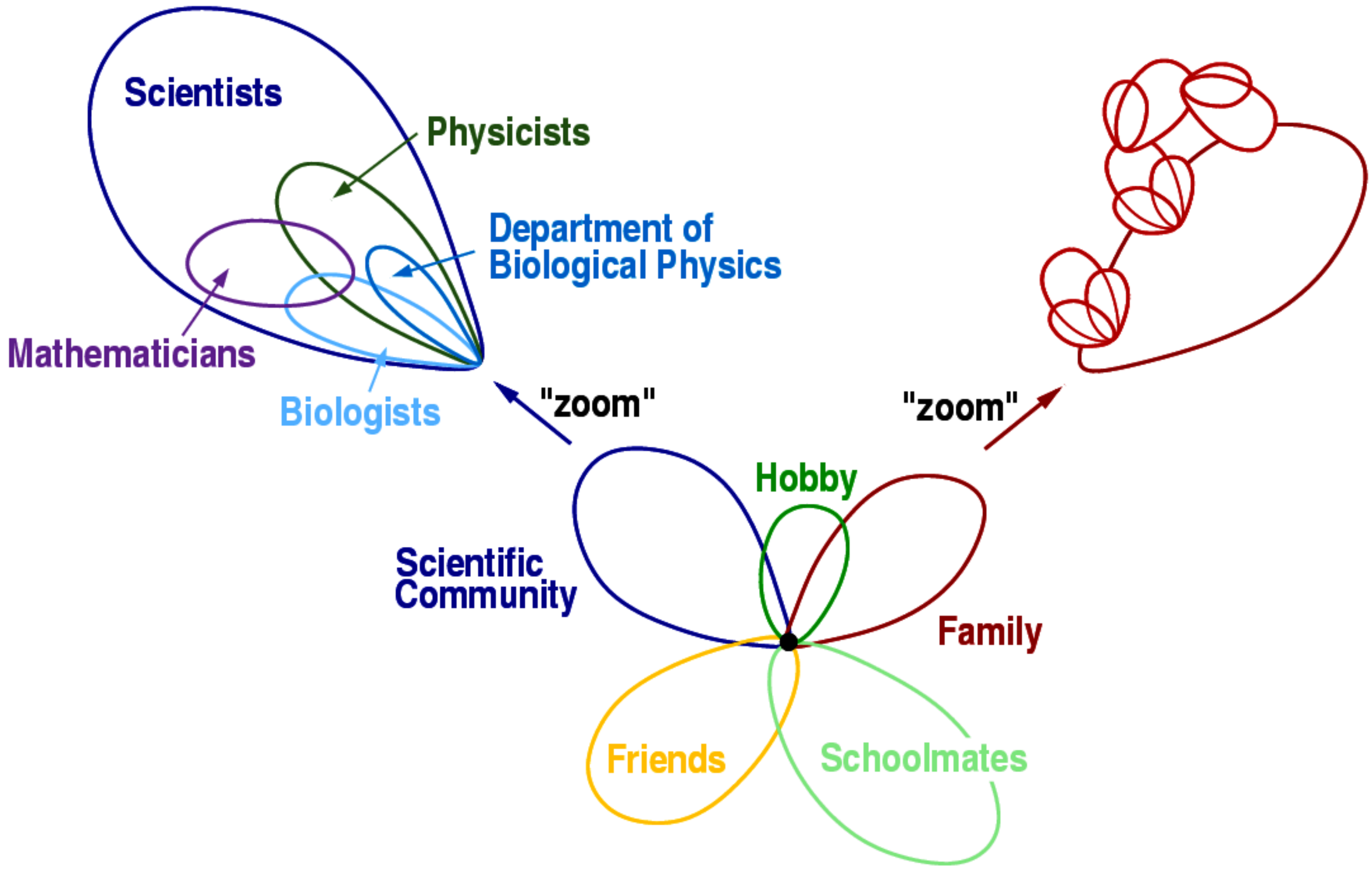




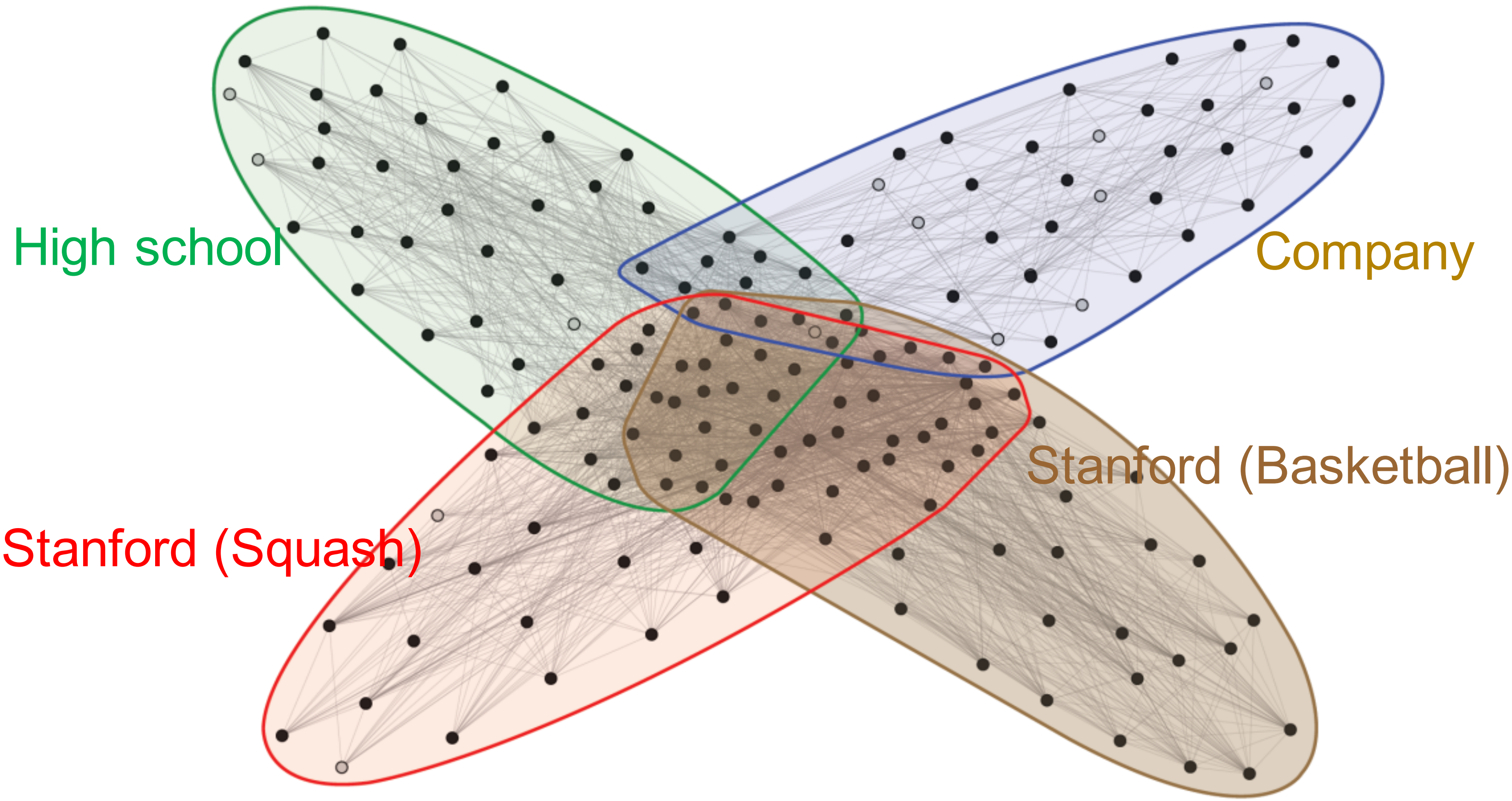
[Palla et al., ‘05]

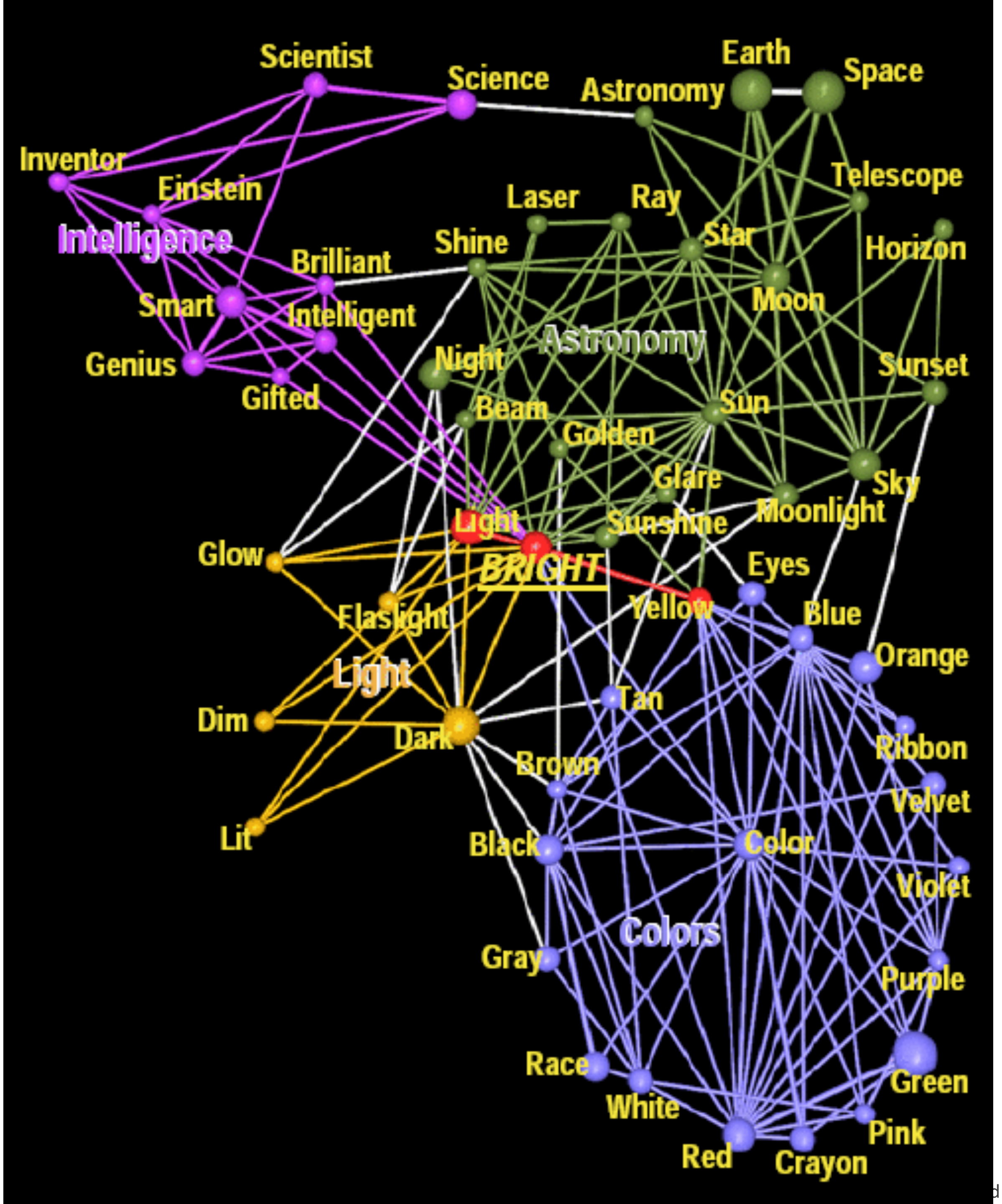
Overlaps of Social Circles

- **A node can belong to many social “circles”**



What if communities overlap?



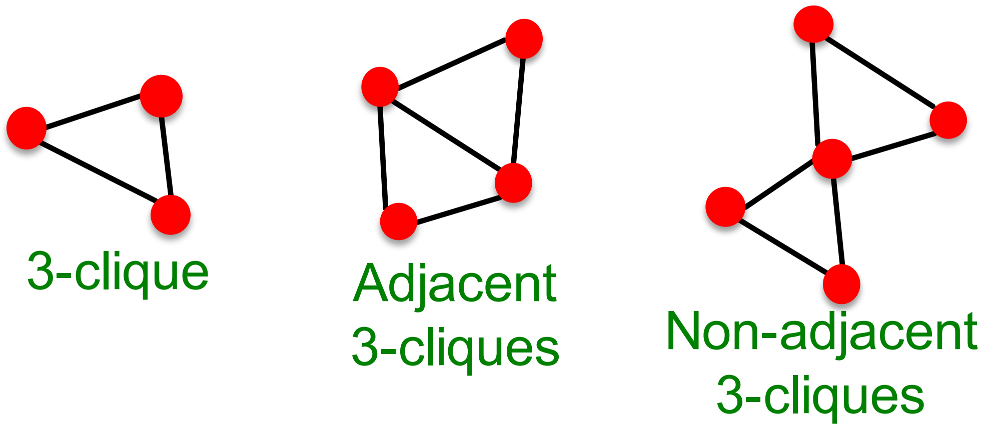


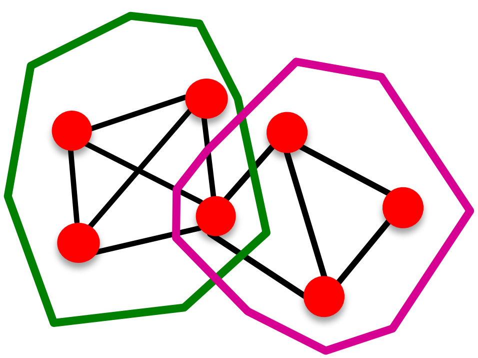
Scientist
Science
Earth
Space
Astronomy
Inventor
Einstein
Telescope
Laser
Ray
Intelligence
Shine
Star
Horizon
Brilliant
Moon
Smart
Intelligent
Astronomy
Genius
Night
Sunset
Gifted
Beam
Sun
Golden
Glare
Sky
Light
Moonlight
Sunshine
Glow
BRIGHT
Eyes
Yellow
Flaslight
Blue
Orange
Light
Tan
Dim
Dark
Ribbon
Brown
Velvet
Lit
Black
Color
Violet
Colors
Gray
Purple
Race
Green
White
Pink
Red
Crayon

[Palla et al., '05]

Clique Percolation Method (CPM)

- **Two nodes belong to the same community if they can be connected through adjacent *k*-cliques:**
  - ***k*-clique:**
    - Fully connected graph on *k* nodes
  - **Adjacent *k*-cliques:**
    - overlap in ***k-1*** nodes
- ***k*-clique community**
  - Set of nodes that can be reached through a sequence of adjacent *k*-cliques



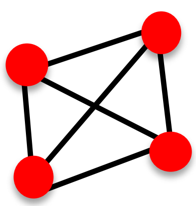


Two overlapping 3-clique communities

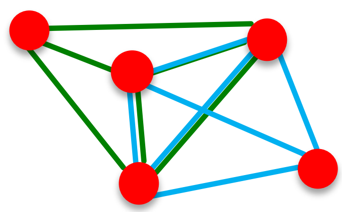
[Palla et al., ‘05]

Clique Percolation Method (CPM)

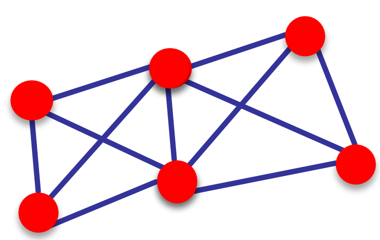
- **Two nodes belong to the same community if they can be connected through adjacent *k*-cliques:**



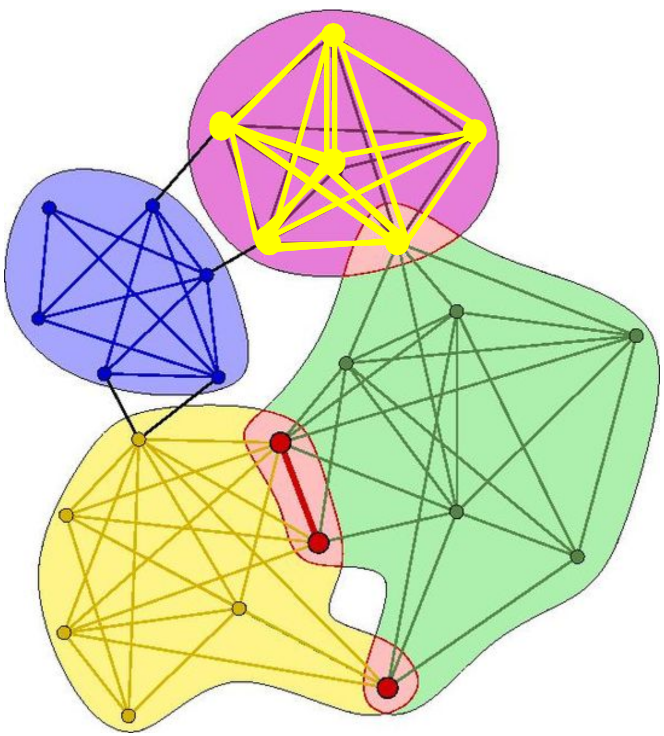
4-clique



Adjacent 4-cliques



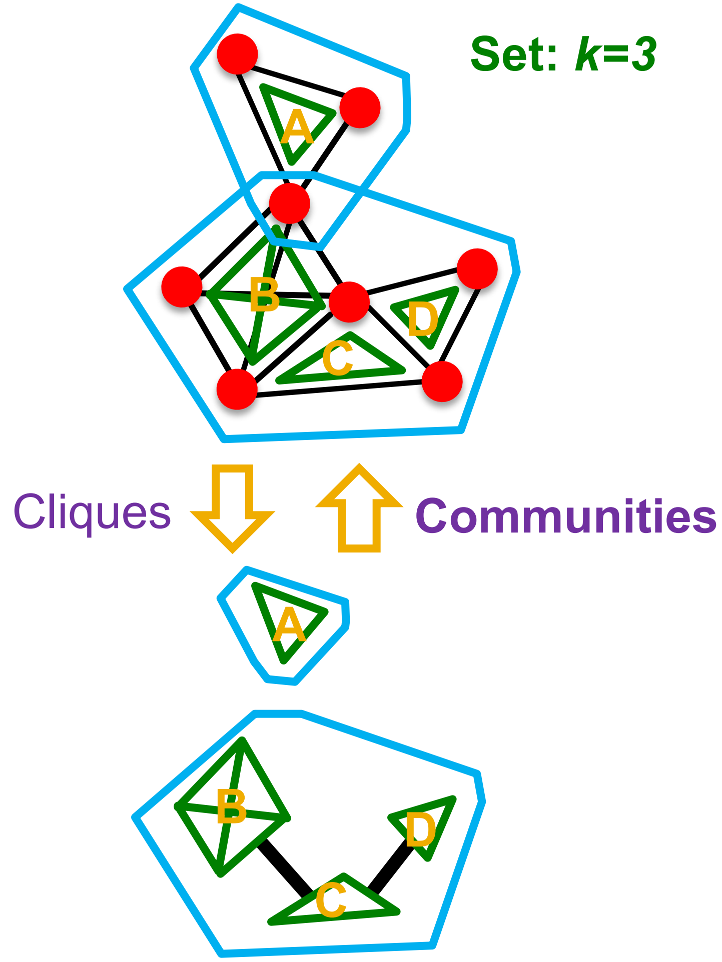
Non-adjacent 4-cliques



Communities for k=4

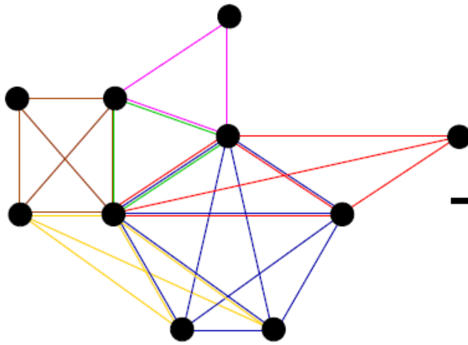
CPM: Steps

- **Clique Percolation Method:**
  - **Find maximal-cliques**
    - Def: Clique is maximal if no superset is a clique
  - **Clique overlap super-graph:**
    - Each clique is a super-node
    - Connect two cliques if they overlap in at least k-1 nodes
  - **Communities:**
    - Connected components of the clique overlap matrix
- **How to set k?**
  - Set k so that we get the "richest" (most widely distributed cluster sizes) community structure



CPM method: Example

- Start with graph
- Find maximal cliques
- Create clique overlap matrix
- Threshold the matrix at value ***k-1***
  - If $a_{ij} < k - 1 \; set \; 0$
- Communities are the connected components of the thresholded matrix



(1) Graph

Overlap size

Cliques

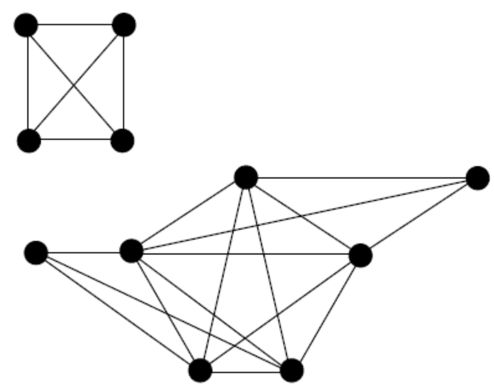
	5	3	2	1	3	1
	3	4	2	1	1	1
	2	2	3	2	1	2
	1	1	2	3	0	1
	3	1	1	0	4	2
	1	1	2	1	2	4

(2) Clique overlap matrix

k=4

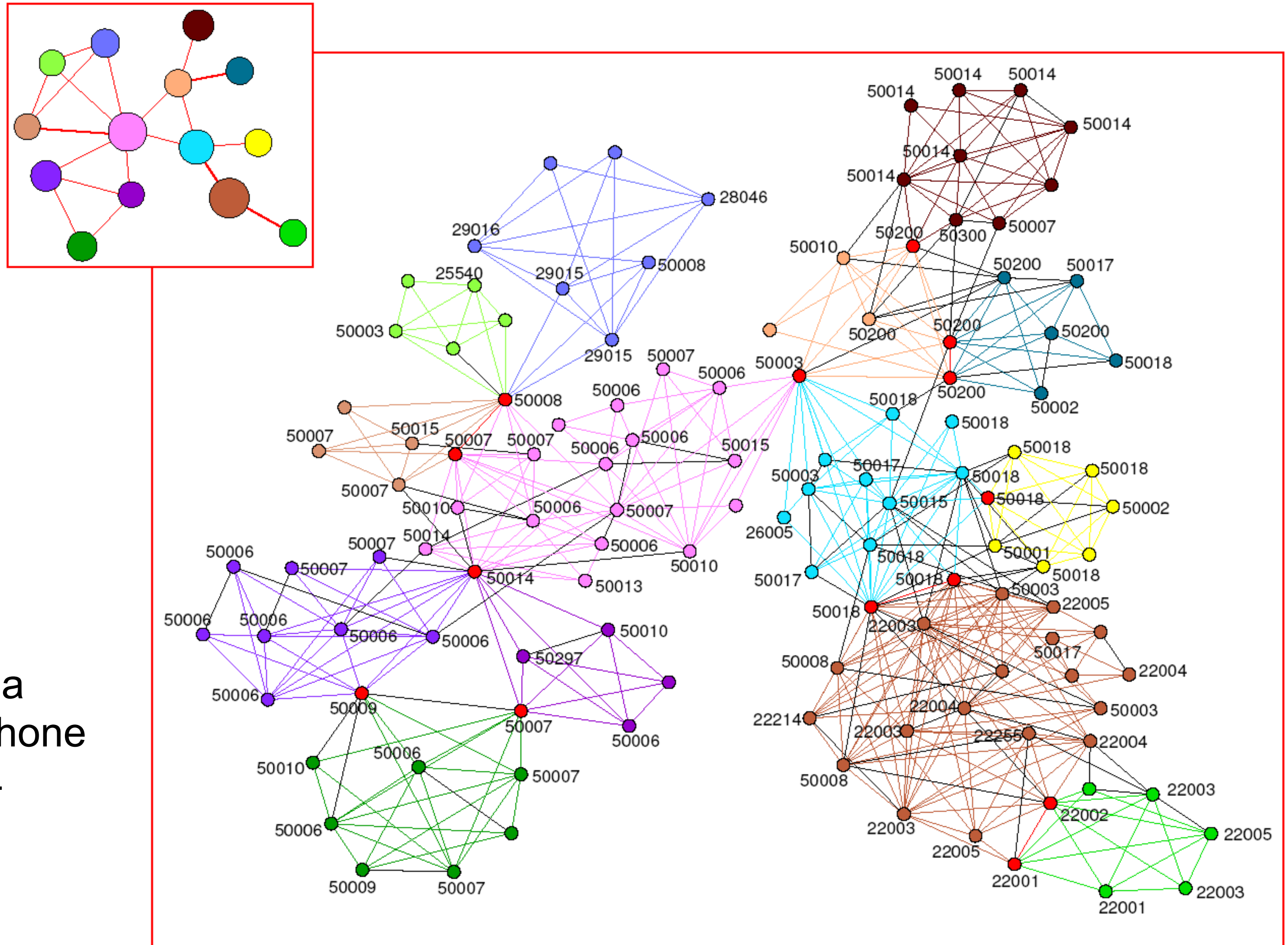
	1	1	0	0	1	0
	1	1	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
	1	0	0	0	1	0
	0	0	0	0	0	1

(3) Thresholded matrix at 3



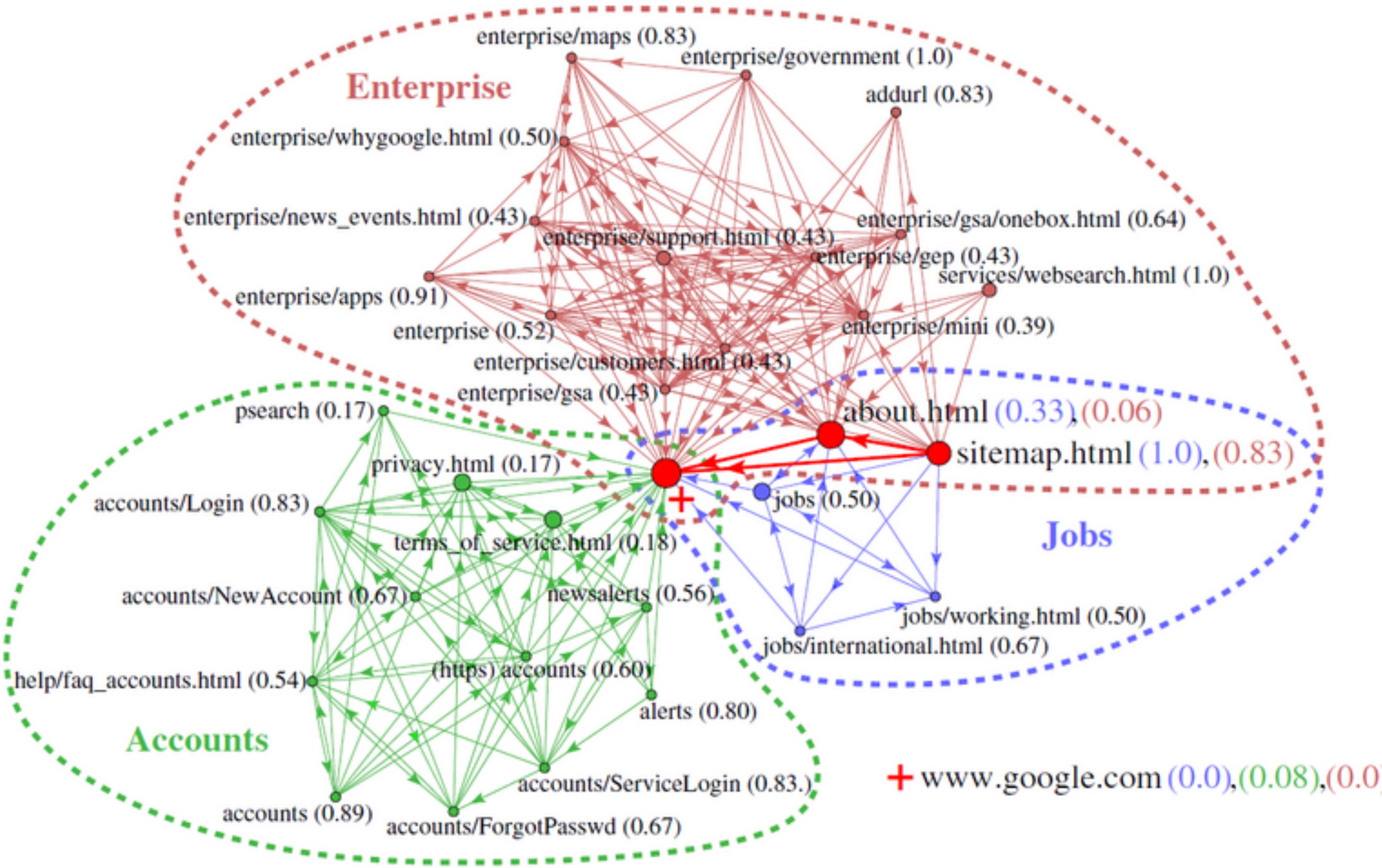
(4) Communities (connected components)

Example: Phone-Call Network



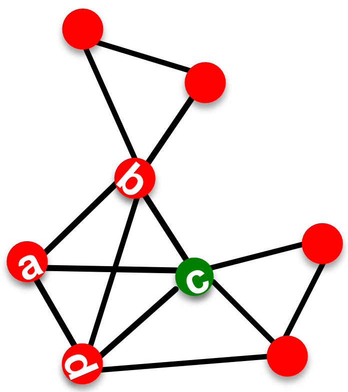
Communities in a "tiny" part of a phone call network of 4 million users [Palla et al., '07]

Example: Website



How to Find Maximal Cliques?

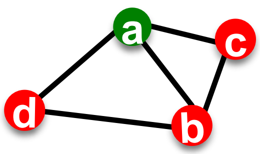
- **No nice way, hard combinatorial problem**
- **Maximal clique:** Clique that can't be extended
  - $\{a, b, c\}$ is a clique but not maximal clique
  - $\{a, b, c, d\}$ is maximal clique



- **Algorithm:** Sketch
  - Start with a seed node
  - Expand the clique around the seed
  - Once the clique cannot be further expanded we found the maximal clique
  - **Note:**
    - This will generate the same clique multiple times

How to Find Maximal Cliques?

- Start with a seed vertex $\boldsymbol{a}$
- **Goal:** Find the max clique $\boldsymbol{Q}$ that $\boldsymbol{a}$ belongs to
  - **Observation:**
    - If some $\boldsymbol{x}$ belongs to $\boldsymbol{Q}$ then it is a neighbor of $\boldsymbol{a}$
      - **Why?** If $\boldsymbol{a}, \boldsymbol{x} \in \boldsymbol{Q}$ but edge $(\boldsymbol{a}, \boldsymbol{x})$ does not exist, $\boldsymbol{Q}$ is not a clique!
- **Recursive algorithm:**
  - $\boldsymbol{Q}$ … current clique
  - $\boldsymbol{R}$ … candidate vertices to expand the clique to
- **Example:** Start with $\boldsymbol{a}$ and expand around it



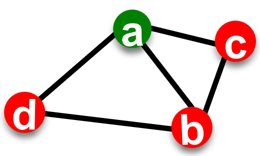
Q=
R=

Steps of the recursive algorithm

Γ(u)…neighbor set of *u*

How to Find Maximal Cliques?

- Start with a seed vertex $\boldsymbol{a}$
- **Goal:** Find the max clique $\boldsymbol{Q}$ that $\boldsymbol{a}$ belongs to
  - **Observation:**
    - If some $\boldsymbol{x}$ belongs to $\boldsymbol{Q}$ then it is a neighbor of $\boldsymbol{a}$
      - **Why?** If $\boldsymbol{a}, \boldsymbol{x} \in \boldsymbol{Q}$ but edge $(\boldsymbol{a}, \boldsymbol{x})$ does not exist, $\boldsymbol{Q}$ is not a clique!
- **Recursive algorithm:**
  - $\boldsymbol{Q}$ … current clique
  - $\boldsymbol{R}$ … candidate vertices to expand the clique to
- **Example:** Start with $\boldsymbol{a}$ and expand around it



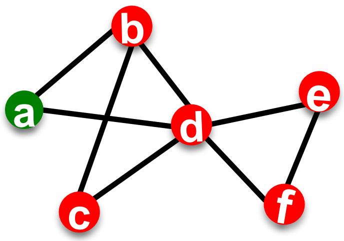
Q=	{a}	{a,b}	{a,b,c}	bktrack	{a,b,d}
R=	{<u>b</u>,c,d}	{b,c,d} $\cap\Gamma$(b)={<u>c</u>,d}	{d}$\cap\Gamma$(c)={}		{c}$\cap\Gamma$(d)={}

Steps of the recursive algorithm

Γ(u)…neighbor set of *u*

How to Find Maximal Cliques?

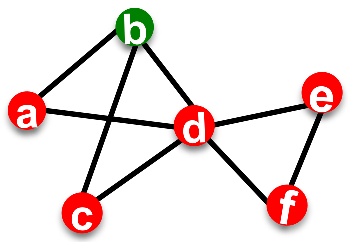
- Q ... current clique
- R ... candidate vertices
- **Expand(R,Q)**
  - **while** $R \neq \{\}$
    - p = vertex in R
    - $Q_p = Q \cup \{p\}$
    - $R_p = R \cap \Gamma(p)$
    - **if** $R_p \neq \{\}$: Expand(R_p,Q_p)
      else: output Q_p
    - $R = R - \{p\}$



How to Find Maximal Cliques?

- Q ... current clique
- R ... candidate vertices

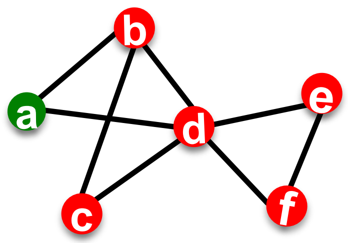
- **Expand(R,Q)**
  - **while** R ≠ {}
    - p = vertex in R
    - Q_p = Q ∪ {p}
    - R_p = R ∩ Γ(p)
    - **if** R_p ≠ {}: Expand(R_p,Q_p)
      else: output Q_p
    - R = R – {p}



Start: Expand(V, {})
R={a,…f}, Q={}
p = {b}
Q_p = {b}
R_p = {a,c,d}
Expand(R_p, Q):
R = {a,c,d}, Q={b}
p = {a}
Q_p = {b,a}
R_p = {d}
Expand(R_p, Q):
R = {d}, Q={b,a}
p = {d}
Q_p = {b,a,d}
R_p = {} : **output {b,a,d}**
p = {c}
Q_p = {b,c}
R_p = {d}
Expand(R_p, Q):
R = {d}, Q={b,c}
p = {d}
Q_p = {b,c,d}
R_p = {} : **output {b,c,d}**

How to Find Maximal Cliques?

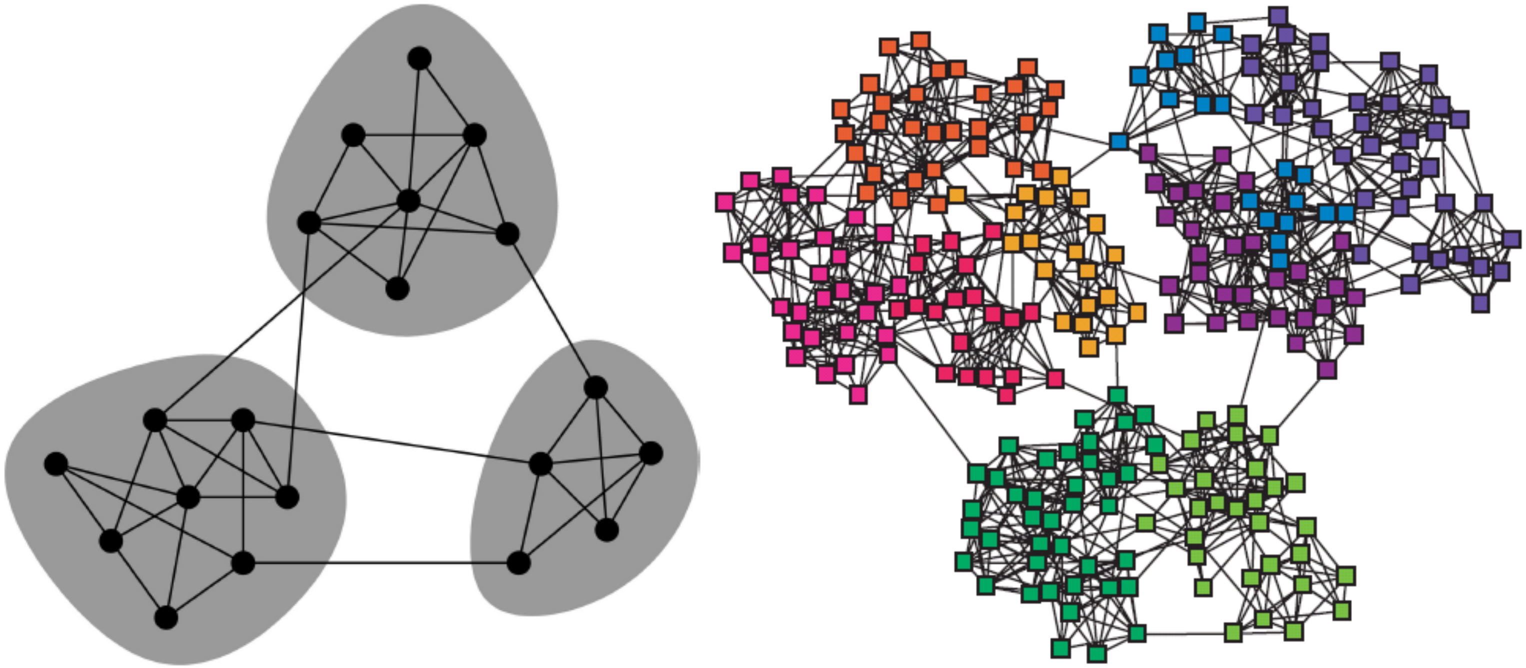
- **How to prevent maximal cliques to be generated multiple times?**
  - Only output cliques that are lexicographically minimum
    - $\{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}\} < \{\boldsymbol{b},\boldsymbol{a},\boldsymbol{c}\}$
  - **Even better:** Only expand to the nodes higher in the lexicographical order



How to Model Networks with Communities?

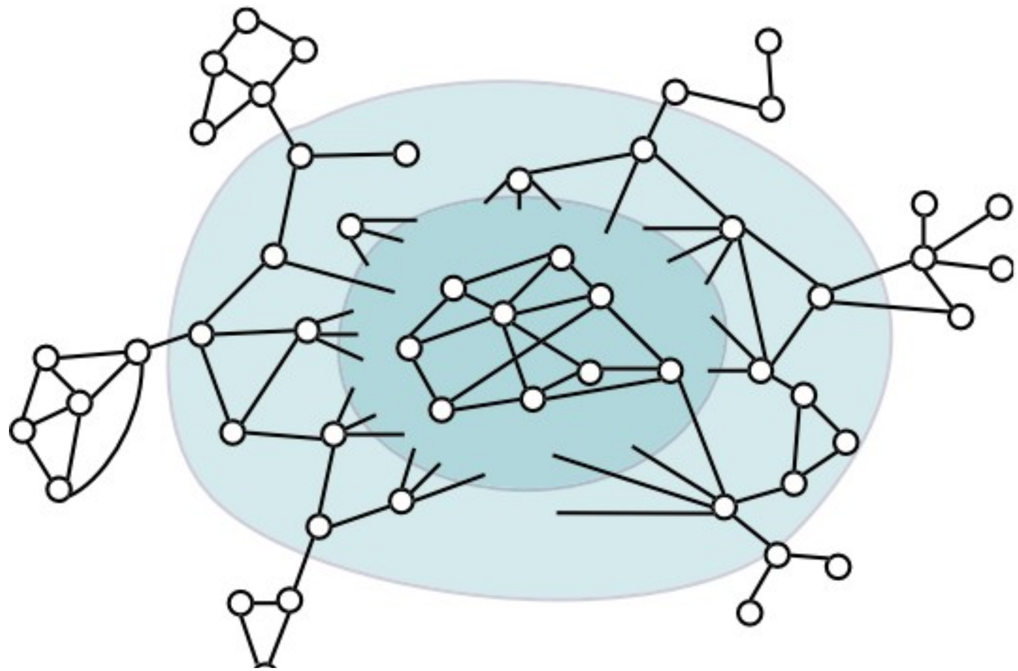
Network and Communities

- **How should we think about large scale organization of clusters in networks?**
  - **Finding:** Community Structure



Network and Communities

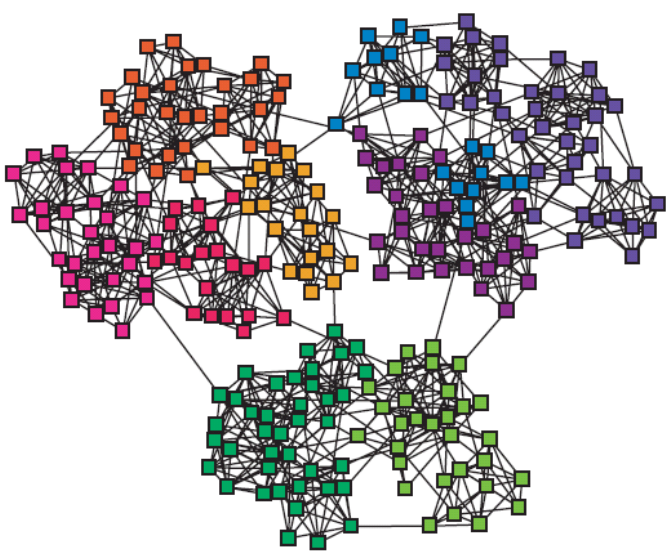
- **How should we think about large scale organization of clusters in networks?**
  - **Finding:** Core-periphery structure



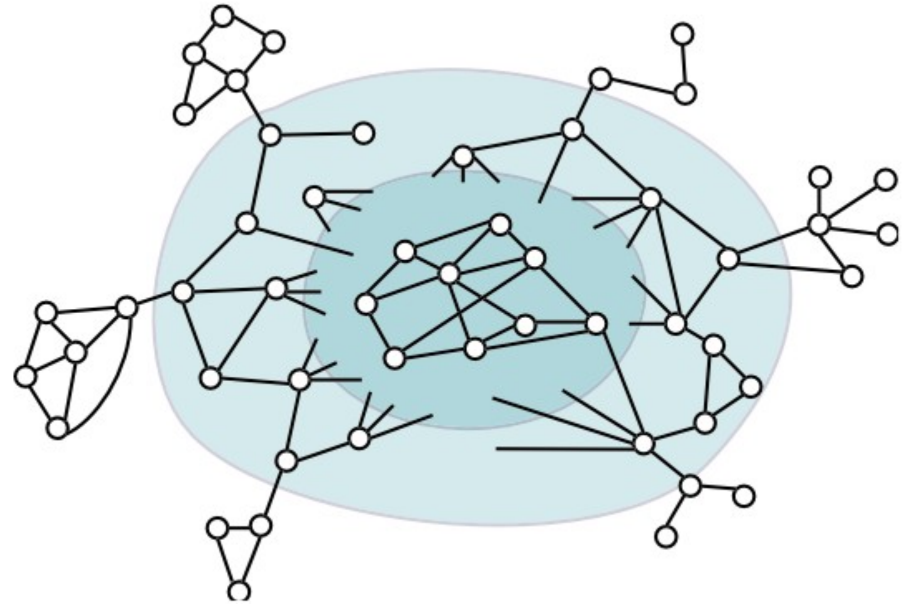
Nested Core-Periphery

Network and Communities

- **How do we reconcile these two views?**
  (and still do community detection)



vs.

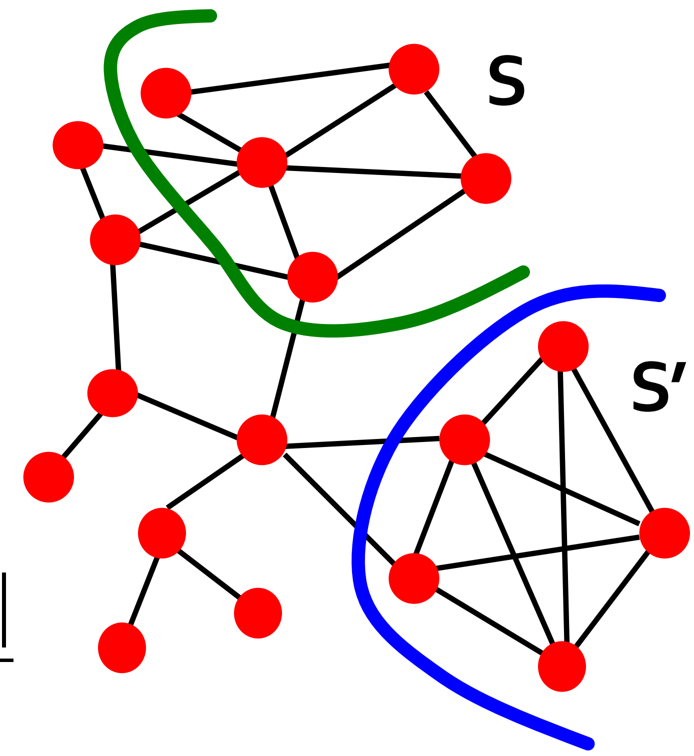


Community structure

Core-periphery

Community Score

- **How community-like is a set of nodes?**
- ***A good cluster S has***
  - Many edges internally
  - Few edges pointing outside
- What's a good metric: **Conductance**



$$\phi(S) = \frac{|\{(i,j) \in E; i \in S, j \notin S\}|}{\sum_{s \in S} d_s}$$

Small conductance corresponds to good clusters
(Note |S| < |V|/2)

Network Community Profile Plot

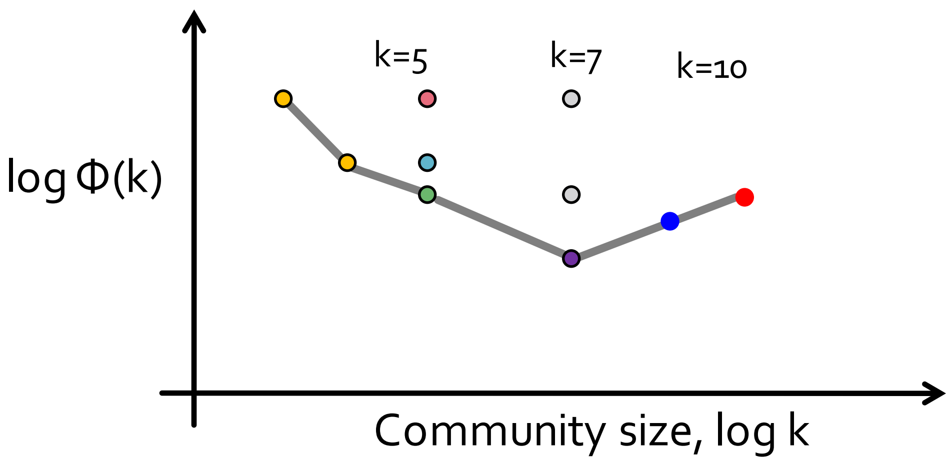
(Note |S| < |V|/2)

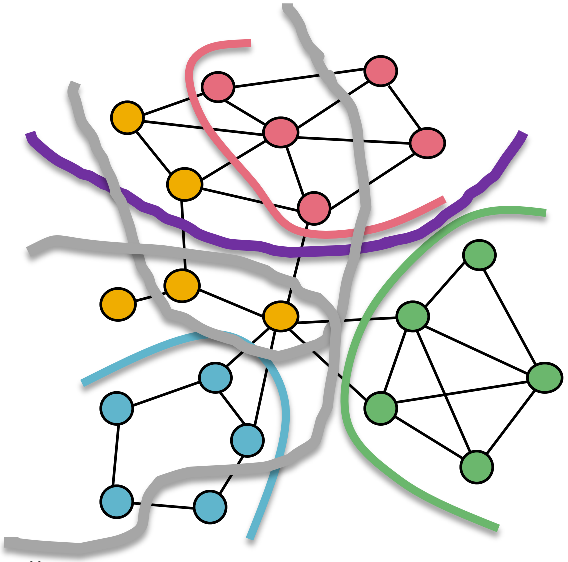
- <u>Define:</u>

  Network community profile (**NCP**) plot

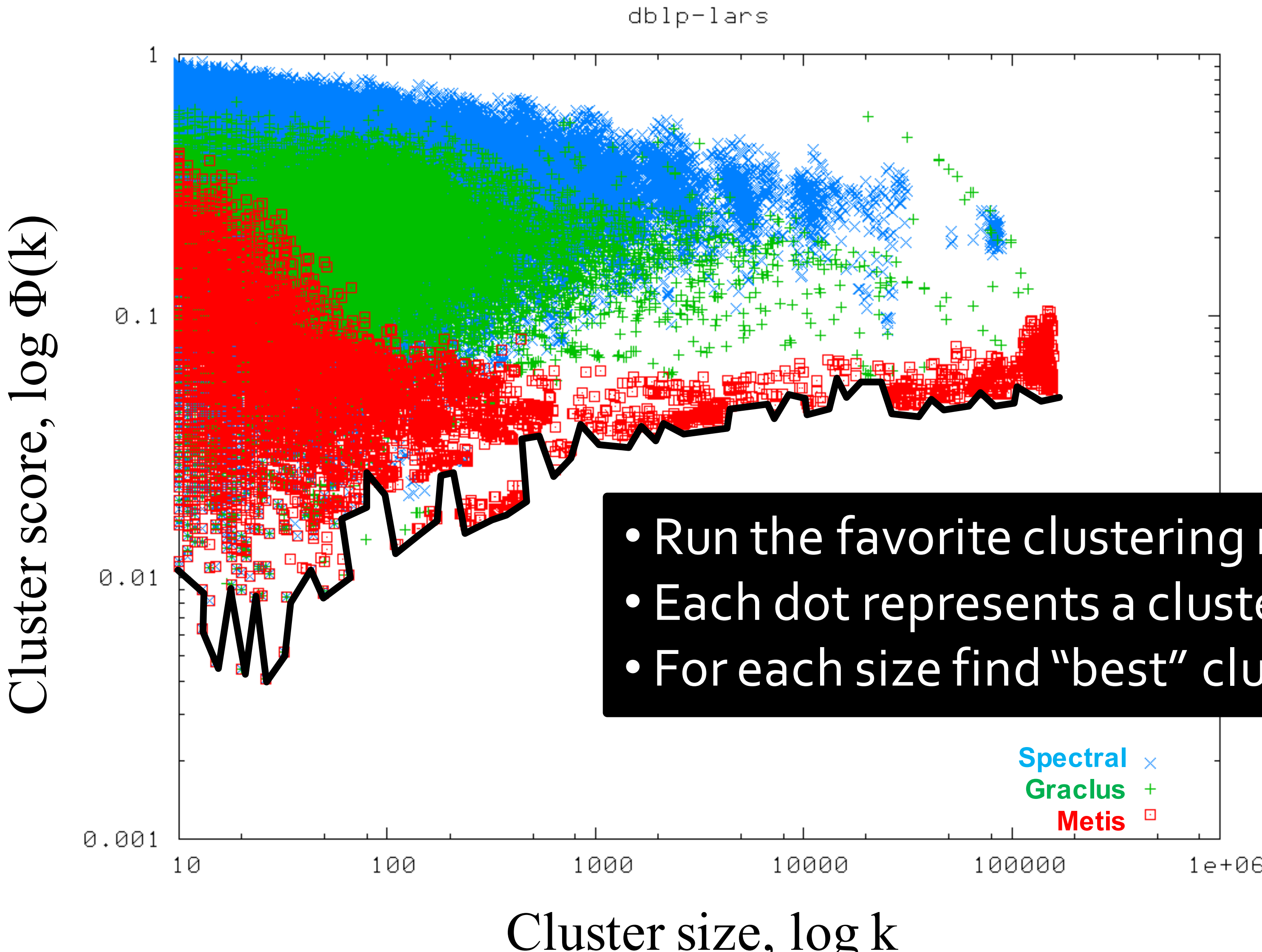
  Plot the score of **best** community of size *k*

$$\Phi(k) = \min_{S \subset V, |S|=k} \phi(S)$$





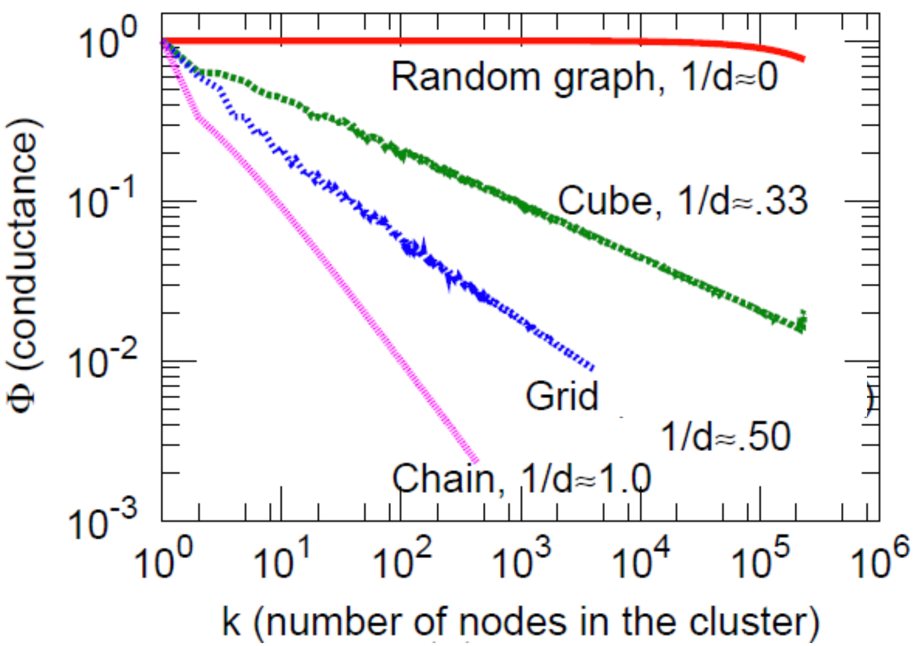
How to (Really) Compute NCP?



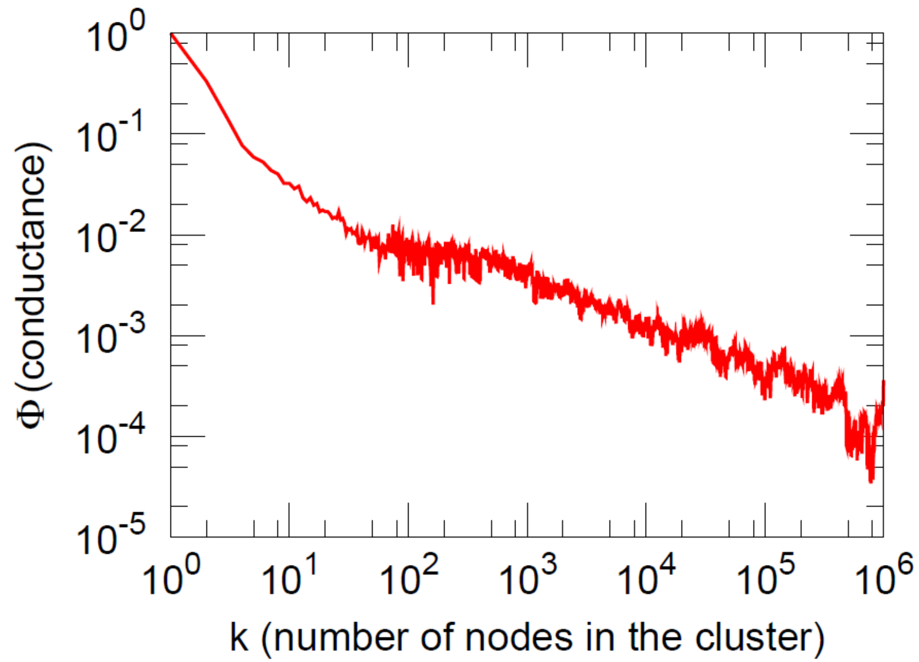
- Run the favorite clustering method
- Each dot represents a cluster
- For each size find "best" cluster

NCP Plot: Meshes

- Meshes, grids, dense random graphs:



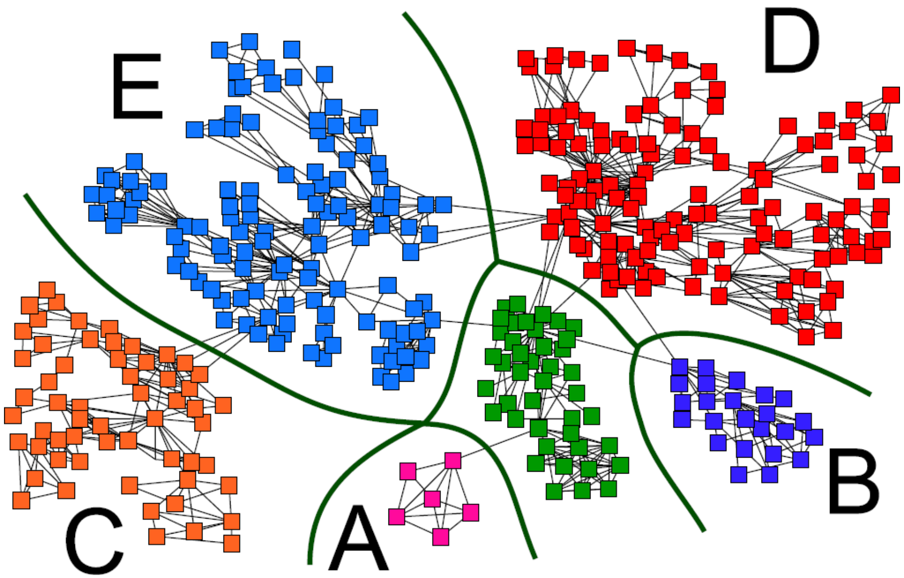
d-dimensional meshes

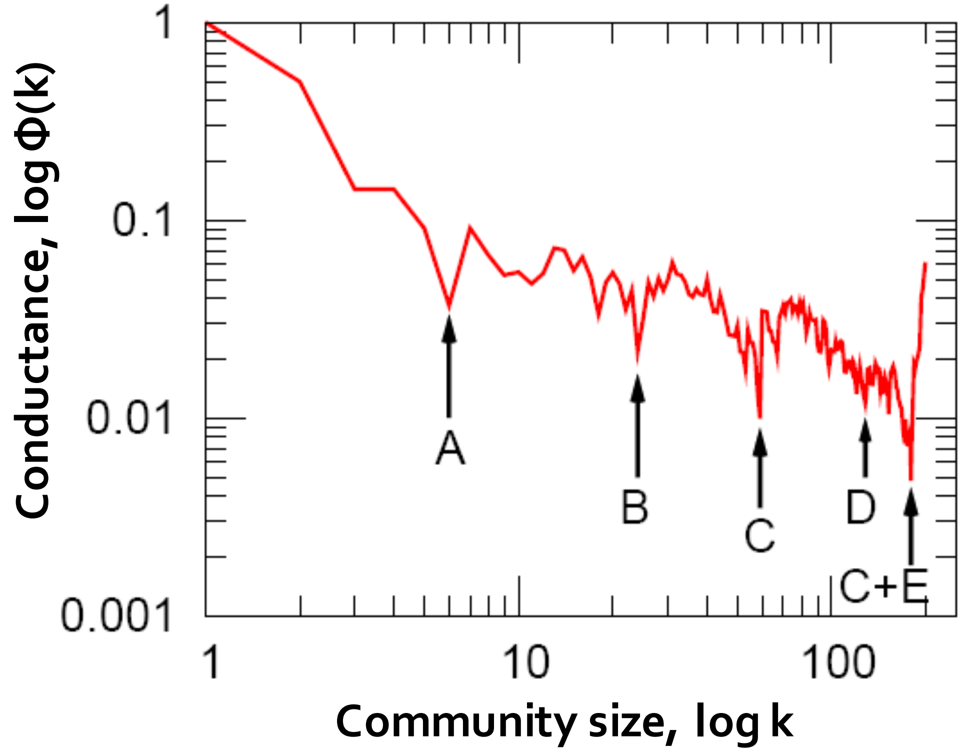


California road network

NCP plot: Network Science

- **Collaborations between scientists in networks** [Newman, 2005]



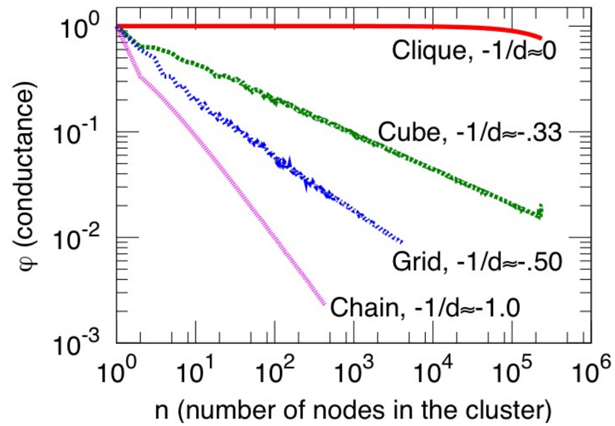


Dips in the conductance graph correspond to the "good" clusters we can visually detect

Natural Hypothesis

Natural hypothesis about NCP:

- NCP of real networks slopes downward
- Slope of the NCP corresponds to the "dimensionality" of the network



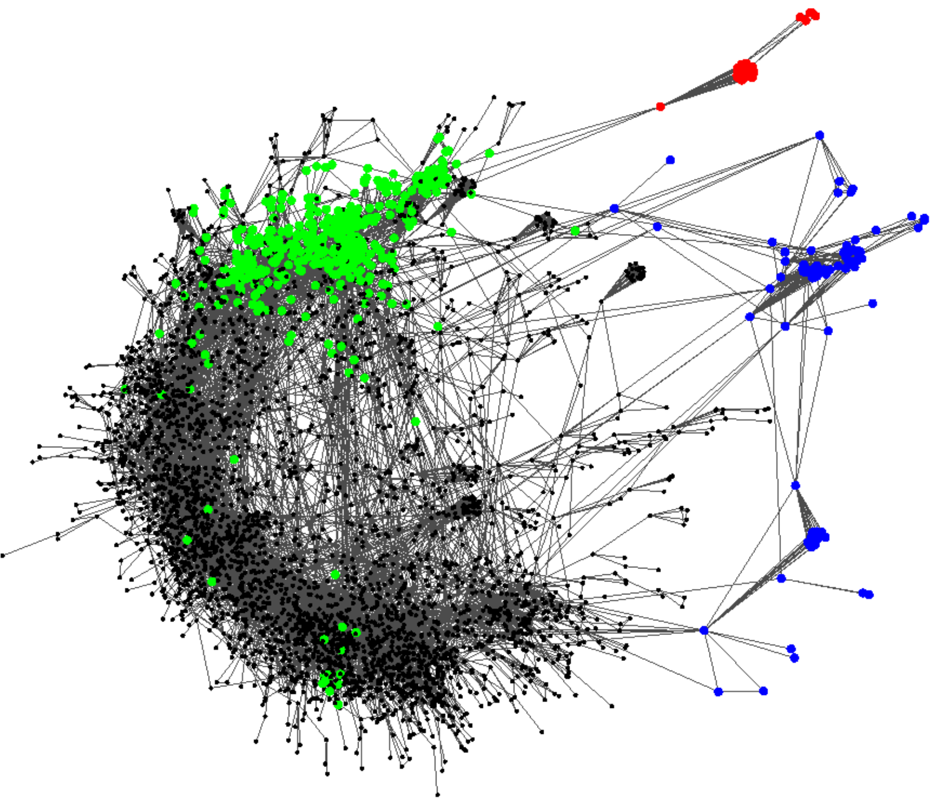
What about large networks?

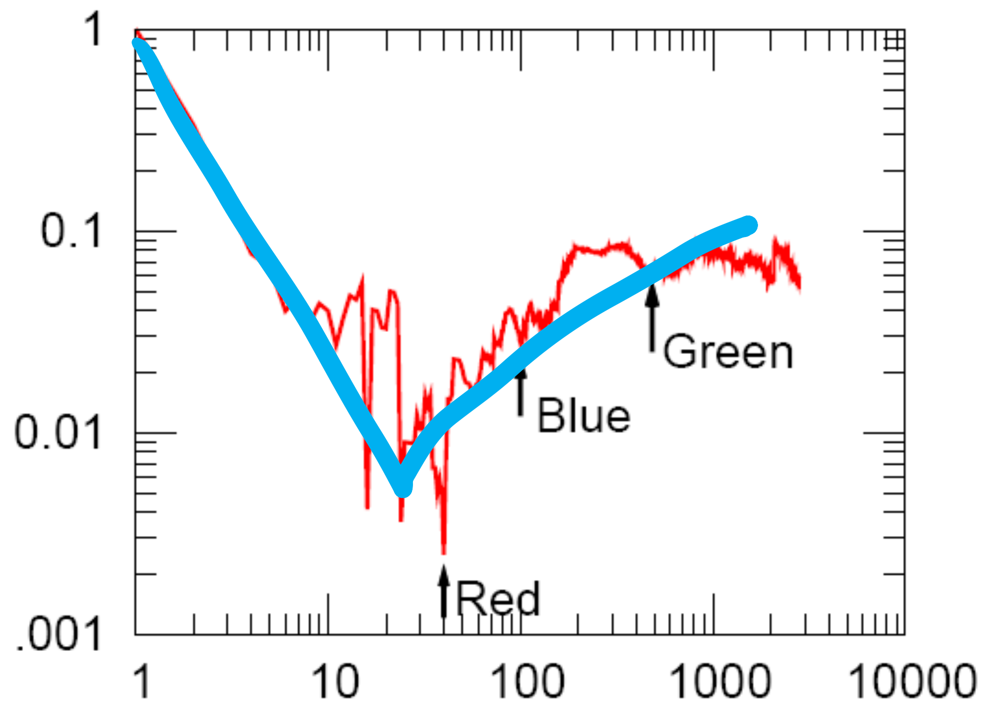
• Social nets	Nodes	Edges	Description
LiveJournal	4,843,953	42,845,684	Blog friendships [5]
Epinions	75,877	405,739	Trust network [28]
CA-DBLP	317,080	1,049,866	Co-authorship [5]
• Information (citation) networks			
Cit-hep-th	27,400	352,021	Arxiv hep-th [14]
AmazonProd	524,371	1,491,793	Amazon products [8]
• Web graphs			
Web-google	855,802	4,291,352	Google web graph
Web-wt10g	1,458,316	6,225,033	TREC WT10G
• Bipartite affiliation (authors-to-papers) networks			
Atp-DBLP	615,678	944,456	DBLP [21]
AtM-Imdb	2,076,978	5,847,693	Actors-to-movies
• Internet networks			
AsSkitter	1,719,037	12,814,089	Autonom. sys.
Gnutella	62,561	147,878	P2P network [29]

[Internet Mathematics'09]

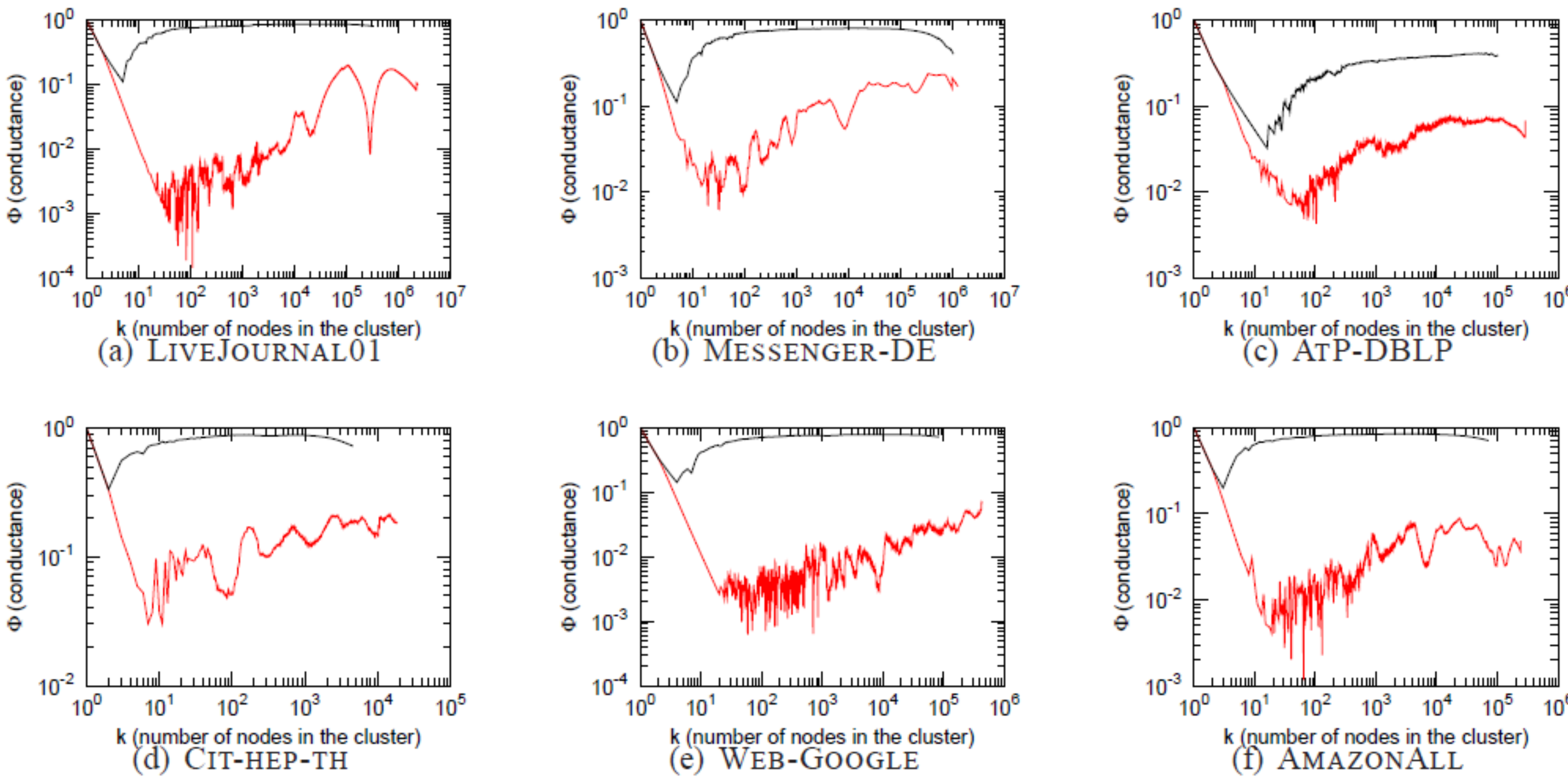
Large Networks: Very Different

Typical example: General Relativity collaborations (n=4,158, m=13,422)



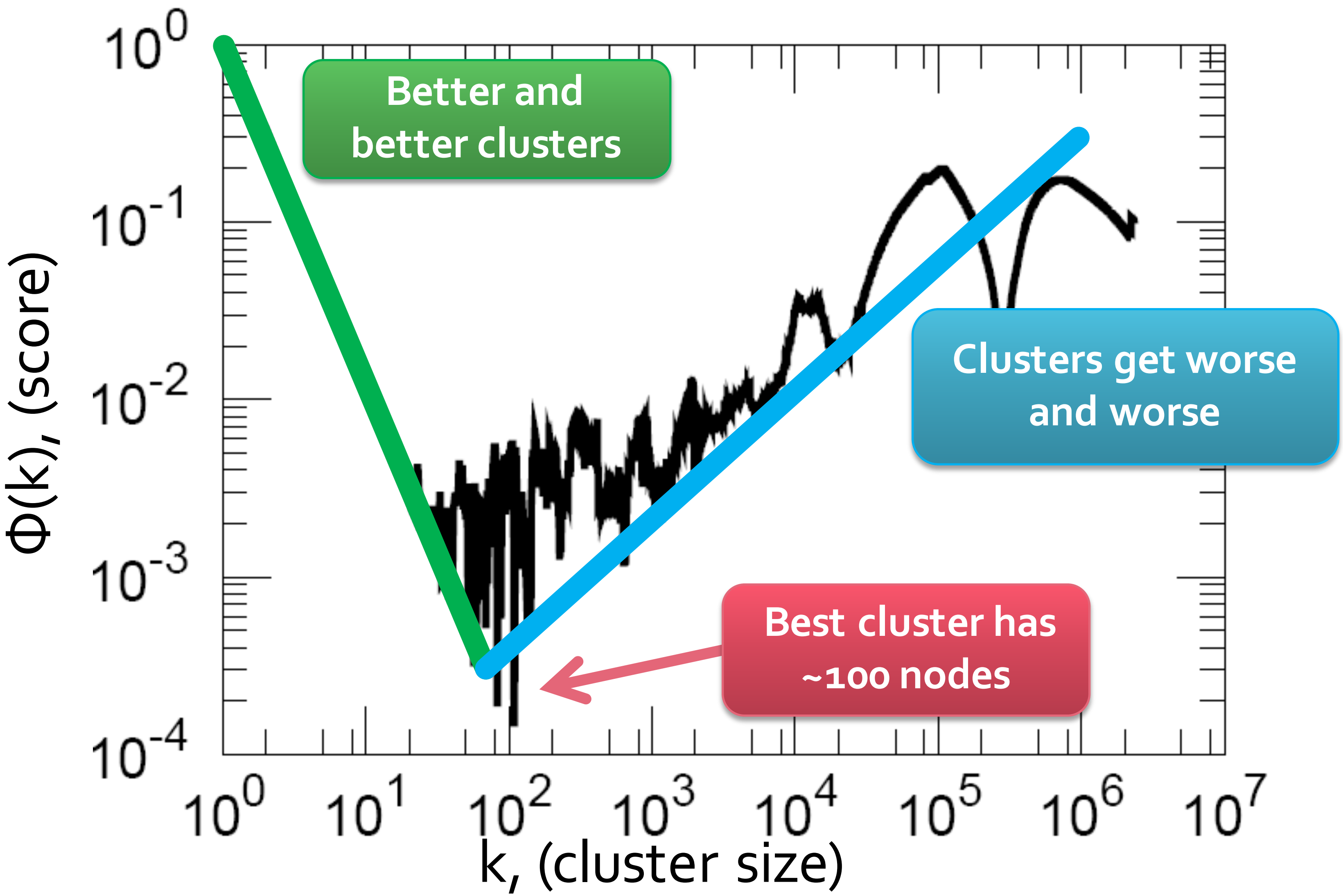


More NCP Plots of Networks



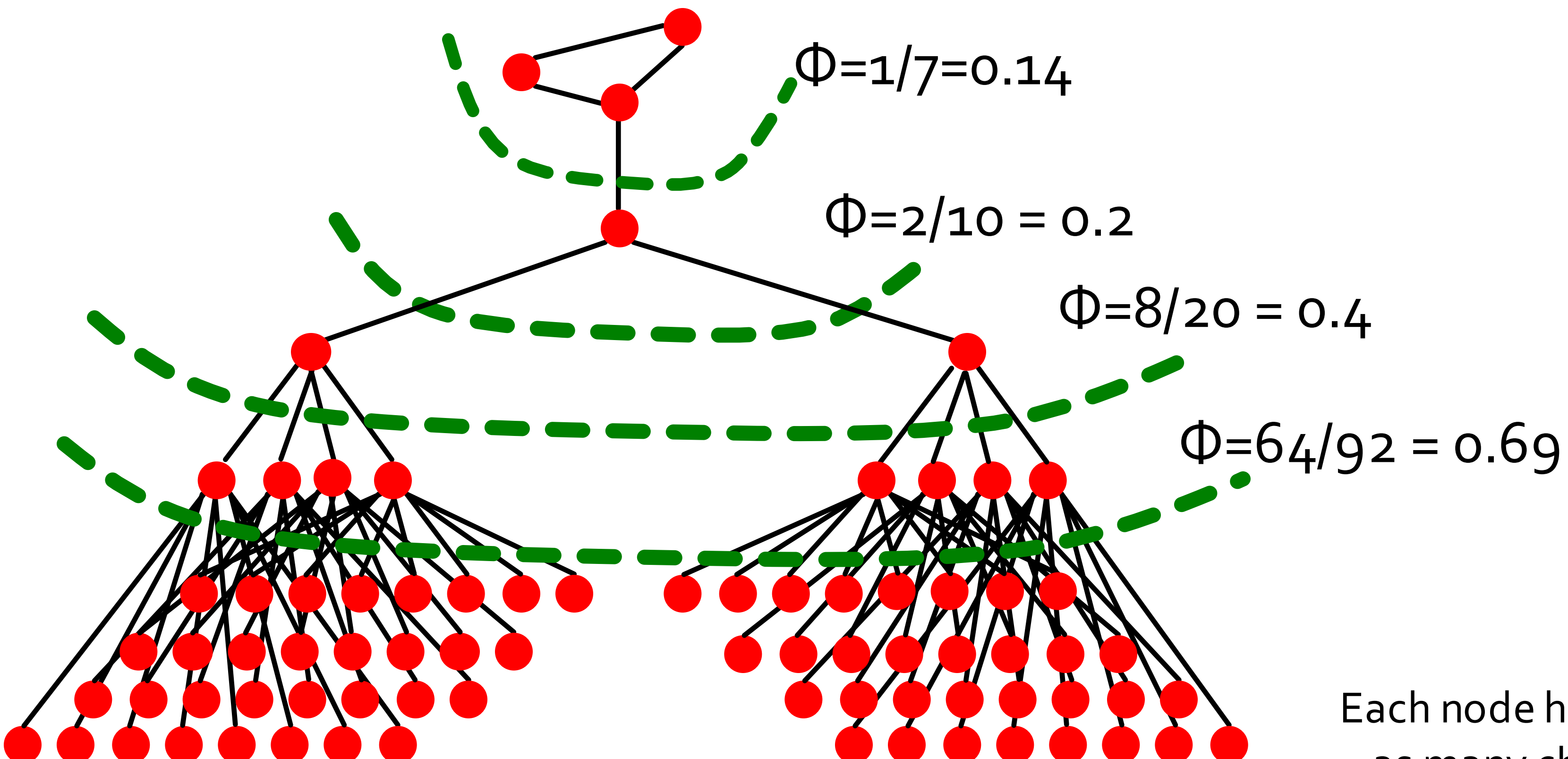
-- Rewired graph
-- Real graph

NCP: LiveJournal (n=5m, m=42m)



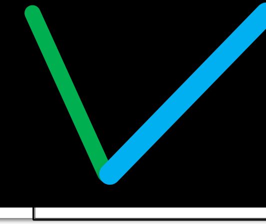
Explanation: The Upward Part

- As clusters grow the number of edges inside grows **slower** that the number crossing

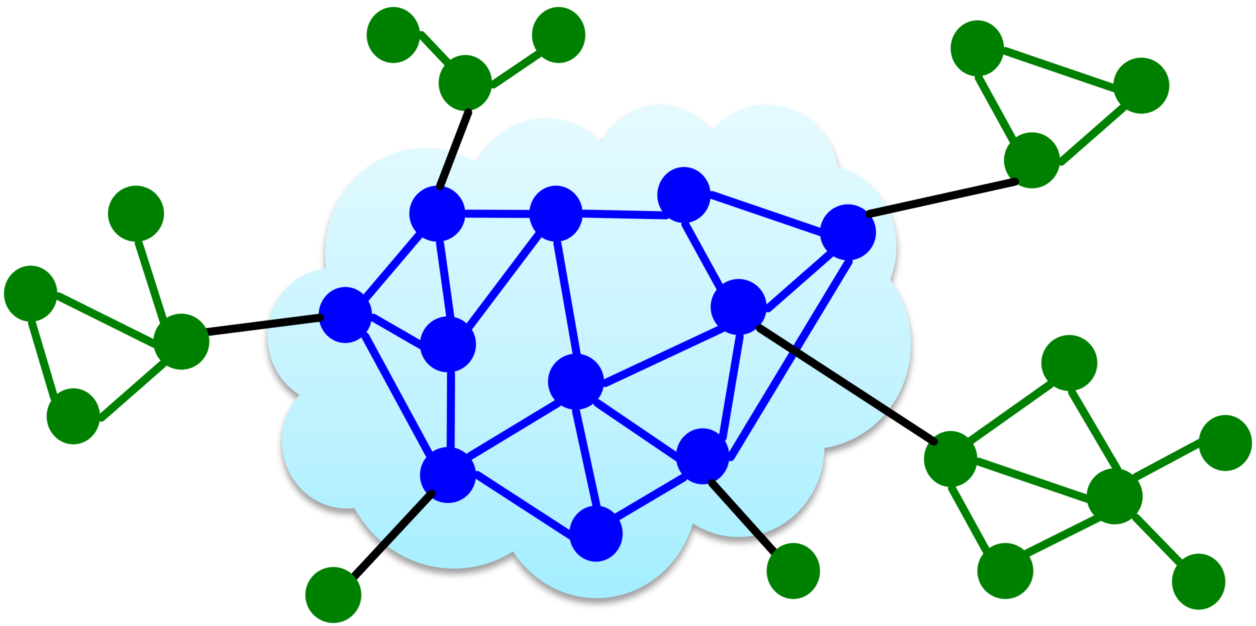


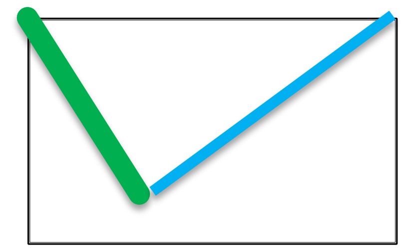
Each node has twice as many children

Explanation: Downward Part



- Empirically we note that **best clusters** (corresponding to **green nodes**) are **barely connected** to the network

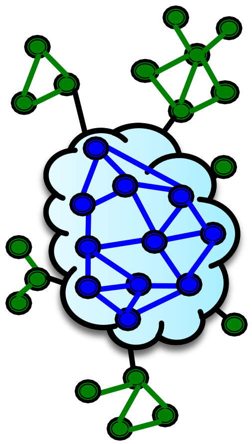


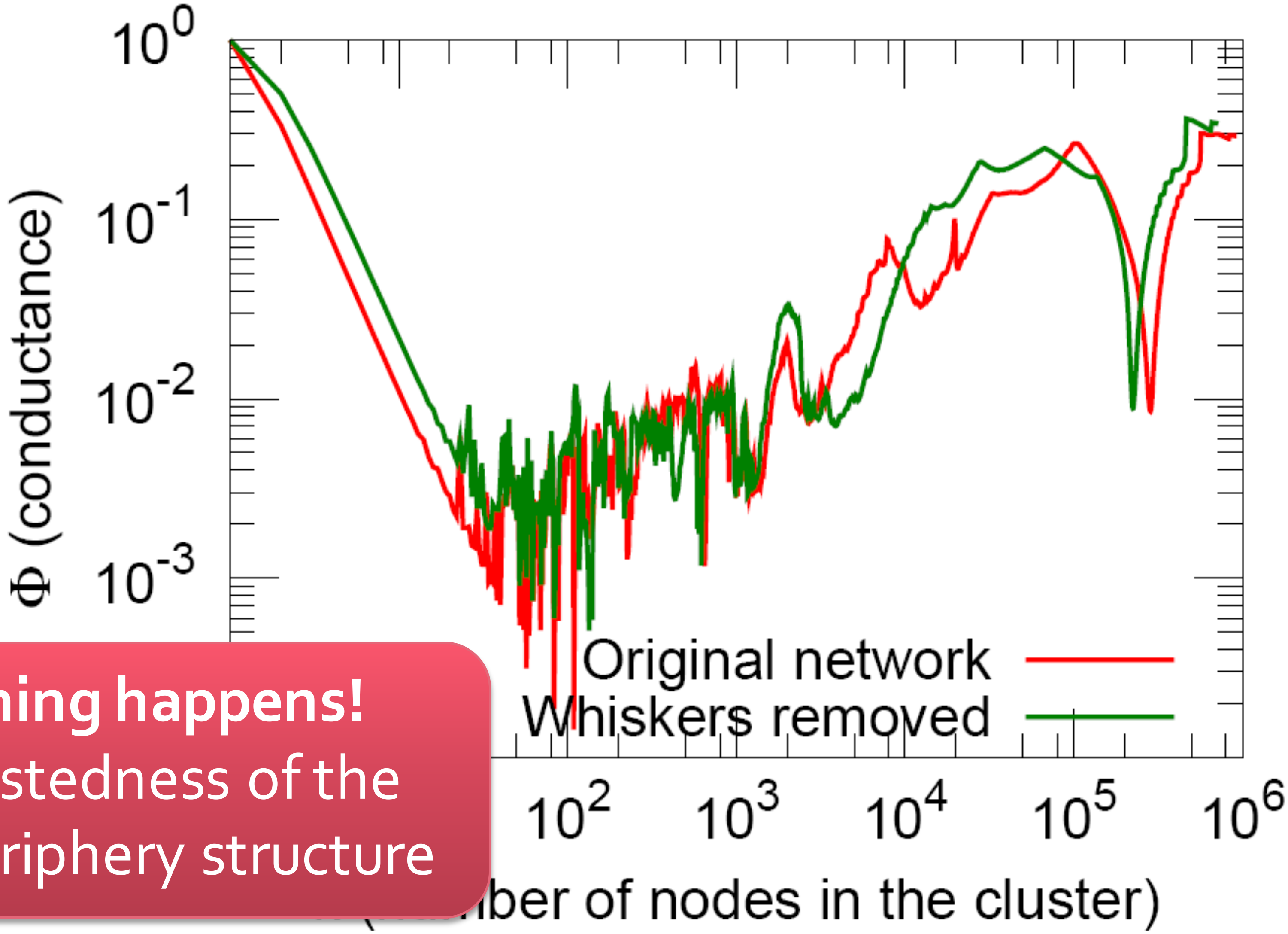


NCP plot

⇒ Core-periphery structure

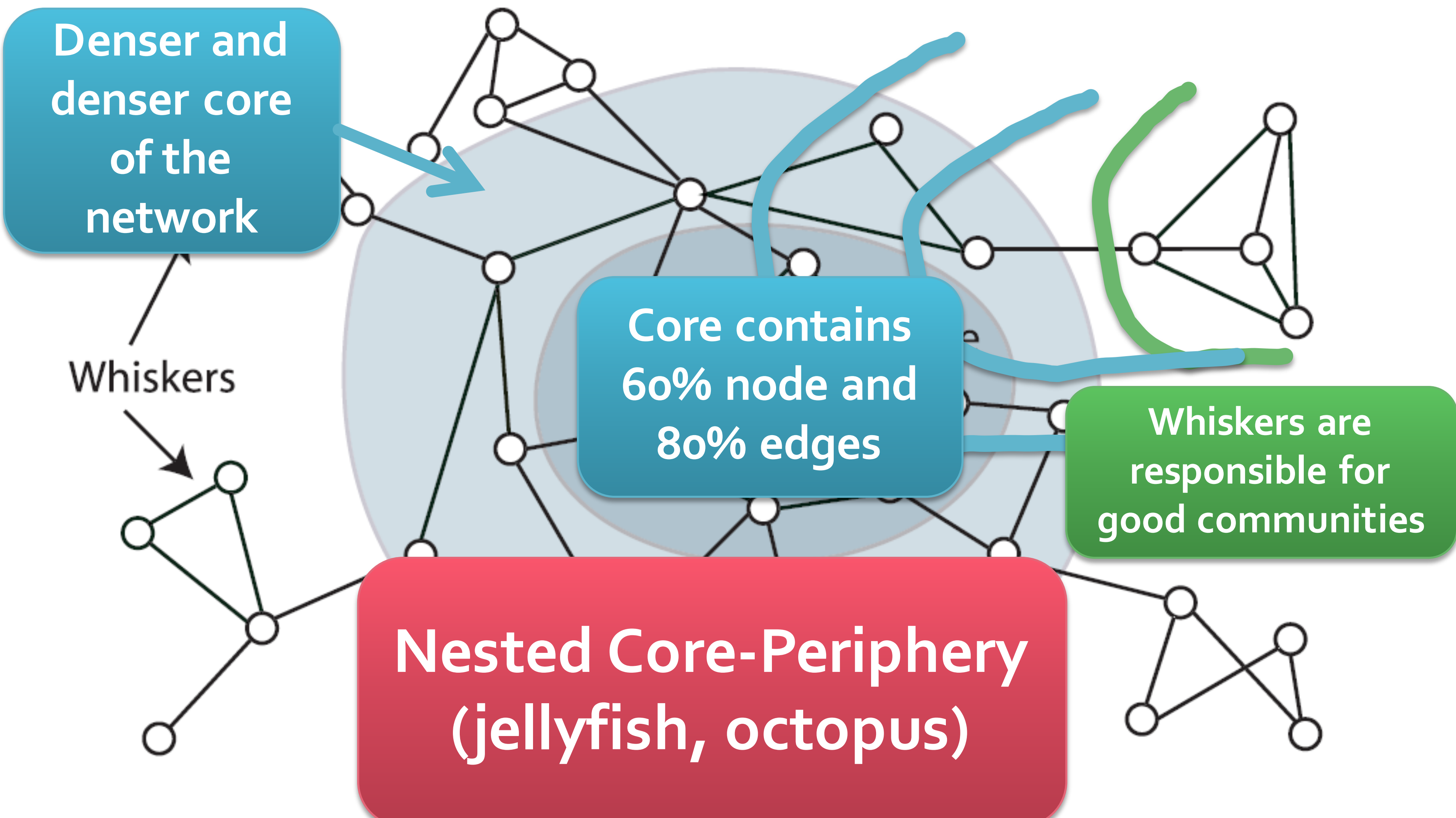
What If We Remove Good Clusters?





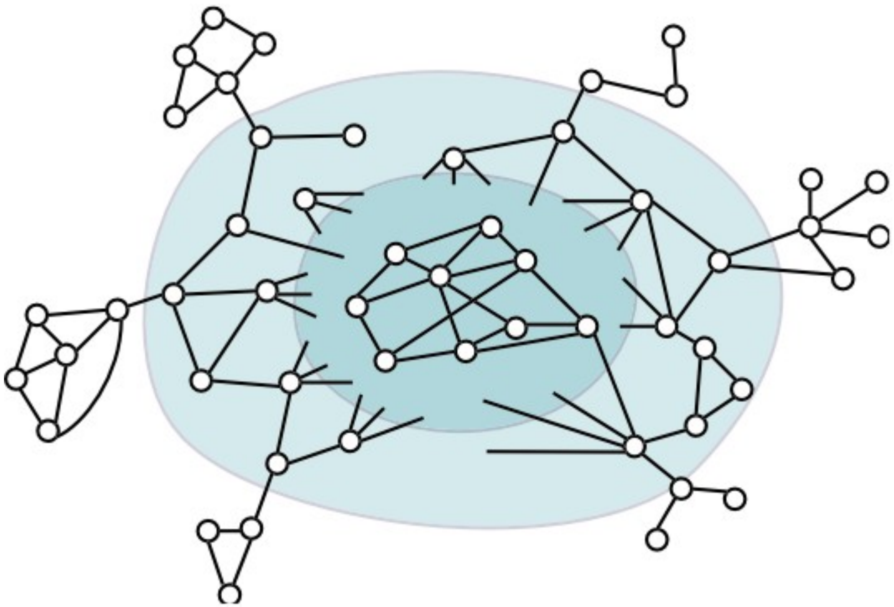
Nothing happens!
⇒ Nestedness of the core-periphery structure

Suggested Network Structure

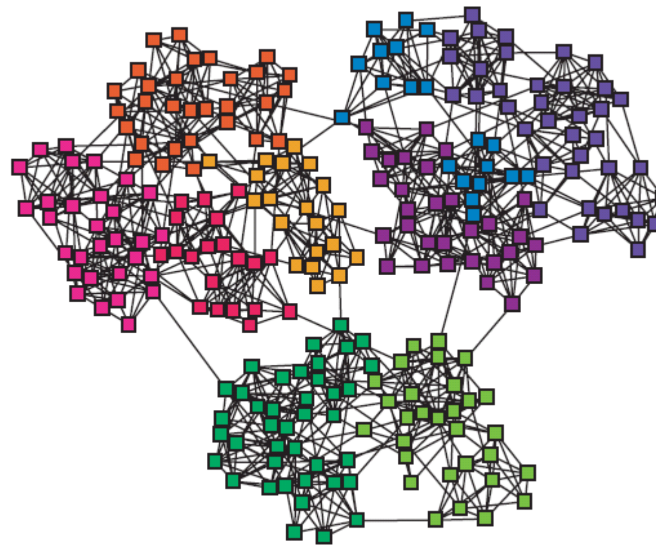


Nested Core-Periphery (jellyfish, octopus)

Part 2: Explanation



vs.



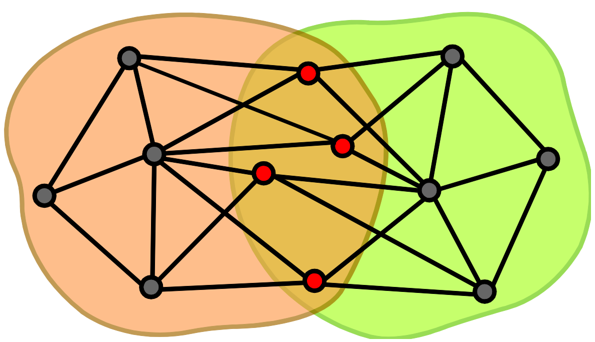
How do we reconcile these two views?

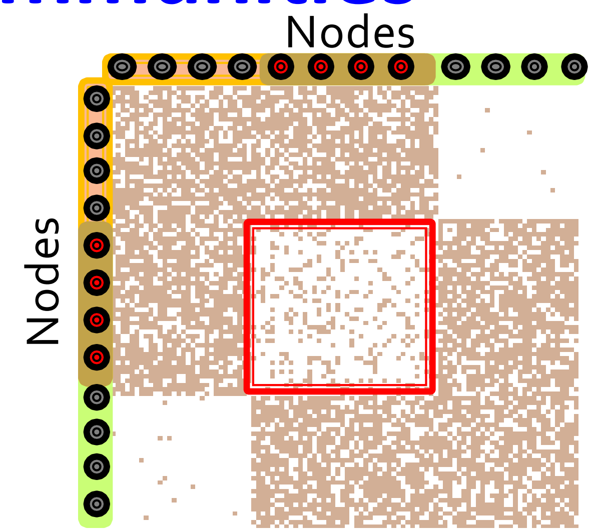
Overlapping Community Detection

- **Many methods for overlapping communities**
  - Clique percolation [Palla et al. '05]
  - Link clustering [Ahn et al. '10] [Evans et al.'09]
  - Clique expansion [Lee et al. '10]
  - Mixed membership stochastic block models [Airoldi et al. '08]
  - Bayesian matrix factorization [Psorakis et al. '11]
- **What do these methods assume about community overlaps?**

Overlapping Communities

- **Many overlapping community detection methods make an implicit assumption:**
  - **Edge probability decreases with the number of shared communities**

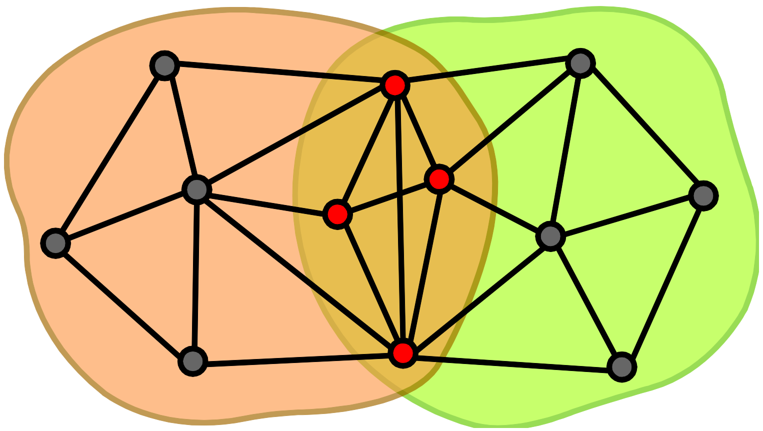


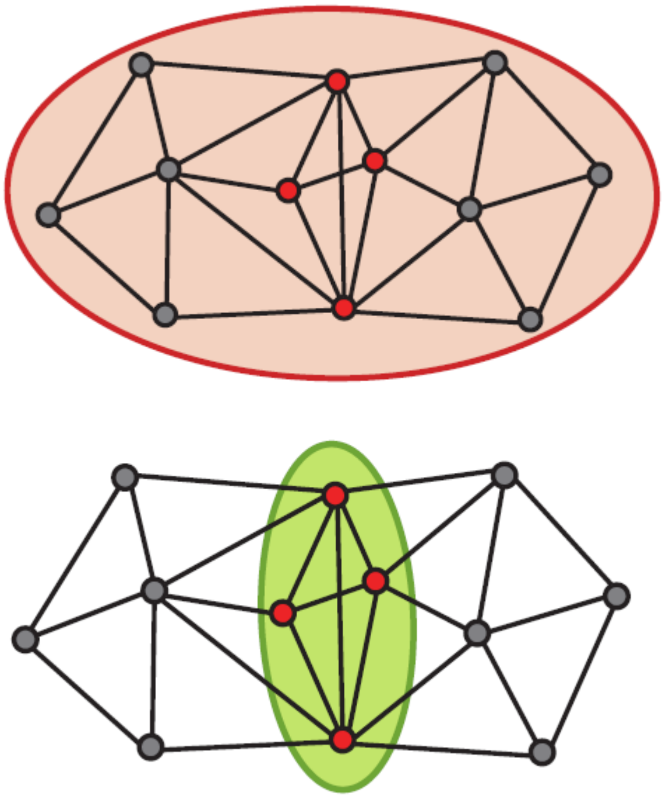


Is this true?

Example: CPM

- **Clique Percolation Method fails to detect dense overlaps:**

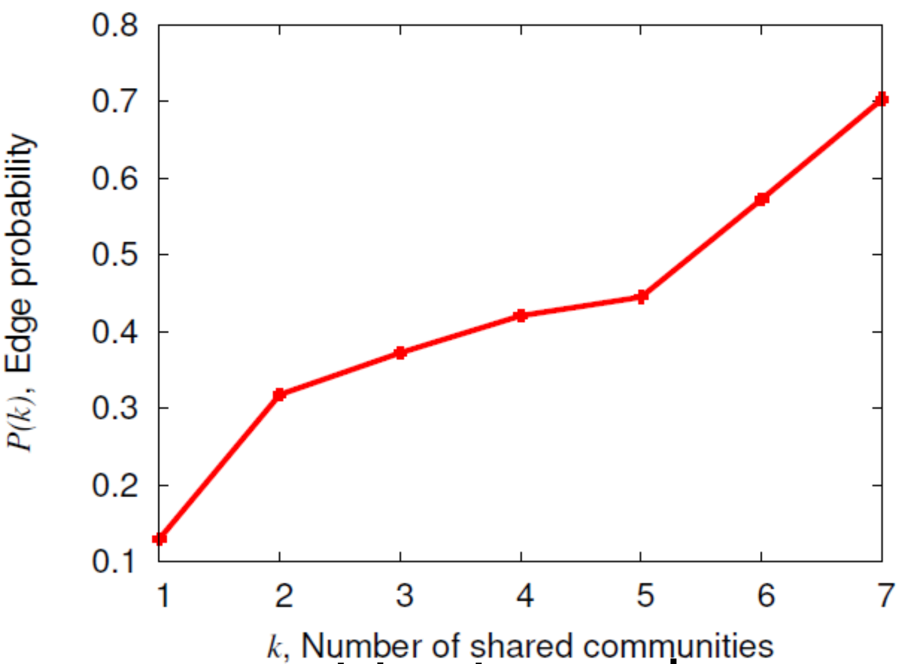




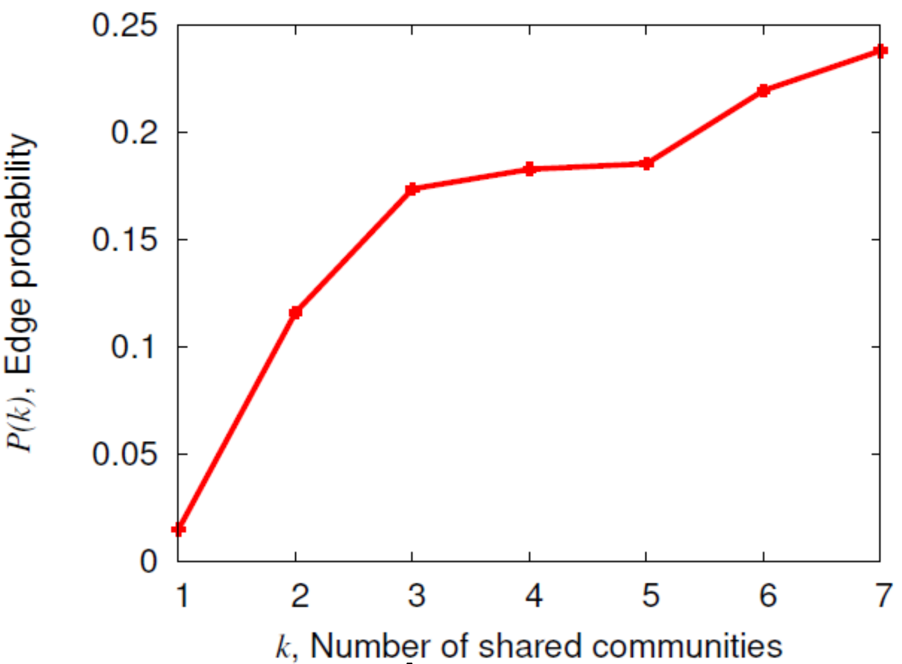
Clique percolation

Ground-truth Communities

- **Basic question: nodes u, v share k communities**
- **What's the edge probability?**



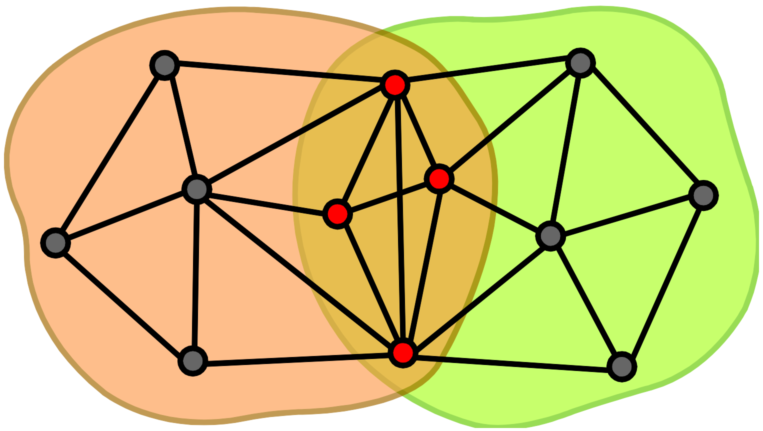
LiveJournal
social network

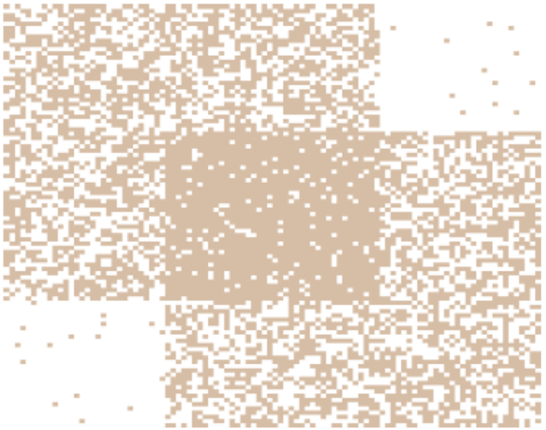


Amazon
product network

Communities as Tiles!

- **Edge density in the overlaps is higher!**

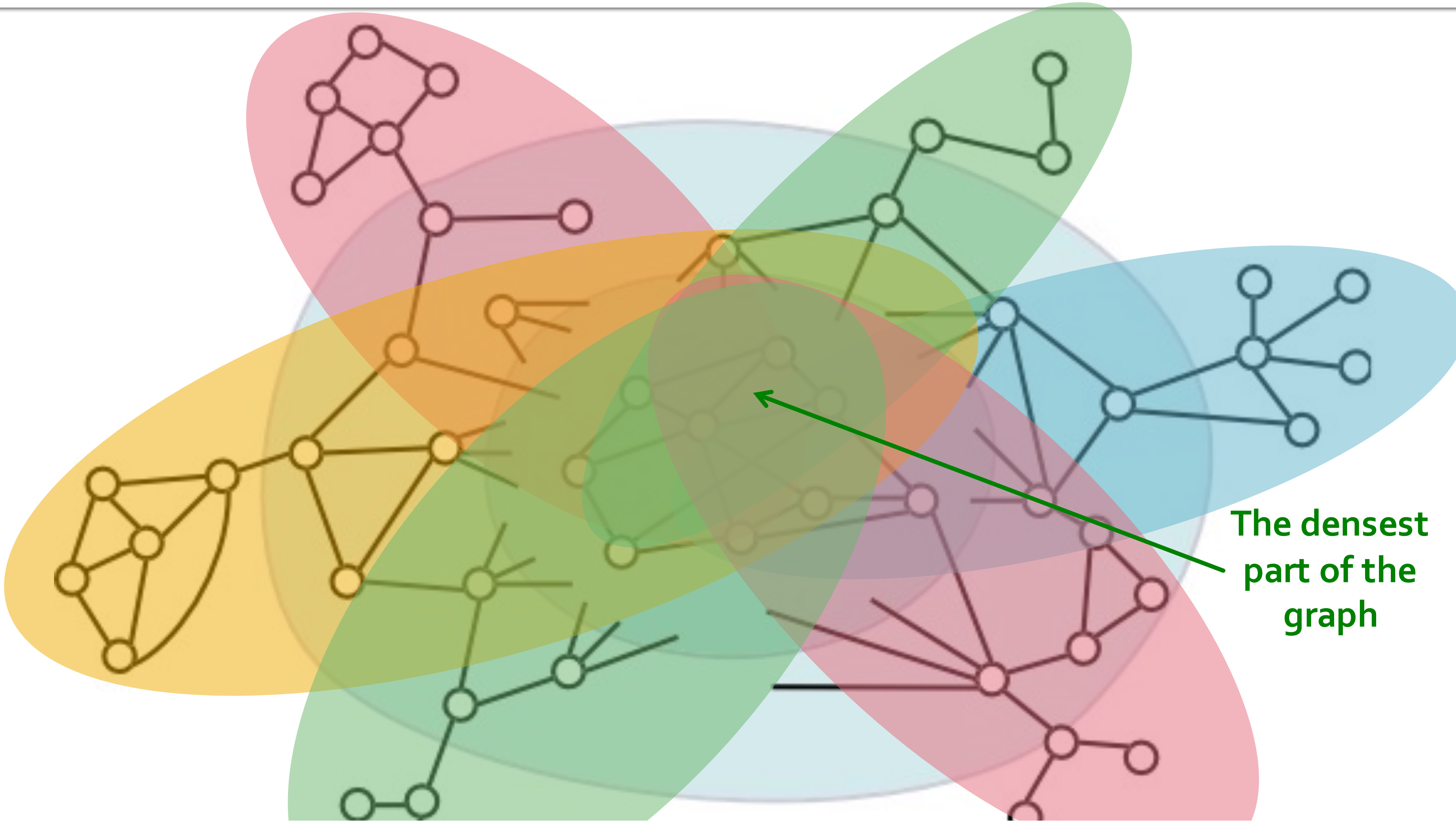




"The more different foci (communities) that two individuals share, the more likely it is that they will be tied" - S. Feld, 1981

Communities as "tiles"

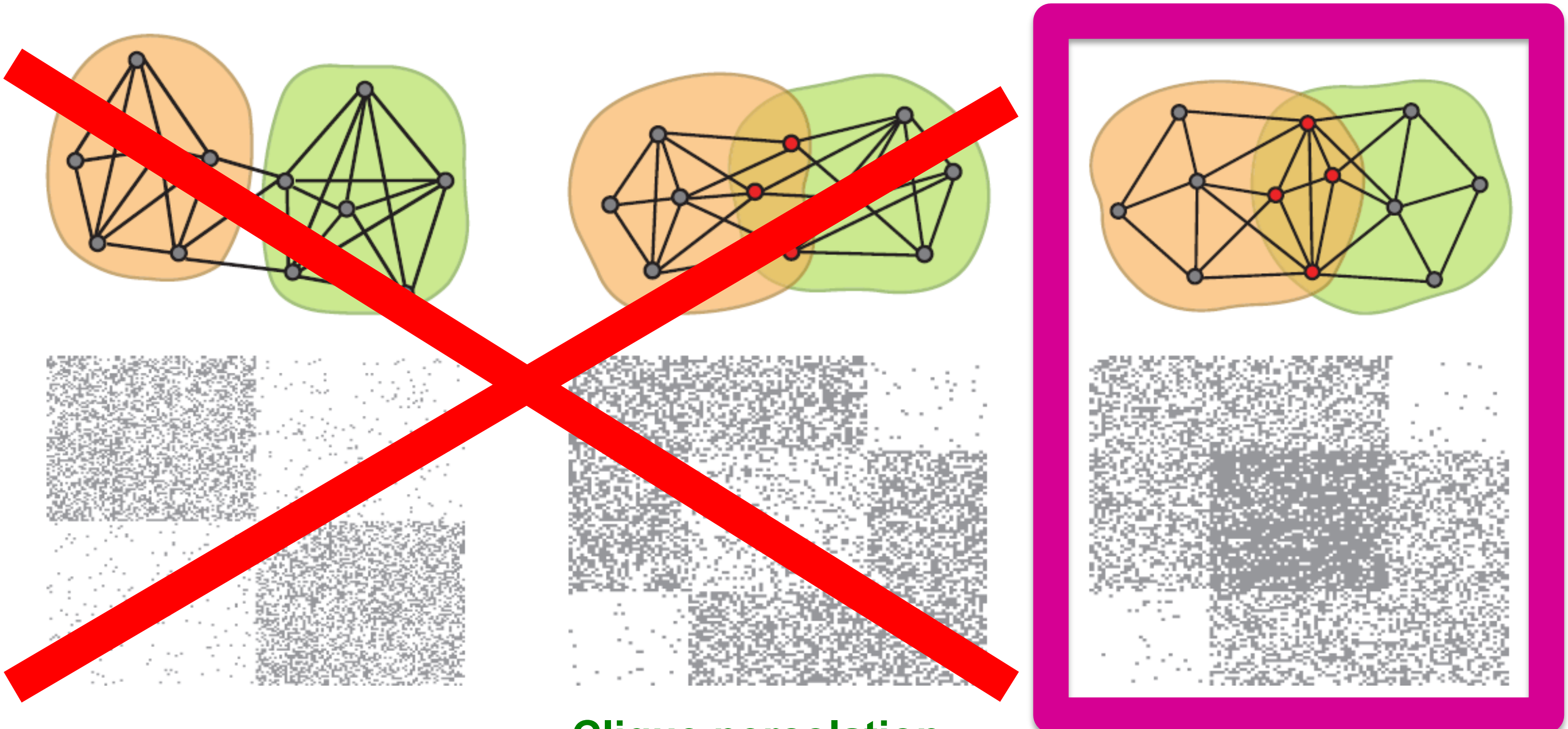
Communities as Tiles/Circles



Communities as overlapping tiles

Communities in Networks

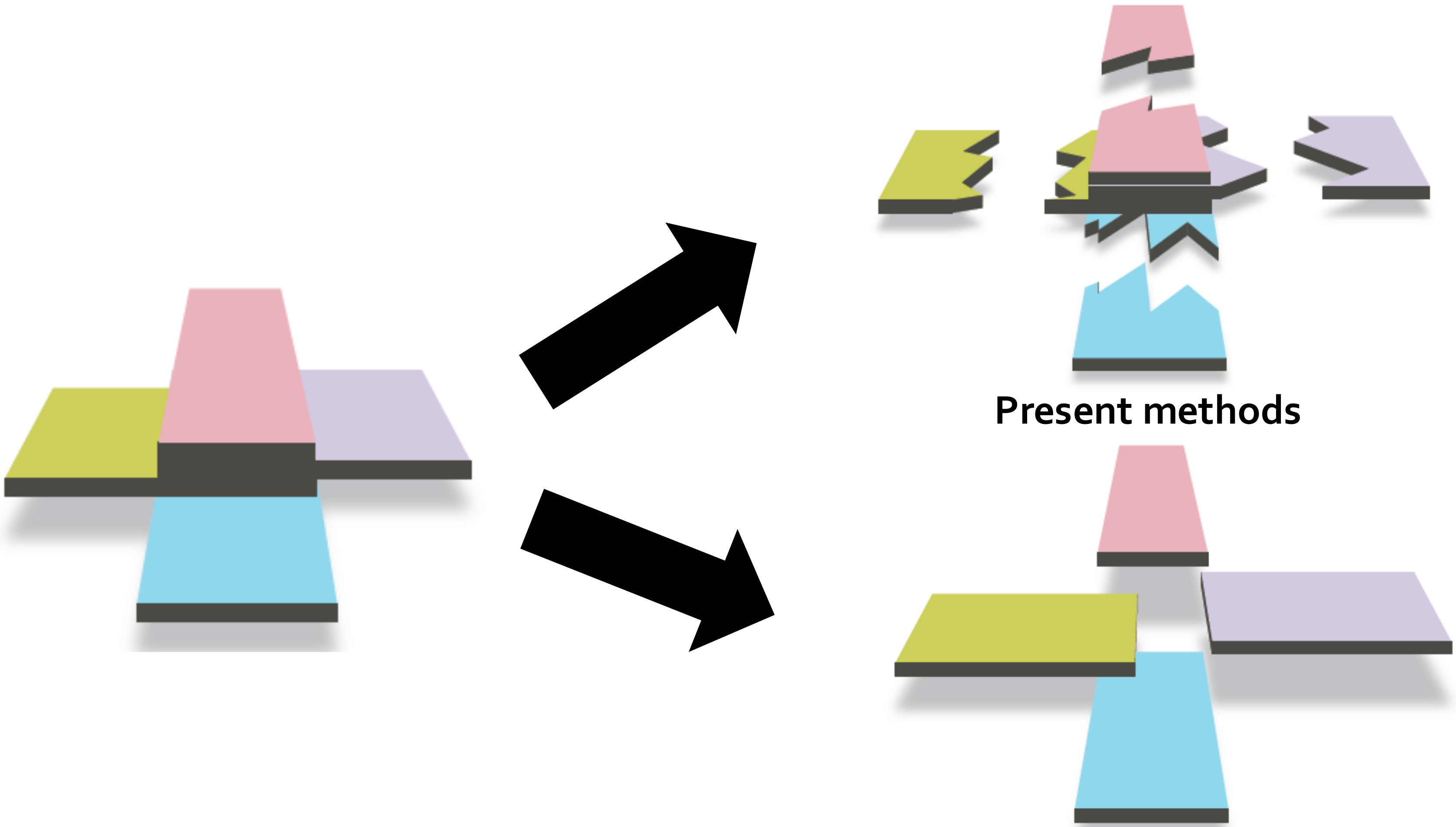
What does this mean?



Non-overlapping methods (spectral, modularity optimization)

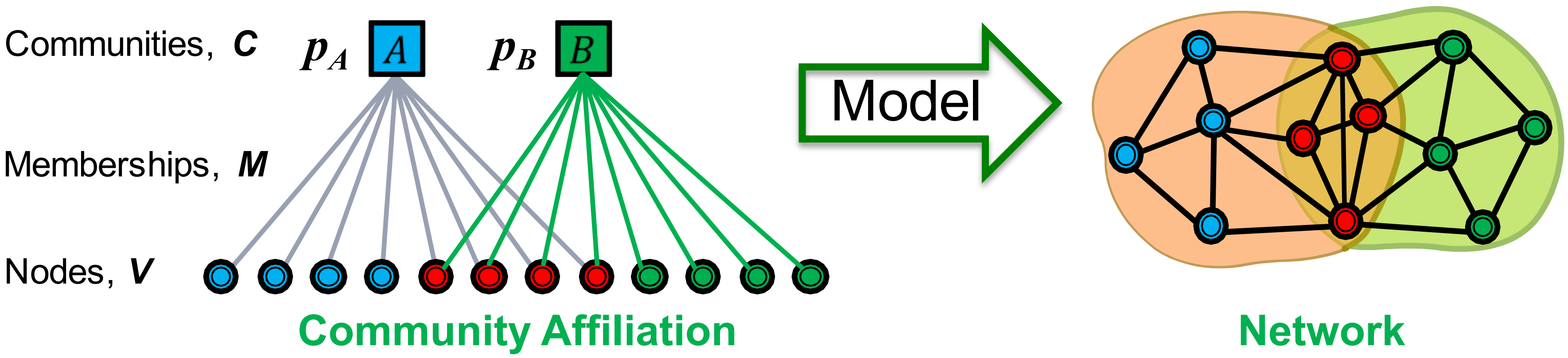
Clique percolation, and many other overlapping methods as well

From Networks to Communities



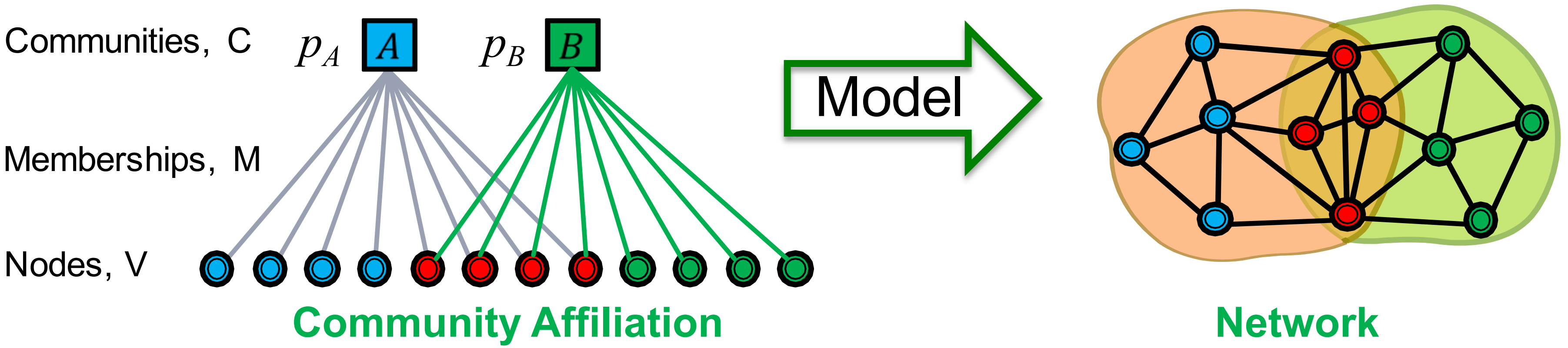
How do we detect communities if they overlap as tiles?

Community-Affiliation Graph Model (AGM)



- **Generative model: How is a network generated from community affiliations?**
- **Model parameters:**
  - Nodes ***V***, Communities ***C***, Memberships ***M***
  - Each community c has a single probability p_c

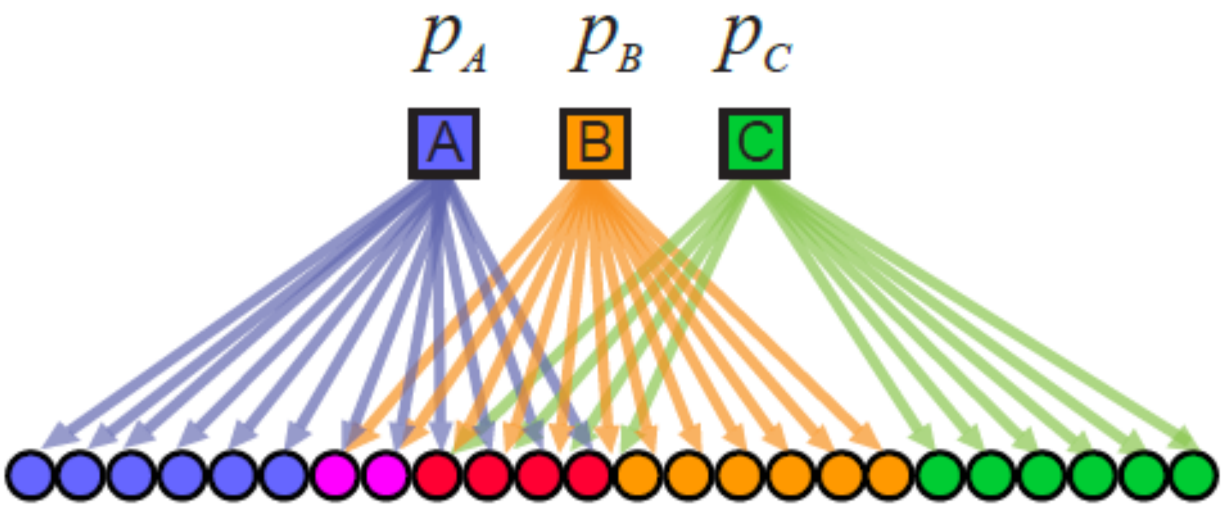
AGM: Generative Process



- **Given parameters ($V, C, M, \{p_c\}$)**
  - Nodes in community c connect to each other by flipping a coin with probability p_c
  - **Nodes that belong to multiple communities have multiple coin flips: Dense community overlaps**
    - If they "miss" the first time, they get another chance through the next community"

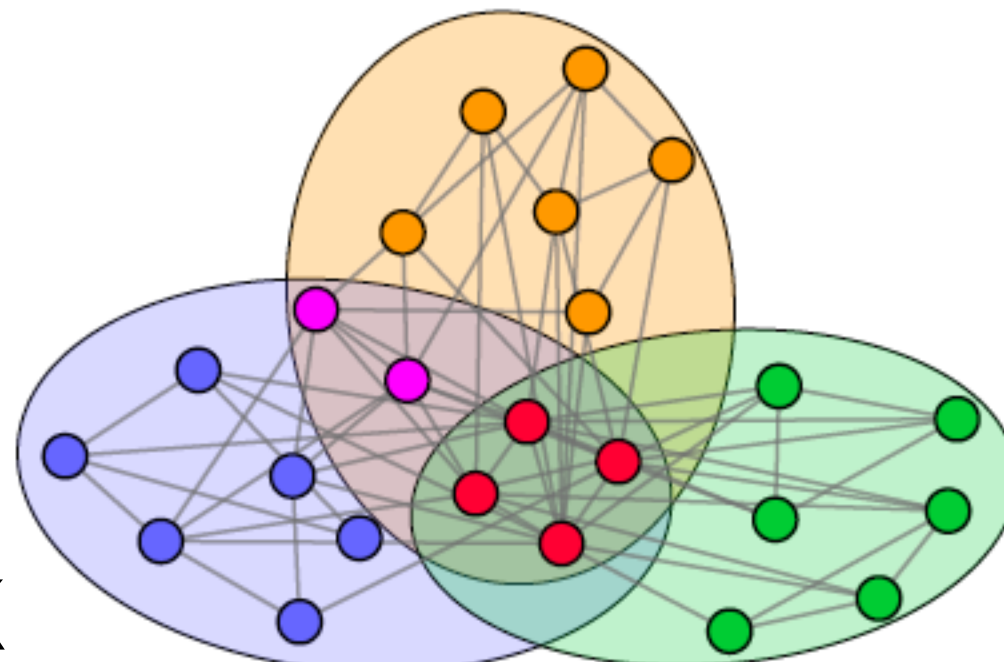
$$p(u,v) = 1 - \prod_{c \in M_u \cap M_v} (1 - p_c)$$

AGM: Dense Overlaps



Model

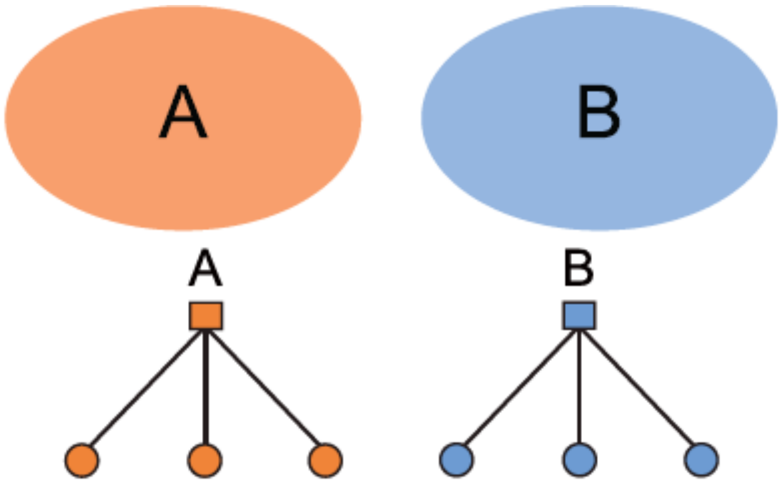
Network

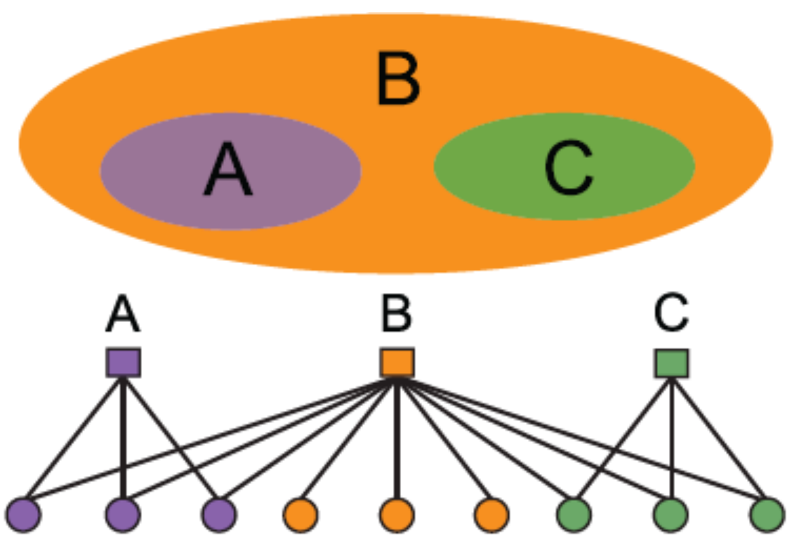


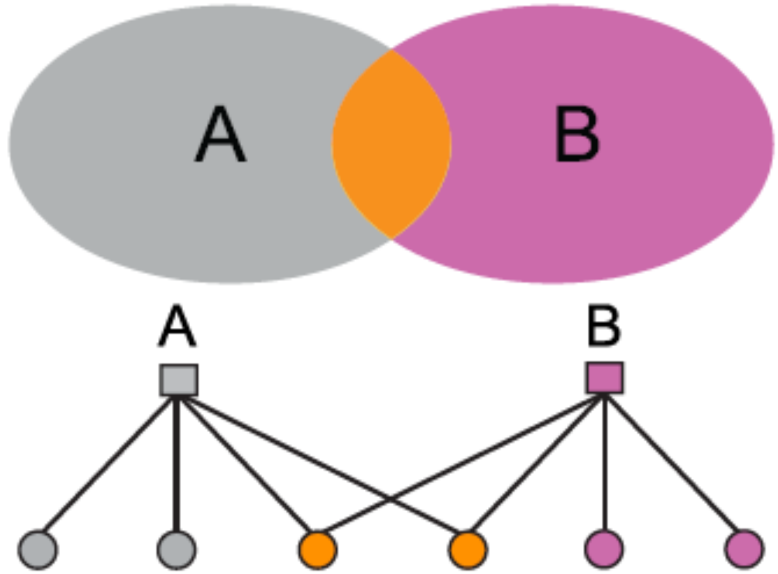
[icdm ‘12]

AGM: Flexibility

- **AGM can express a variety of community structures:** Non-overlapping, Overlapping, Nested

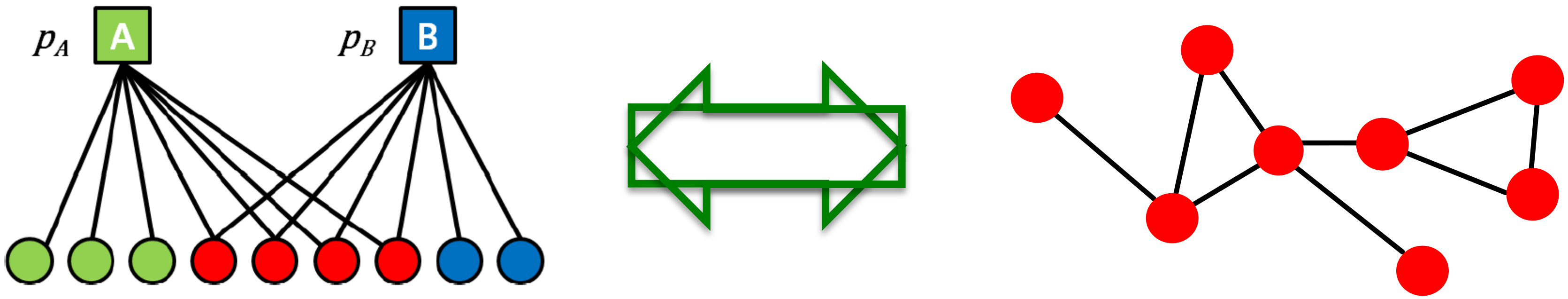






Detecting Communities

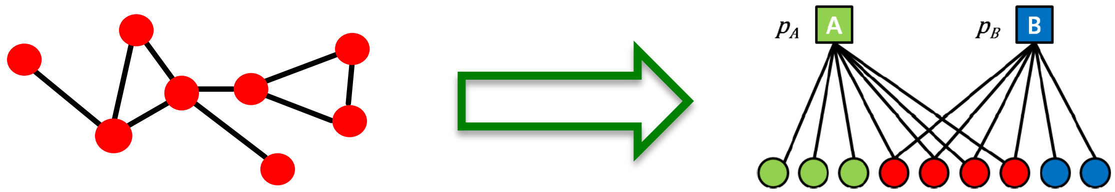
- **Detecting communities with AGM:**



Given a Graph, find the Model

1) Affiliation graph ***M***
2) Number of communities **C**
3) Parameters $\boldsymbol{p_c}$

MAG Model Fitting

- Task: 

  - Given network *G(V,E)*. Find *B(V, C, M, {p_c})*
- Optimization problem (MLE)

$$\arg\max_B P(G \mid B) = \prod_{(i,j)\in E} P(i,j) \prod_{(i,j)\notin E} (1 - P(i,j))$$

$$P(i,j) = 1 - \prod_{c \in M_i \cap M_j} (1 - p_c)$$

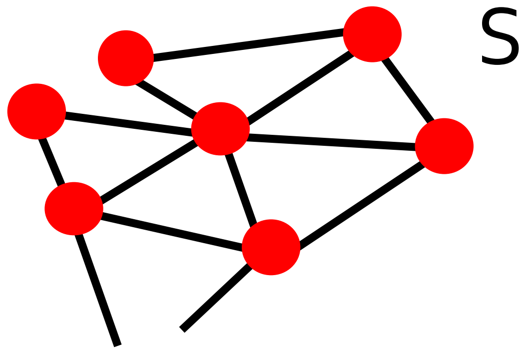
- How to solve?
  - Approach: Coordinate ascent
    - **(1)** Stochastic search over *B*, while keeping *{p_c}* fixed
    - **(2)** Optimize *{p_c}*, while keeping *B* fixed (**convex**!)
  - Works relatively well in practice!

Communities:
Issues and Questions

Communities: Issues and Questions

- **Some issues with community detection:**
  - Many different formalizations of clustering objective functions
  - Objectives are NP-hard to optimize exactly
  - Methods can find clusters that are systematically "biased"
- **Questions:**
  - **How well do algorithms optimize objectives?**
  - **What clusters do different methods find?**

Many Different Objective Functions



n: nodes in S
m: edges in S
c: edges pointing outside S

- **Single-criterion:**
  - Modularity: $m\text{-}E(m)$
  - Edges cut: c
- **Multi-criterion:**
  - Conductance: $c/(2m+c)$
  - Expansion: c/n
  - Density: $1\text{-}m/n^2$
  - CutRatio: $c/n(N\text{-}n)$
  - Normalized Cut: $c/(2m+c) + c/2(M\text{-}m)+c$
  - Flake-ODF: *frac. of nodes with more than ½ edges pointing outside S*

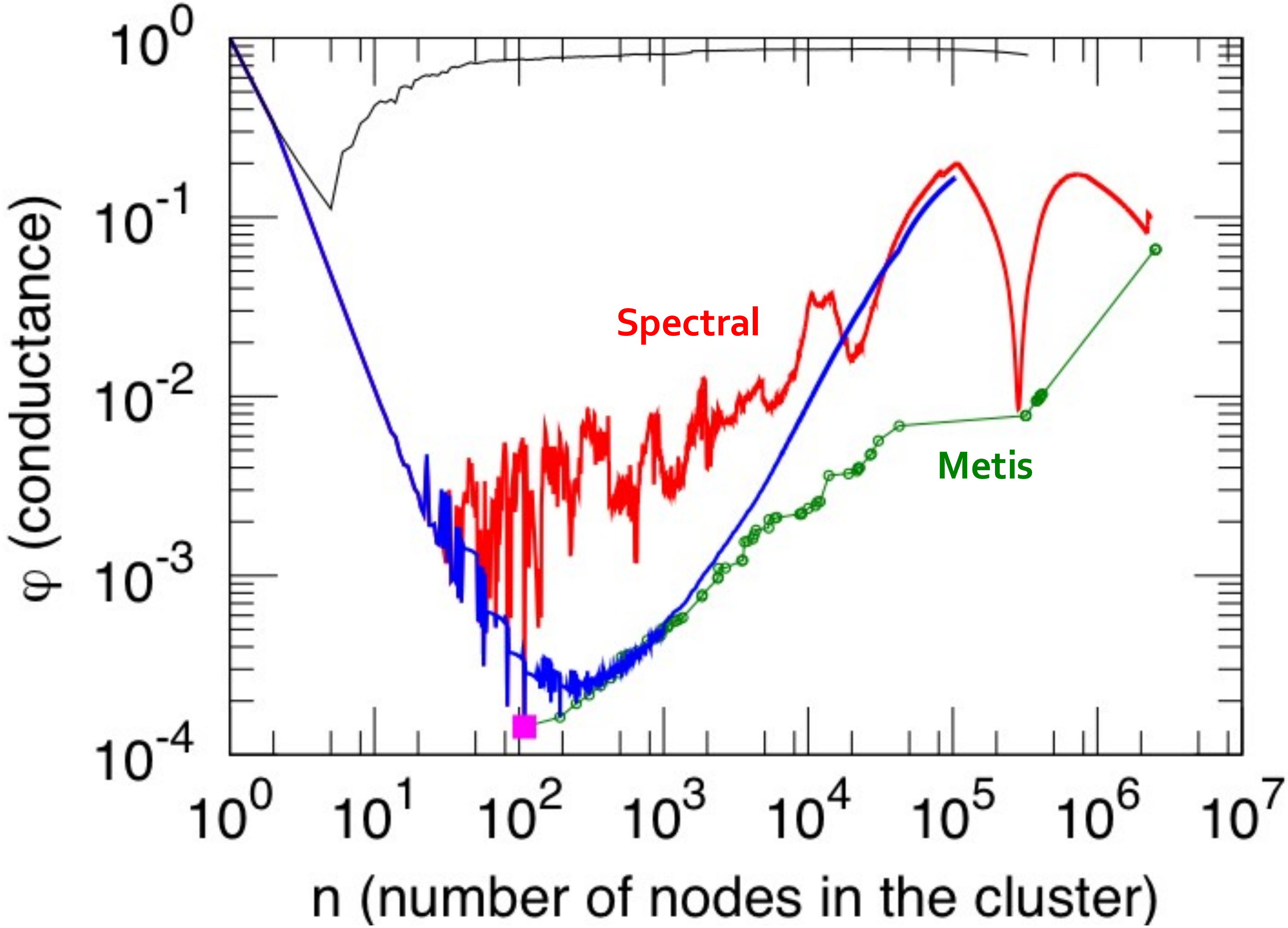
[WWW '09]

Many Classes of Algorithms

Many algorithms to that implicitly or explicitly optimize objectives and extract communities:

- **Heuristics:**
  - Girvan-Newman, Modularity optimization: popular heuristics
  - Metis: multi-resolution heuristic [Karypis-Kumar '98]
- **Theoretical approximation algorithms:**
  - Spectral partitioning

NCP: Live Journal

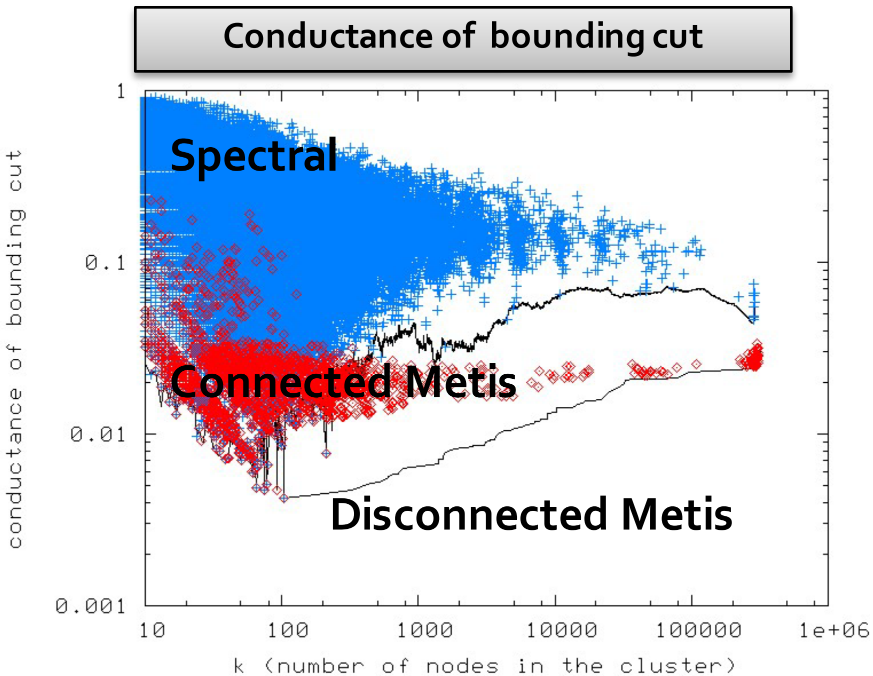


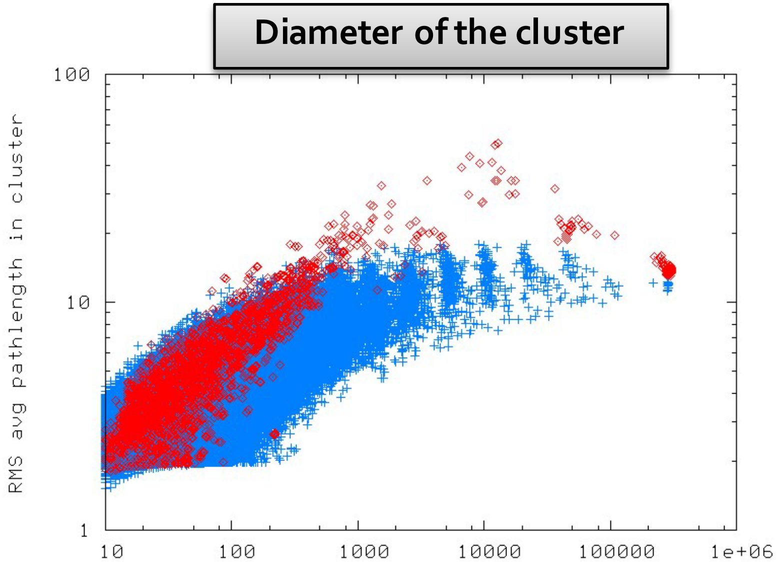
Properties of Clusters (1)

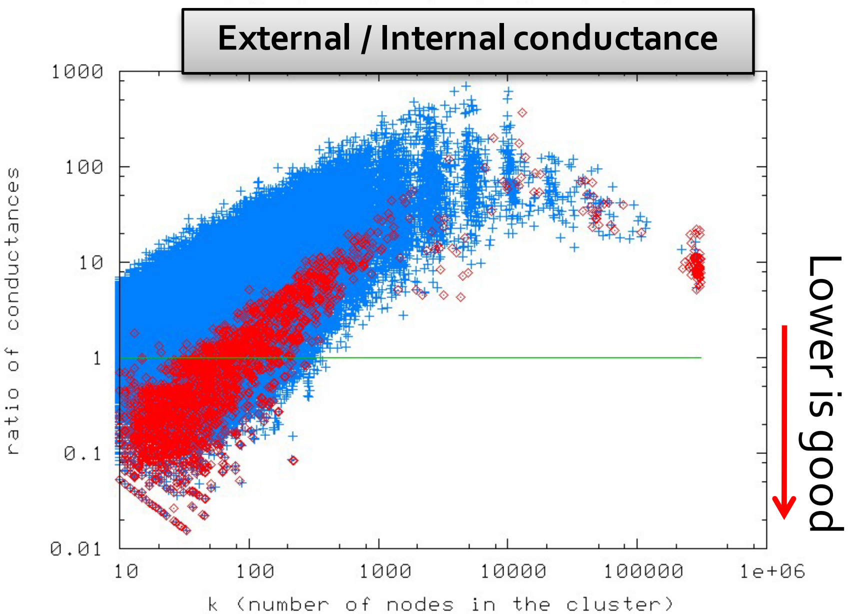
500 node communities from Spectral:

500 node communities from Metis:

Properties of Clusters (2)

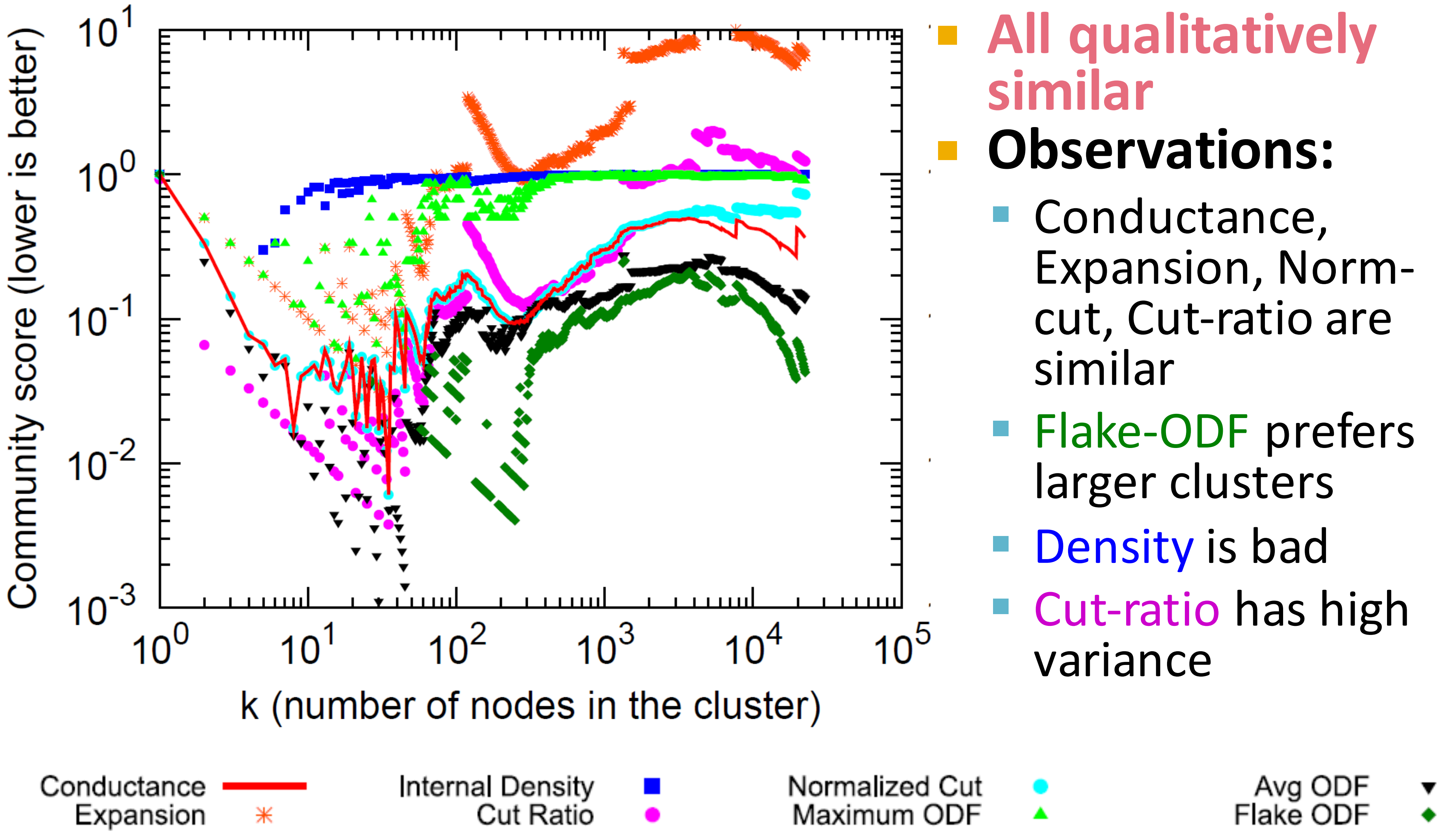






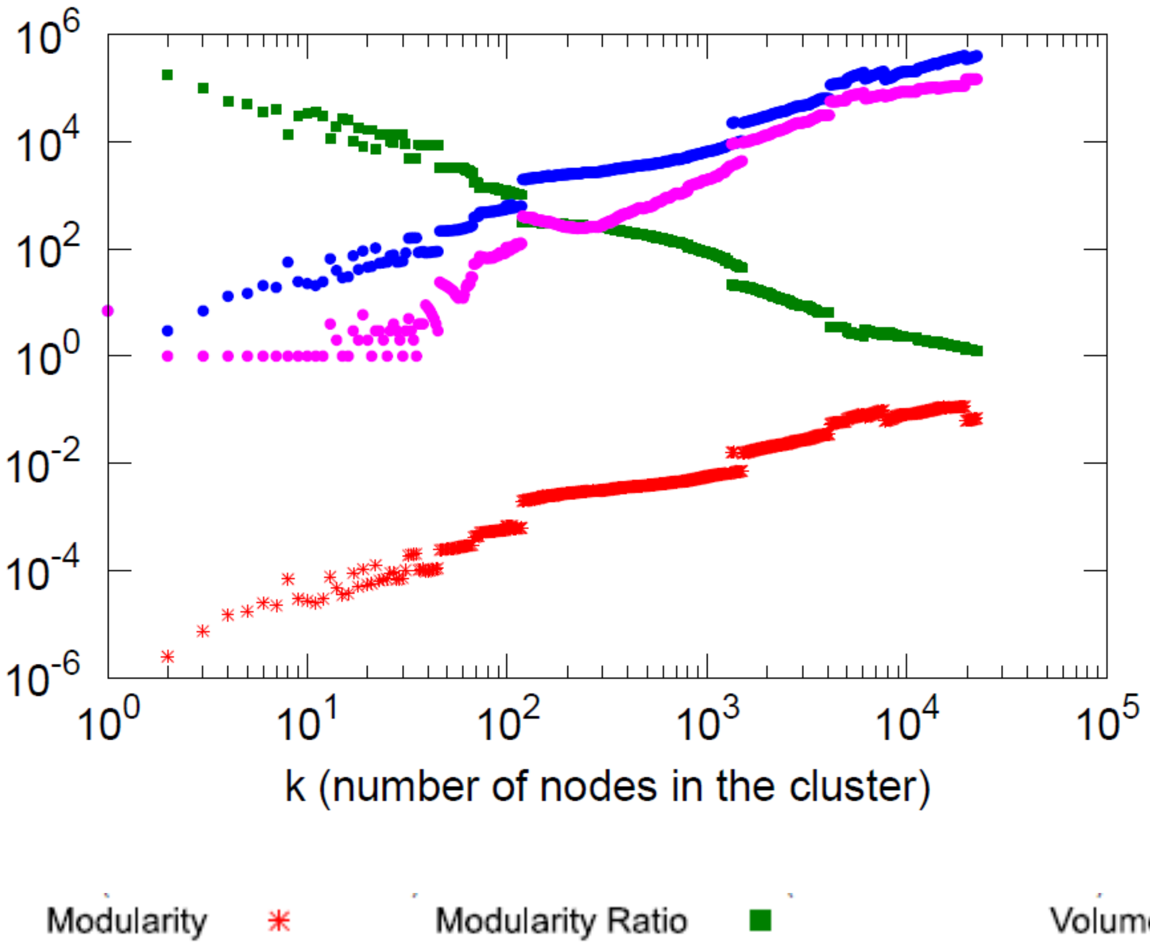
- Metis gives sets with better conductance
- Spectral gives tighter and more well-rounded sets

Multi-criterion Objectives



- **All qualitatively similar**
- **Observations:**
  - Conductance, Expansion, Norm-cut, Cut-ratio are similar
  - Flake-ODF prefers larger clusters
  - Density is bad
  - Cut-ratio has high variance

Single-criterion Objectives



Observations:

- All measures are monotonic
- **Modularity**
  - prefers large clusters
  - Ignores small clusters