

Community Structure



Network Theory and Applications

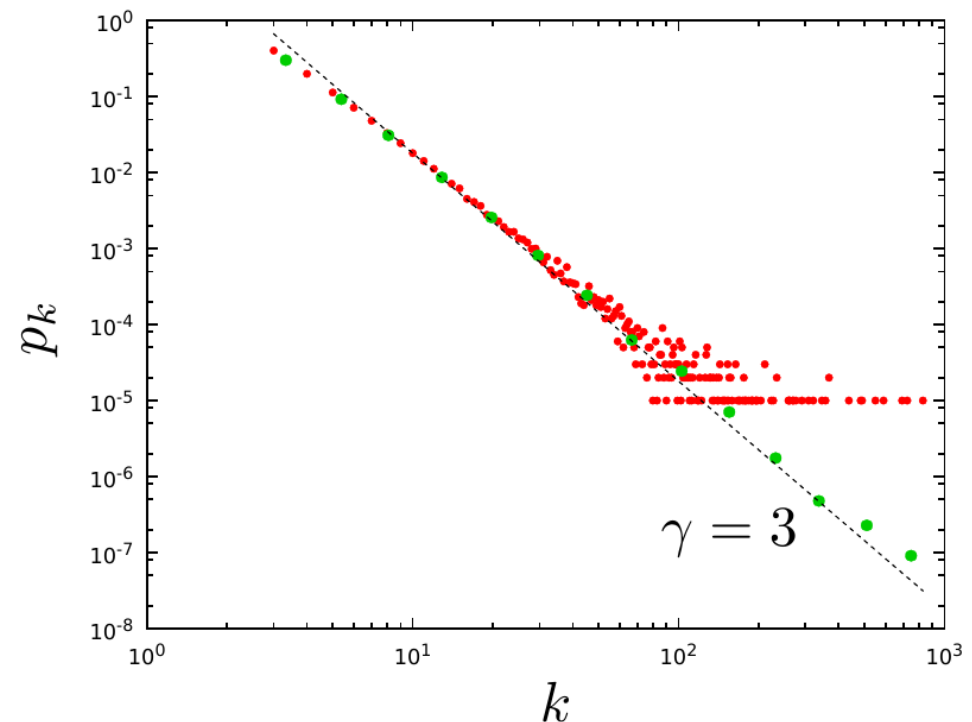
ECS 253 / MAE 253

Spring 2025

Marton Posfai

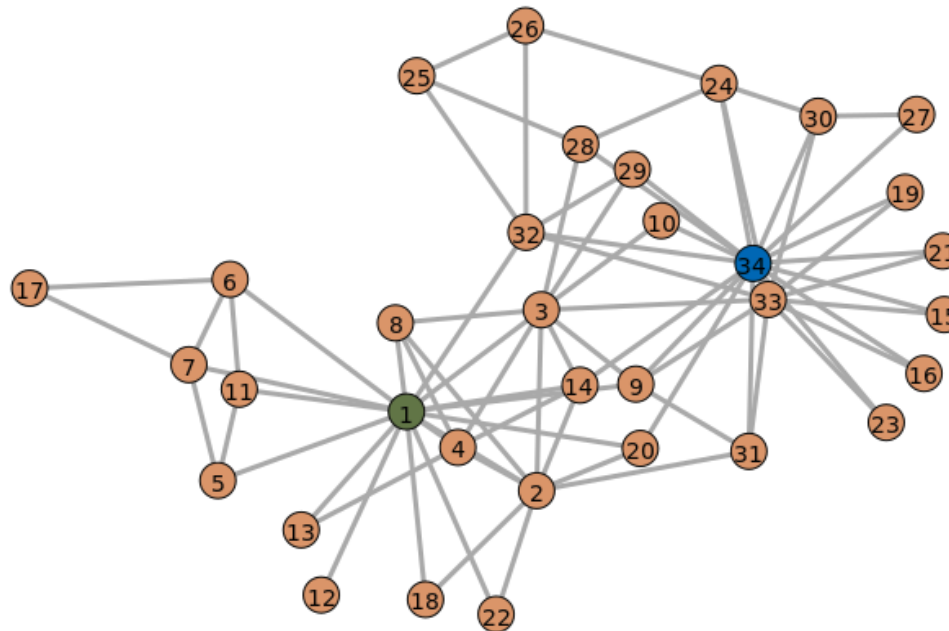
Communities

- So far:
 - Properties of nodes: degree, centralities, triangles
 - Statistics of local properties: degree distribution
- How about large-scale organization?
 - Core-periphery, hierarchy,...
 - Communities



Introduction – Zachary's karate club

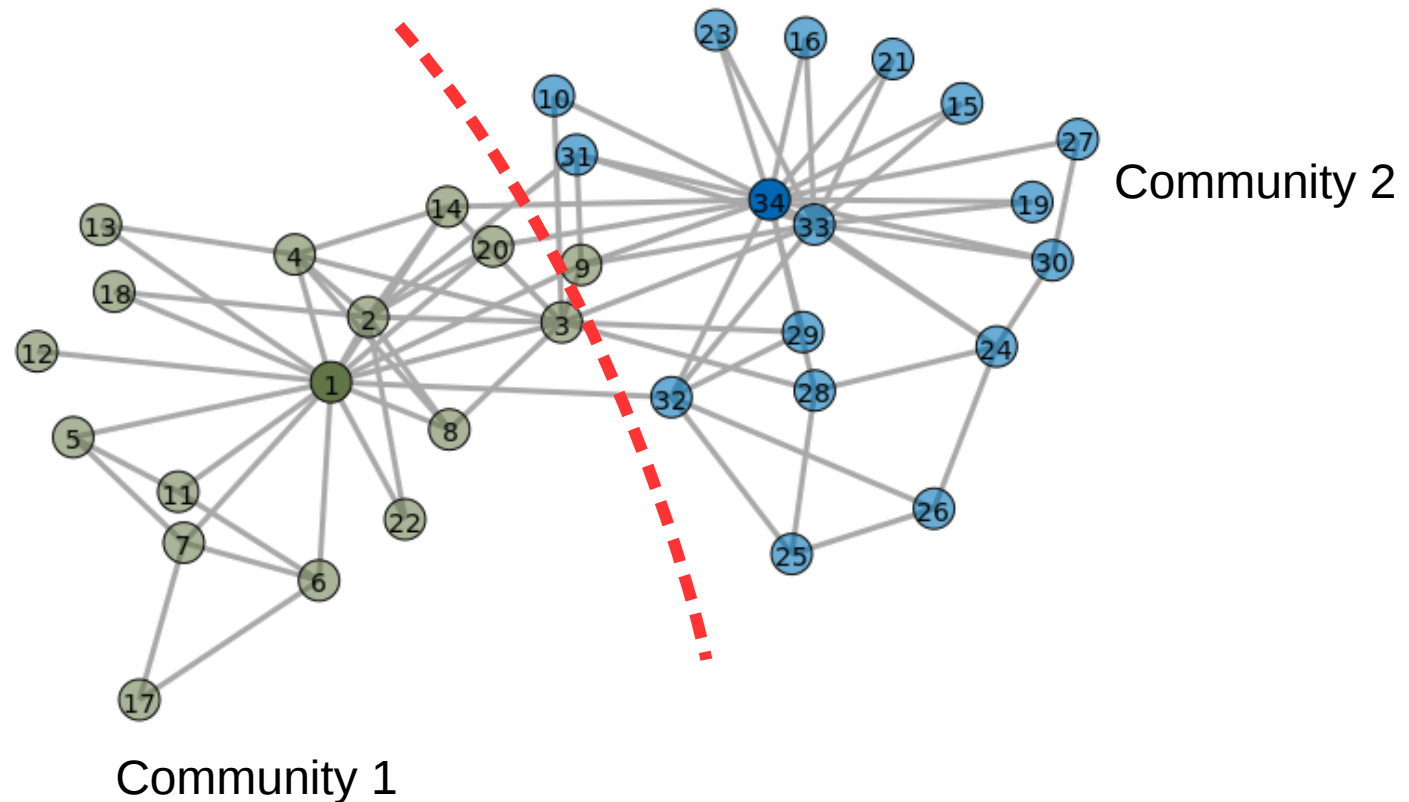
- Social scientist Wayne W. Zachary collected friendship data at a karate club from 1970 to 72
- Edges represent friendship (activities outside the club)



- Conflict between instructor (node 1) and administrator (node 34) group broke up into two
- Q: Can we predict the groups based on network structure?

Introduction – Zachary's karate club

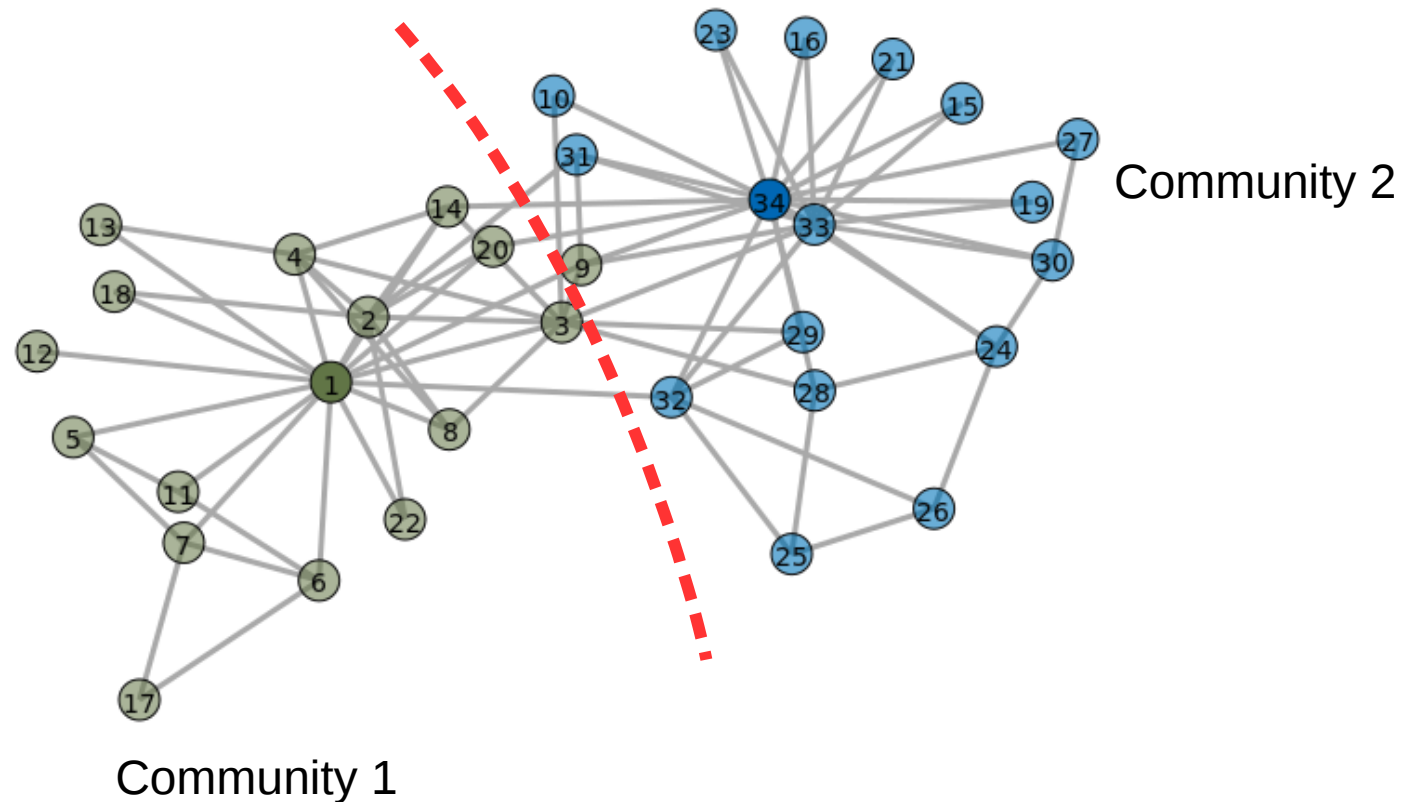
- Zachary's solution: network flow, source=node 1, sink=node 34
- Cut: flow bottleneck



- All but node 9 correct

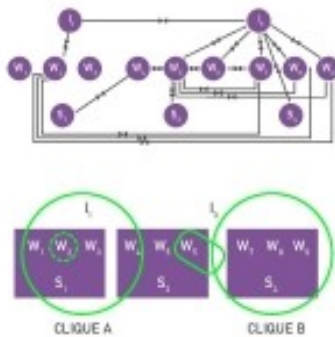
Introduction – Zachary's karate club

- Zachary's solution: network flow, source=node 1, sink=node 34
- Cut: flow bottleneck



- All but node 9 correct

History



Early Communities

George Homans recorded the communication of bank tellers [top], identifying their communities (bottom) [3].



Graph Partitioning

Predecessors to community finding, graph partitioning algorithms optimize the layout of integrated circuits



Michelle Girvan and Mark Newman

propose the hierarchical divisive algorithm, igniting an explosive interest in community identification [9]. They also introduce modularity in 2004 [23].



Erzsébet Ravasz

proposes a hierarchical agglomerative algorithm, unleashing an explosion of research within systems biology [11].



The sociologist **Stuart Rice** uses voting patterns to identify communities in political bodies [4].

Duncan R Luce and **Albert D Perry** define communities as cliques [5].

Robert Weiss and **Eugene Jacobson** identify communities by removing individuals linked to multiple groups [6].

Brian Wilson Kerningham and **Shen Lin** develop a graph partitioning algorithm [18], widely used in chip design (BOX 9.1)

Mark Granovetter explores the interplay between communities and weak ties [62].



Wayne W. Zachary maps out the karate club, that a quarter of a century later becomes a test bed for community identification [7].

Gary Flake, Steve Lawrence and **Lee Giles** define a WWW community as documents that have more links to each other than to documents outside their community [15].

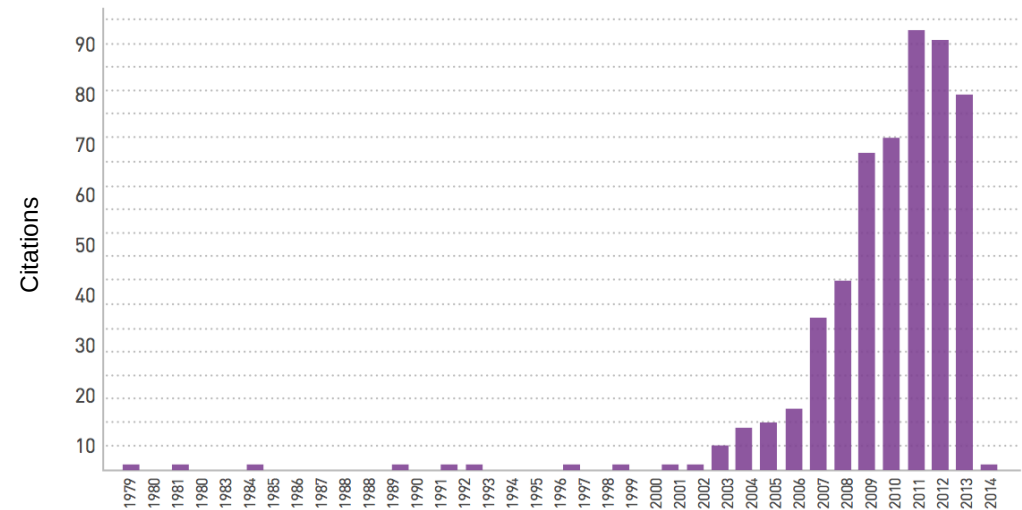
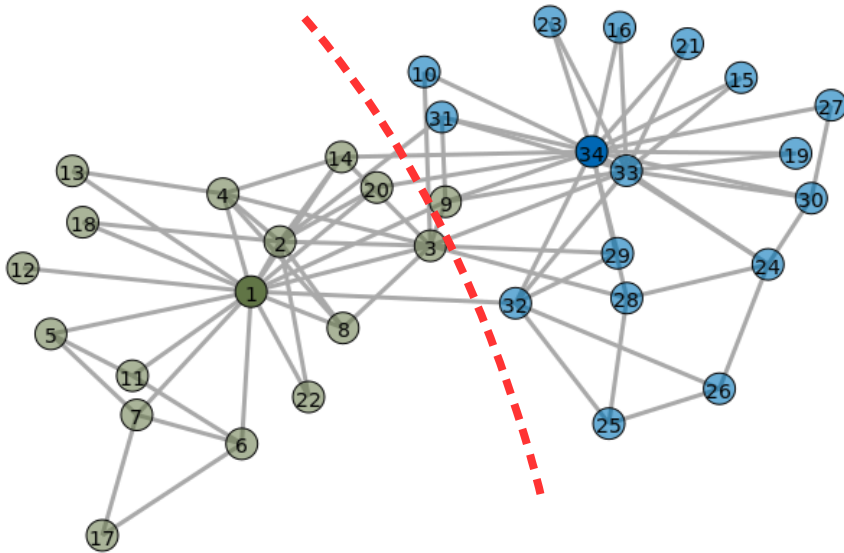


Tamás Vicsek

introduces the CFinder algorithm to identify overlapping communities [36].

Introduction – Zachary's karate club

- Communities = locally dense subgraphs
- Modern network community detection from 2000s



Zachary Karate Club Club

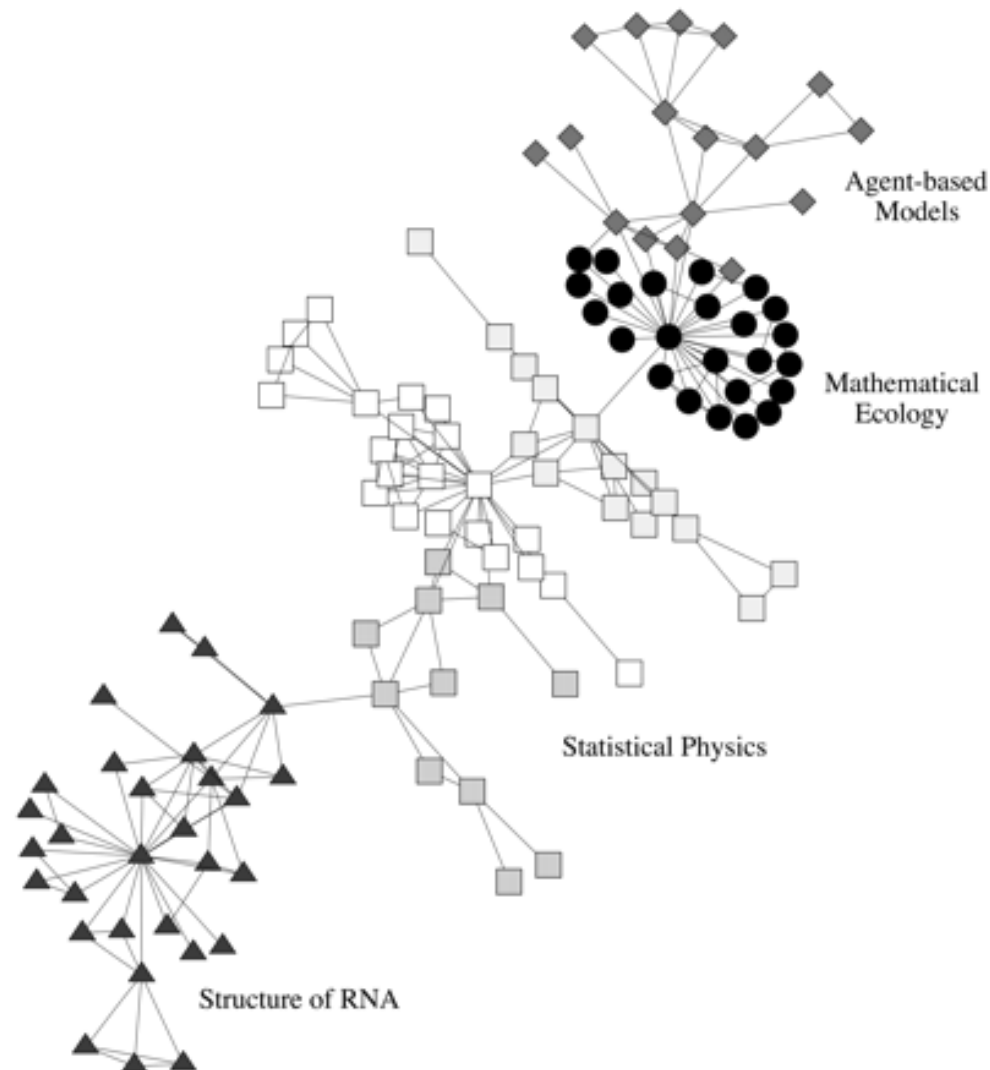
First presenter to mention the ZKC at a conference gets the trophy.

<http://networkkarate.tumblr.com/>



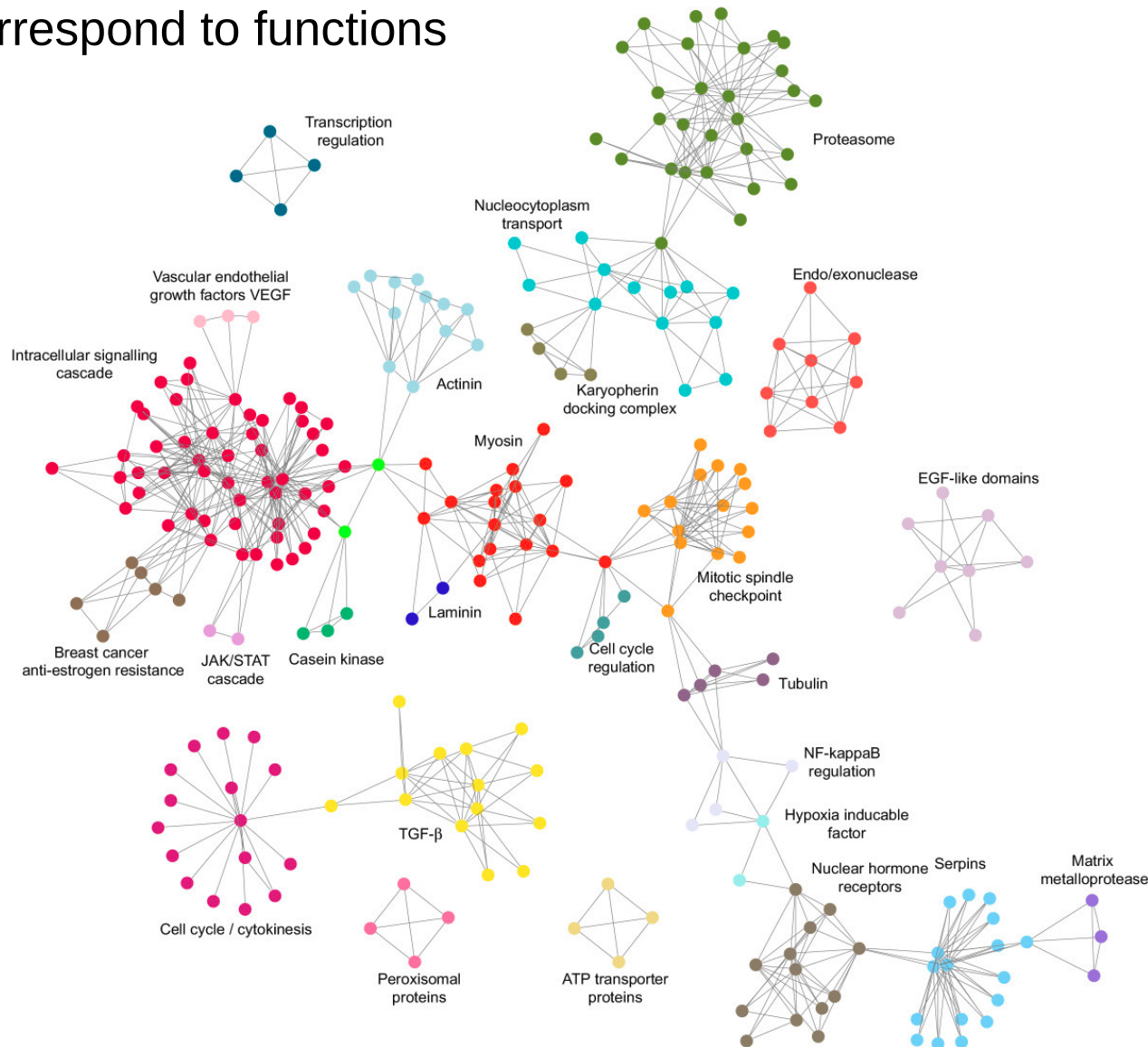
Scientific collaboration at SFI

- Link (A – B) : A and B coauthored a paper
- Node classification by role according to position in community



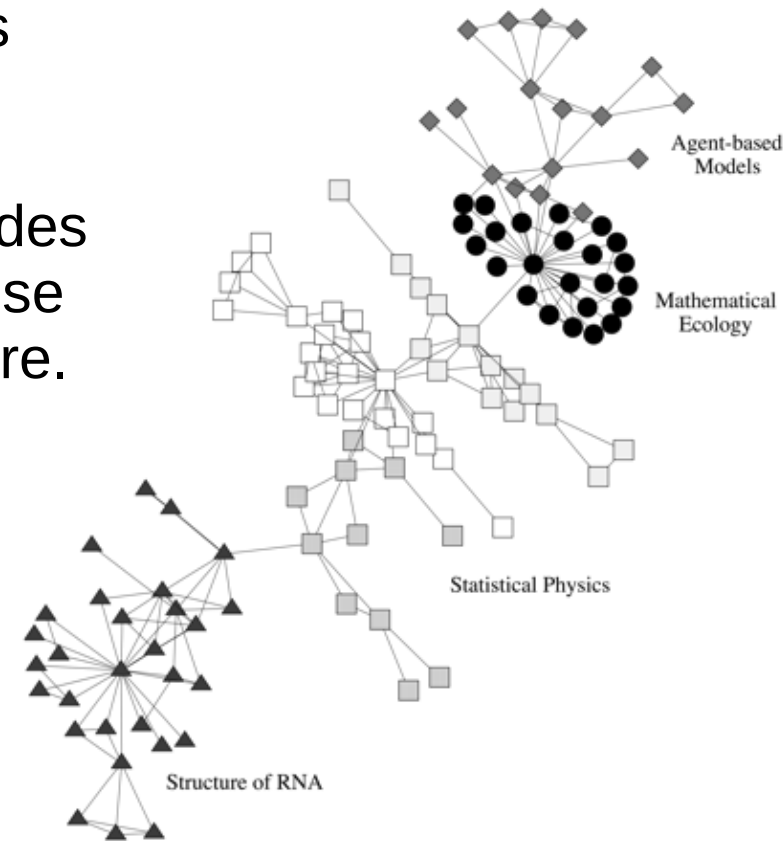
Rat protein-protein interaction network

- Link (A – B) : A and B proteins physically interact
- Modules correspond to functions



Basics

- What is a community?
 - Intuitively: densely connected subnetworks
- Why is it interesting?
 - Nodes that participate in same function/nodes with similar attributes form groups and these groups are represented in network structure.
- Challenges:
 - No single clear definition.
 - Many competing options.
 - Large networks, different features.
 - Even more algorithms.
 - Which one to choose?
 - How to evaluate performance of method?



Outline

- How to identify communities?
- How to assess the quality of a community division?
- How assess the quality of a method?

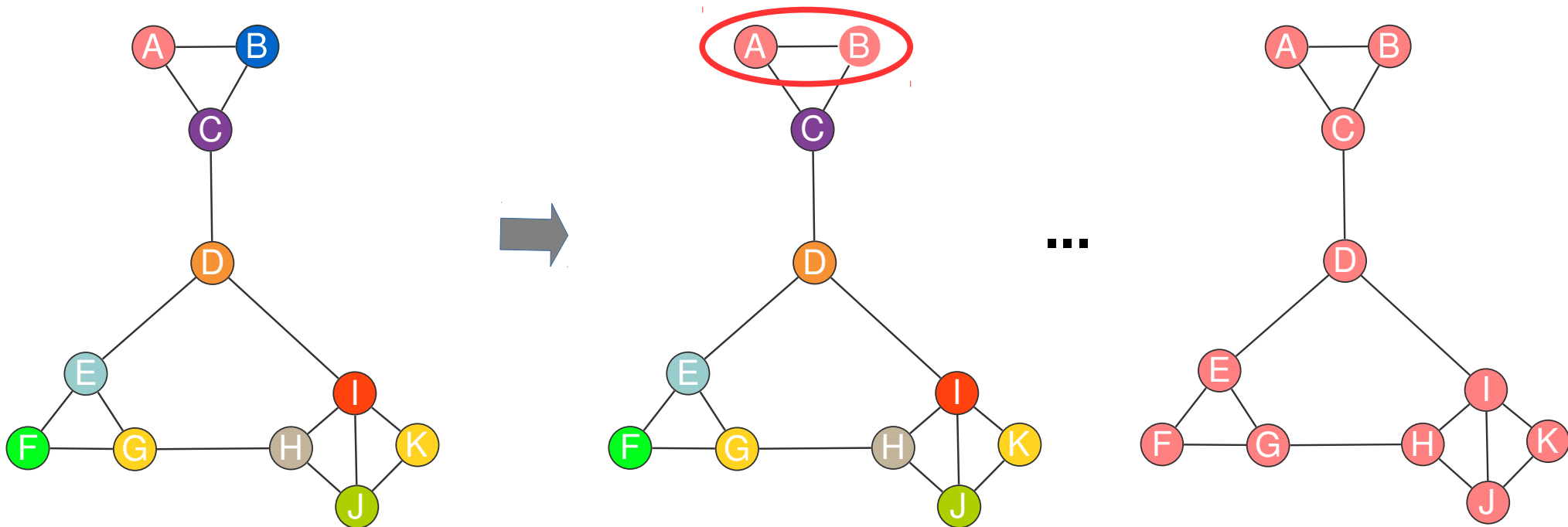
- A lot of competing methods and measures
 - Here: selection that shows the development of the field
- And some guidelines to navigate the field

Method 1: Hierarchical Clustering



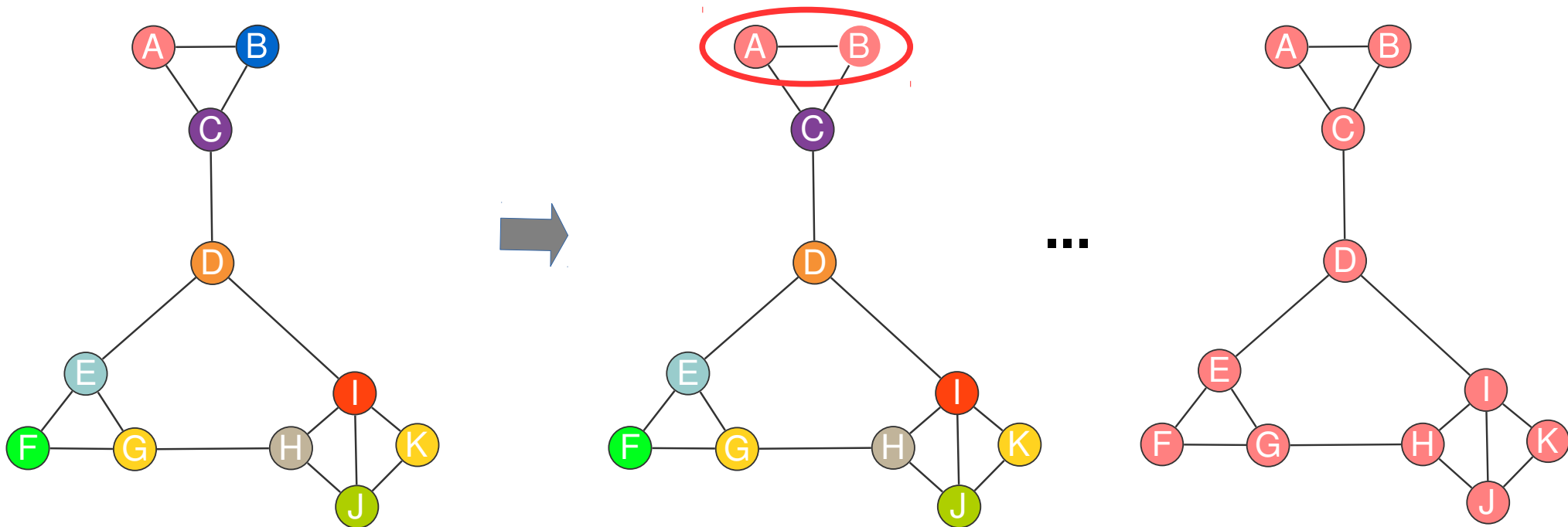
Hierarchical clustering

- Traditional method used by social scientists.
 - 0) Define a distance metric σ_{ij} between nodes based on network
- 1) Each node in its own community.
 - 2) Calculate a distance between pairs of communities according to some rule.
 - 3) Join closest pair.
 - 4) Go to step 2.



Hierarchical clustering

- Traditional method used by social scientists.
- 0) Define a distance metric σ_{ij} between nodes based on network
- 1) Each node in its own community.
- 2) Calculate a distance between pairs of communities according to some rule.
- 3) Join closest pair.
- 4) Go to step 2.

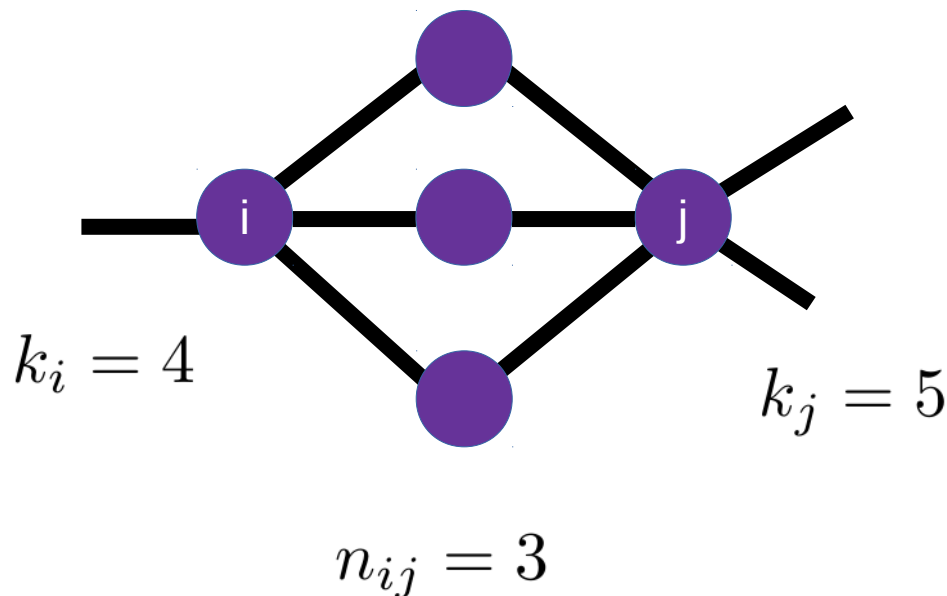


Hierarchical clustering: distance

- Node distance should be low if nodes are in a community.
- Popular choice:

$$\sigma_{ij} = \frac{n_{ij}}{\sqrt{k_i k_j}}$$

#common neighbors
degree

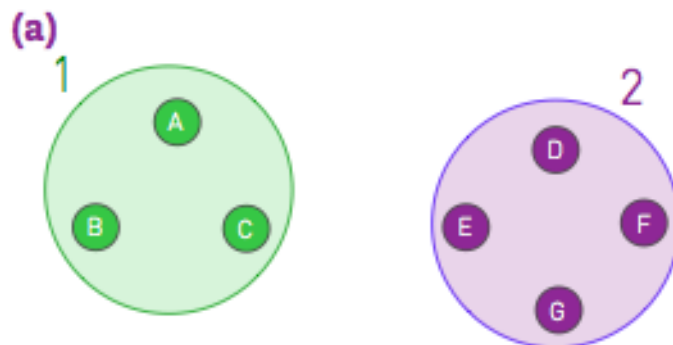


$$\sigma_{ij} = \frac{3}{\sqrt{4 \cdot 5}} \approx 0.67$$

- Other distances possible, e.g. number of independent paths connecting i and j

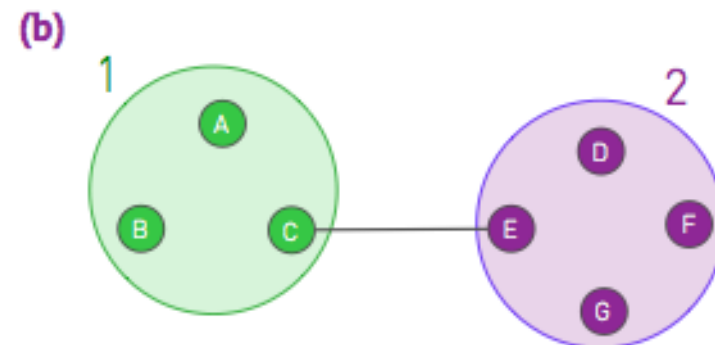
Hierarchical clustering: distance

- Distance between communities with more than one node:

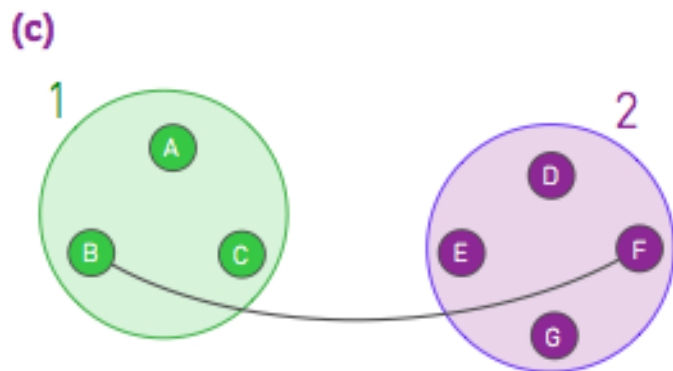


$x_{ij} = r_{ij} =$

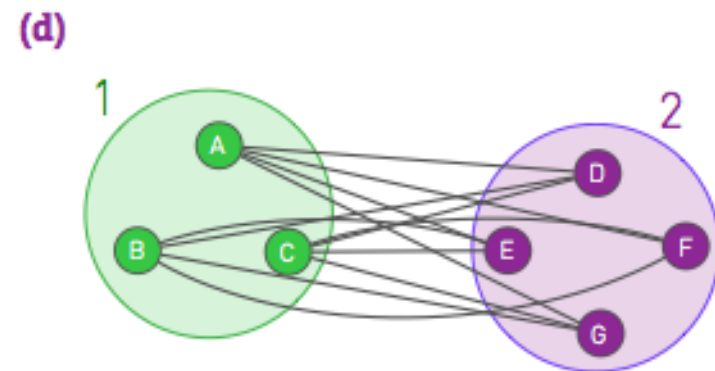
	D	E	F	G
A	2.75	2.22	3.46	3.08
B	3.38	2.68	3.97	3.40
C	2.31	1.59	2.88	2.34



Single Linkage: $x_{12} = 1.59$



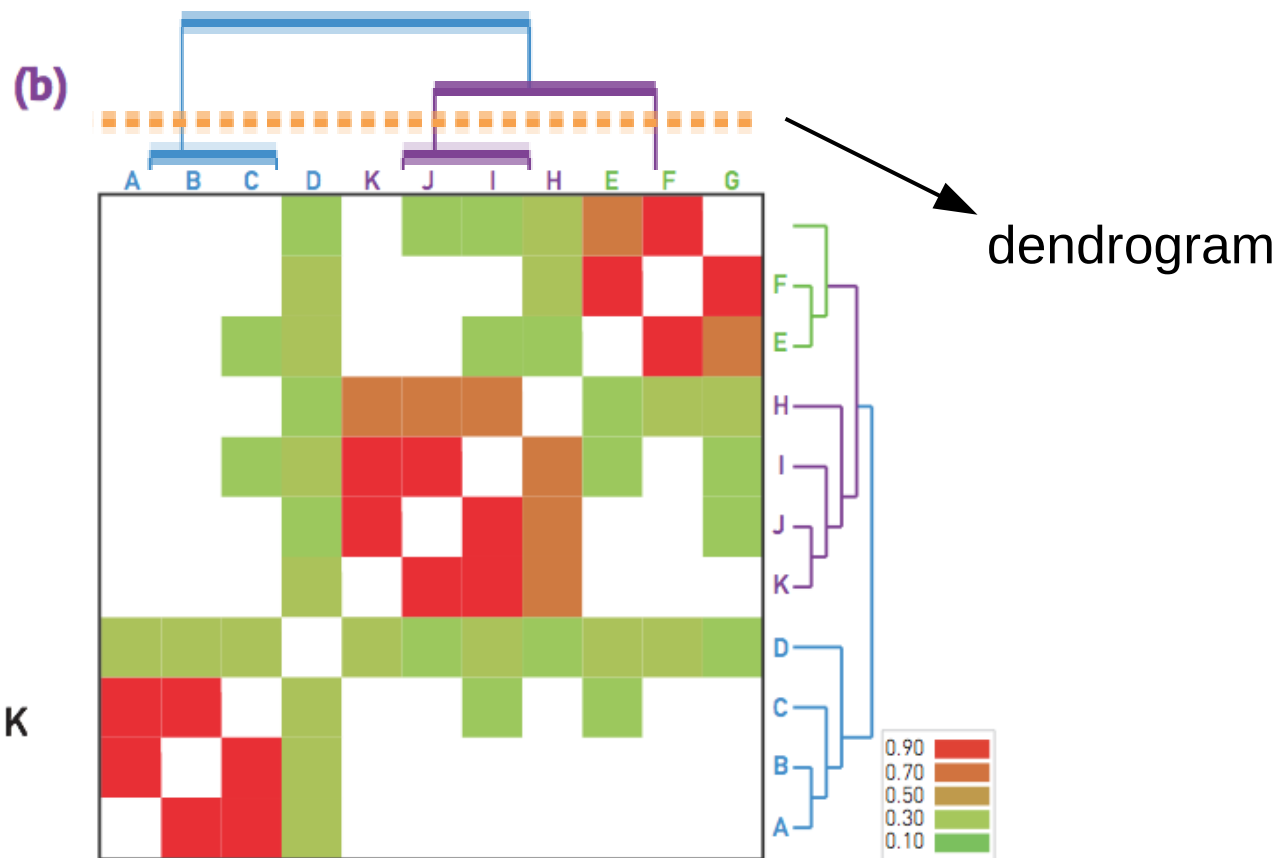
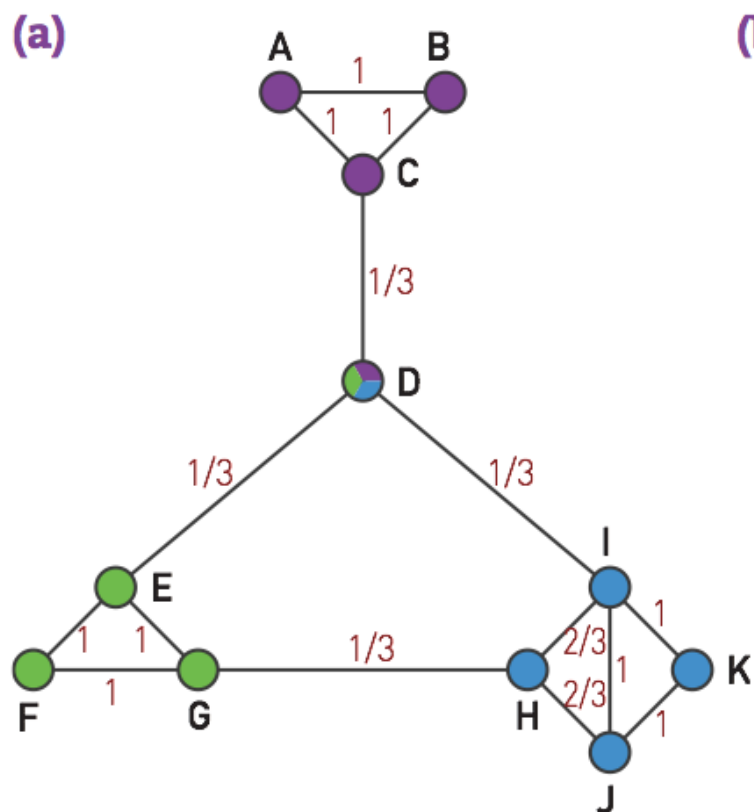
Complete Linkage: $x_{12} = 3.97$



Average Linkage: $x_{12} = 2.84$

Hierarchical clustering: result

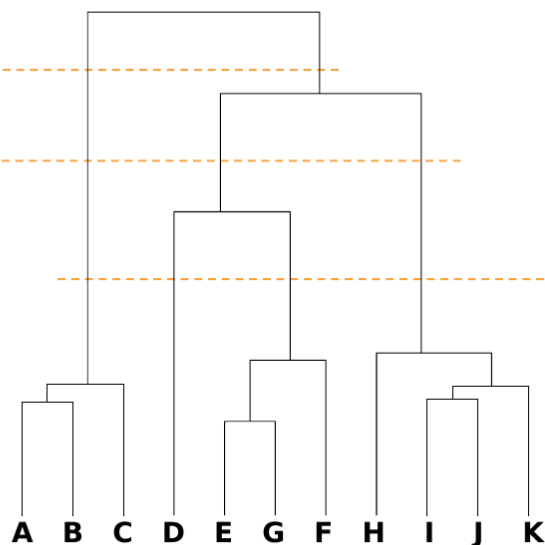
- Example artificial network
(slightly different definition of distance)



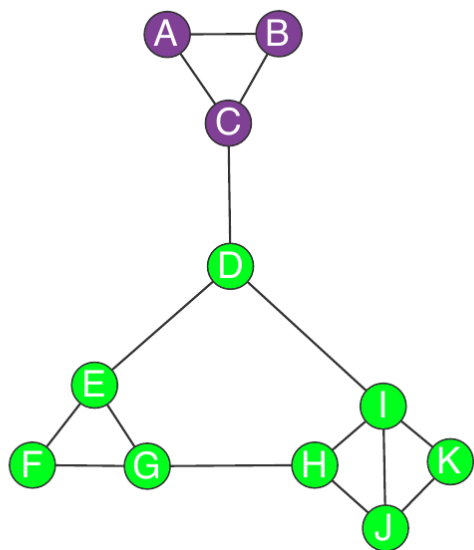
Hierarchical clustering: result

- Dendrogram

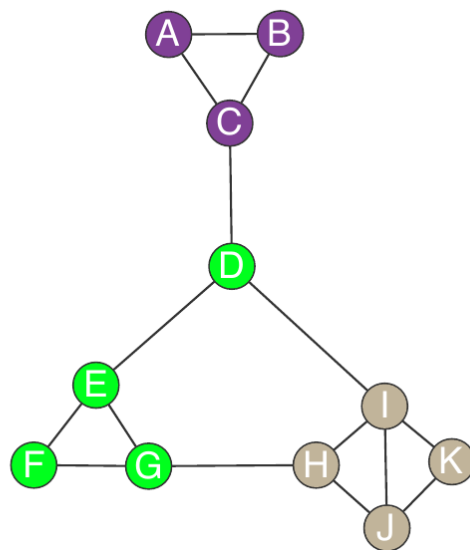
(a)



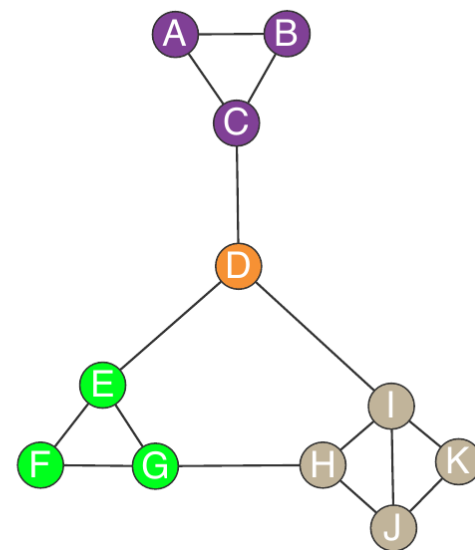
(b)



(c)

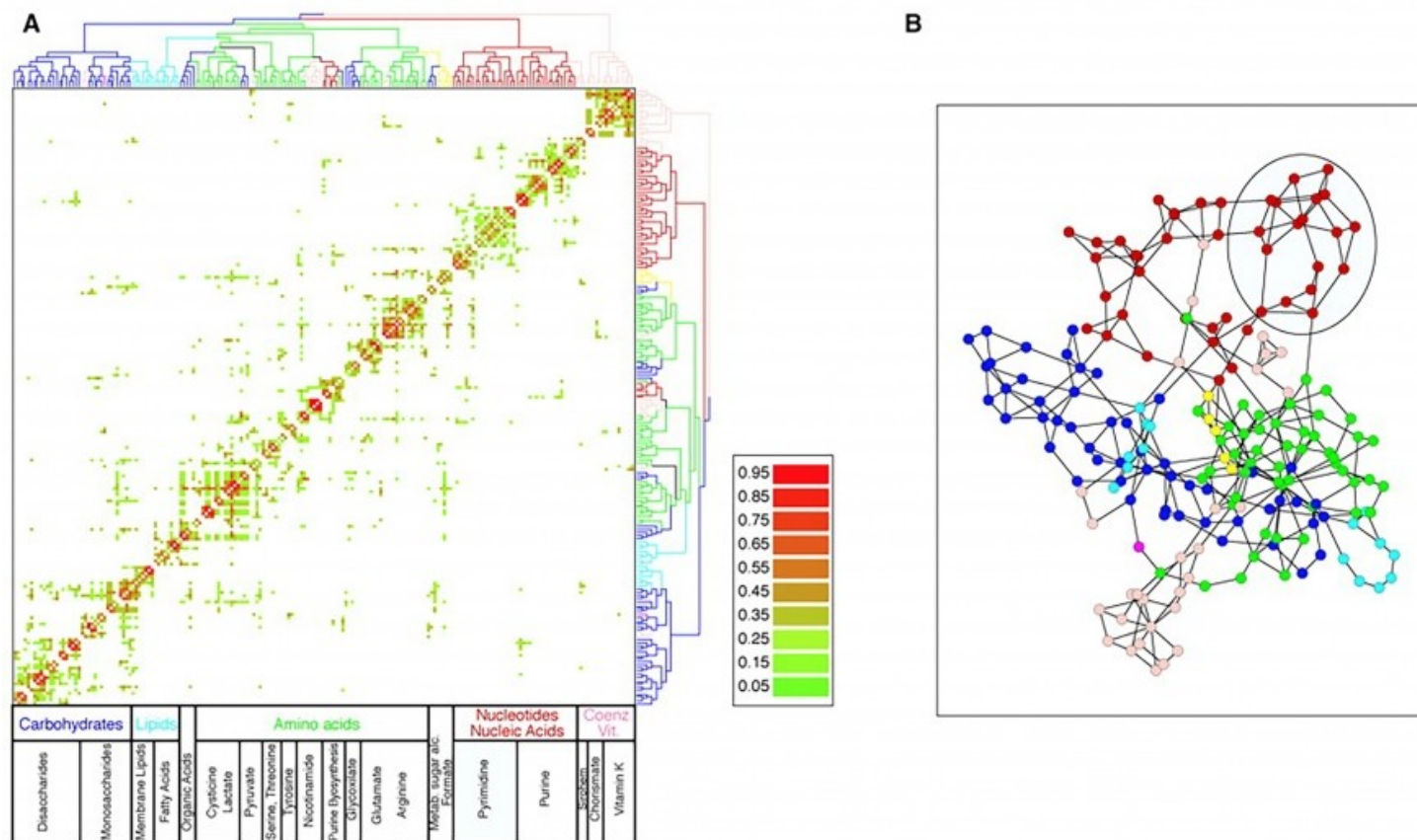


(d)



Hierarchical clustering: application

- E. coli metabolism



- The color of each node, capturing the predominant biochemical class to which it belongs, indicates that different functional classes are segregated in distinct network neighborhoods.
- The highlighted region selects the nodes that belong to the pyrimidine metabolism, one of the predicted communities.

Hierarchical clustering: issues

- Advantages:
 - Easy to understand
 - Easy to implement
- Disadvantages:
 - Slow(ish), number of steps to evaluate: $\sim N^2$ or $\sim N^3$, depending on linkage
 - “Tends to group together those nodes with the strongest connections but leave out those with weaker connections → divisions consist of a few dense cores surrounded by a periphery of unattached nodes”
 - (Results depend on distance and linkage)
- Open question: where to cut the dendrogram?

Method 2: Betweenness based division



Betweenness based division

- Alternative method: instead of agglomerating communities, breaking one large into smaller ones

- 1) Start from one large community
- 2) Calculate a centrality measure for each link
- 3) Remove link with highest centrality
- 4) Go to step 2

- Centralities: **betweenness** for edge i

$$B(i) = \sum_{s \neq t} \frac{\sigma_{st}(i)}{\sigma_{st}}$$

#shortest path between
s and t traversing i

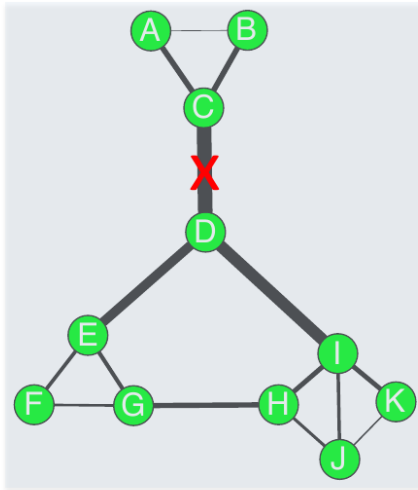
#shortest path between
s and t

- Other: random walk, ...

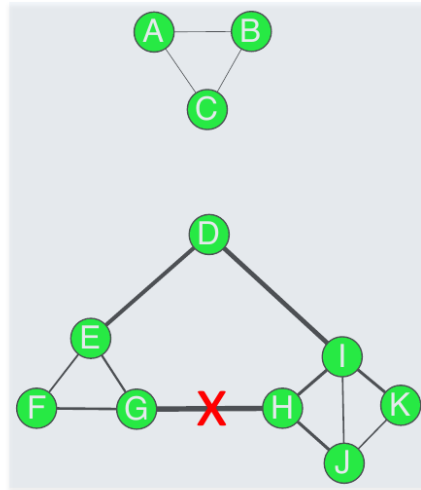
Betweenness based division

- Our little example:

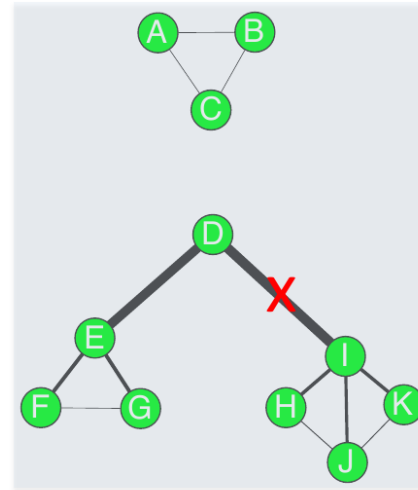
(a)



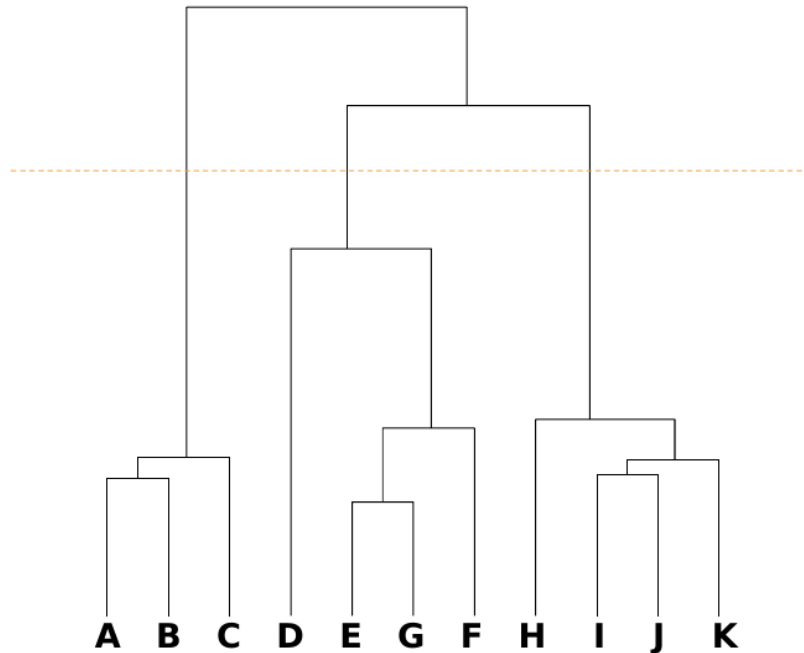
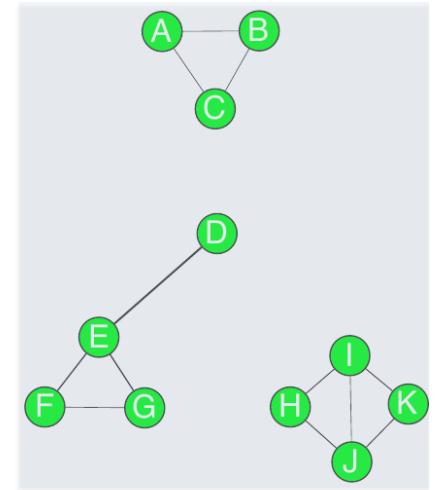
(b)



(c)

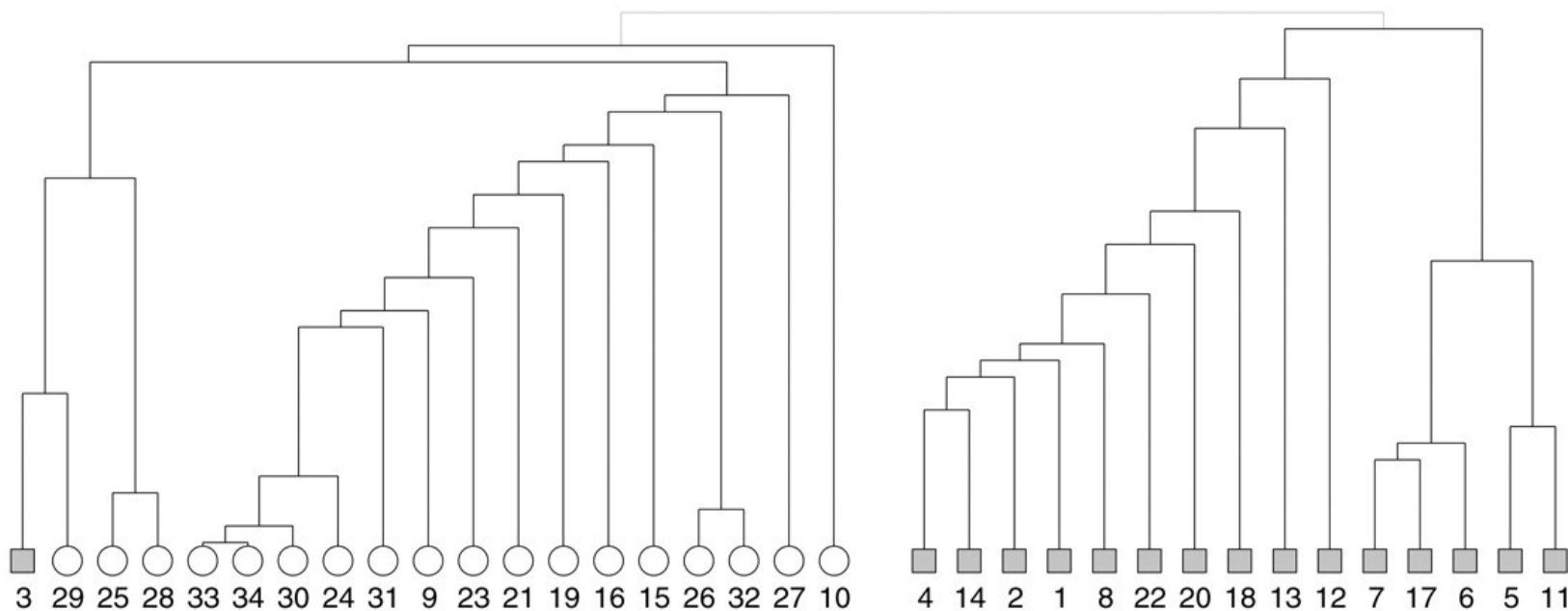
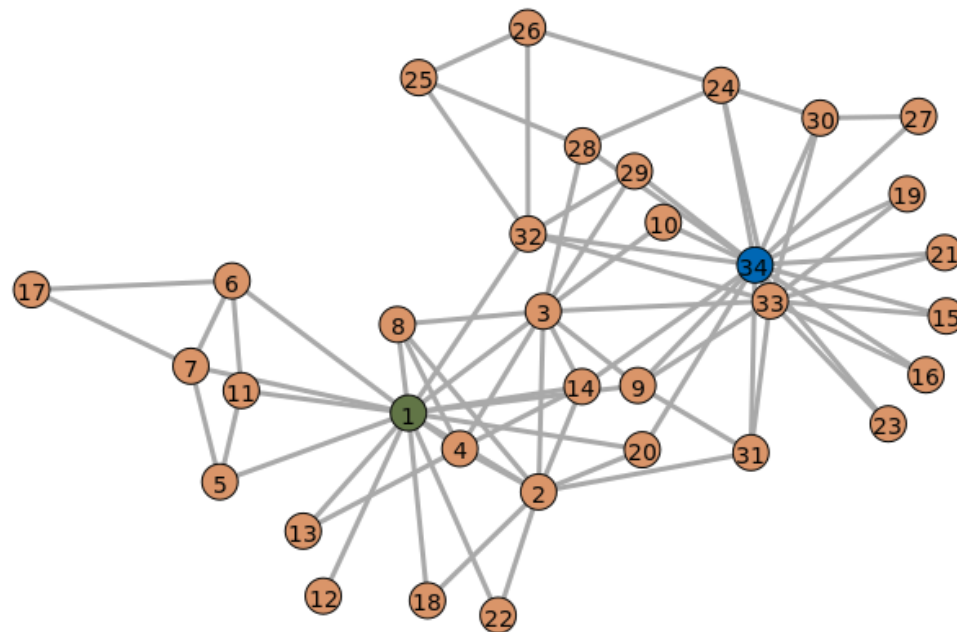


(d)



Betweenness based division: application

- Zachary's Karate Club



Betweenness based division

- Advantages:
 - Easy to understand
 - Easy to implement
 - Perhaps less decisions have to be made
- Disadvantages:
 - Slow, number of steps to evaluate: $\sim NL^2$ or faster if we don't recalculate the betweenness in each step
 - (Results depend on centrality)
- **Still open question: where to cut the dendrogram?**

Quality of community division



Modularity

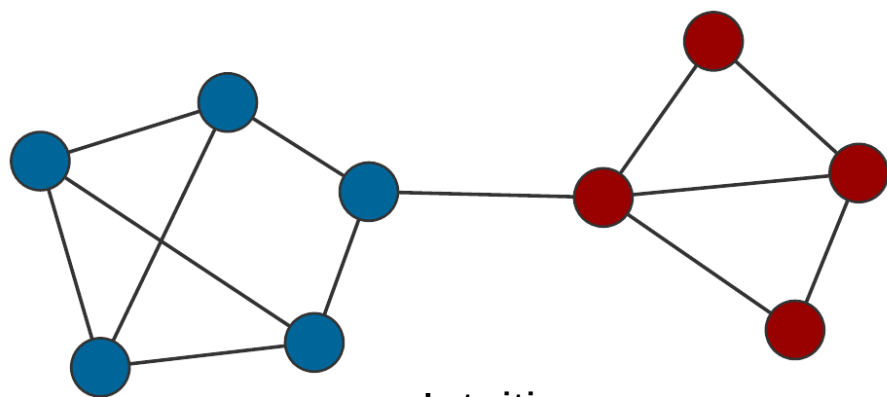
- Again: many existing measures.
- Naïve: fraction of links that are inside communities

$$Q = \frac{1}{2m} \sum_{ij} A_{ij} \delta_{s_i, s_j}$$

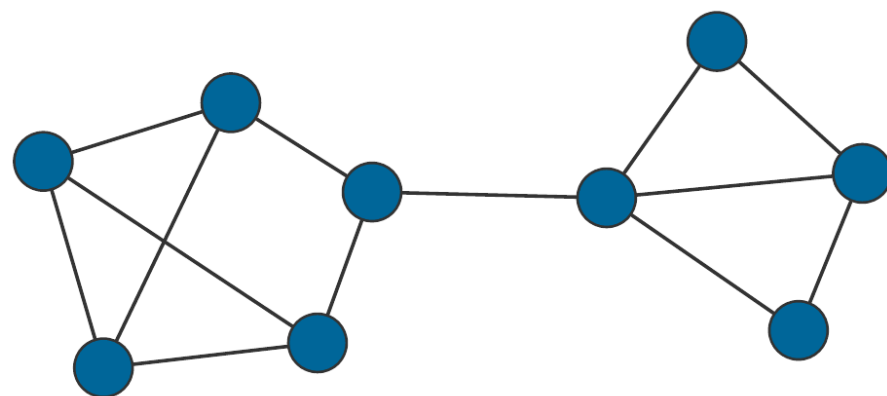
- m : #links in the network
- A_{ij} : adjacency matrix, 1 if i and j are connected, 0 if not
- $\delta_{s_1 s_2}$: 1 if in the same community, 0 if not

Modularity

$$Q = \frac{1}{2m} \sum_{ij} A_{ij} \delta_{s_i, s_j}$$



Intuitive.

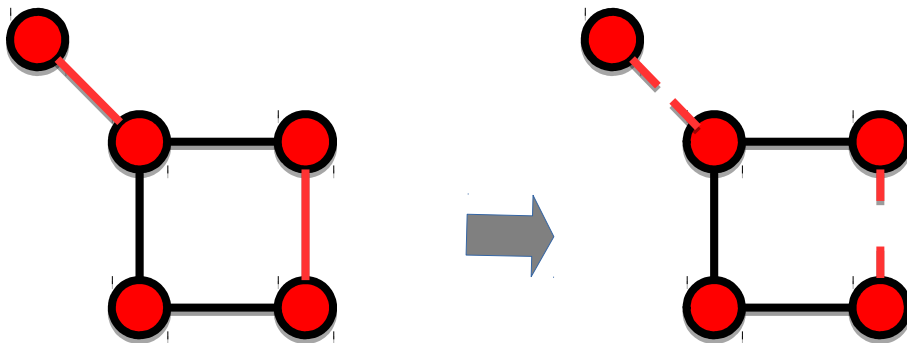


Optimal.

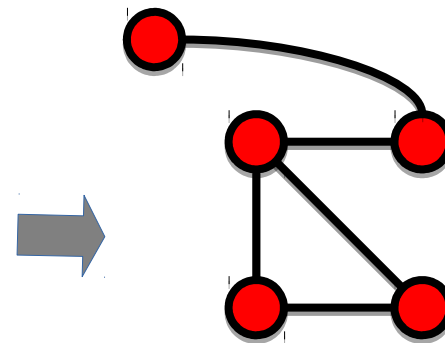
- Entire network one community: even if links are randomly placed, all links are inside the community
- Instead: fraction of links inside communities compared to what you would expect by chance
- What is by chance?

What is by chance?

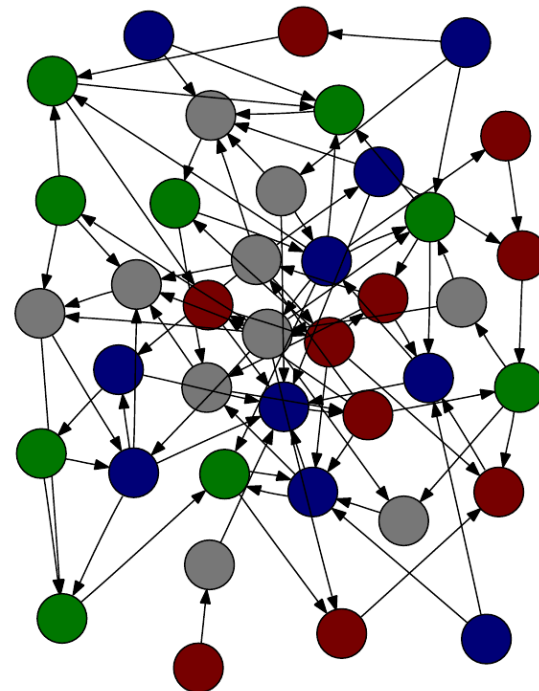
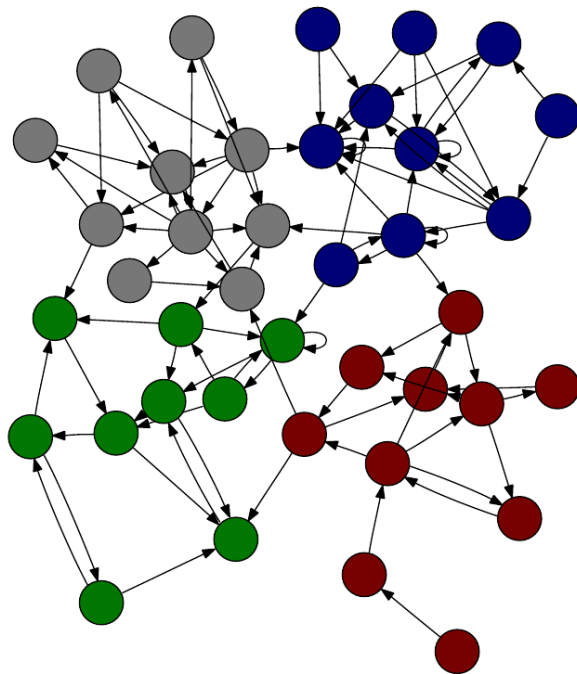
- Degree preserved randomization



Original



What we compare to



Modularity

- Modularity:

$$M = Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta_{s_i, s_j}$$

Real link

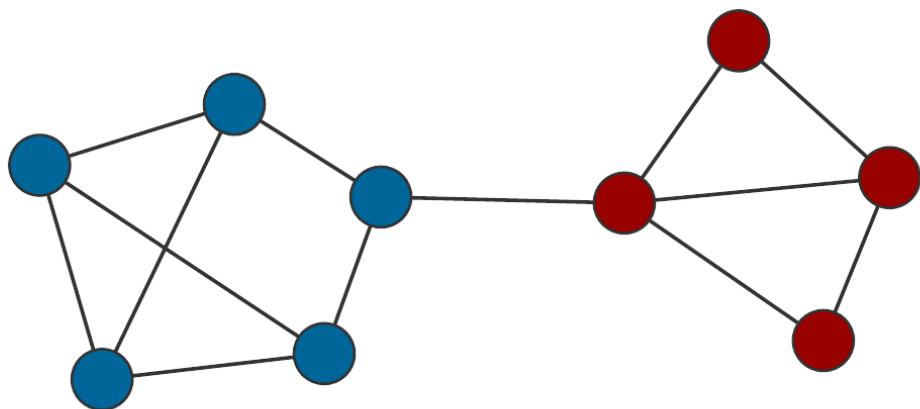
Probability of link in randomized version

- m : #links in the network
- A_{ij} : adjacency matrix, 1 if i and j are connected, 0 if not
- k_i : degree of node i
- $\delta_{s_1 s_2}$: 1 if in the same community, 0 if not
- High $M \rightarrow$ good division

Modularity

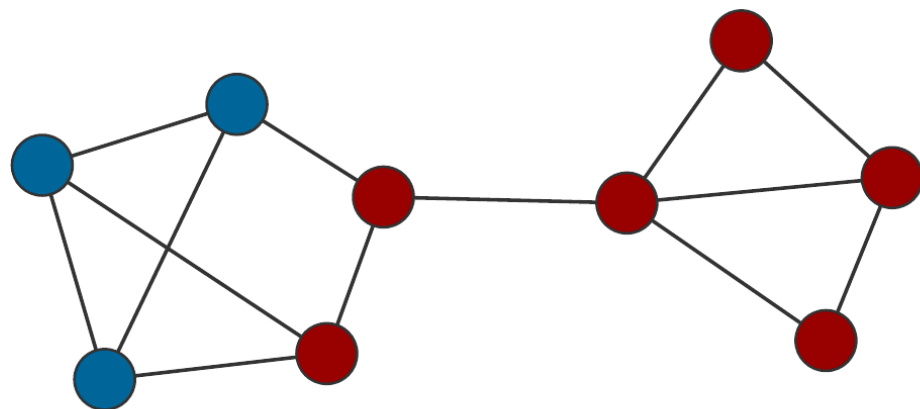
(a)

Optimal Partition
 $M = 0.41$



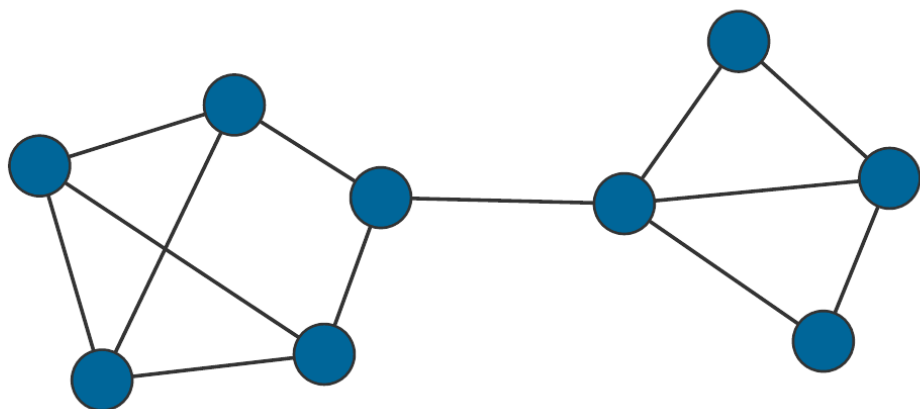
(b)

Suboptimal Partition
 $M = 0.22$



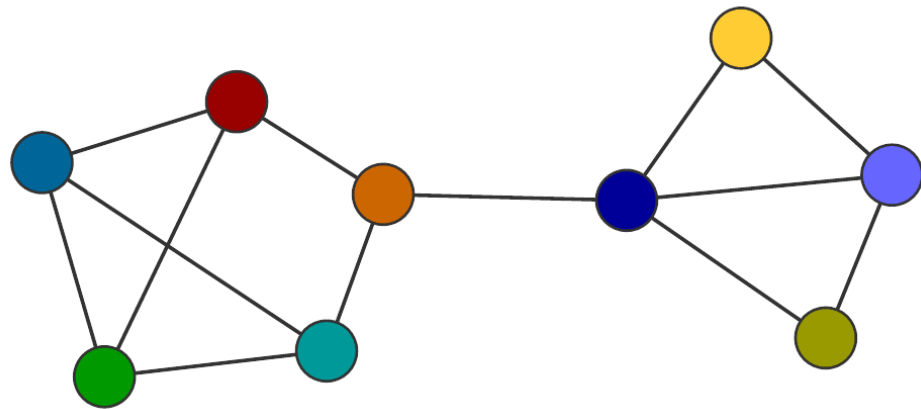
(c)

Single Community
 $M = 0$



(d)

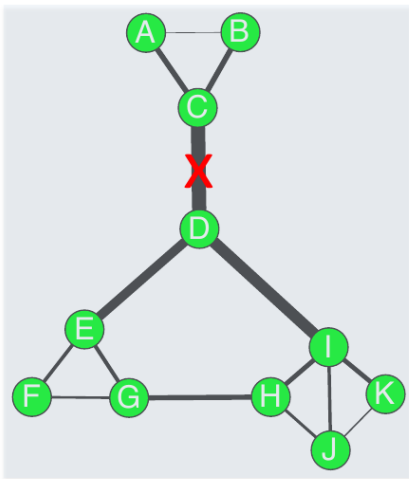
Negative Modularity
 $M = -0.12$



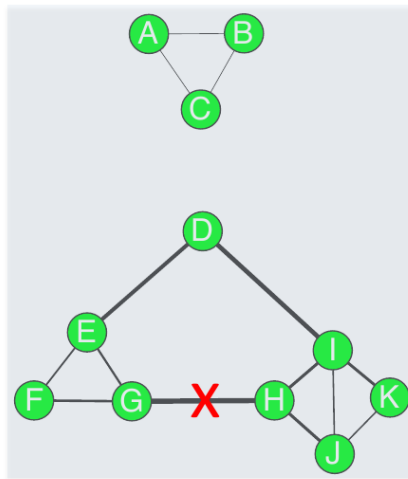
Modularity

- Where to cut dendrogram? At maximum Q!

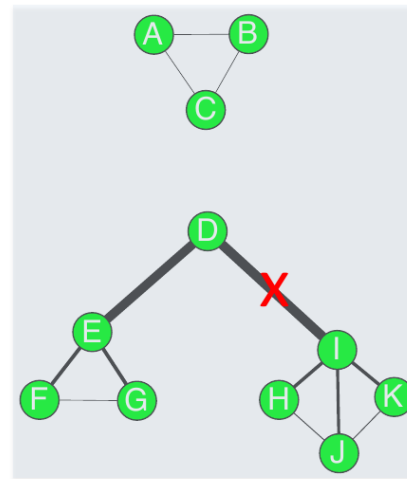
(a)



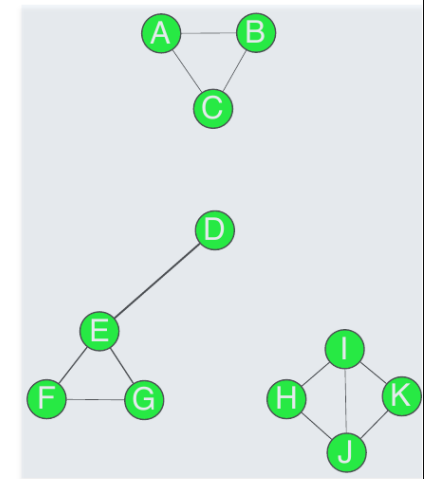
(b)



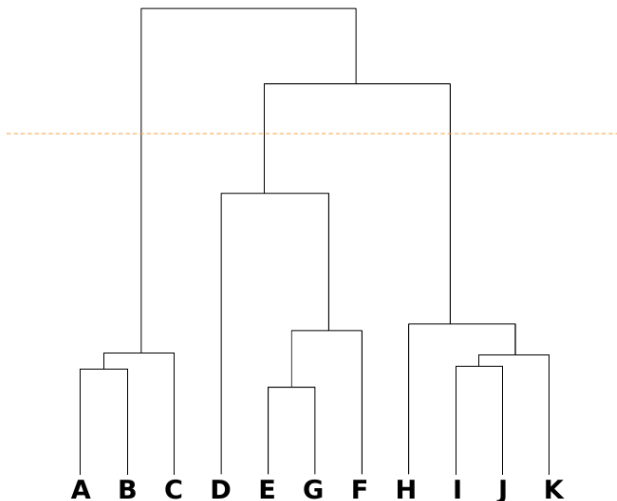
(c)



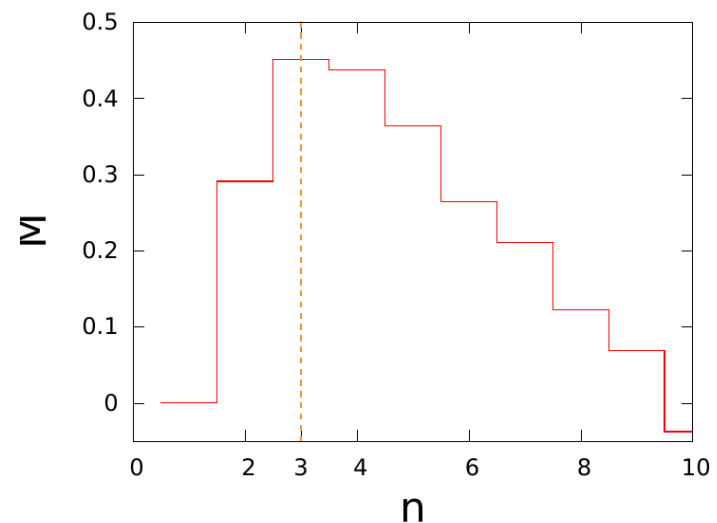
(d)



(e)



(f)



Method 3: Direct optimization of Q

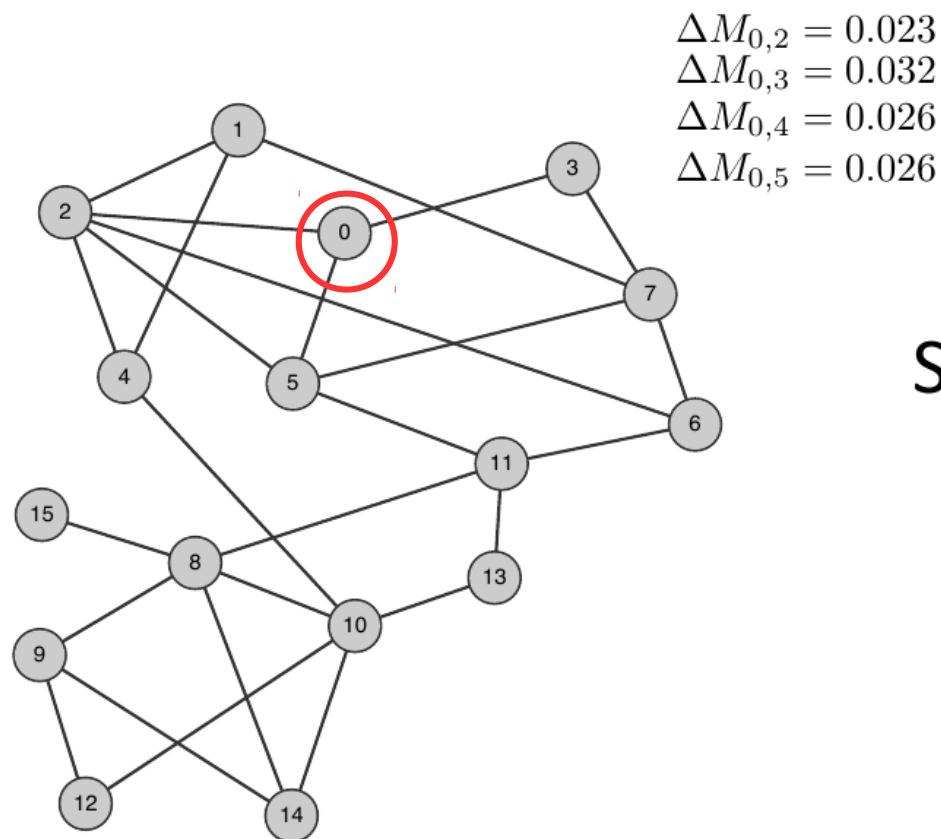


Direct optimization of modularity

- Exact maximum of $Q \rightarrow$ NP-complete (exponentially increasingly difficult with N)
- Approximation methods: a lot to choose from
- Louvain method
 - Fast, $\sim L$
 - Typically preforms well on tests
- Two steps applied iteratively:
 - 1) Find local maximum.
 - 2) Coarse grain network.

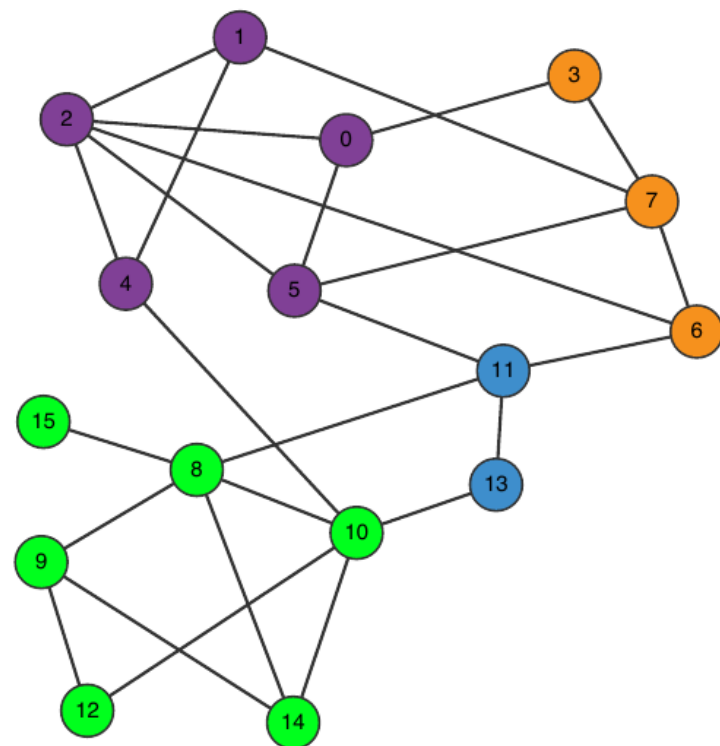
Louvain method: Step 1

- 1) For node i , calculate ΔQ for each neighbor j if node i is removed from its community and placed in the community of j . Coarse grain network.
- 2) Move i to community that maximizes ΔQ , if $\Delta Q > 0$.
- 3) Repeat while Q increases.



Start with each node in its own community

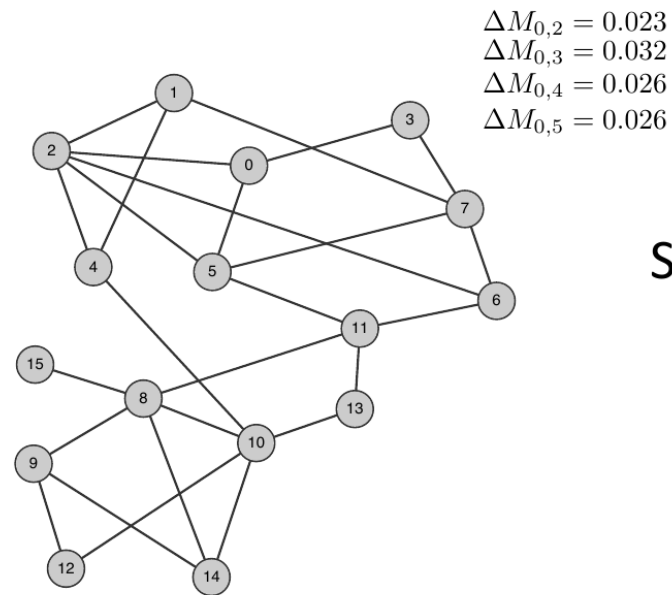
Step 1



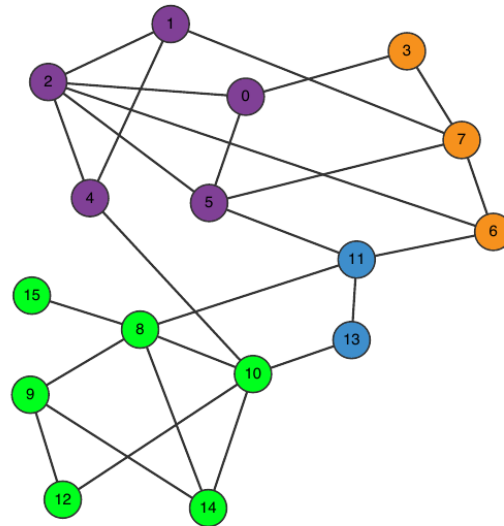
V.D. Blondel et al, *J. Stat. Mech.* P10008 (2008).

Louvain method

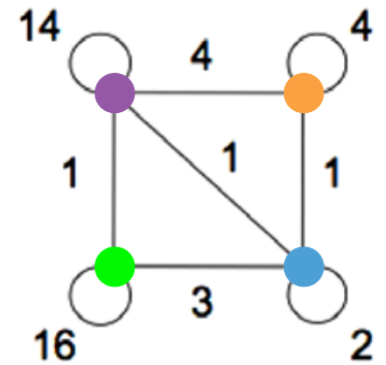
1st Pass



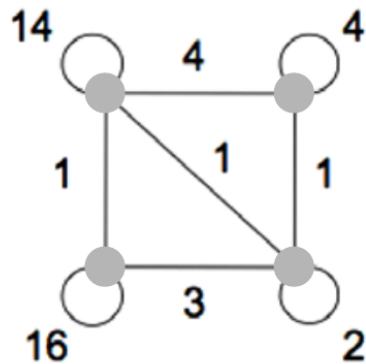
Step I



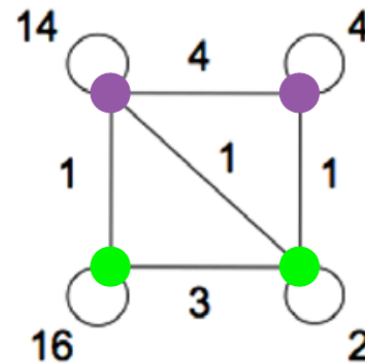
Step II



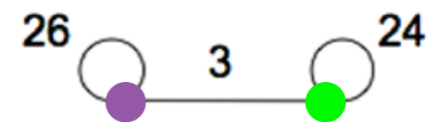
2nd Pass



Step I

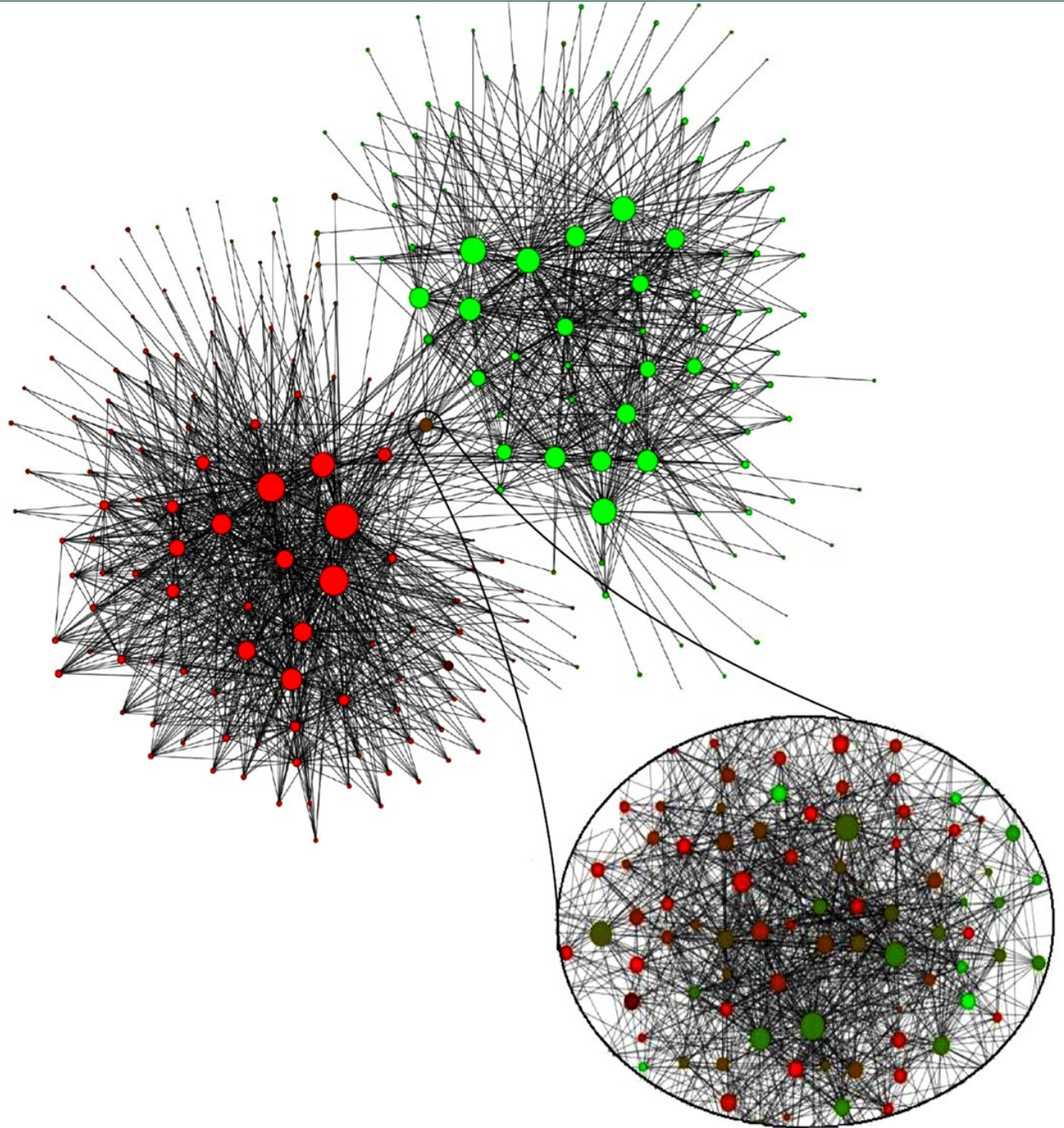


Step II



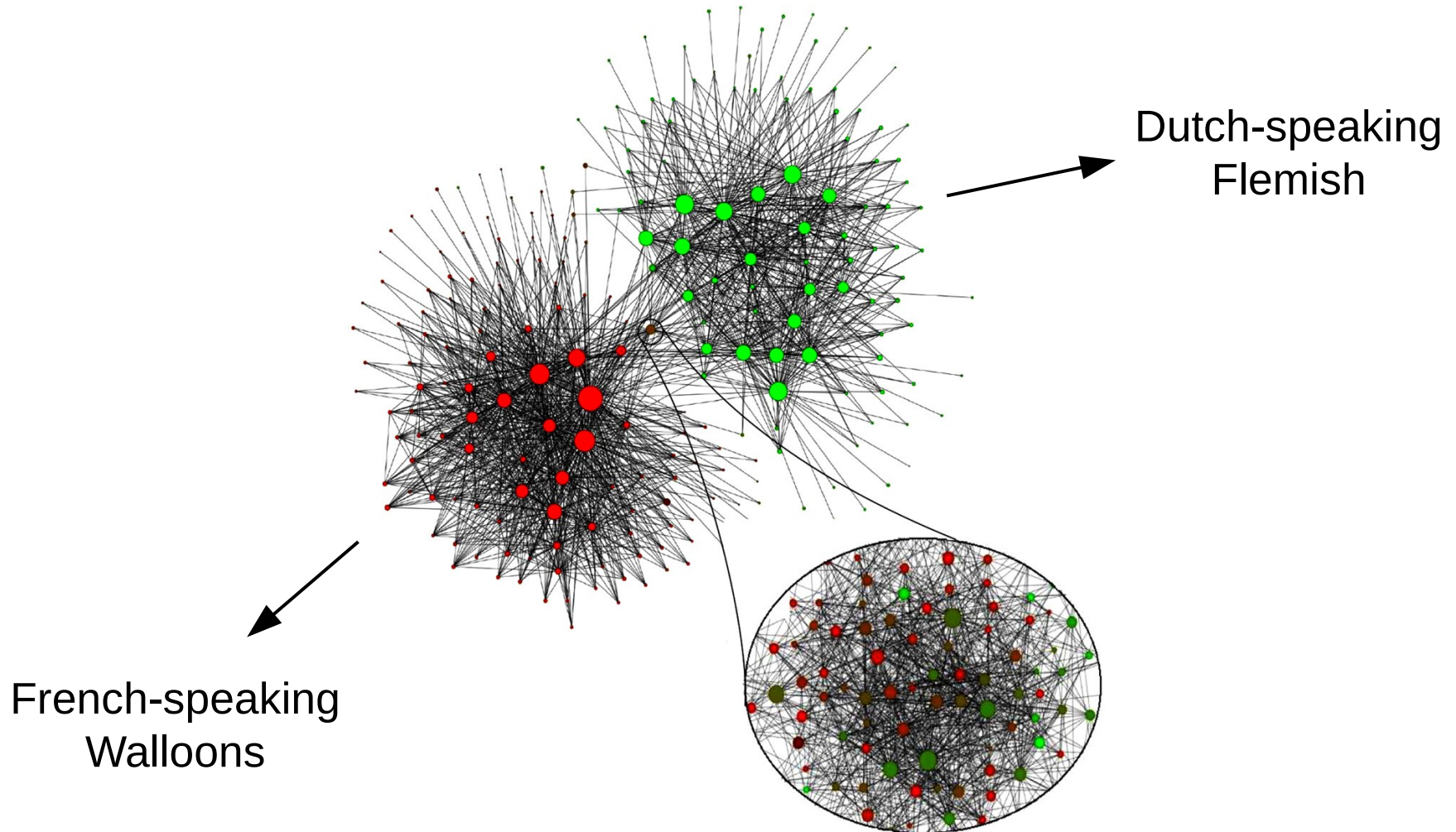
Application: Belgium phone call network

- Link (A – B) : A and B talk frequently on the phone
- Phone calls of ~2 million customers



Belgium phone call network

- Link (A – B) : A and B talk frequently on the phone
- Phone calls of ~2 million customers



Comparing methods

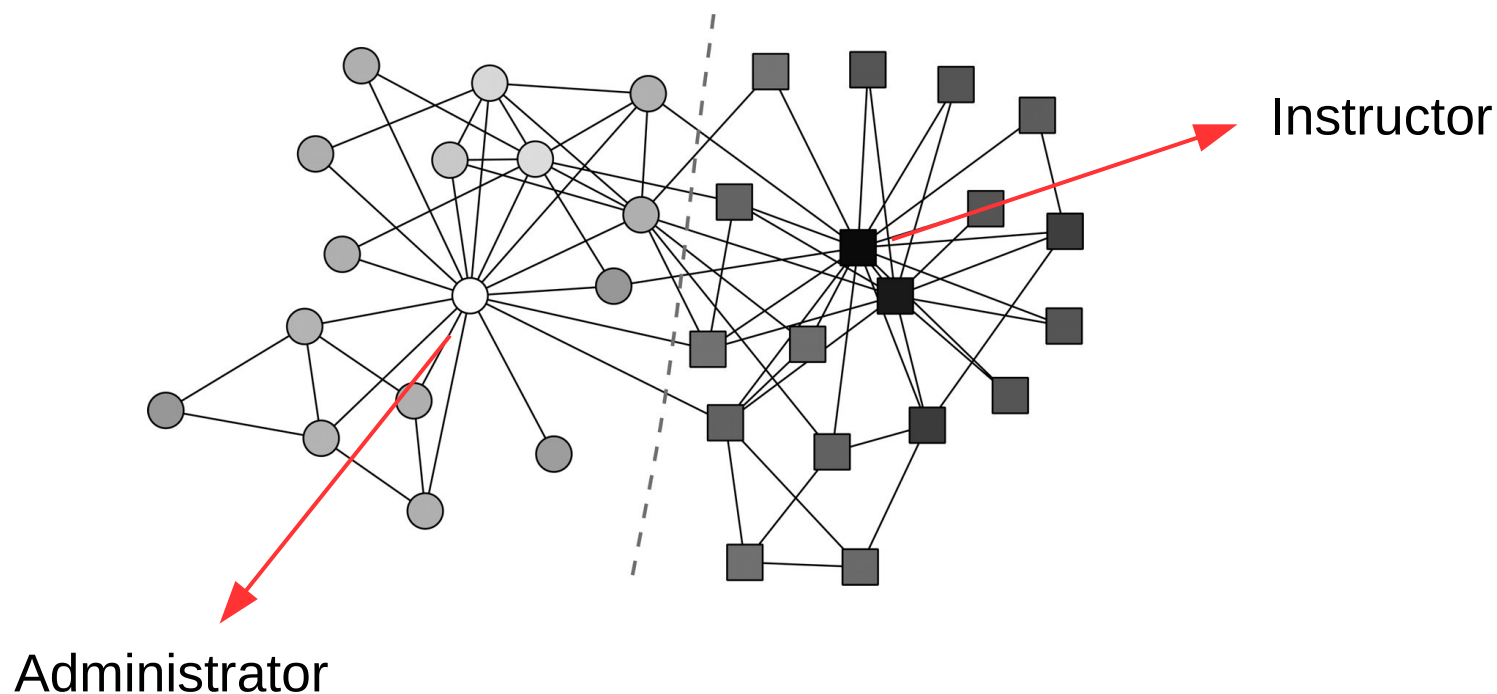


Comparing methods

- We need a network where we know the true community.
- Option I: Real networks with known ground truth
- Option II: Model networks with built-in communities

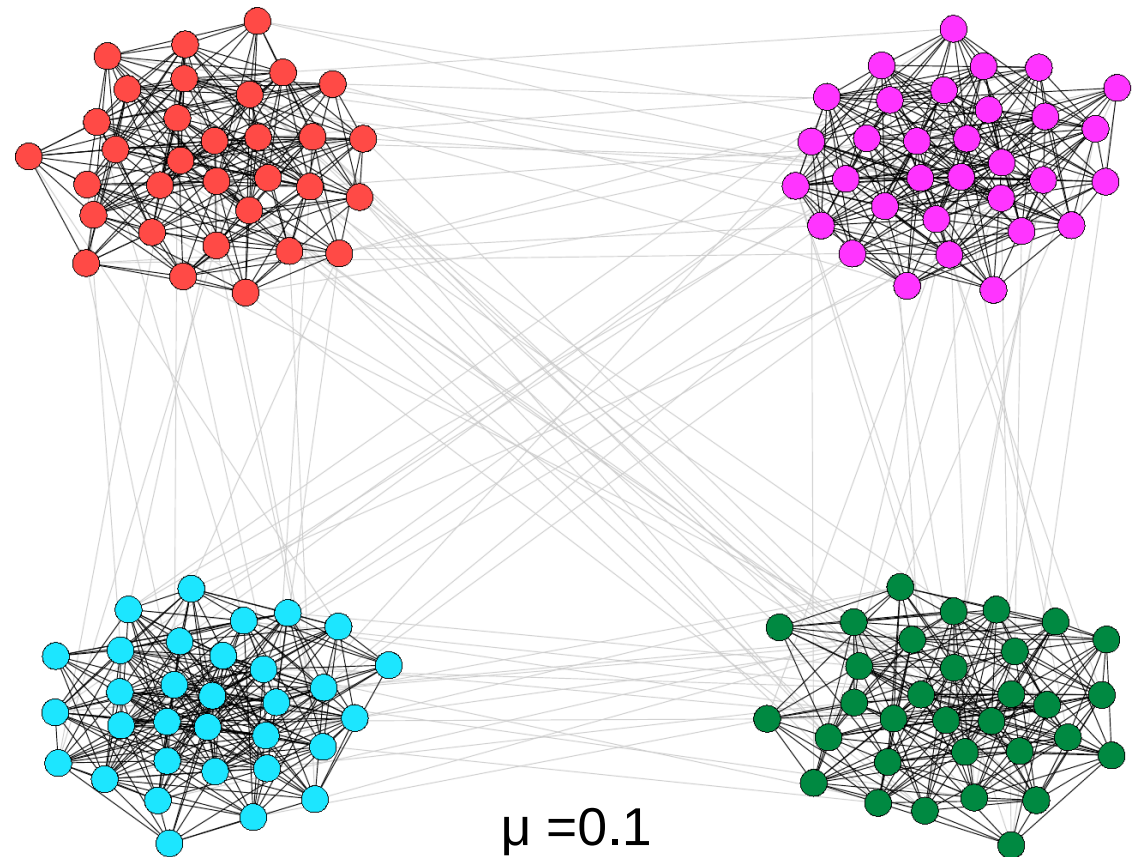
Comparing methods

- We need a network where we know the true community.
- Option I: Real networks
- Zachary Karate Club



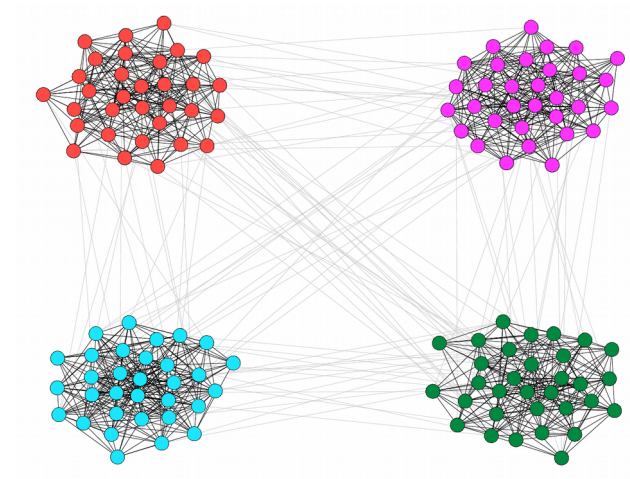
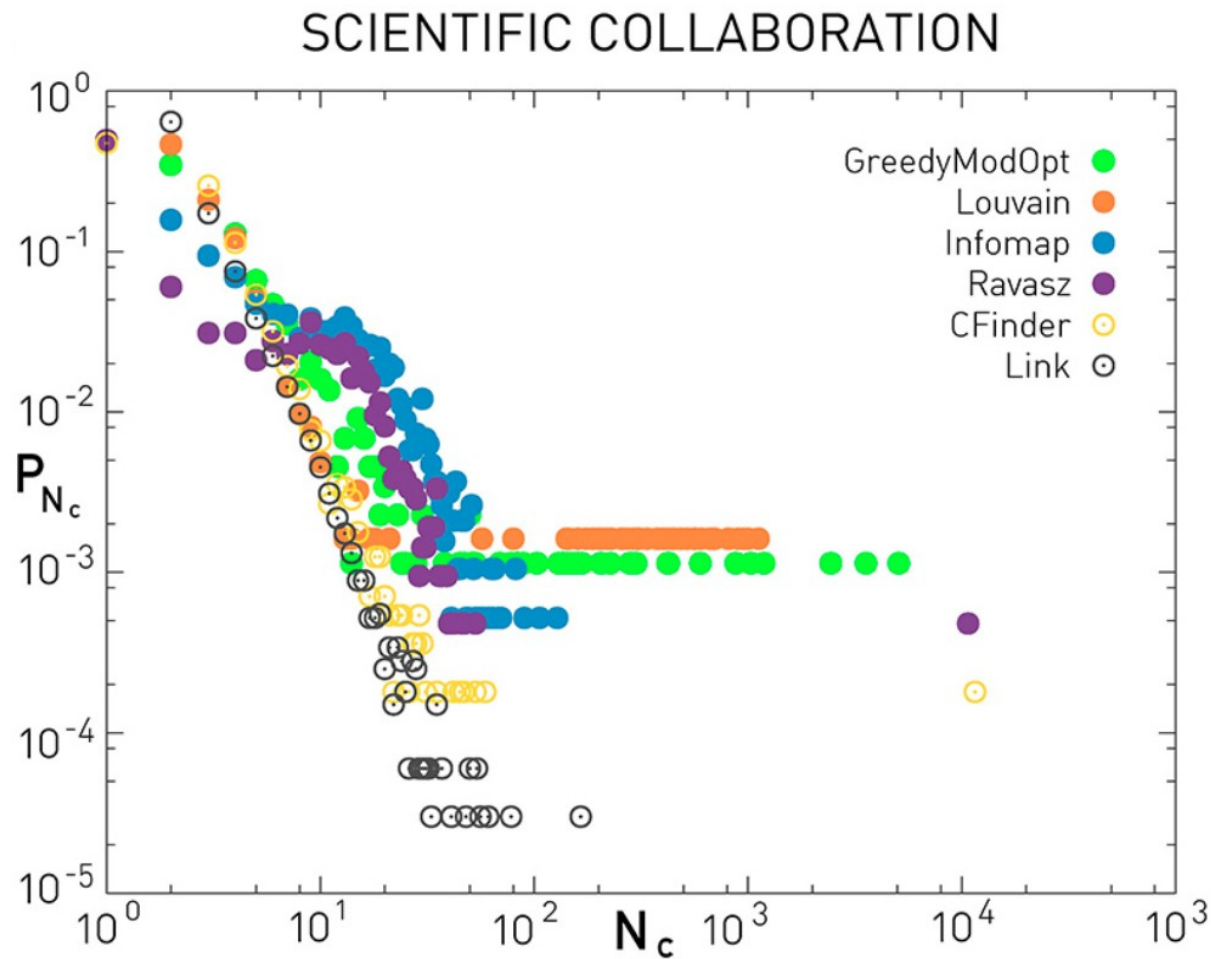
Artificial benchmarks

- Girvan-Newman benchmark
- $N=128$ node divided into 4 groups, $\langle k \rangle=16$
- p_{in} = prob. that two nodes in the same group are connected
- p_{out} = prob. that two nodes in different groups are connected (not independent)
- μ = fraction of external links = $3p_{out}/(p_{in}+3p_{out})$
- No communities:
 $p_{in}=p_{out}$ or $\mu=0.75$



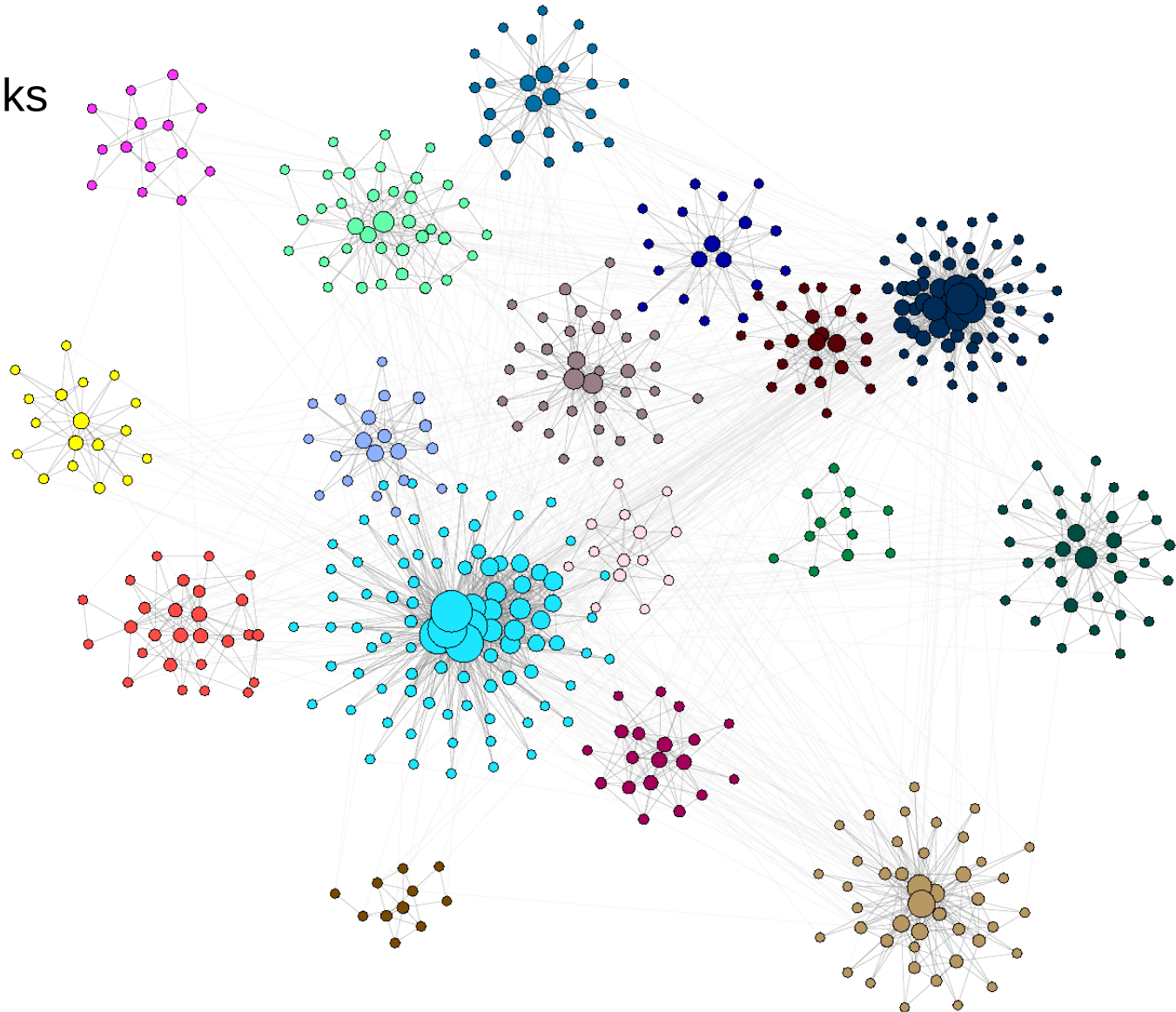
Artificial benchmarks

- Is the Girvan-Newman benchmark realistic?
- Community size distribution



Artificial benchmarks

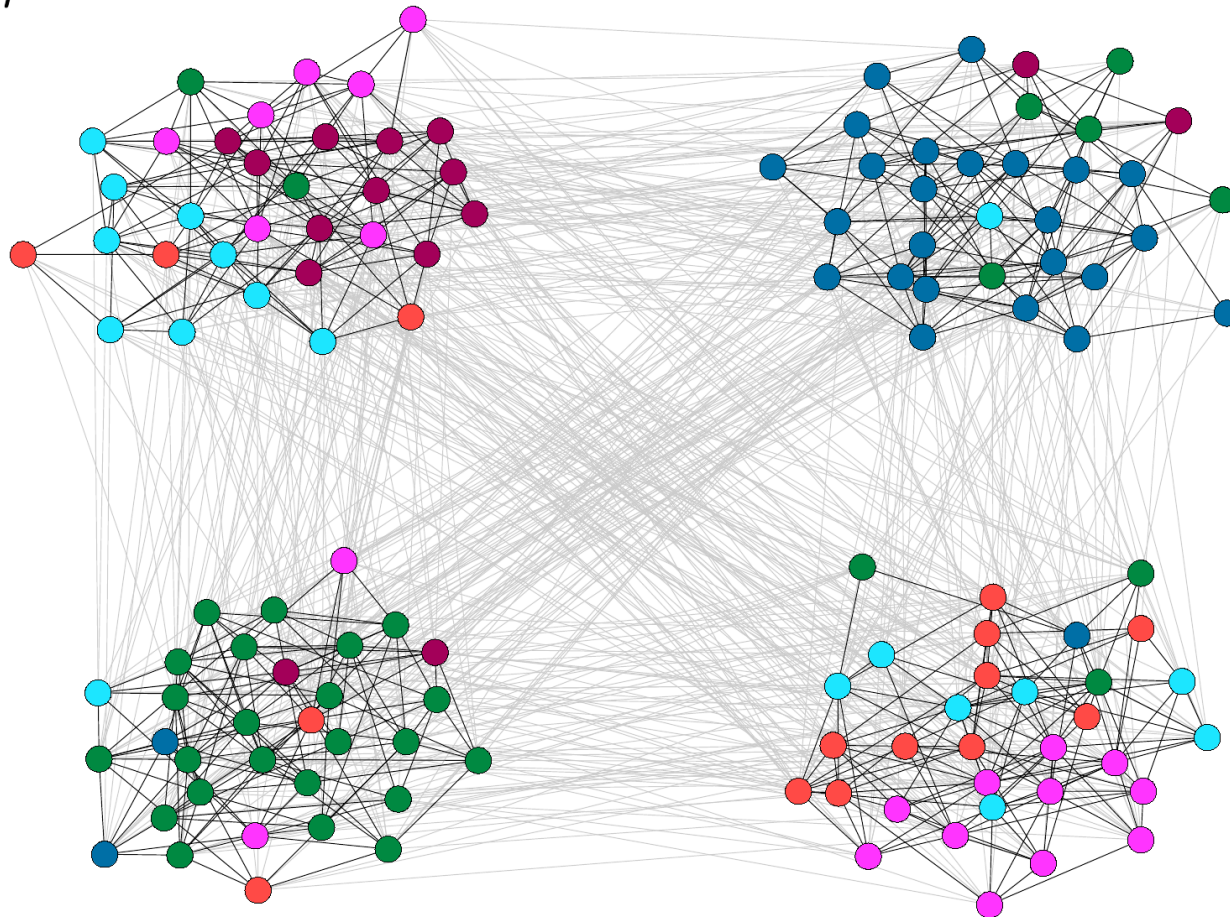
- Lancichinetti–Fortunato–Radicchi (LFR) benchmark
- N nodes, N_c communities
- μ = fraction of external links
- Power-law degree distribution
- Power-law community size distribution



Comparing community divisions

- We know what we should get.
- How to systematically compare what we found?
- Again, a lot of possibilities
- Our choice now: Normalized mutual information

$$\mu = 0.50$$



Normalized mutual information

Information theory approach: if two partitions are similar, one needs very little information to infer one partition given the other. We can use **the mutual information**

Shannon entropy:

$$H(\{C_1\}) = -\sum_{C_1} p(C_1) \log p(C_1)$$

Measures the amount of information in a string of random variables drawn from $p(C_1)$

Probability that a randomly chosen node belongs to community C_i

$$p(C_1) = \frac{\text{how many nodes belong to } C_1}{\text{sum over all partitions}} = \frac{N_{C_1}}{\sum_C N_C}$$

Normalized mutual information

Information theory approach: if two partitions are similar, one needs very little information to infer one partition given the other. We can use **the mutual information**

$$I(\{C_1\}, \{C_2\}) = \sum_{C_1} \sum_{C_2} p(C_1, C_2) \log \frac{p(C_1, C_2)}{p(C_1) p(C_2)}$$

Joint probability that a randomly chosen node belongs to community C_1 in the first partition and C_2 in the second

$$p(C_1, C_2) = \frac{\text{how many nodes that are in } C_1 \text{ are also in } C_2}{\text{sum over all possible pairs } C_1 \text{ and } C_2} = \frac{N_{C_1 C_2}}{\sum_{C_1, C_2} N_{C_1 C_2}}$$

Probability that a randomly chosen node belongs to community C_1

$$p(C_1) = \frac{\text{how many nodes belong to } C_1}{\text{sum over all partitions}} = \frac{N_{C_1}}{\sum_C N_C}$$

Normalization by average Shannon entropy:

$$I_n(\{C_1\}, \{C_2\}) = \frac{2I(\{C_1\}, \{C_2\})}{H(\{C_1\}) + H(\{C_2\})}$$

Normalized mutual information

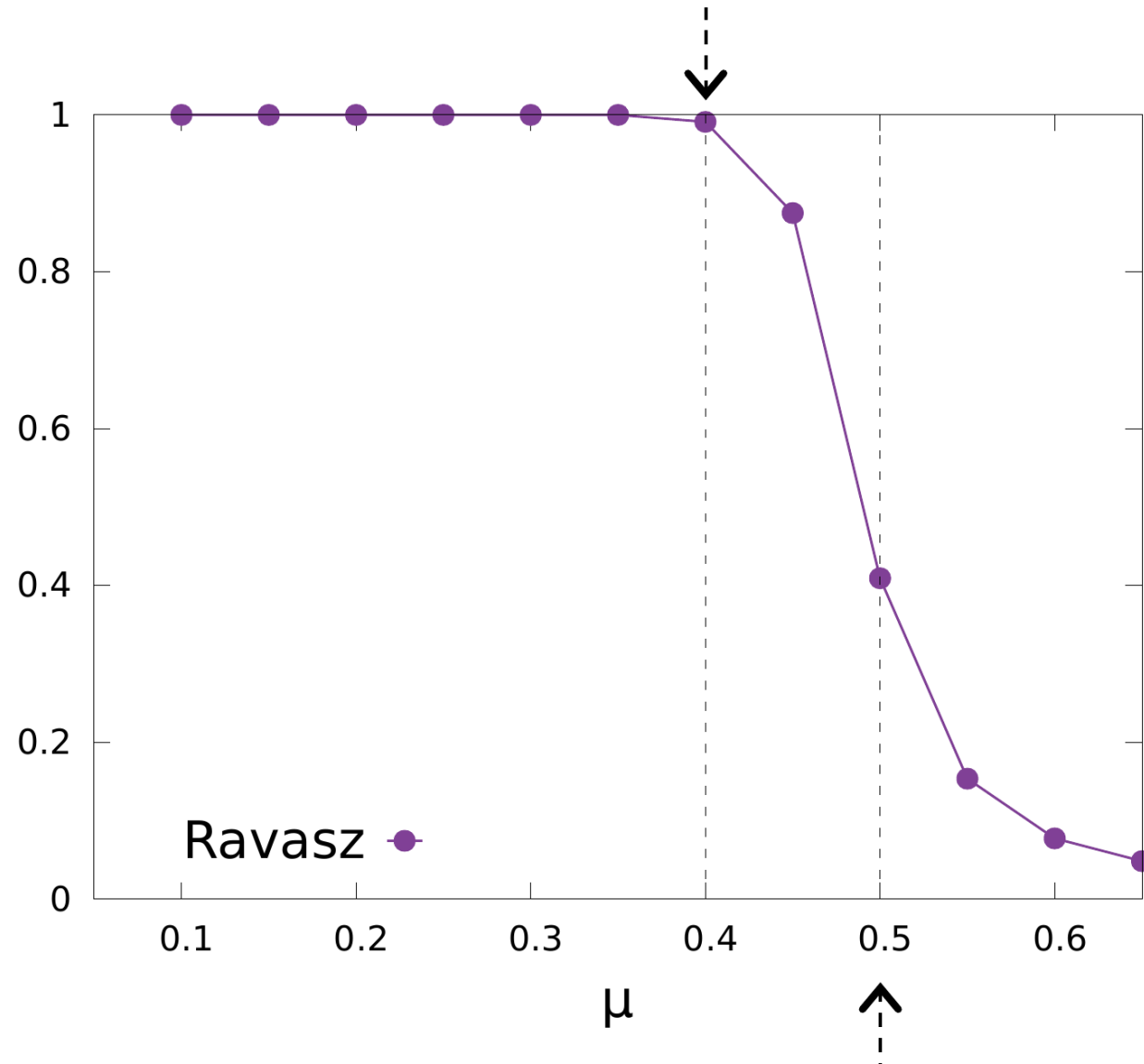
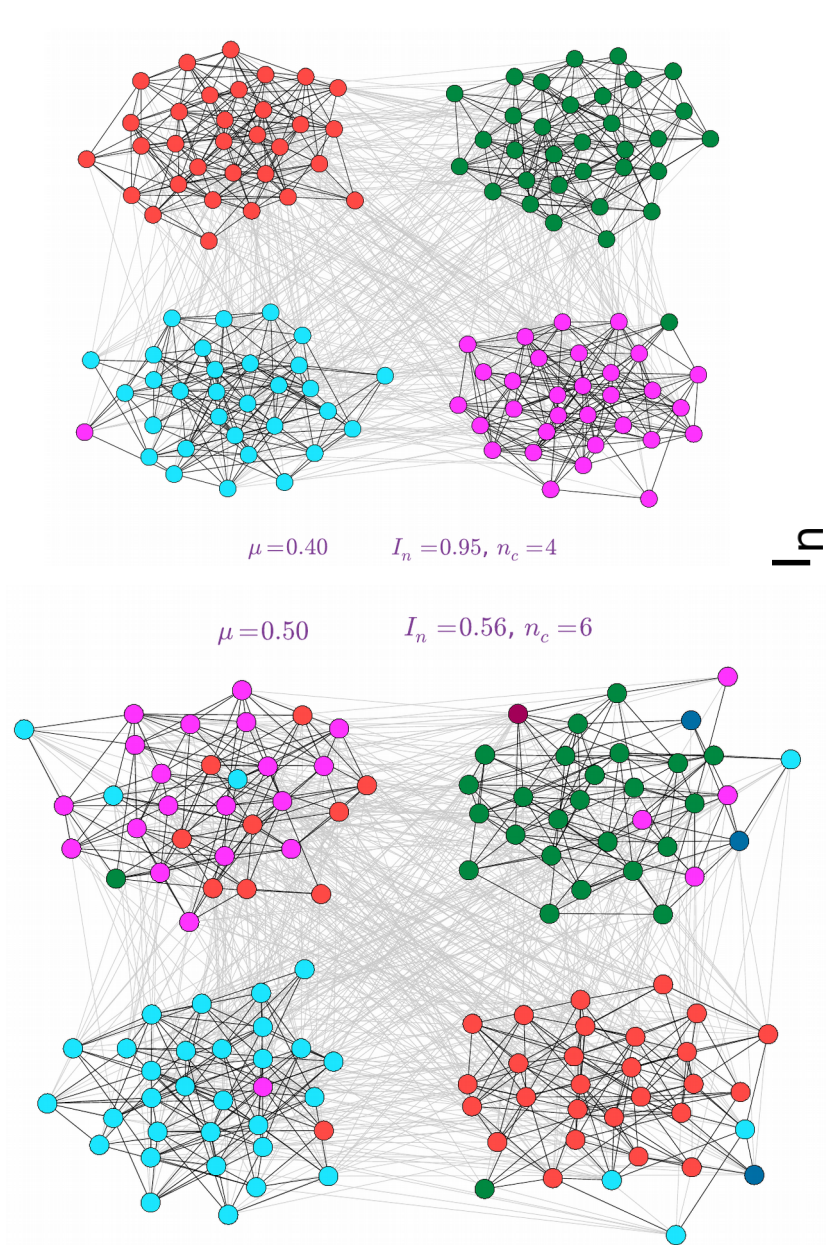
- In summary:

$$I_n(\{C_1\}, \{C_2\}) = \frac{2I(\{C_1\}, \{C_2\})}{H(\{C_1\}) + H(\{C_2\})}$$

- it quantifies the "amount of information" (in units such as bits) obtained about one random variable, through the other random variable (wiki)
- $I_n = 1 \rightarrow$ same division
- $I_n = 0 \rightarrow$ two divisions independent from each other

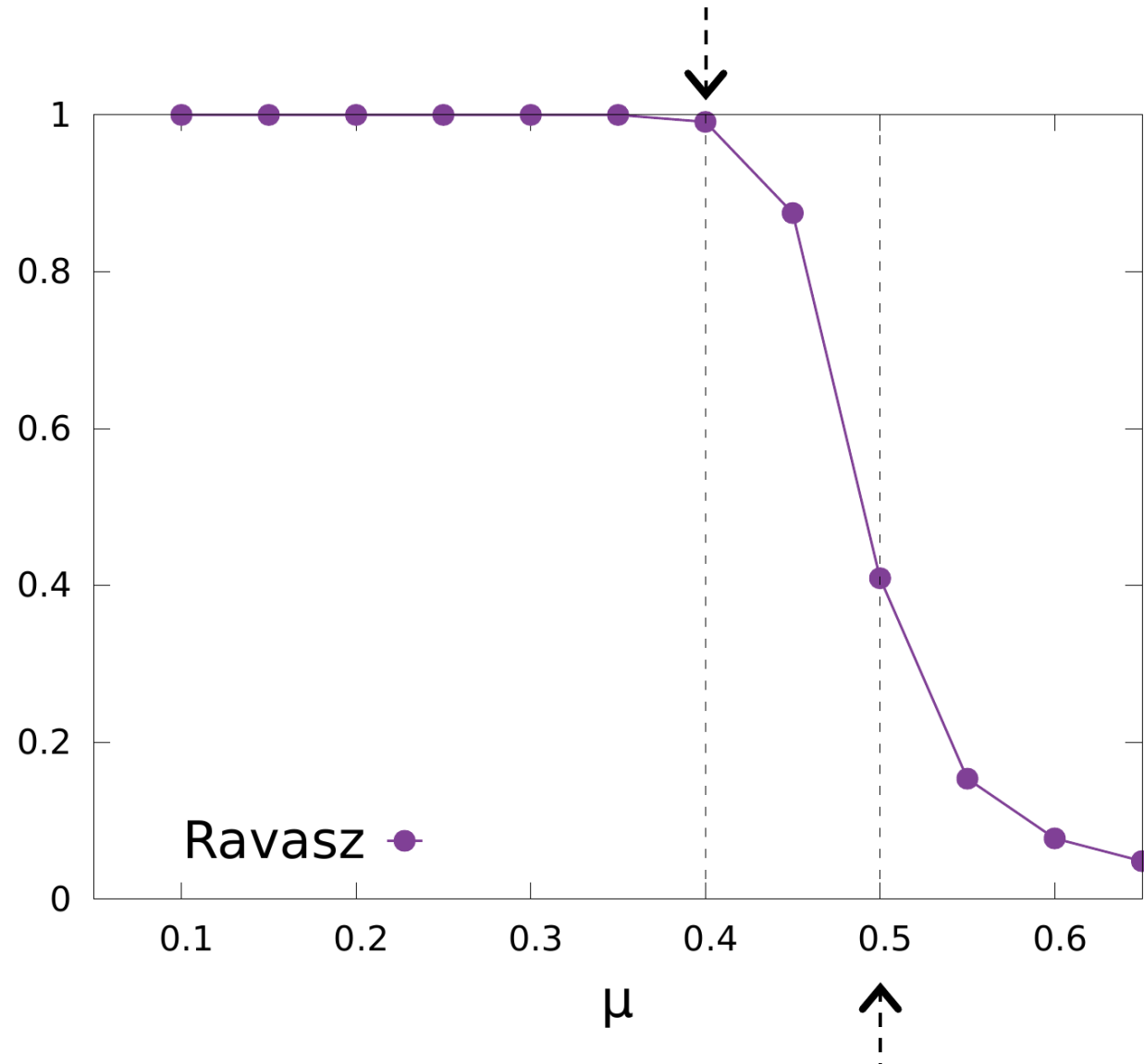
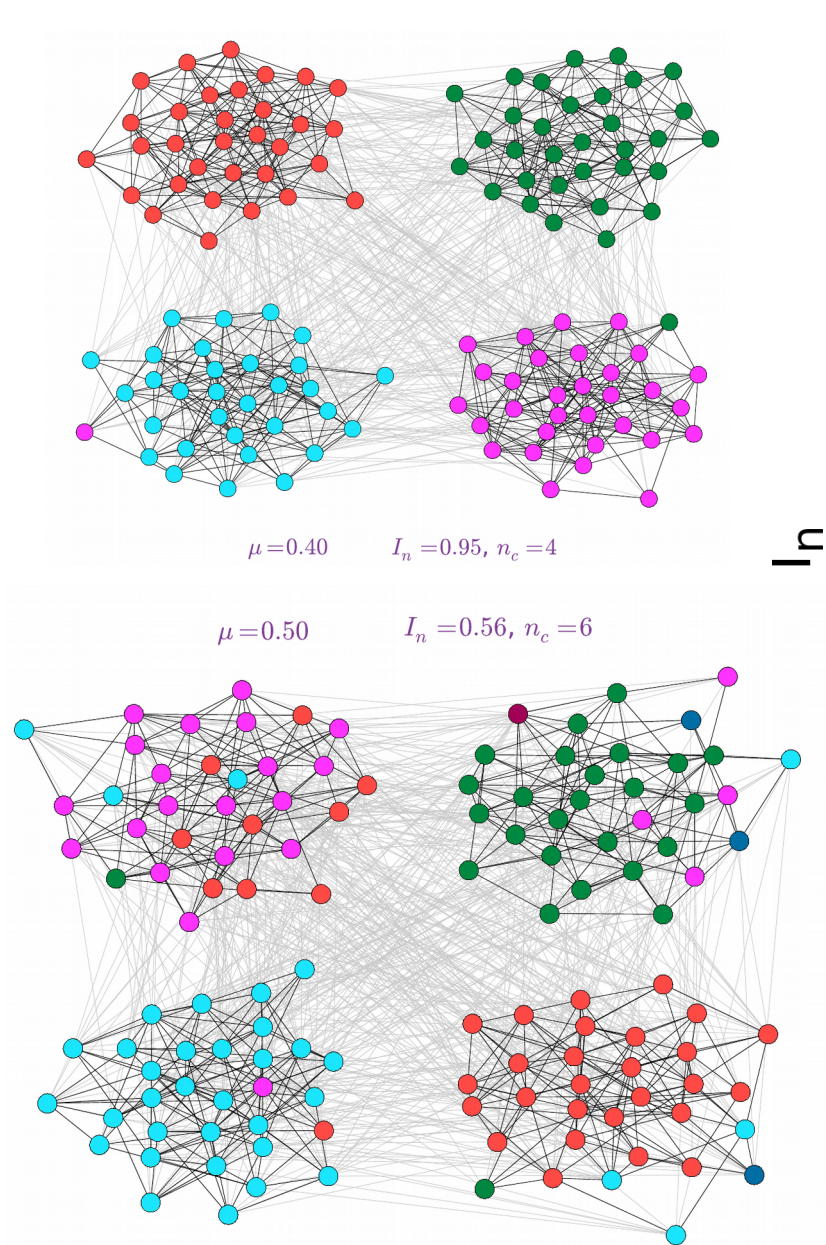
Benchmarks and NMI in action

- NG benchmark, hierarchical clustering



Benchmarks and NMI in action

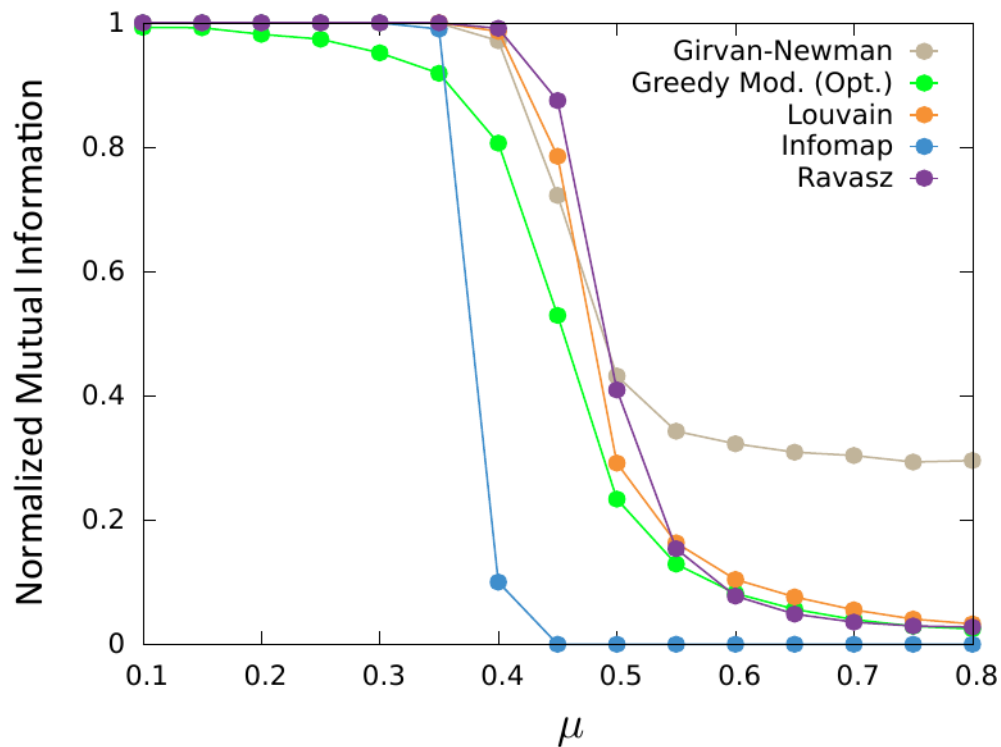
- NG benchmark, hierarchical clustering



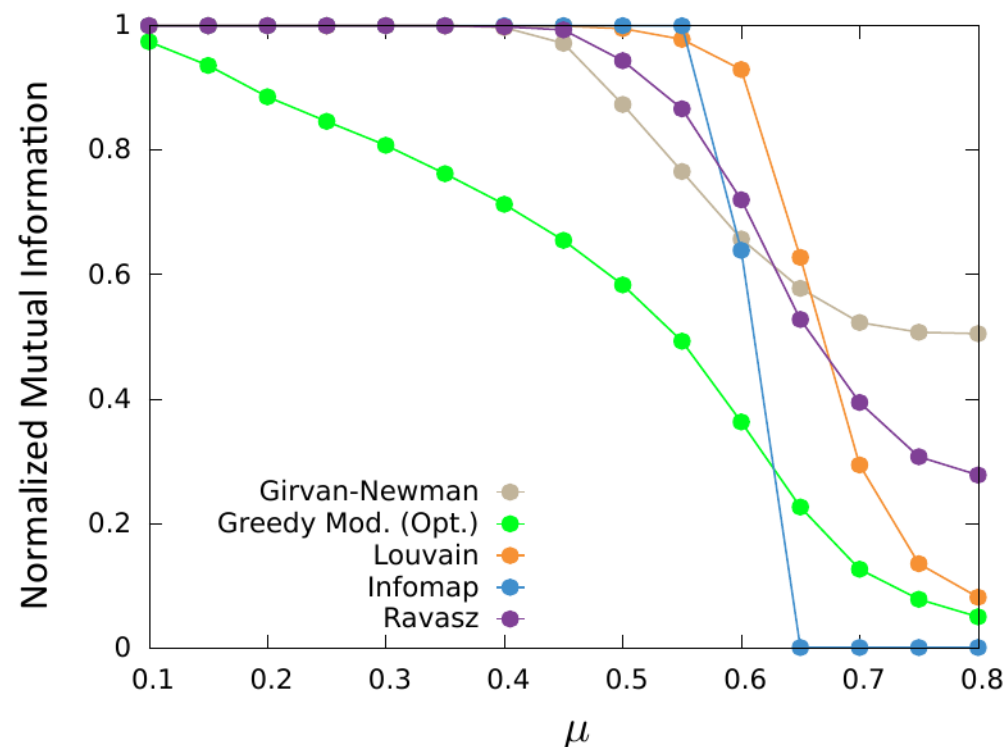
Benchmarks and NMI in action

- Purple: Hierarchical; Orange: Louvain; Gray: Betweenness

NG Benchmark



LFR Benchmark



LFR parameters: $N = 1000$; degree exp. = 2; max degree = 50; comm. size exponent = 1, min comm. size = 20, max = 100

Short list of other methods

- Many other methods to find communities:
 - Local: instead of finding global division, find the community a given node belongs to
 - Spectral: based on spectrum of graph Laplacian
 - Dynamical: Potts-model, oscillators, random walks
 - Stochastic block models: find best fit using maximum likelihood fit of benchmark-like model → mathematically principled results

How to choose method?

- What is the best method?
 - No clear answer.
- Better question: What is the method that fits my needs?
 - Network features: Size? Directed? Weighted? Bipartite?
 - What do we expect to find? Overlapping communities? Size of the groups?

How to choose method?

- What is the best method?
 - No clear answer.
- Better question: What is the method that fits my needs?
 - Network features: Size? Directed? Weighted? Bipartite?
 - What do we expect to find? Overlapping communities? Size of the groups?

How to choose method?

	Name	Overlap	Dir	Weight	Dyn	NoPar	MDim	Incr	Multip	Complexity	BESn	BESm	Year	Ref
Feature Distance	Evolutionary*				✓			✓		$\mathcal{O}(n^2)$	5k	?	2006	[21]
	MSN-BD			✓					✓	$\mathcal{O}(n^2 ck)$	6k	3M	2006	[22]
	SocDim	✓		✓			✓			$\mathcal{O}(n^2 \log n)^*$	80k	6M	2009	[23]
	PMM			✓			✓			$\mathcal{O}(mn^2)$	15k	27M	2009	[24]
	MRGC		✓		✓		✓		✓	$\mathcal{O}(mD)$	40k	?	2007	[25]
	Infinite Relational					✓	✓			$\mathcal{O}(n^{2c} D)$	160	?	2006	[26]
	Find-Tribes				✓				✓	$\mathcal{O}(mnK^2)$	26k	100k	2007	[27]
	AutoPart		✓			✓		✓		$\mathcal{O}(mk^2)$	75k	500k	2004	[28]
	Timefall				✓	✓			✓	$\mathcal{O}(mk)$	7.5M	53M	2008	[29]
	Context-specific Cluster Tree					✓			✓	$\mathcal{O}(mk)$	37k	367k	2008	[30]
IntDensity	Modularity	✓	✓	✓			✓	✓	✓	$\mathcal{O}(mk \log n)$	118M	1B	2004	[18]
	MetaFac				✓		✓			$\mathcal{O}(mnD)$	?	2M	2009	[31]
	Variational Bayes		✓			✓				$\mathcal{O}(mk)$	115	613	2008	[32]
	$LA \rightarrow IS^{2*}$	✓	✓							$\mathcal{O}(mk + n)$	16k	?	2005	[33]
	Local Density		✓			✓		✓		$\mathcal{O}(nK \log n)$	108k	330k	2005	[34]
Bridge	Edge Betweenness		✓	✓						$\mathcal{O}(m^2 n)$	271	1k	2002	[4]
	CONGO*	✓		✓						$\mathcal{O}(n \log n)$	30k	116k	2008	[35]
	L-Shell	✓						✓		$\mathcal{O}(n^3)$	77	254	2005	[36]
	Internal-External Degree	✓								$\mathcal{O}(n^2 \log n)$	775k	4.7M	2009	[37]
Diffusion	Label Propagation			✓		✓		✓		$\mathcal{O}(m + n)$	374k	30M	2007	[38]
	Node Colouring				✓				✓	$\mathcal{O}(ntk^2)$	2k	?	2007	[39]
	Kirchhoff	✓		✓						$\mathcal{O}(m + n)$	115	613	2004	[40]
	Communication Dynamic	✓	✓		✓			✓		$\mathcal{O}(mnt)$	160k	530k	2008	[41]
	GuruMine		✓		✓					$\mathcal{O}(TAn^2)$	217k	212k	2008	[9]
	DegreeDiscountIC		✓							$\mathcal{O}(k \log n + m)$	37k	230k	2009	[42]
	MMSB	✓	✓							$\mathcal{O}(nk)$	871	2k	2007	[43]
Close	Walktrap			✓						$\mathcal{O}(mn^2)$	160k	1.8M	2006	[44]
	DOCS	✓								?	325k	1M	2009	[45]
	Infomap		✓	✓		✓				$\mathcal{O}(m \log^2 n)$	6k	6M	2008	[46]
Structure	K-Clique	✓								$\mathcal{O}(m^{\frac{n-m}{10}})$	20k	127k	2005	[3]
	S-Plexes Enumeration	✓								$\mathcal{O}(kmn)$	?	?	2009	[47]
	Bi-Clique	✓							✓	$\mathcal{O}(m^2)$	200k	500k	2008	[48]
	EAGLE	✓	✓	✓						$\mathcal{O}(3^{\frac{n}{3}})$	16k	31k	2009	[49]
Link	Link modularity	✓		✓					✓	$\mathcal{O}(2mk \log n)$	20k	127k	2009	[50]
	HLC*	✓		✓					✓	$\mathcal{O}(n\bar{K}^2)$	885k	5.5M	2010	[51]
	Link Maximum Likelihood	✓								$\mathcal{O}(mk)$	4.8M	42M	2011	[52]
NoD	Hybrid*	✓	✓	✓		✓				$\mathcal{O}(nkK)$	325k	1.5M	2010	[53]
	Multi-relational Regression			✓			✓			?	?	?	2005	[54]
	Hierarchical Bayes									$\mathcal{O}(n^2)$	1k	4k	2008	[55]
	Expectation Maximization		✓							?	112	?	2007	[7]

Literature

- Where to start reading?

- 1) Newman, M. E., Communities, modules and large-scale structure in networks. *Nature Physics*, 8(1):25-31 (2012).
→ Short, big picture
- 2) Fortunato, S., Community detection in graphs. *Physics reports*, 486(3):75-174 (2010).
→ >100 pages, complete at the time, good for looking up methods
- 3) Coscia, M., Giannotti, F., & Pedreschi, D., A classification for community discovery methods in complex networks. *Statistical Analysis and Data Mining*, 4(5):512-546 (2011).
→ shorter, compares a lot of methods
- 4) A.-L. Barabási, *Network Science*, Chapter 9
<http://barabasi.com/networksciencebook/>
Appears in print in May.
→ Lot of figures of the lecture are from here. Easy to read, tells a detailed story, but does not cover everything.

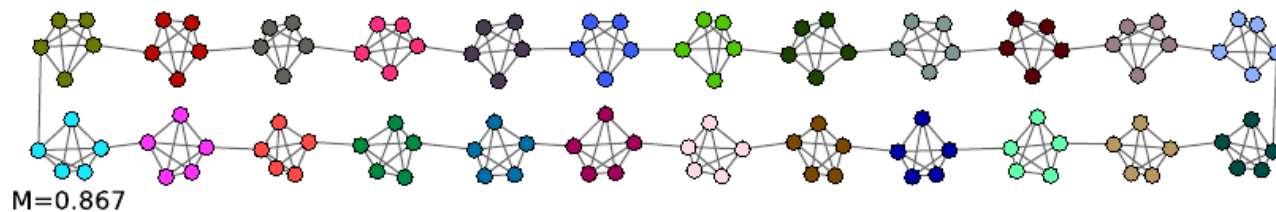
Extra time:

Problems with modularity

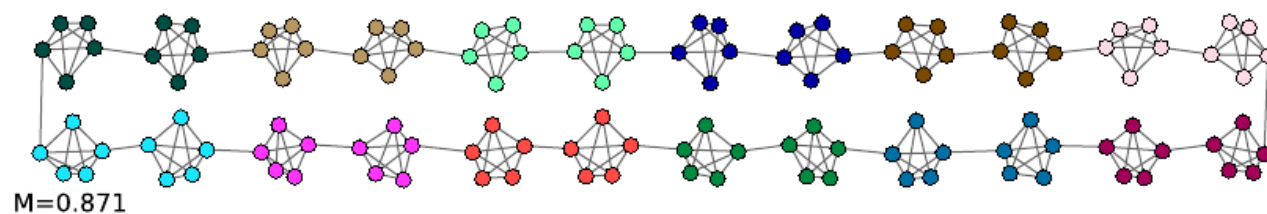


Is the maximum unique?

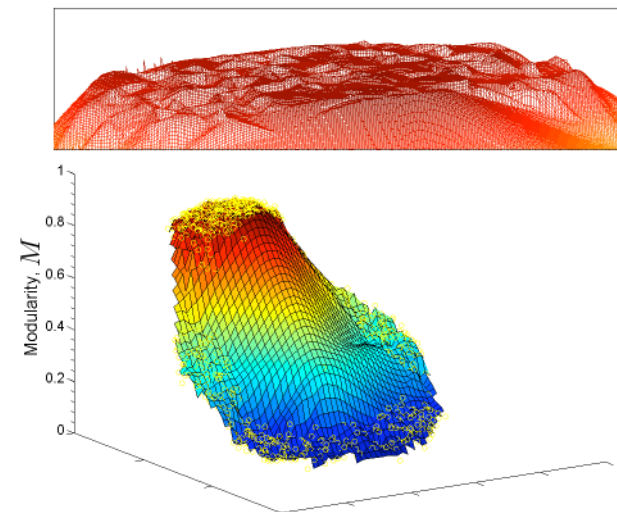
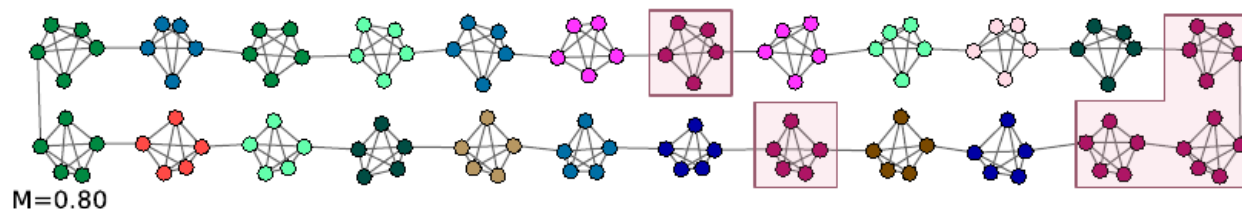
- Should we merge two communities?
- Intuitive:



- Global maximum:

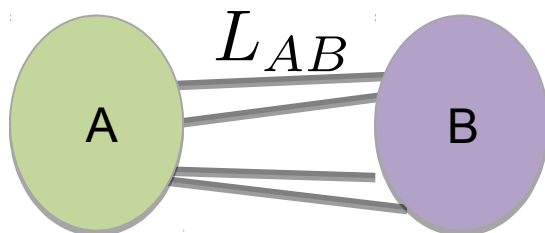


- Random:



Resolution limit

- Should we merge two communities?

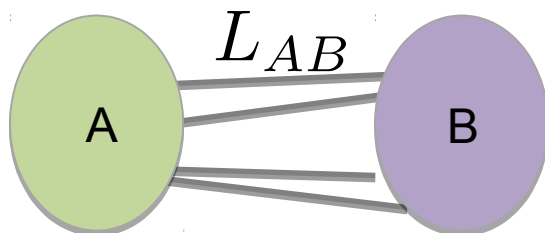


$$\Delta M_A = \frac{1}{2L} \sum_{i,j \in A} A_{ij} - \frac{k_i k_j}{2L} = \frac{2k_A^{\text{int}}}{2L} - \frac{1}{2L} \sum_{i,j \in A} \frac{k_i k_j}{2L}$$

$$\begin{aligned} \Delta M_{AB} &= -M_A - M_B + M_{AB} = \\ &= -\left(\frac{k_A^{\text{int}}}{L} - \sum_{i,j \in A} \frac{k_i k_j}{(2L)^2} \right) - \left(\frac{k_B^{\text{int}}}{L} - \sum_{i,j \in B} \frac{k_i k_j}{(2L)^2} \right) + + \\ &+ \left(\frac{k_A^{\text{int}} + k_A^{\text{int}} + L_{AB}}{L} - \sum_{i,j \in A} \frac{k_i k_j}{(2L)^2} - \sum_{i,j \in B} \frac{k_i k_j}{(2L)^2} - 2 \sum_{i \in A, j \in B} \frac{k_i k_j}{(2L)^2} \right) = \\ &= \frac{L_{AB}}{L} - \frac{k_A k_B}{2L^2} \end{aligned}$$

Resolution limit

- Should we merge two communities?



$$\Delta M_{AB} = \frac{L_{AB}}{L} - \frac{k_A k_B}{2L^2},$$

k_A and k_B total degree in A and B

If $\frac{k_A k_B}{2L} < 1$ and $L_{AB} \geq 1$ $\Rightarrow \Delta M_{AB} > 0$ We merge A and B to maximize modularity.

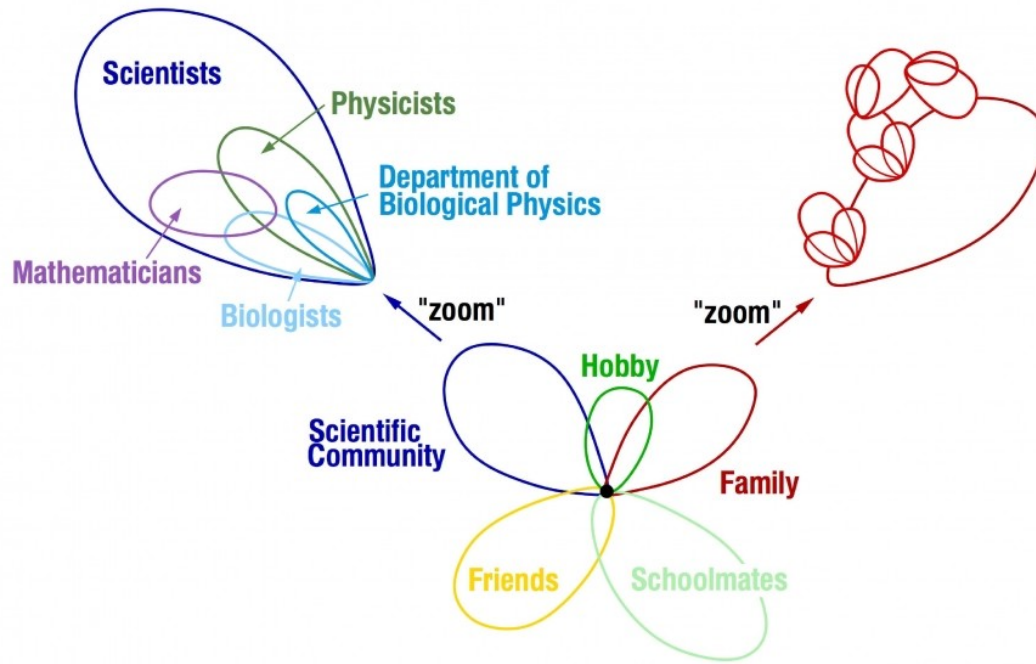
Assuming $k_A \sim k_B = k$ $\Rightarrow k \leq \sqrt{2L}$

Modularity has a resolution limit, as it cannot detect communities smaller than this size.

Even more time:
Link communities



Link communities



Nodes tend to belong to multiple communities

Links tend to be specific, capturing the nature of the relationship between two nodes.

Social networks, a link may indicate:

- they are in the same family; they work together; they share a hobby.

Biological networks:

each interaction of a protein is responsible for a different function, uniquely defining the protein's role in the cell



Define a hierarchical algorithm based on similarity of links

Link communities

1. Define link similarity

$$S(e_{ik}, e_{jk}) = \frac{|n_+(i) \cap n_+(j)|}{|n_+(i) \cup n_+(j)|}$$

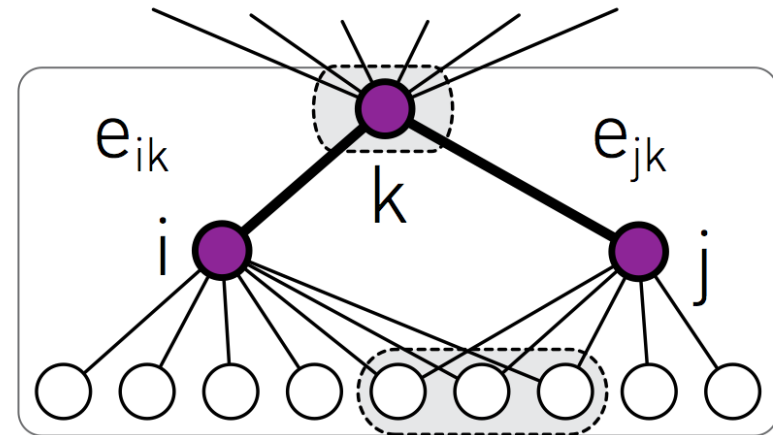
$n_+(i)$: the list of the neighbors of node i , including itself.

S measures the relative number of common neighbors i and j have.

$$\begin{aligned} |n_+(i) \cap n_+(j)| &= 4 \\ |n_+(i) \cup n_+(j)| &= 12 \end{aligned}$$

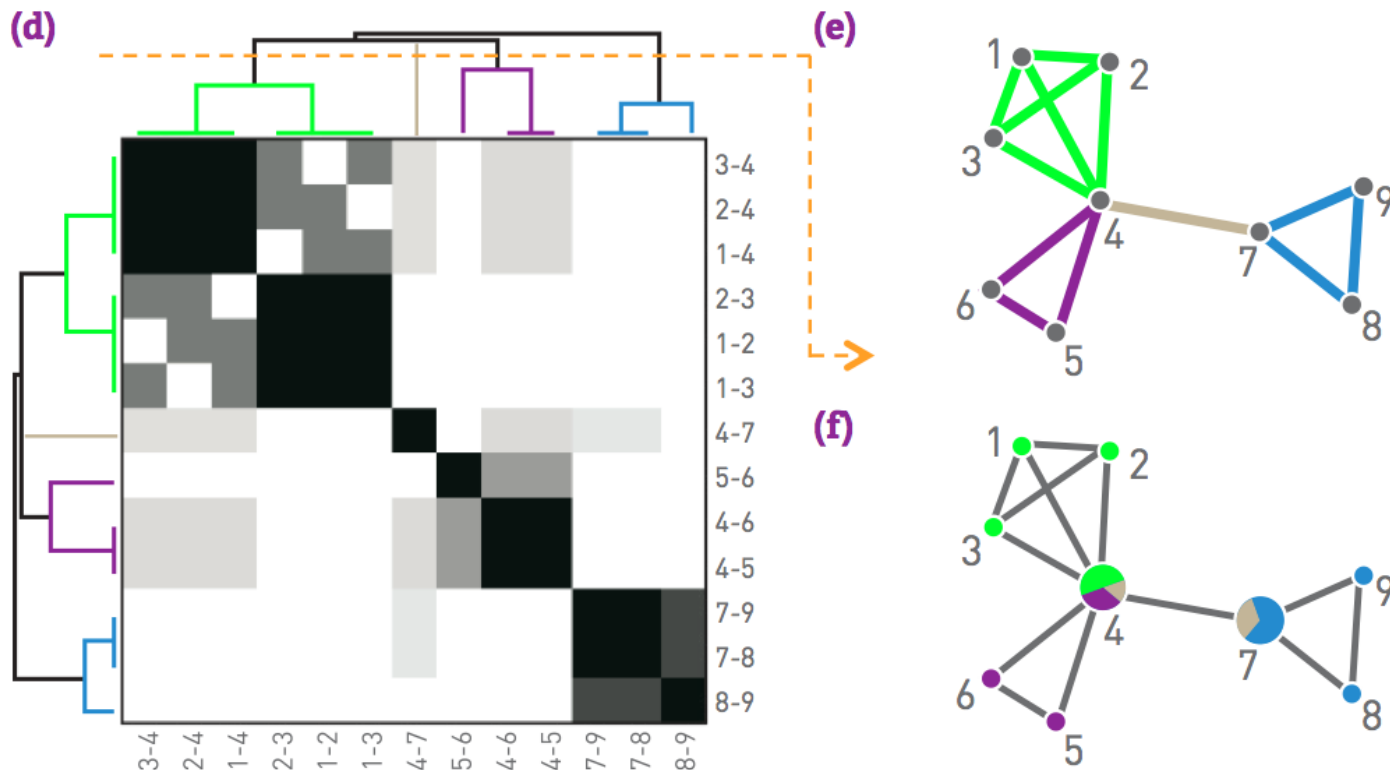
$$S(e_{ik}, e_{jk}) = \frac{1}{3}$$

(a)

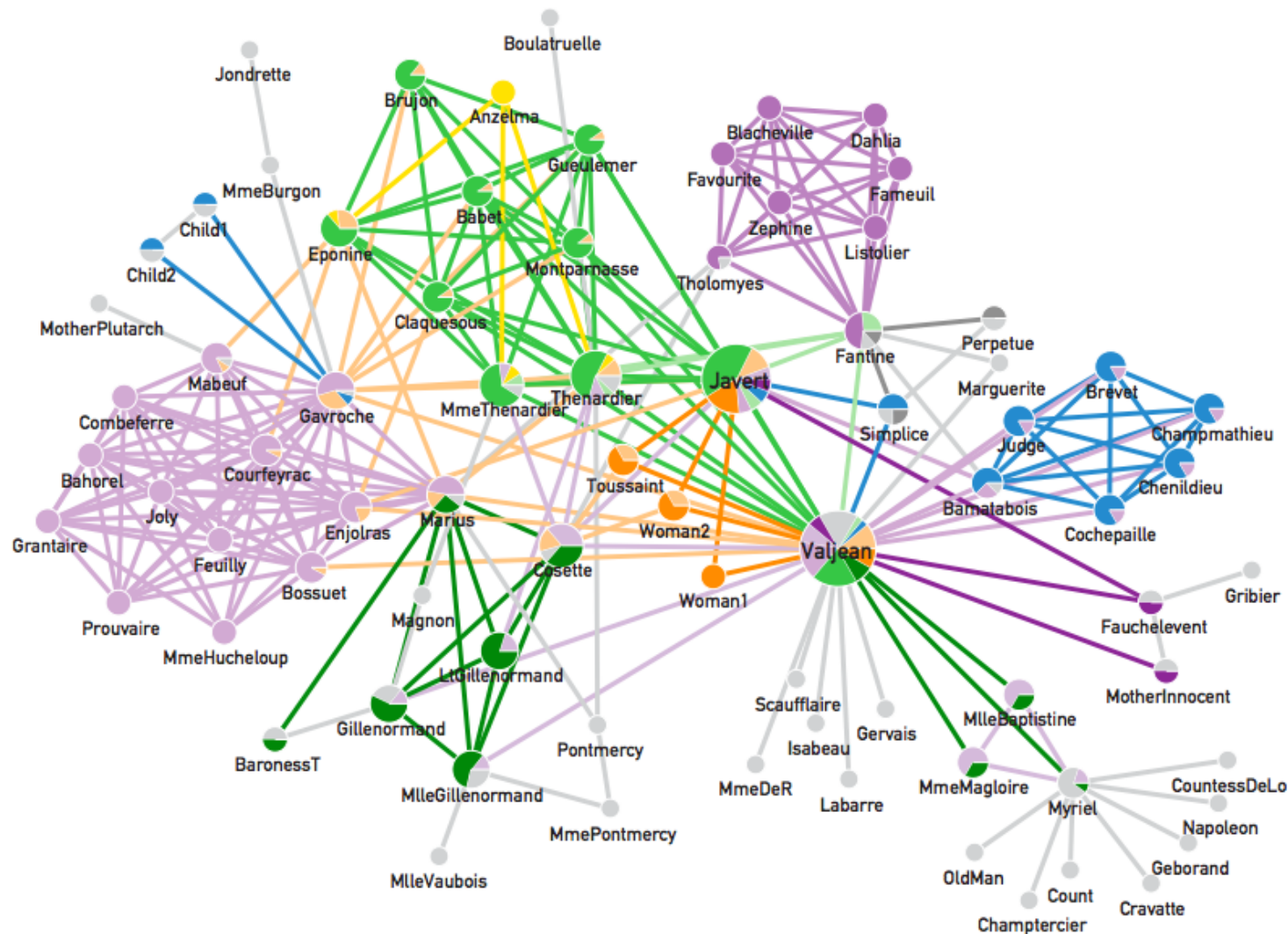


Link communities

2. Apply hierarchical clustering (agglomerative, single linkage)



Link communities



The network of characters in Victor Hugo's 1862 novel *Les Misérables*. Two characters are connected if they interact directly with each other in the story. The link colors indicate the clusters, grey nodes corresponding to single-link clusters. Each node is depicted as a pie-chart, illustrating its membership in multiple communities. Not surprisingly, the main character, Jean Valjean, has the most diverse community membership