Network Science

Communities Part 1

Sean P. Cornelius

With

Emma K. Towlson and Albert-László Barabási

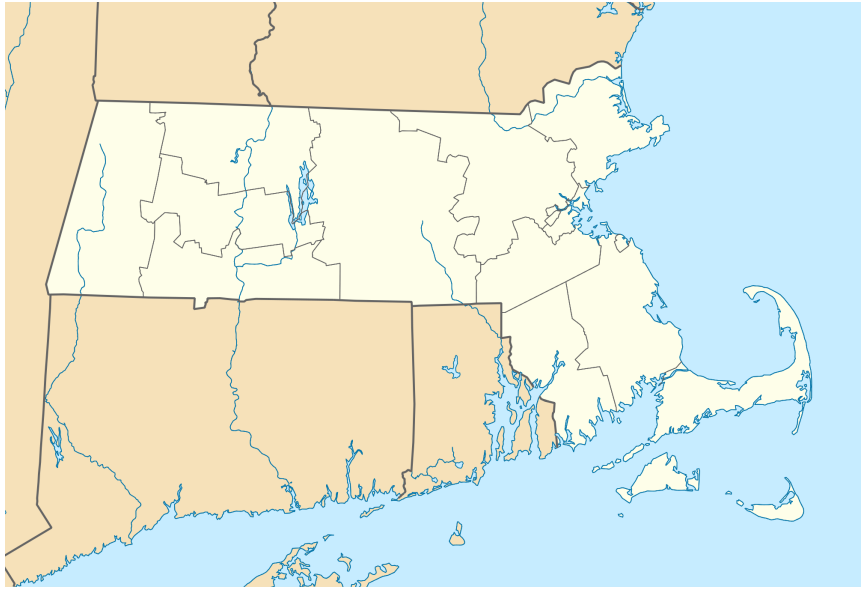
www.BarabasiLab.com

Introduction

Section 1

Introduction: Belgium

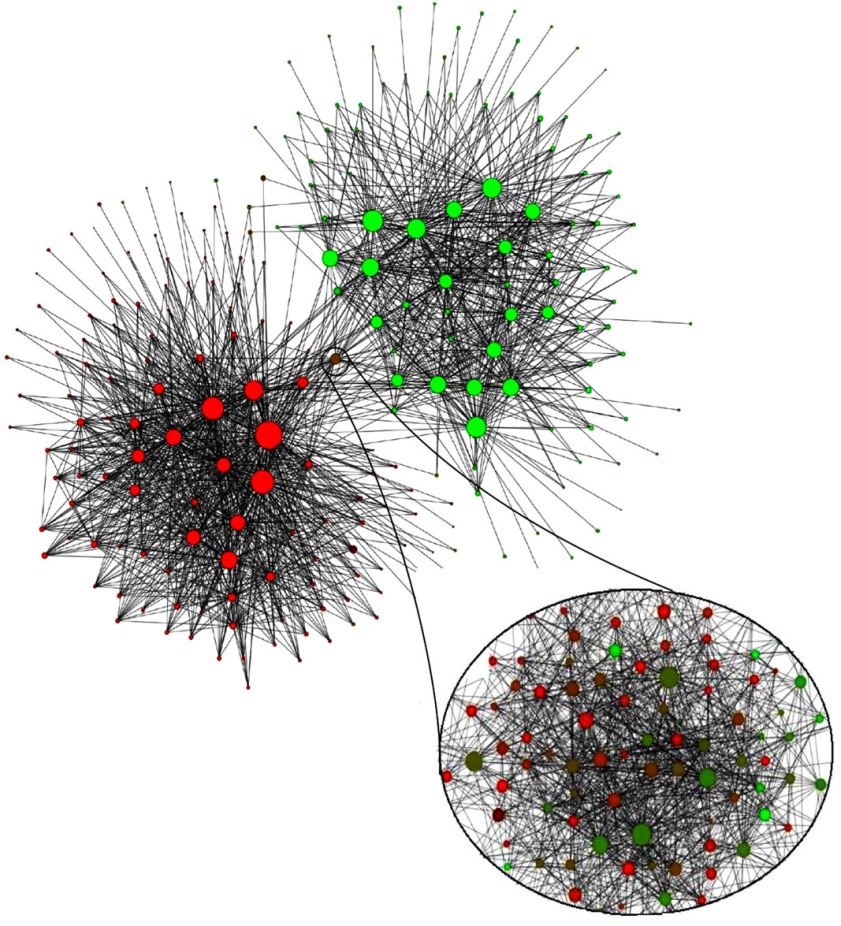
Section 1 Introduction: Belgium



Same area as Massachusetts (~12,000 sq miles)
Same population as Ohio (~11.5 millions)

Section 1

Introduction: Belgium



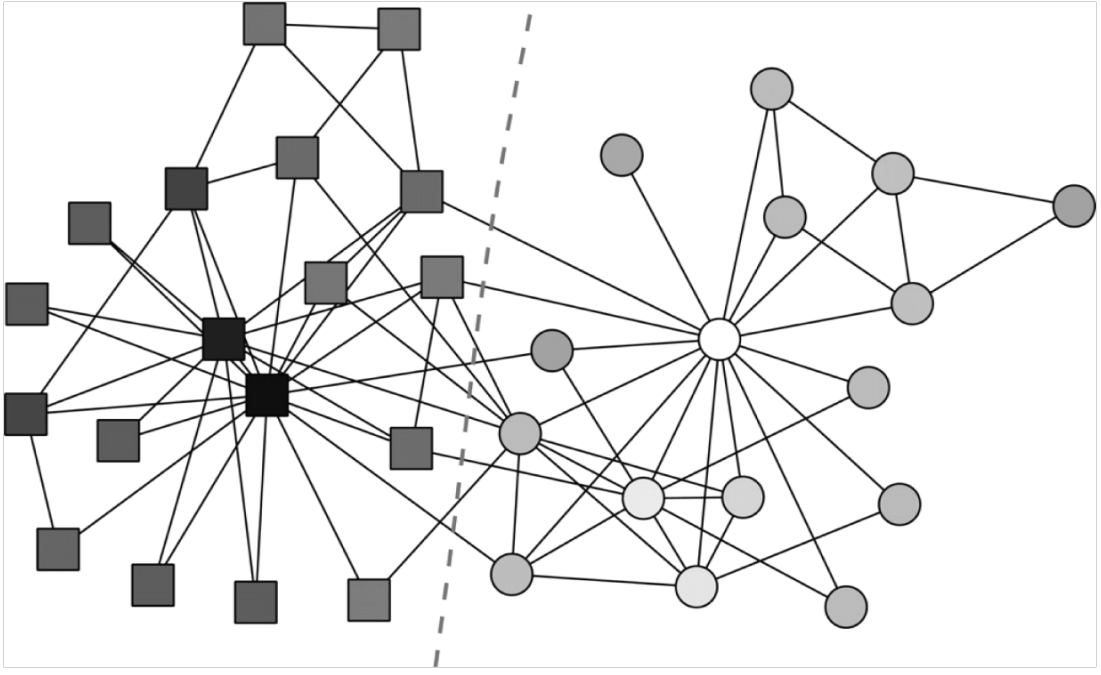
V.D. Blondel et al, *J. Stat. Mech.* P10008 (2008).

A.-L. Barabási, *Network Science: Communities.*

Examples of communities

Section 2

Zachary's Karate Club

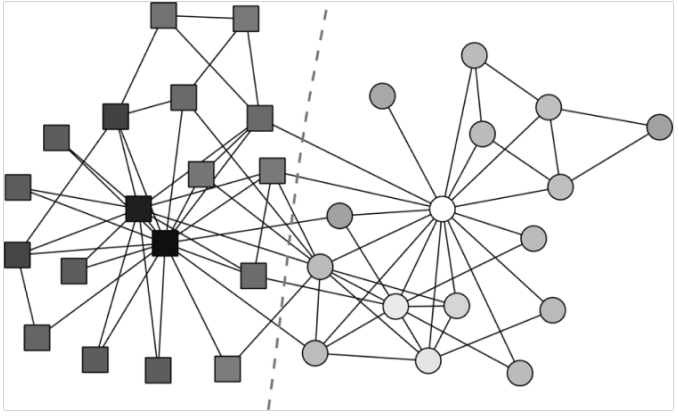


W.W. Zachary, *J. Anthropol. Res.* 33:452-473 (1977).

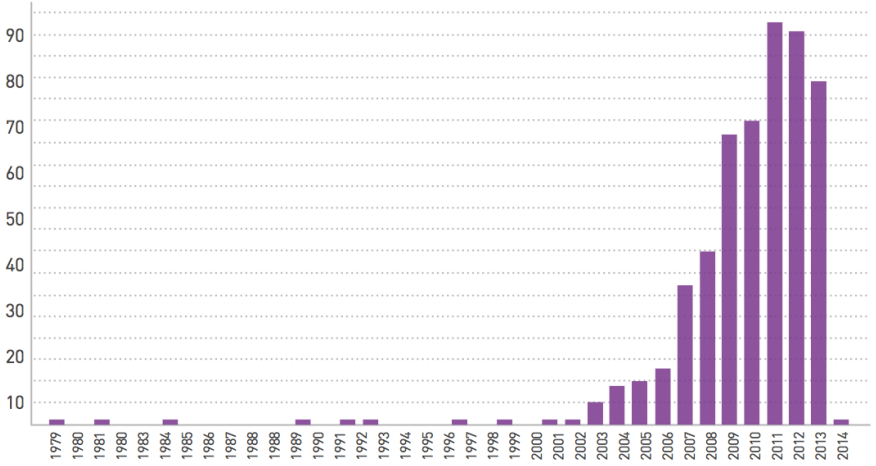
A.-L. Barabási, *Network Science: Communities.*

Section 2

Zachary's Karate Club



Citation history
of the Zachary's Karate club paper



W.W. Zachary, *J. Anthropol. Res.* 33:452-473 (1977).

A.-L. Barabási, *Network Science: Communities.*

Section 2

Zachary Karate Club Club

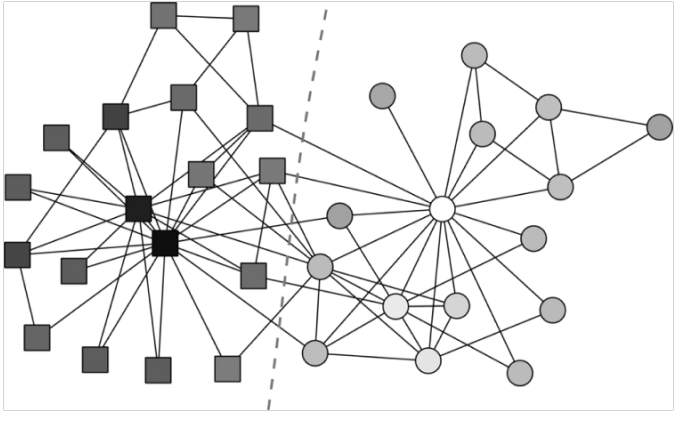
The first scientist at any conference on networks who uses Zachary's karate club as an example is inducted into the Zachary Karate Club Club, and awarded a prize.

Chris Moore (9 May 2013).
Mason Porter (NetSci, June 2013).
Yong-Year Ahn (Oxford University, July 2013)
Marián Boguñá (ECCS, September 2013).
Mark Newman (Netsci, June 2014)

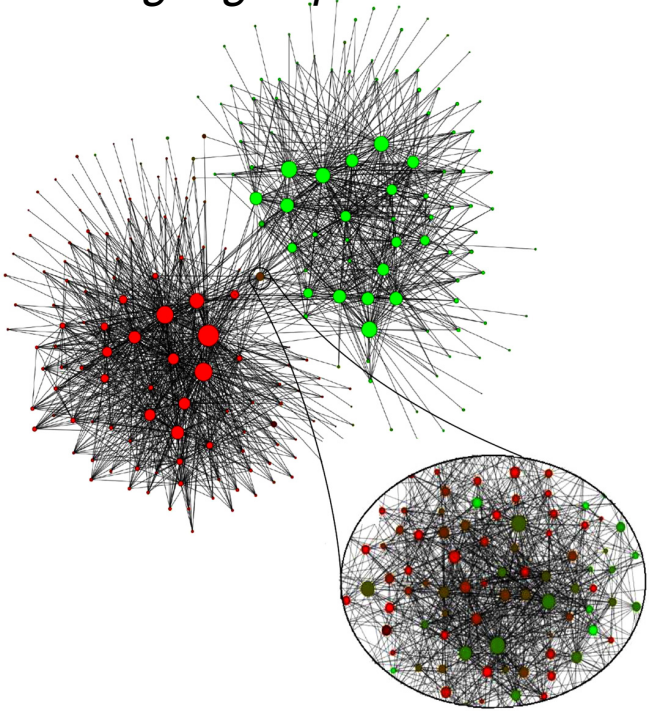
http://networkkarate.tumblr.com/)

Section 2 Auxiliary information

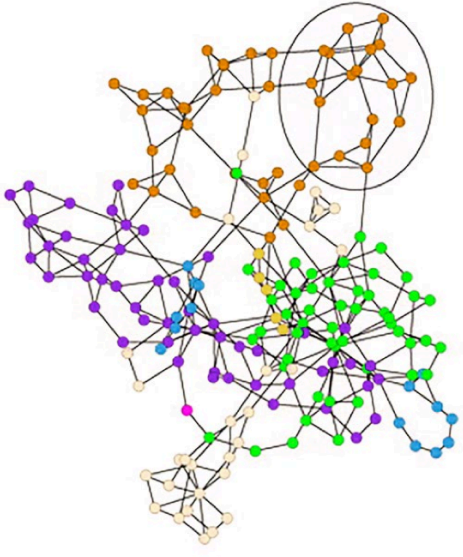
→ *Karate Club:*
Breakup of the club

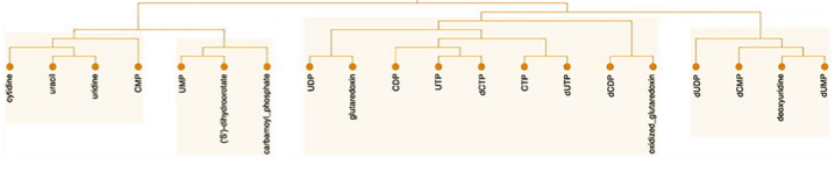


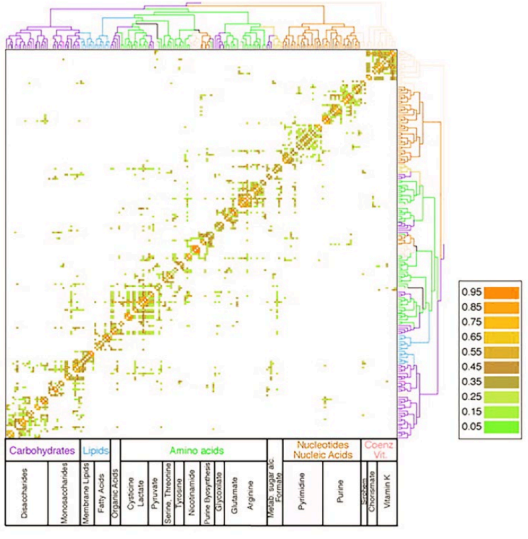
→ *Belgian Phone Data:*
Language spoken

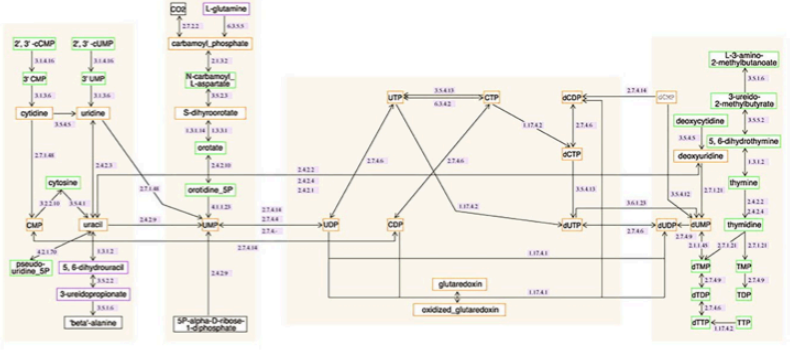


Section 2 Biological Modules





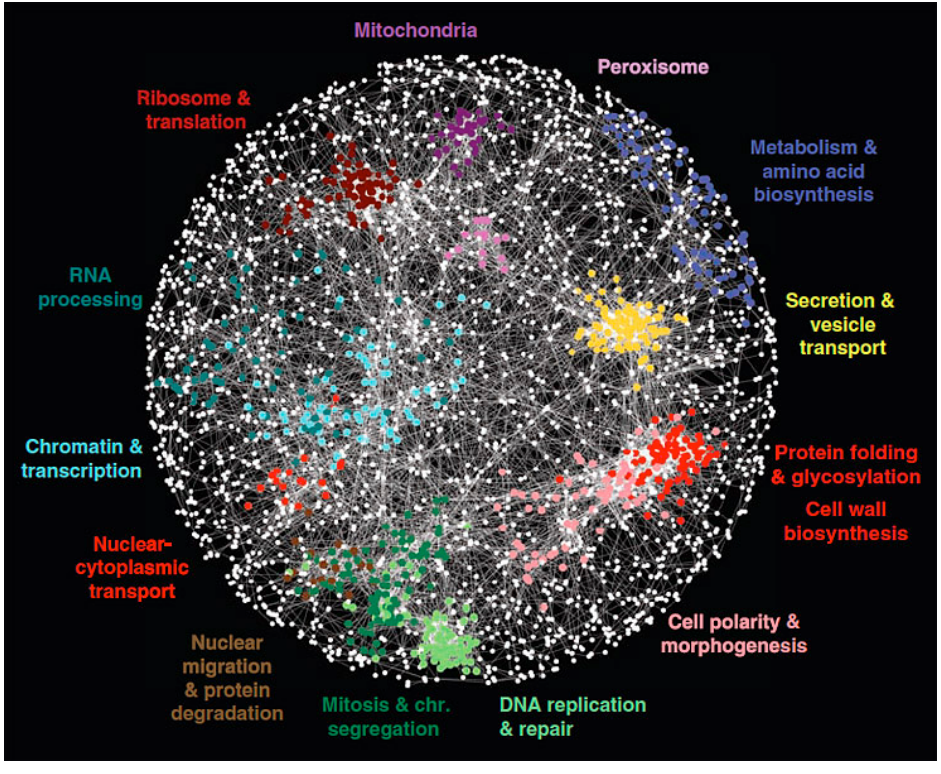




E. Ravasz et al., *Science* 297 (2002).

A.-L. Barabási, *Network Science: Communities.*

Section 2 Biological Modules

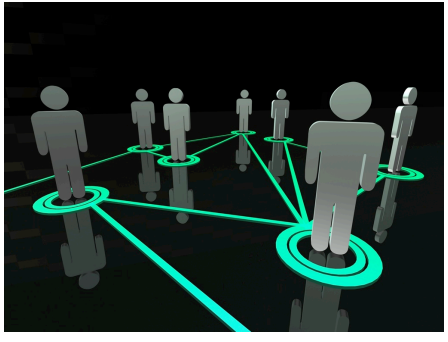


M. Costanzo et al, *Science,* 22 (2010).

A.-L. Barabási, *Network Science: Communities.*

Basics of communities

We focus on the mesoscopic scale of the network



Microscopic Mesoscopic Macroscopic

H1: A network's community structure is uniquely encoded in its wiring diagram

H2: *Connectedness Hypothesis*

A community corresponds to a connected subgraph.

H3: *Density Hypothesis*

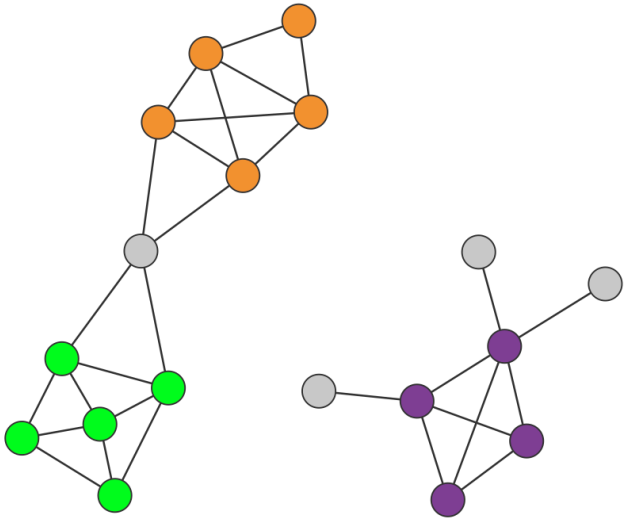
Communities correspond to locally dense neighborhoods of a network.

H2: *Connectedness Hypothesis*

A community corresponds to a connected subgraph.

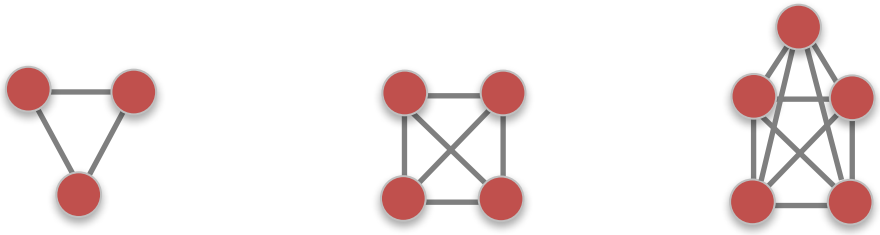
H3: *Density Hypothesis*

Communities correspond to locally dense neighborhoods of a network.



Cliques as communities

A clique is a complete subgraph of *k*-nodes

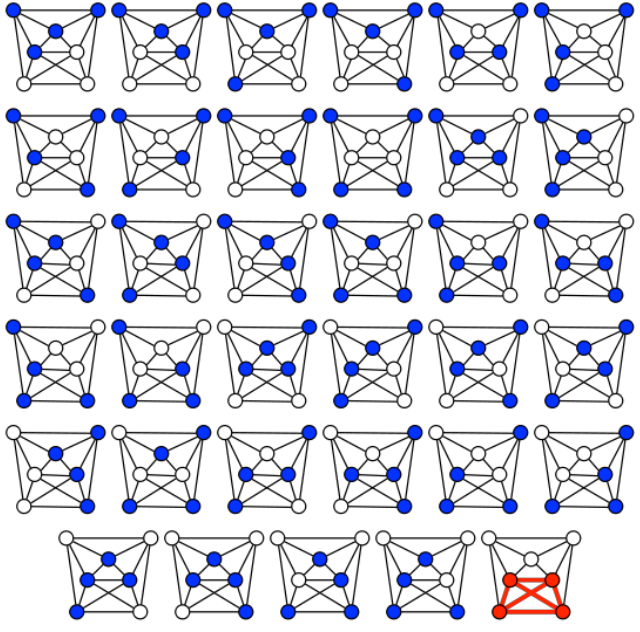


R.D. Luce & A.D. Perry, *Psychometrika* 14 (1949)

A.-L. Barabási, *Network Science: Communities.*

Cliques as communities

- Triangles are frequent; larger cliques are rare.
- Communities do not necessarily correspond to complete subgraphs, as many of their nodes do not link directly to each other.
- Finding the cliques of a network is computationally rather demanding, being a so-called NP-complete problem.



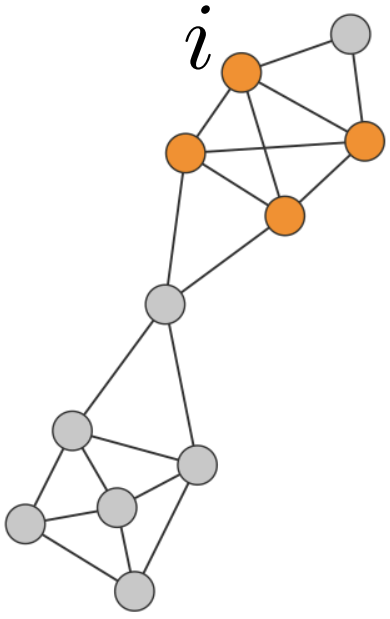
Strong and weak communities

$$k_i^{int} = 3$$
$$k_i^{ext} = 1$$

Consider a connected subgraph *C* of N_c nodes

Internal degree, k_i^{int} *:* set of links of node *i* that connects to other nodes of the same community *C*.

External degree k_i^{ext}*:* the set of links of node *i* that connects to the rest of the network.



If $k_i^{ext}=0$*:* all neighbors of *i* belong to C, and *C* is a good community for *i*.

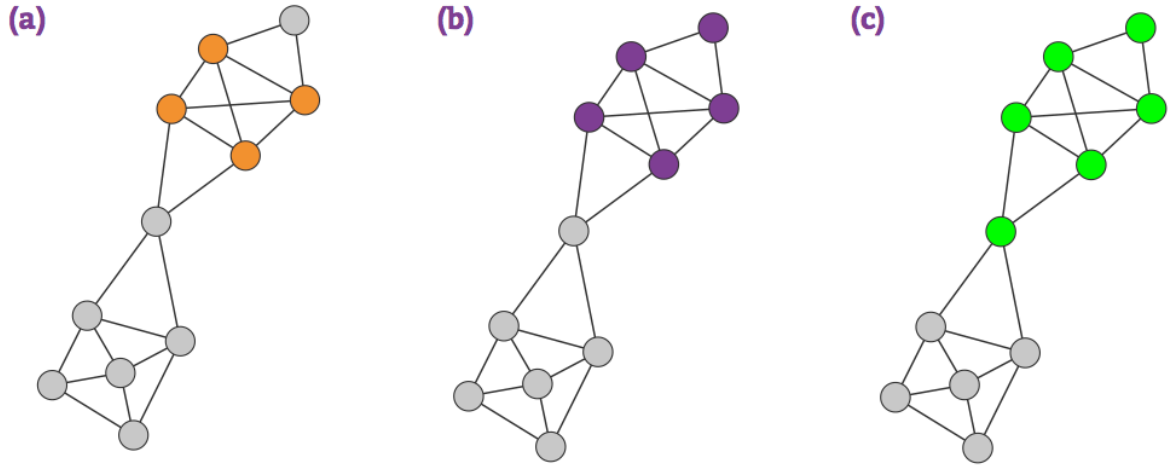
If $k_i^{int}=0$, all neighbors of *i* belong to other communities, then *i* should be assigned to a different community.

Strong community:
Each node of *C* has more links within the community than with the rest of the graph.

$$k_i^{\text{int}}(C) > k_i^{\text{ext}}(C)$$

Weak community:
The total internal degree of *C* exceeds its total external degree,

$$\sum_{i \in C} k_i^{\text{in}}(C) > \sum_{i \in C} k_i^{\text{out}}(C)$$



Clique *Strong* *Weak*

A.-L. Barabási, *Network Science: Communities.*

How many ways can we partition a network into 2 communities?

Graph bisection

Divide a network into two equal non-overlapping subgraphs, such that the number of links between the nodes in the two groups is minimized.

Two subgroups of size n_1 and n_2. Total number of combinations: $\frac{N!}{n_1!n_2!}$

$$n! \simeq \sqrt{2\pi n}(n/e)^n$$

$$\frac{N!}{n_1!n_2!} \simeq \frac{\sqrt{2\pi N}(N/e)^n}{\sqrt{2\pi n_1}(n_1/e)^{n_1}\sqrt{2\pi n_2}(n_2/e)^{n_2}} \simeq \frac{N^{N+1/2}}{n_1^{n_1+1/2}n_2^{n_2+1/2}}$$

$$n_1 = n_2 = N/2$$

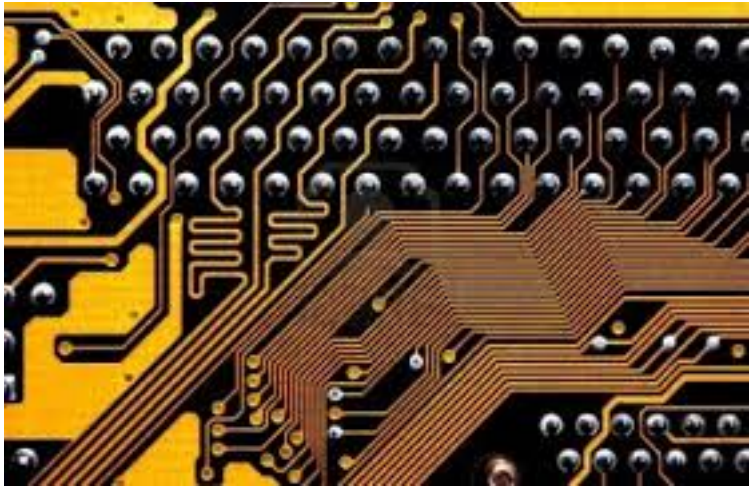
$$\frac{2^{N+1}}{\sqrt{N}} = e^{(N+1)\ln 2/\sqrt{N}}$$

N=10 → 256 partitions (1 ms)

N=100→ 10^{26} partitions (10^{21} years)

Graph Partitioning

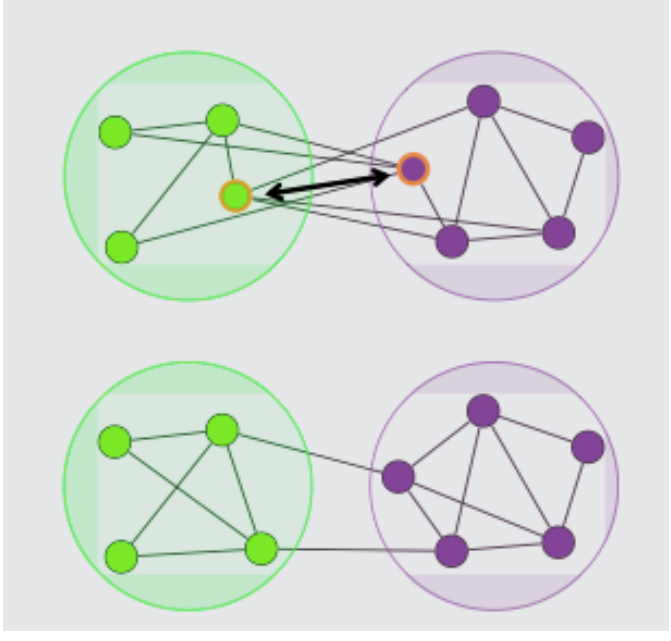
partition the full wiring diagram of an integrated circuit into smaller subgraphs, so that they minimize the number of connections between them.



2.5 billion transistors

Kerninghan-Lin Algorithm for graph bisection

- Partition a network into two groups of predefined size. This partition is called **cut**.
- Inspect each a pair of nodes, one from each group. Identify the pair that results in the largest reduction of the cut size (links between the two groups) if we swap them
- Swap them.
- If no pair deduces the cut size, we swap the pair that increases the cut size the least.
- The process is repeated until each node is moved once.



Community detection

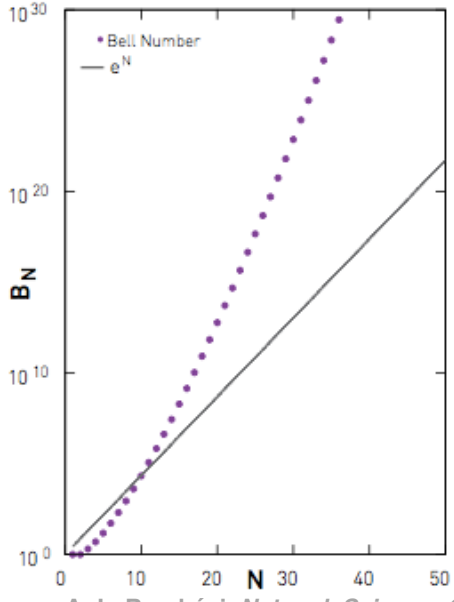
The *number* and *size* of the communities are unknown at the beginning.

Partition

Division of a network into groups of nodes, so that each node belongs to one group.

$$B_N = \frac{1}{e}\sum_{j=0}^{\infty}\frac{j^N}{j!} \quad .$$

Bell Number: number of possible partitions of *N* nodes



A.-L. Barabási, *Network Science: Communities.*

Hierarchical Clustering

Section 4 Hierarchical Clustering

1. Build a similarity matrix for the network

2. *Similarity matrix*: how similar two nodes are to each other → we need to determine from the adjacency matrix

3. Hierarchical clustering iteratively identifies groups of nodes with high similarity, following one of two distinct strategies:

   Agglomerative algorithms merge nodes and communities with high similarity.

   Divisive algorithms split communities by removing links that connect nodes with low similarity.

4. *Hierarchical tree* or *dendrogram*: visualize the history of the merging or splitting process the algorithm follows. Horizontal cuts of this tree offer various community partitions.

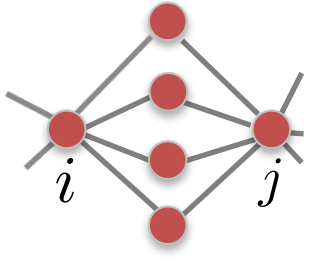
Agglomerative algorithms merge nodes and communities with high similarity.

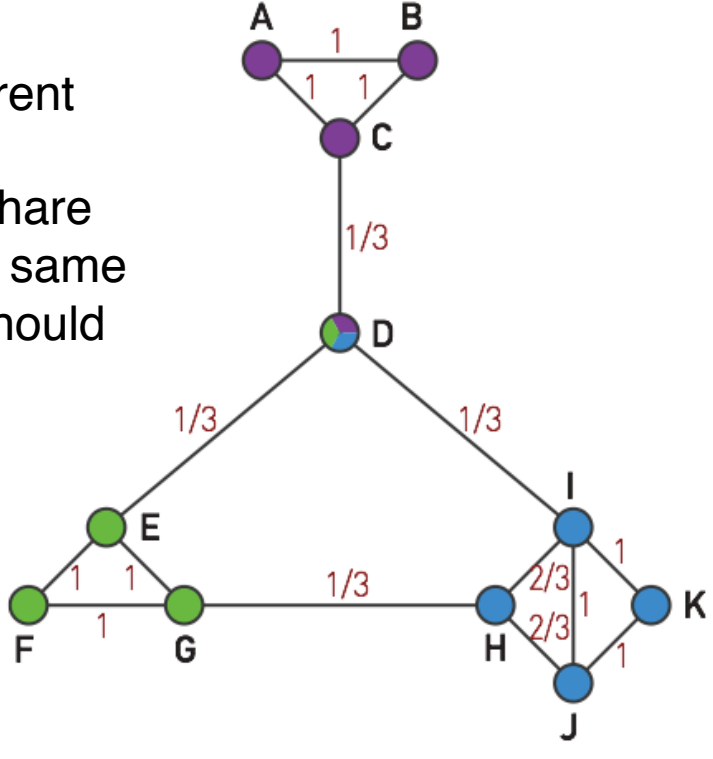
Step 1: Define the Similarity Matrix (Ravasz algorithm)

- High for node pairs that likely belong to the same community, low for those that likely belong to different communities.
- Nodes that connect directly to each other and/or share multiple neighbors are more likely to belong to the same dense local neighborhood, hence their similarity should be large.

Topological overlap matrix: $x_{ij}^{o} = \frac{J_N(i,j)}{\min(k_i, k_j)}$

$J_N(i,j)$: number of common neighbors of node i and j; (+1) if there is a direct link between i and j;



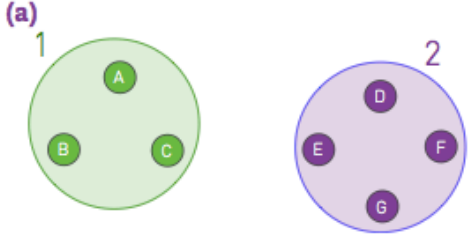


E. Ravasz et al., *Science* 297 (2002).

A.-L. Barabási, *Network Science: Communities.*

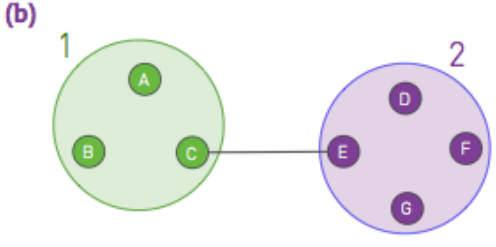
Step 2: Decide Group Similarity

- Groups are merged based on their mutual similarity through *single*, *complete* or *average cluster linkage*

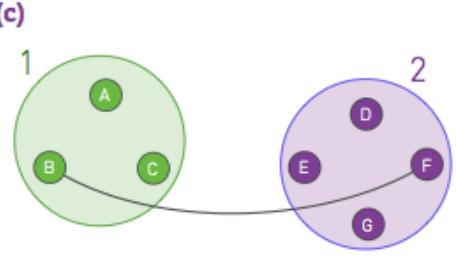


$$\frac{1}{x_{ij}} = r_{ij} =$$

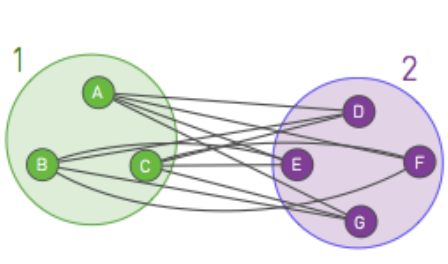
	D	E	F	G
A	2.75	2.22	3.46	3.08
B	3.38	2.68	3.97	3.40
C	2.31	1.59	2.88	2.34



Single Linkage: $r_{12} = 1.59$



Complete Linkage: $r_{12} = 3.97$



Average Linkage: $r_{12} = 2.84$

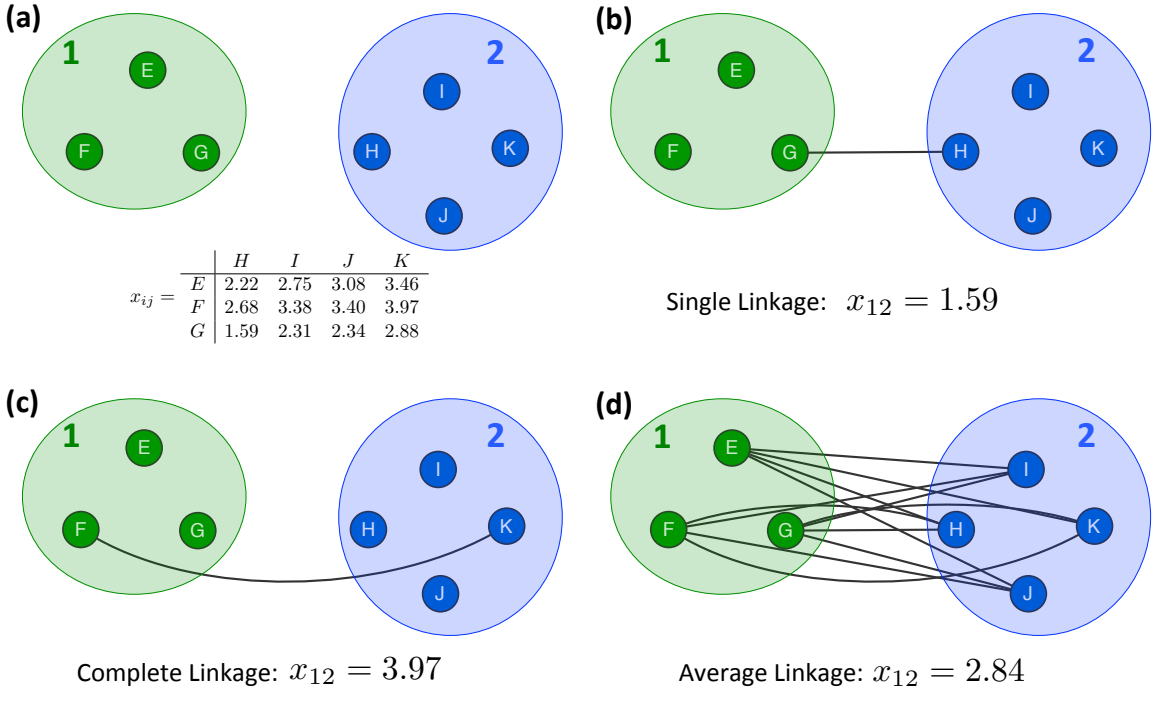
E. Ravasz et al., *Science* 297 (2002).

A.-L. Barabási, *Network Science: Communities.*

Step 2: Decide Group Similarity.

- groups are merged based on their mutual similarity:

single, *complete* and *average cluster similarity*



$x_{ij} =$

	H	I	J	K
E	2.22	2.75	3.08	3.46
F	2.68	3.38	3.40	3.97
G	1.59	2.31	2.34	2.88

Step 3: Apply Hierarchical Clustering

- Assign each node to a community of its own and evaluate the similarity for all node pairs. The initial similarities between these “communities” are simply the node similarities.
- Find the community pair with the highest similarity and merge them to form a single community.
- Calculate the similarity between the new community and all other communities.
- Repeat from Step 2 until all nodes are merged into a single community.

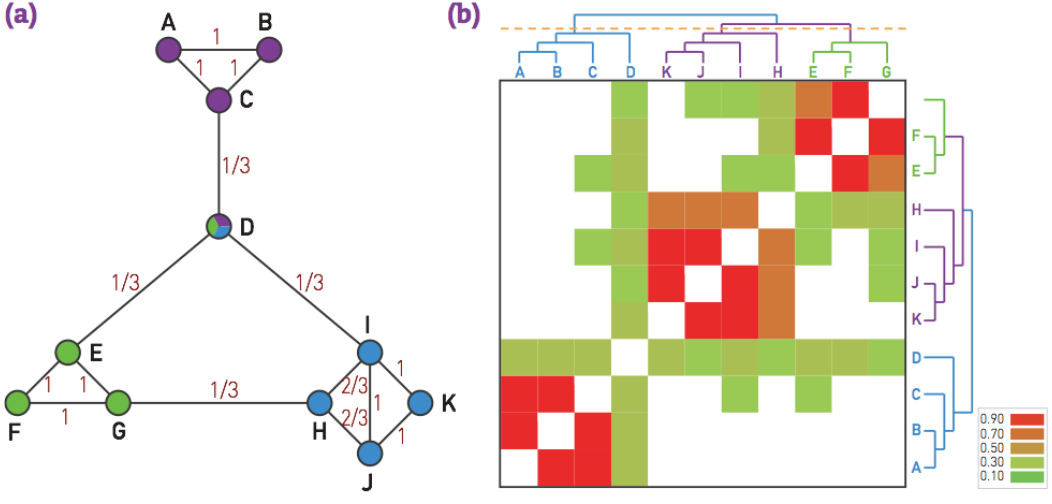
Step 4: Build Dendrogram

- Describes the precise order in which the nodes are assigned to communities.

E. Ravasz et al., *Science* 297 (2002).

A.-L. Barabási, *Network Science: Communities*.

Agglomerative Algorithms



Computational complexity:

- Step 1 (calculation similarity matrix): $O\left(N^2\right)$
- Step 2-3 (group similarity): $O\left(N^2\right)$
- Step 4 (dendrogram): $O\left(N \log N\right)$

$O\left(N^2\right)$

E. Ravasz et al., *Science* 297 (2002).

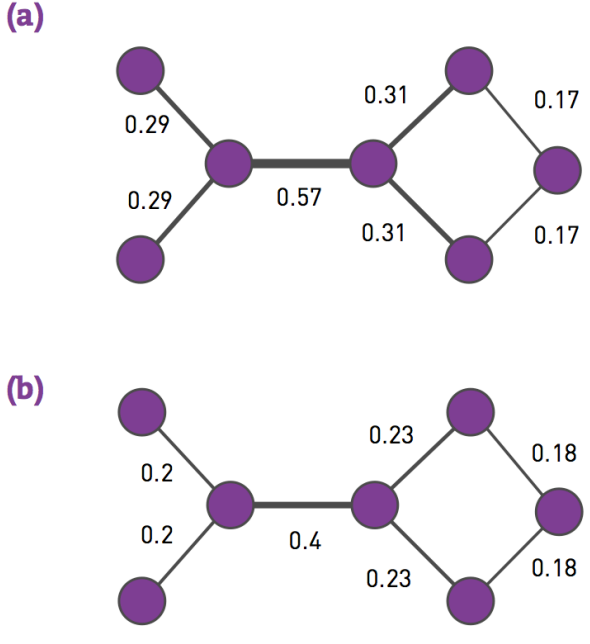
A.-L. Barabási, *Network Science: Communities.*

Section 4 Divisive Algorithms

Divisive algorithms split communities by removing links that connect nodes with low similarity.

Step 1: Define a Centrality Measure (Girvan-Newman algorithm)

- *Link betweenness* is the number of shortest paths between all node pairs that run along a link.
- *Random-walk betweenness*. A pair of nodes m and n are chosen at random. A walker starts at m, following each adjacent link with equal probability until it reaches n. Random walk betweenness x_{ij} is the probability that the link $i \rightarrow j$ was crossed by the walker after averaging over all possible choices for the starting nodes m and n

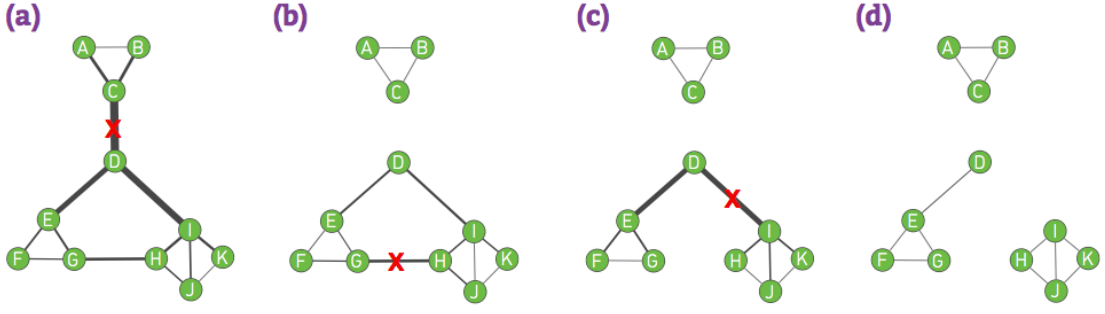


M. Girvan & M.E.J. Newman, *PNAS* 99 (2002).

A.-L. Barabási, *Network Science: Communities.*

Step 2: Hierarchical Clustering

a) Compute of the centrality of each link.
b) Remove the link with the largest centrality; in case of a tie, choose one randomly.
c) Recalculate the centrality of each link for the altered network.
d) Repeat until all links are removed (yields a dendrogram).

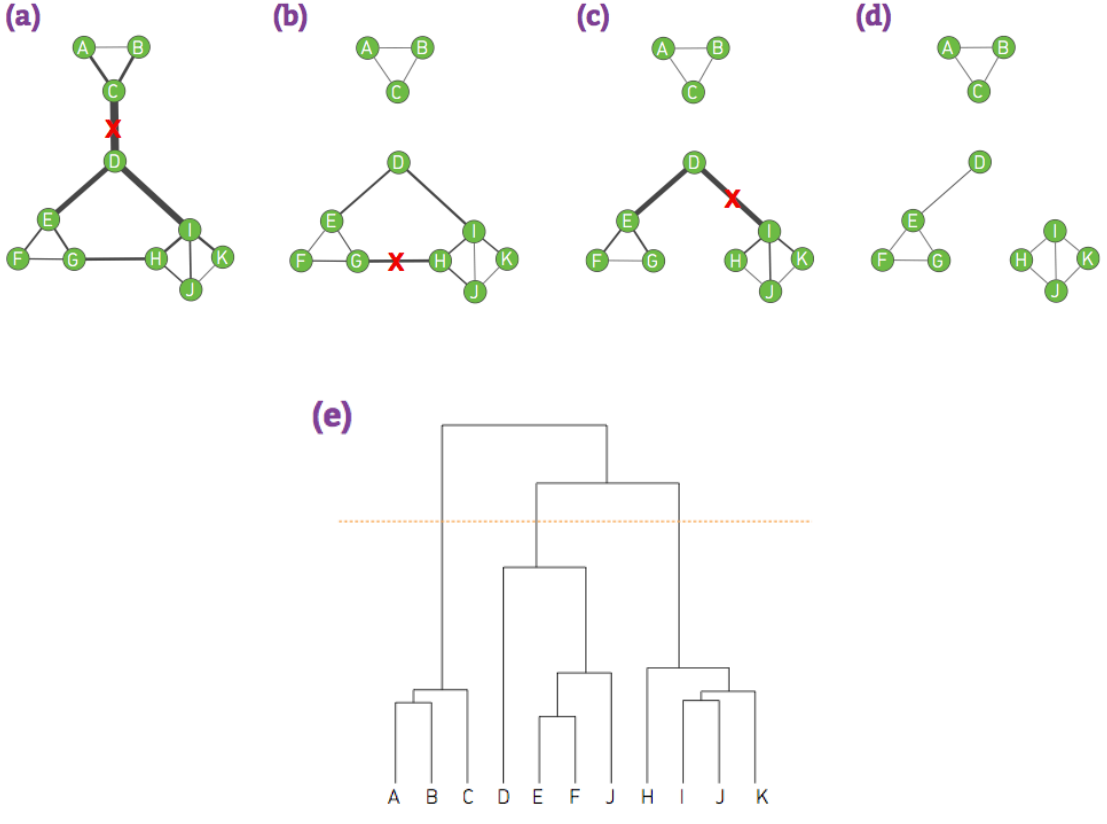


M. Girvan & M.E.J. Newman, *PNAS* 99 (2002).

A.-L. Barabási, *Network Science: Communities.*

Step 2: Hierarchical Clustering

a) Compute of the centrality of each link.
b) Remove the link with the largest centrality; in case of a tie, choose one randomly.
c) Recalculate the centrality of each link for the altered network.
d) Repeat until all links are removed (yields a dendrogram).

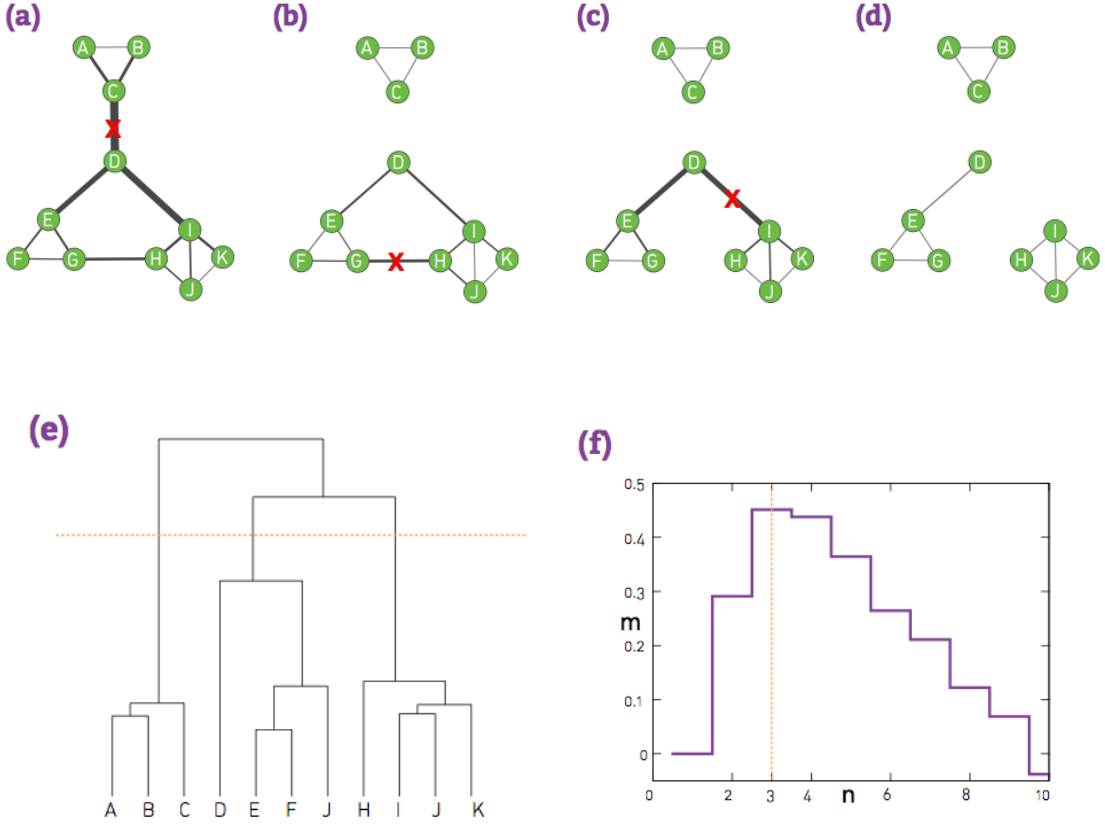


M. Girvan & M.E.J. Newman, *PNAS* 99 (2002).

A.-L. Barabási, *Network Science: Communities.*

Step 2: Hierarchical Clustering

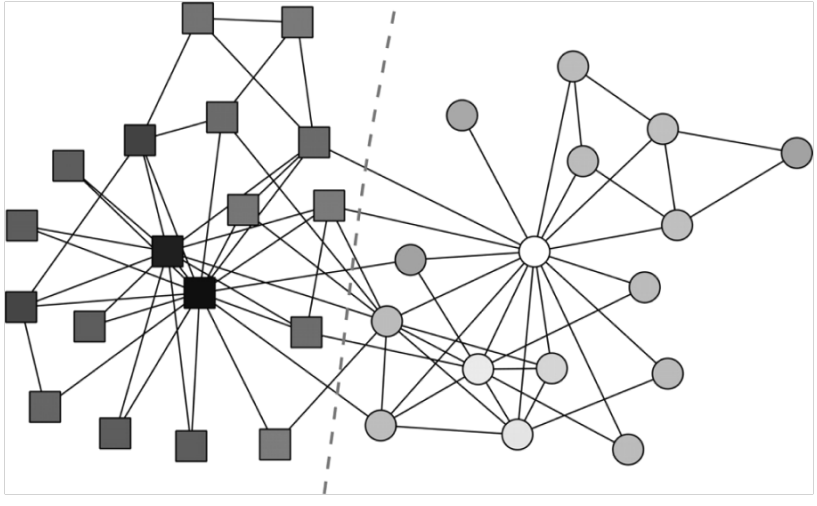
a) Compute of the centrality of each link.
b) Remove the link with the largest centrality; in case of a tie, choose one randomly.
c) Recalculate the centrality of each link for the altered network.
d) Repeat until all links are removed (yields a dendrogram).



M. Girvan & M.E.J. Newman, *PNAS* 99 (2002).

A.-L. Barabási, *Network Science: Communities.*

Divisive Algorithm



Computational complexity:

- Step 1a (calculation betweenness centrality): $O\left(N^2\right)$
- Step 1b (Recalculation of betweenness centrality for all links): $O\left(LN^2\right)$

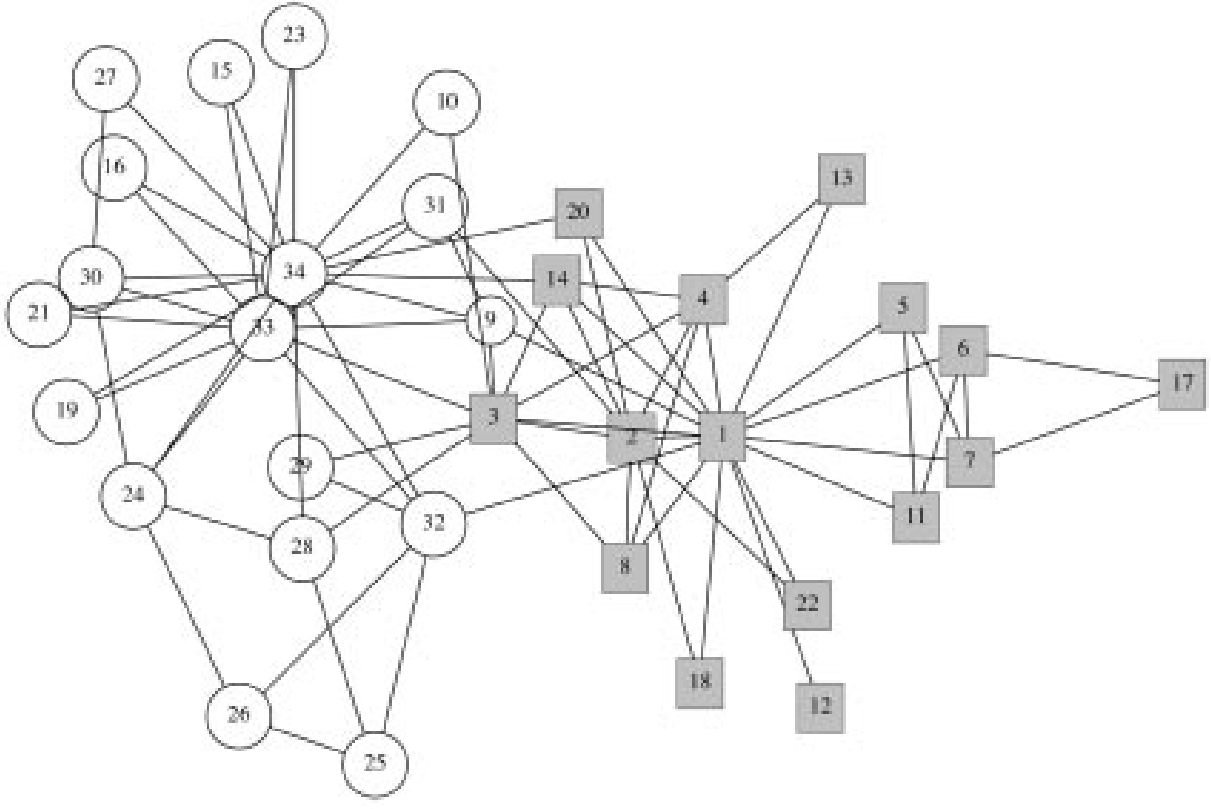
for sparse networks → $O\left(N^3\right)$

M. Girvan & M.E.J. Newman, *PNAS* 99 (2002).

A.-L. Barabási, *Network Science: Communities.*

Section 4

Divisive Algorithm

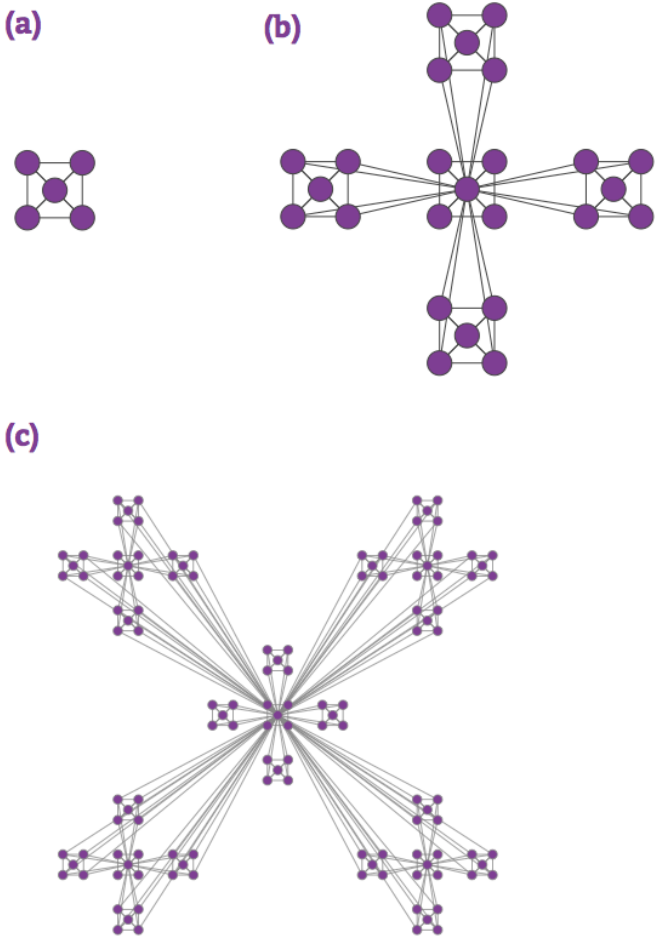


i. Nested Communities

It assumes that communities are organized in a hierarchical fashion, i.e. small modules are nested into larger ones. This hierarchical nesting is captured by the dendrogram (Figures 9.12a and 9.15e). How do we know, however, if such hierarchy is indeed present in a network? Could this hierarchy be imposed by the algorithm, whether or not the underlying network has a nested community structure?

ii. Communities and the Scale-Free Property

The density hypothesis states that a network can be partitioned into a collection of subgraphs that are only weakly linked to other subgraphs. How can we have somewhat isolated communities in a scale-free network, whose hubs inevitably connect to nodes that can belong to different communities?

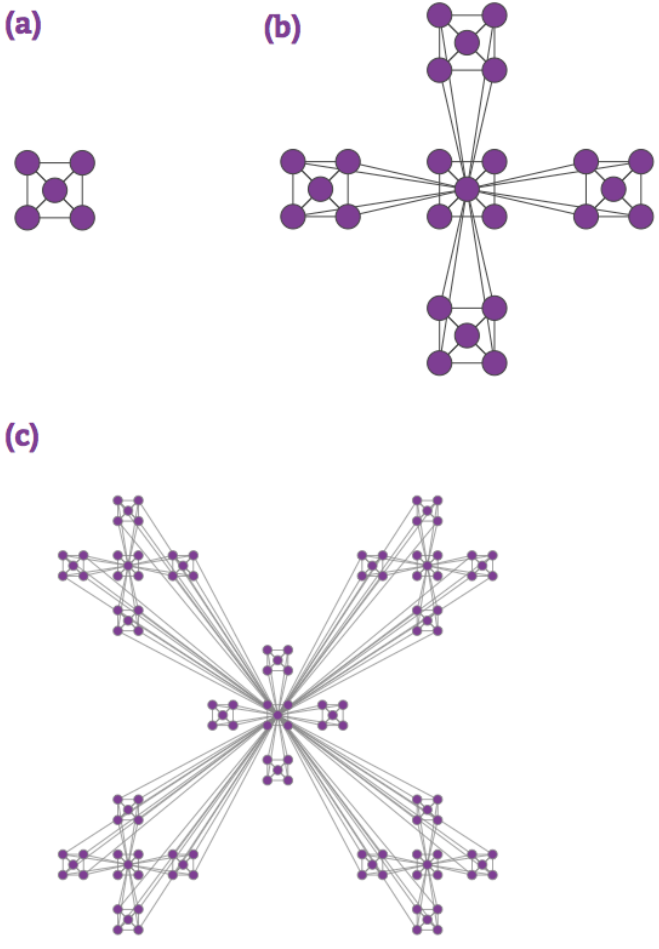


(1) Scale-free property

The obtained network is scale-free, its degree distribution following a power-law with $\gamma = 1 + \frac{\ln 5}{\ln 4} \simeq 2.16$

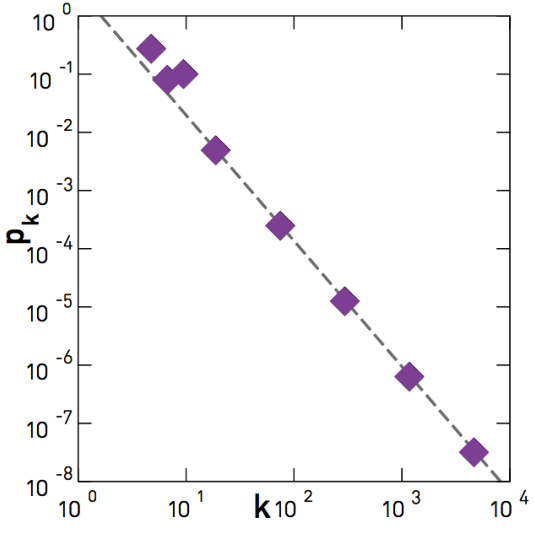
E. Ravasz & A.-L. Barabási, *PRE* 67 (2003).

A.-L. Barabási, *Network Science: Communities.*



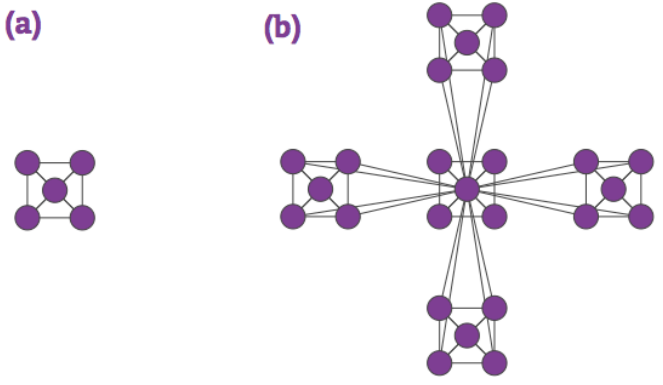
(1) Scale-free property

The obtained network is scale-free, its degree distribution following a power-law with $\gamma = 1 + \frac{\ln 5}{\ln 4} \simeq 2.16$



E. Ravasz & A.-L. Barabási, *PRE* 67 (2003).

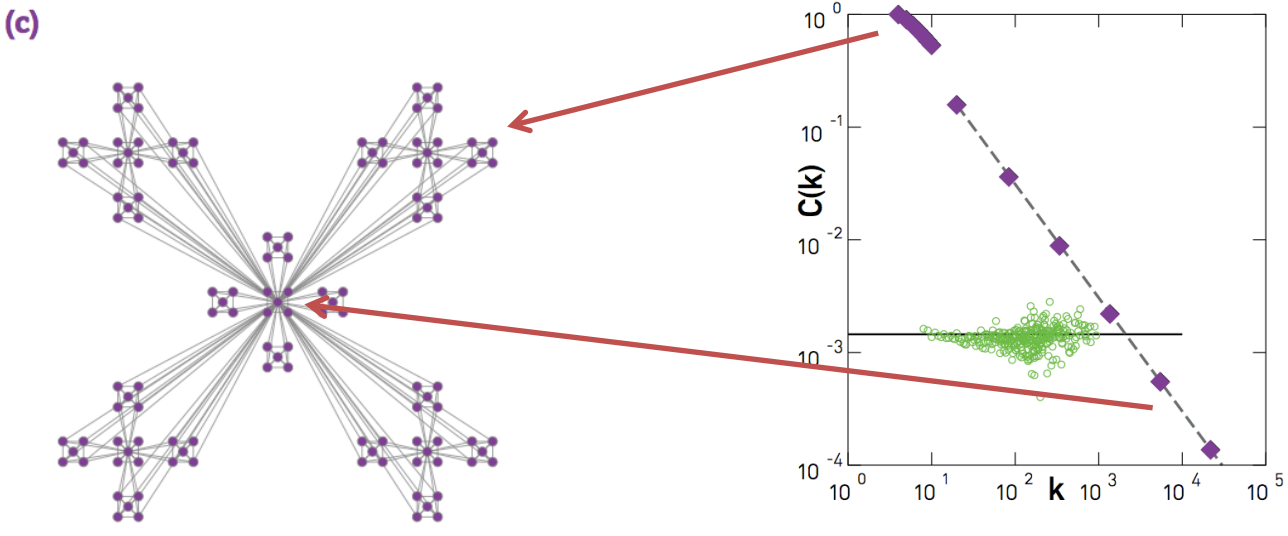
A.-L. Barabási, *Network Science: Communities.*



(2) Clustering coefficient scaling with k

$$C(k) = \frac{\#\text{ between } k \text{ neighbors}}{k(k-1)/2} =$$

$$C(k) \sim k^{-1}$$

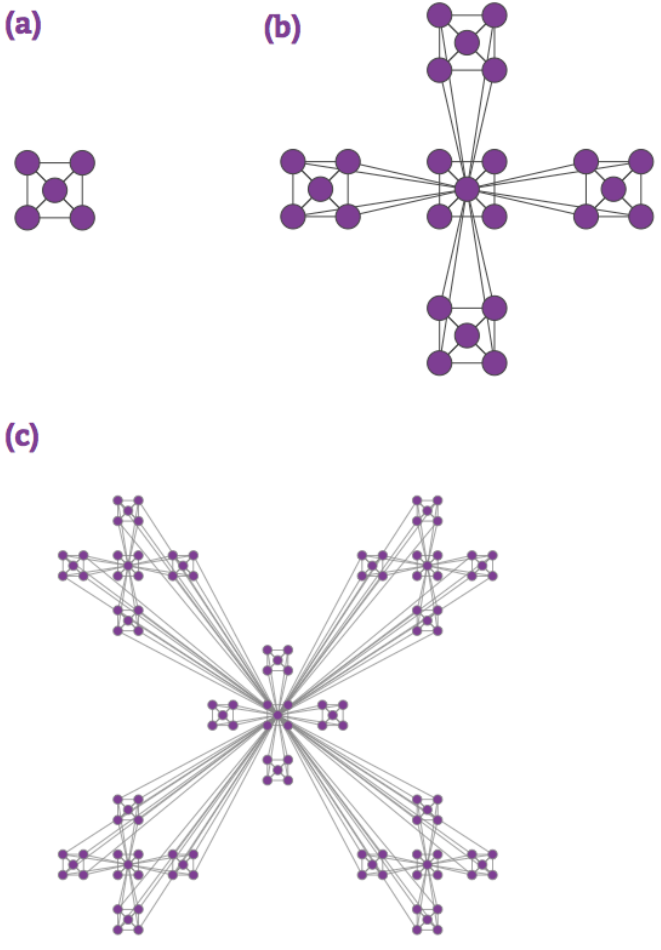


Small *k* nodes:
*high clustering coefficient;
*their neighbors tend to link to each other in highly interlinked, compact communities.

High *k* nodes (hubs):
*small clustering coefficient;
*connect independent communities.

E. Ravasz & A.-L. Barabási, *PRE* 67 (2003).

A.-L. Barabási, *Network Science: Communities.*

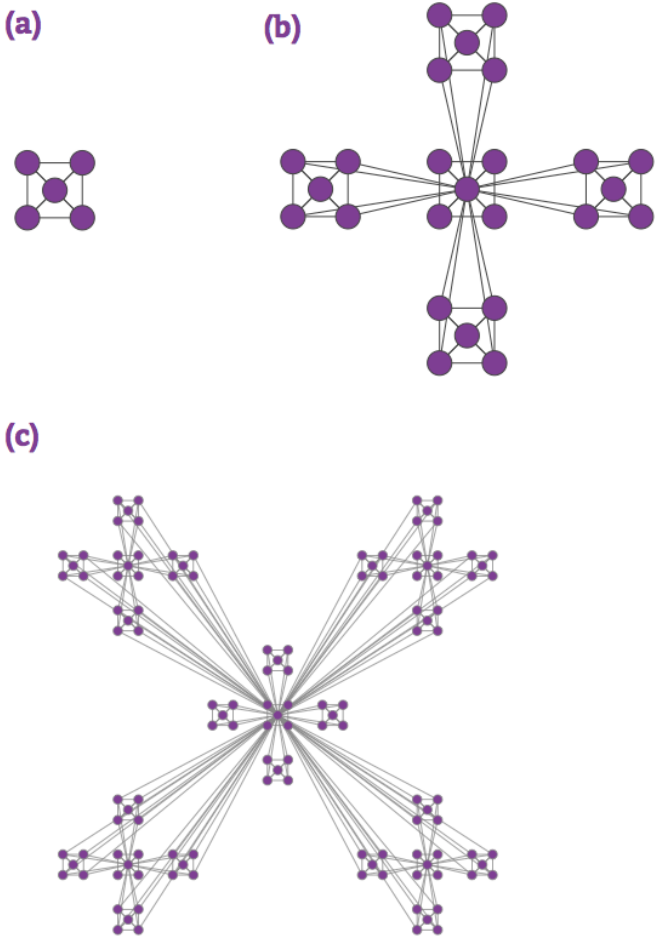


(3) Clustering coefficient independent of N

$$C = 0.743$$

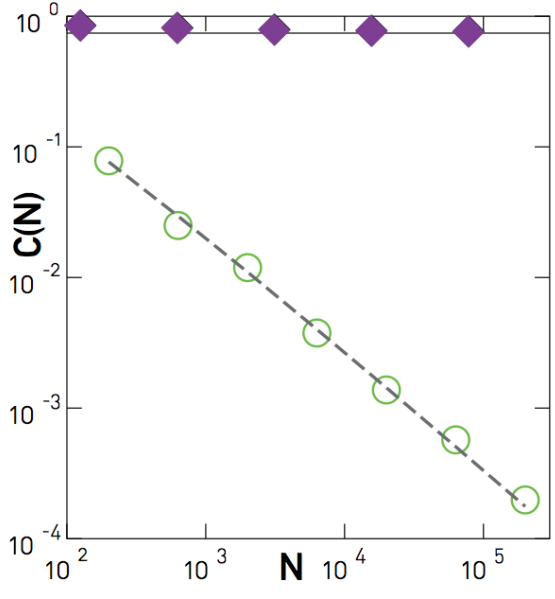
E. Ravasz & A.-L. Barabási, *PRE* 67 (2003).

A.-L. Barabási, *Network Science: Communities.*



(3) Clustering coefficient independent of N

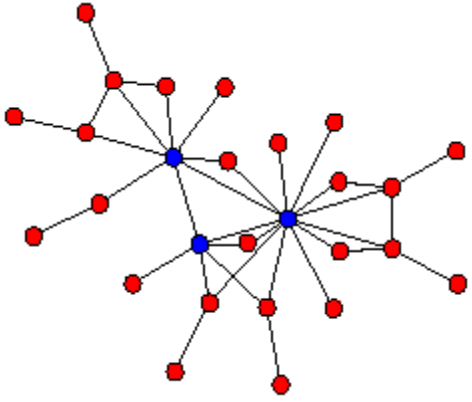
$$C = 0.743$$



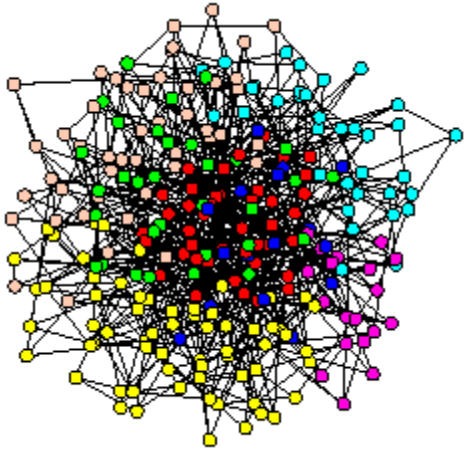
E. Ravasz & A.-L. Barabási, *PRE* 67 (2003).

A.-L. Barabási, *Network Science: Communities.*

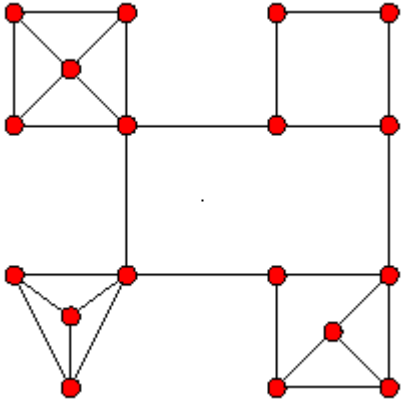
(a)



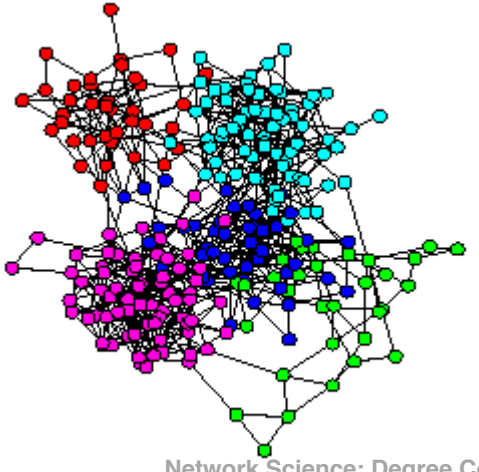
Scale-free



(b)

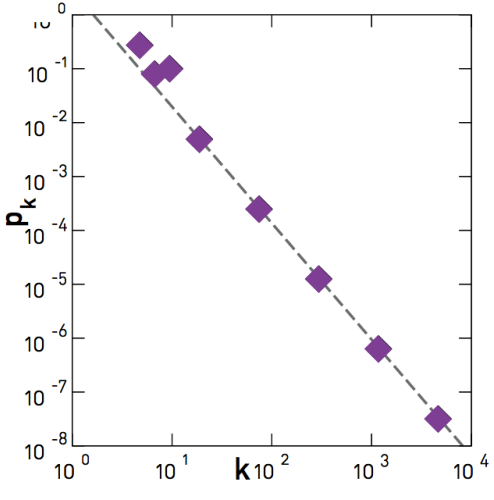


Modular



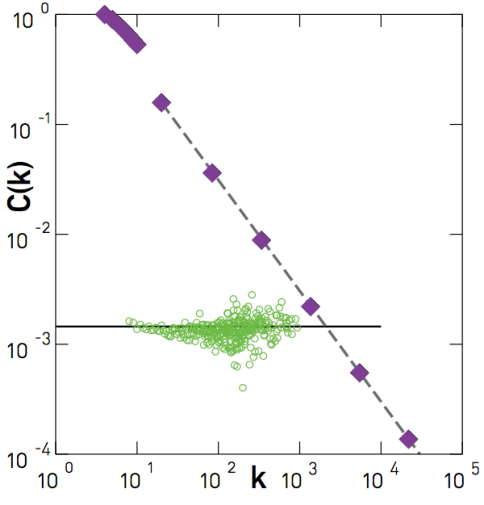
1. Scale-free

$$\gamma = 1 + \frac{\ln 5}{\ln 4} = 2.161$$



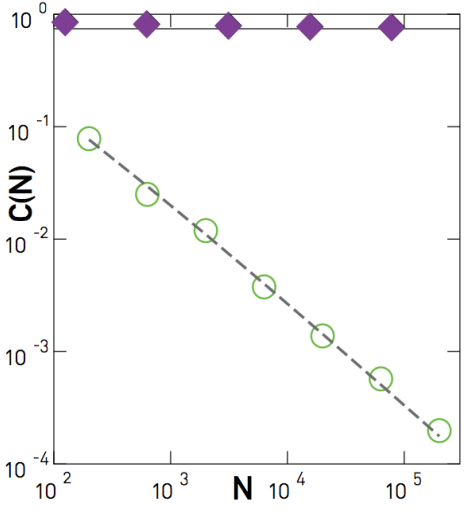
2. Scaling clustering coefficient (DGM)

$$C(k) \sim k^{-1}$$



3. Clustering coefficient independent of N

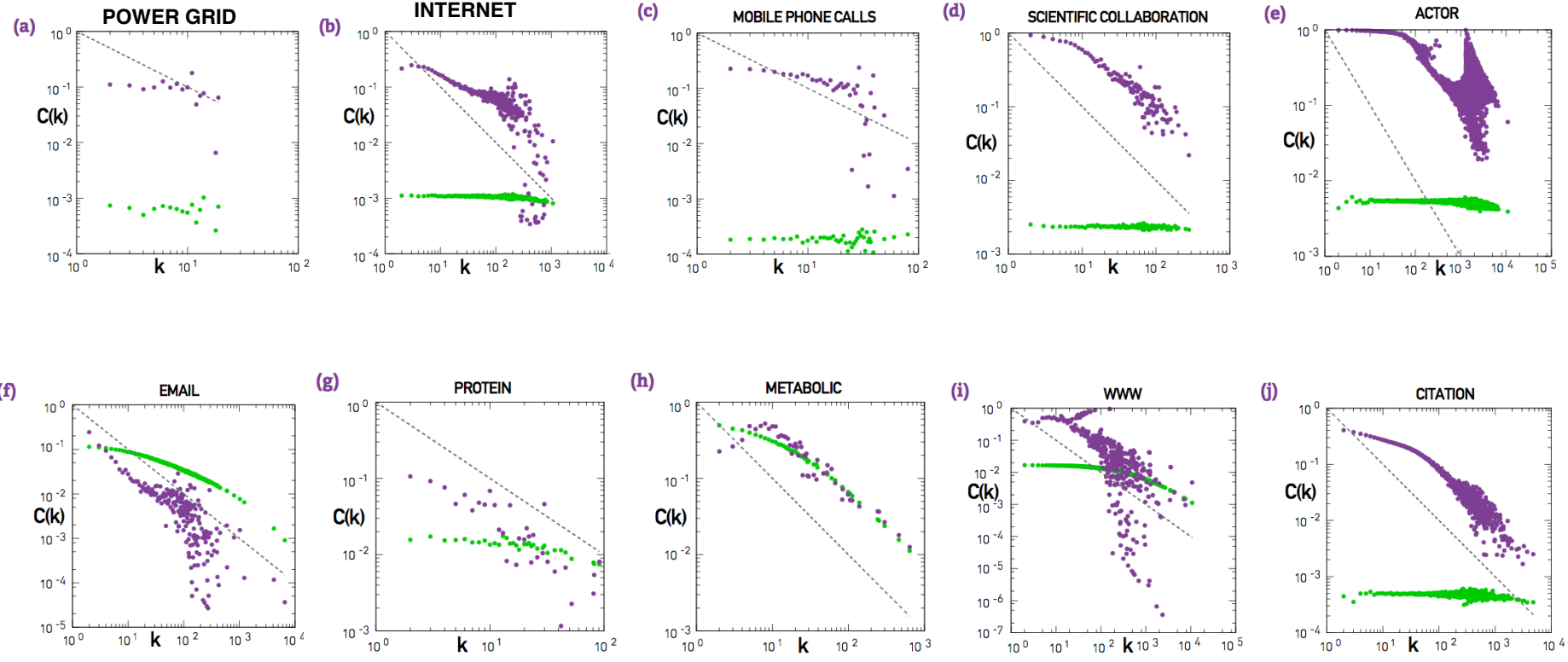
$$C(N) = \text{const.}$$



E. Ravasz & A.-L. Barabási, *PRE* 67 (2003).

A.-L. Barabási, *Network Science: Communities.*

Section 4 Hierarchy in real networks

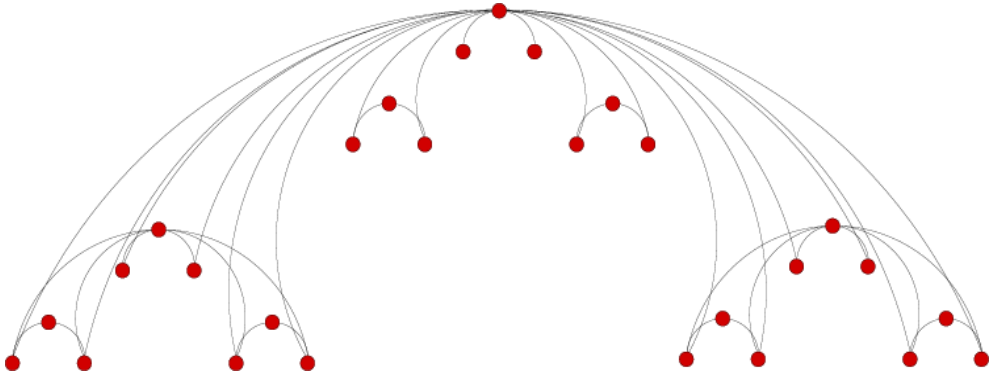


A.-L. Barabási, *Network Science: Communities.*

HIERARCHICAL MODELS

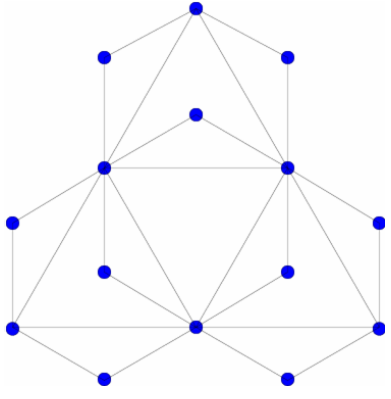
$$\gamma = \frac{\ln 3}{\ln 2}$$

- Barabási, Ravasz, Vicsek, Physica A 2003



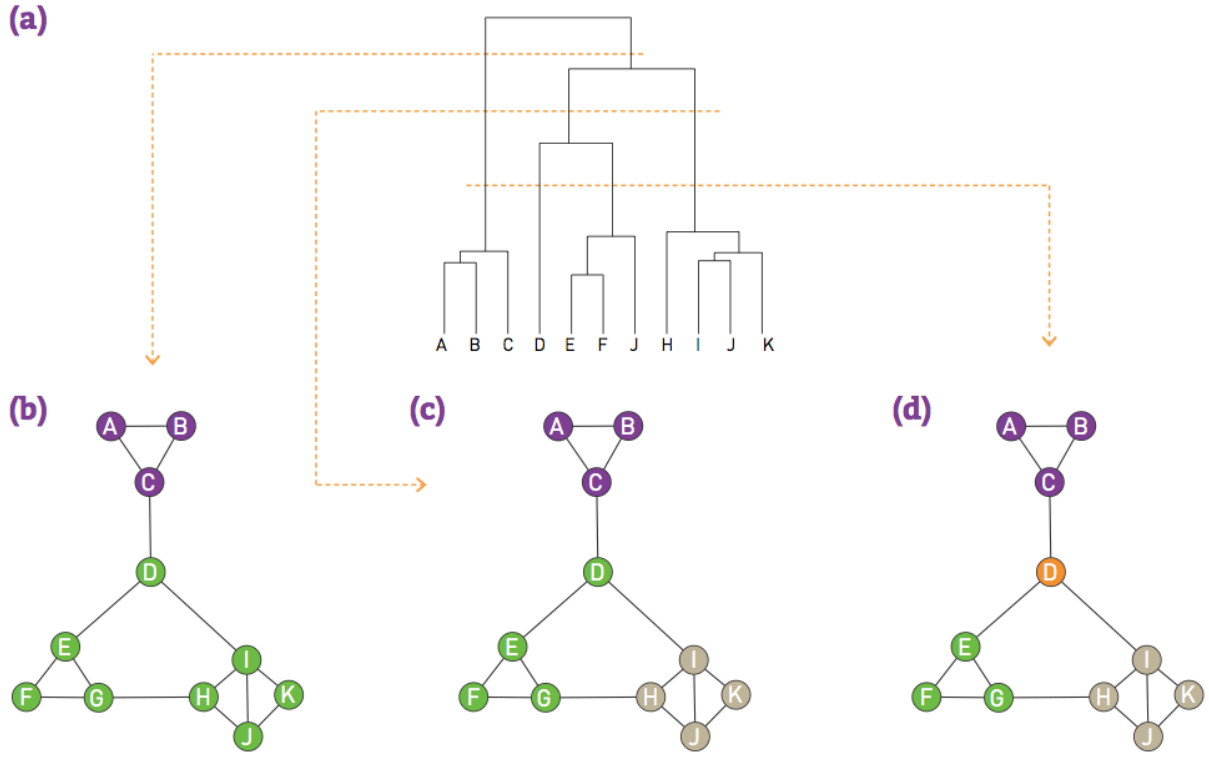
- Dorogovtsev, Goltsev, Mendes, 2001

$$\gamma = 1 + \frac{\ln 3}{\ln 2}$$



Ambiguity in Hierarchical clustering

Where to “cut”?

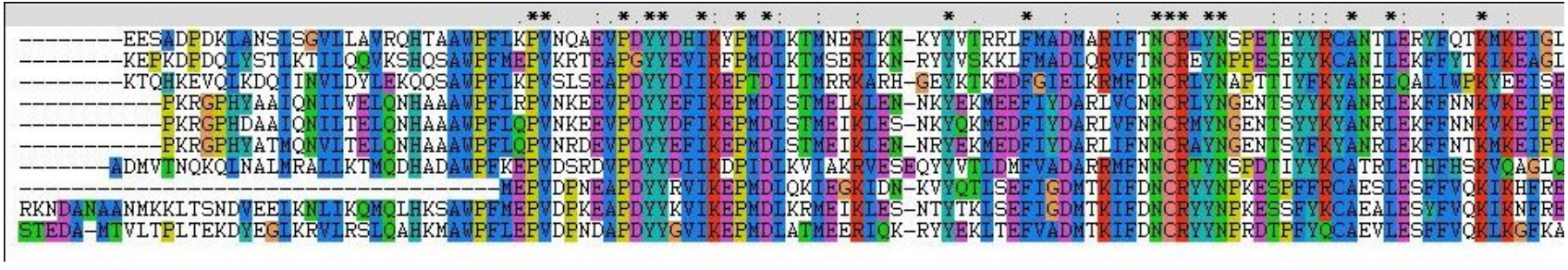


A.-L. Barabási, *Network Science: Communities.*

Phylogenetic dendrograms

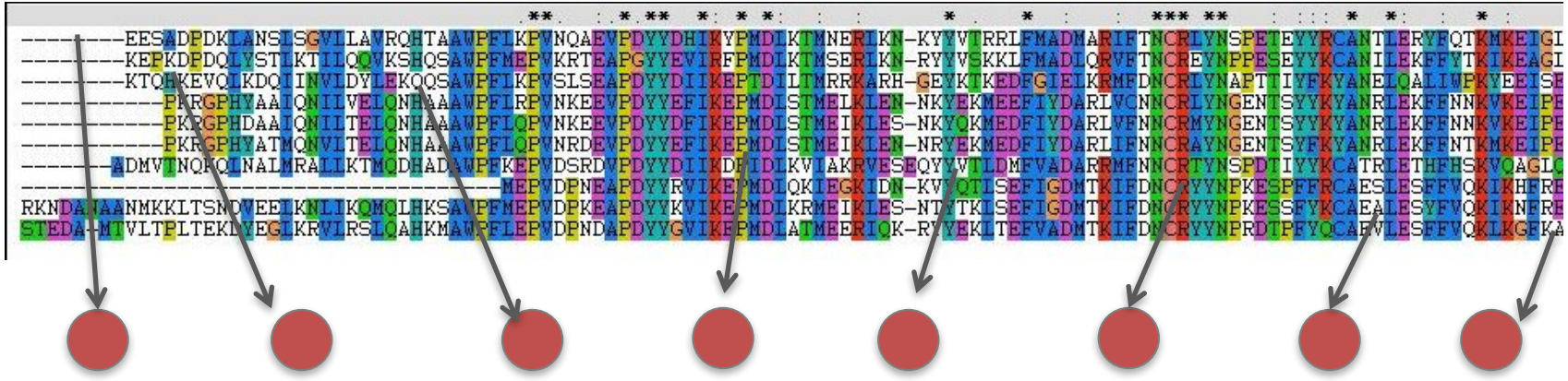
In bioinformatrics, clusters and dendrograms have been studied for a long time.

For example, the sequences of the same protein or gene in different species are selected, and compared with each other.



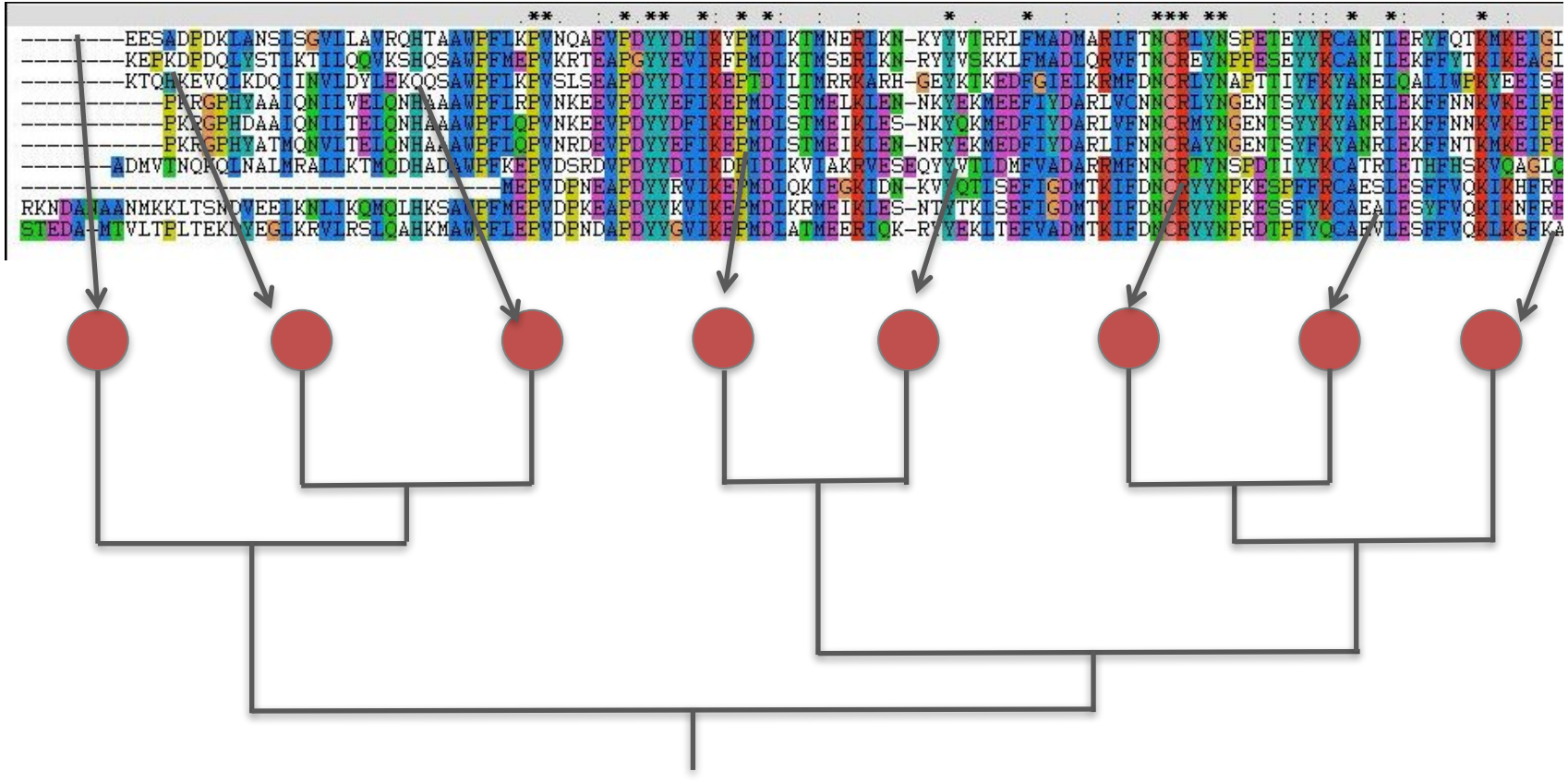
Phylogenetic dendrograms

A similarity matrix is constructed between these sequences,
by looking at how many aminoacids/nucleotides stay in place

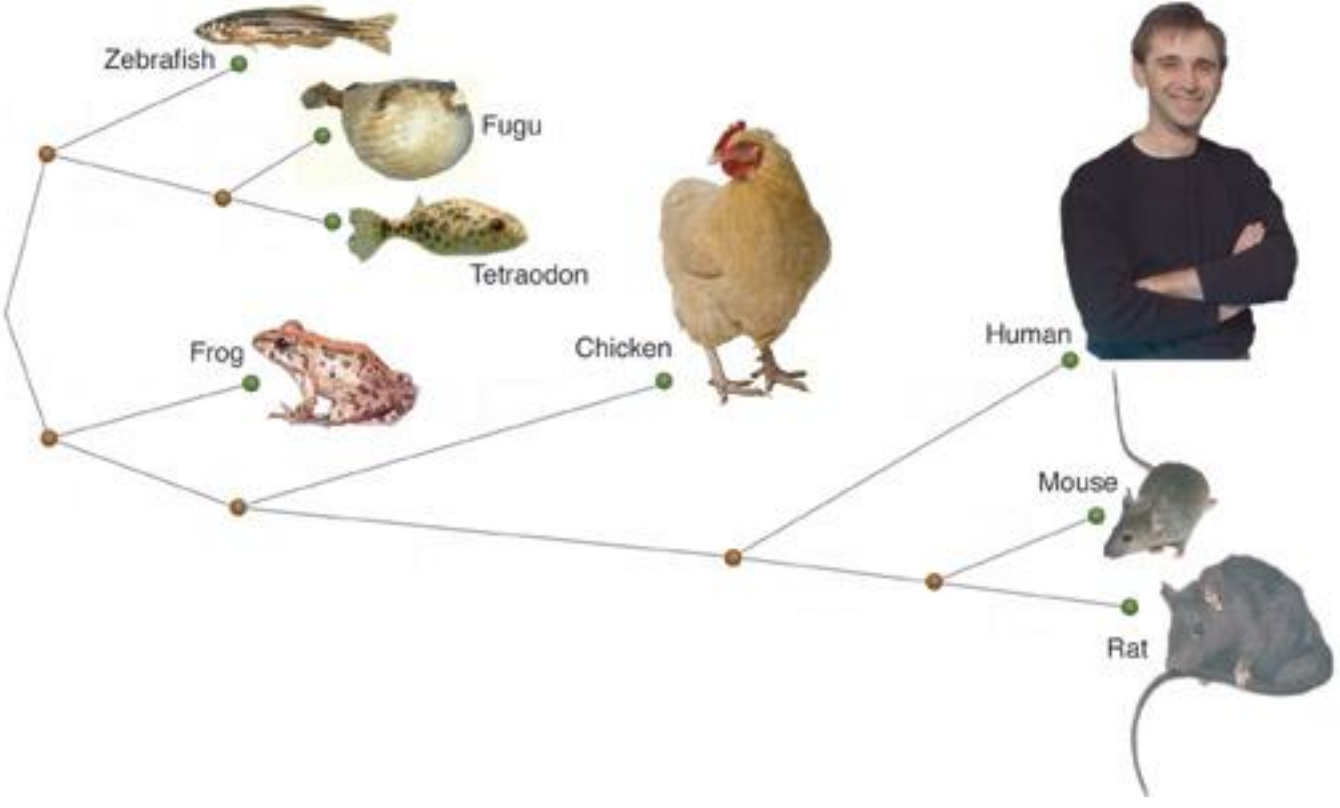


Phylogenetic dendrograms

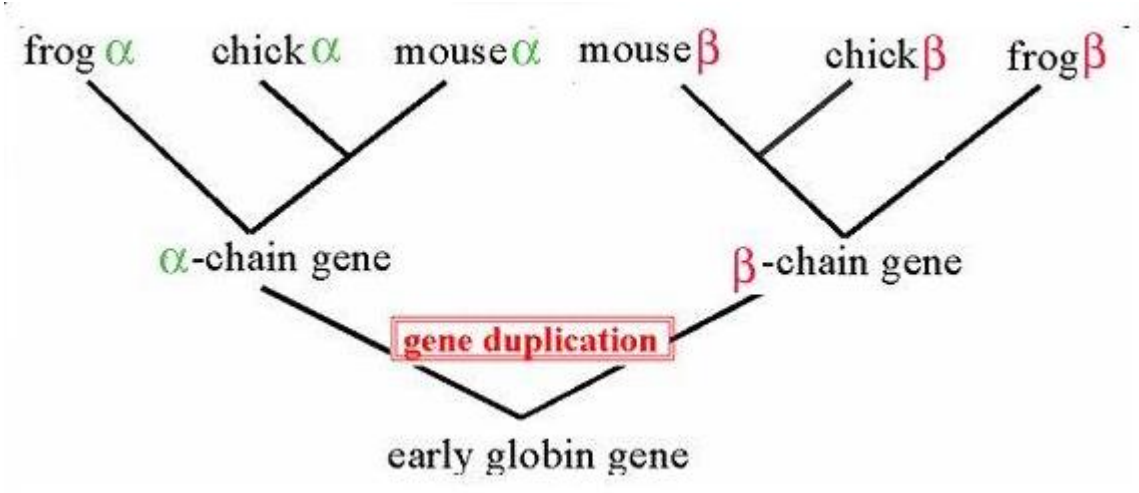
A similarity matrix is constructed between these sequences,
by looking at how many aminoacids/nucleotides stay in place



Phylogenetic dendrograms



Phylogenetic dendrograms



Modularity

H4: *Random Hypothesis*

Randomly wired networks are not expected to have a community structure.

MEJ Newman, *PNAS* 103 (2006).

A.-L. Barabási, *Network Science: Communities.*

H4: *Random Hypothesis*

Randomly wired networks are not expected to have a community structure.

Imagine a partition in n_c communities $\{C_c,\ c = 1,\ n_c\}$

Modularity $$M\left(C_c\right) = \frac{1}{2L}\sum_{i,j=1}^{N}(A_{ij} - P_{ij})\delta(C_i - C_j)$$

MEJ Newman, *PNAS* 103 (2006).

A.-L. Barabási, *Network Science: Communities.*

H4: *Random Hypothesis*

Randomly wired networks are not expected to have a community structure.

Imagine a partition in n_c communities $\{C_c,\ c = 1,\ n_c\}$

Modularity $$M\left(C_c\right) = \frac{1}{2L}\sum_{i,j=1}^{N}(A_{ij} - P_{ij})\delta(C_i - C_j)$$

Original data

MEJ Newman, *PNAS* 103 (2006).

A.-L. Barabási, *Network Science: Communities.*

H4: *Random Hypothesis*

Randomly wired networks are not expected to have a community structure.

Imagine a partition in n_c communities $\{C_c,\ c = 1,\ n_c\}$

Modularity $$M\left(C_c\right) = \frac{1}{2L}\sum_{i,j=1}^{N}(A_{ij} - P_{ij})\delta(C_i - C_j)$$

Original data

Expected connections, a model

MEJ Newman, *PNAS* 103 (2006).

A.-L. Barabási, *Network Science: Communities.*

H4: *Random Hypothesis*

Randomly wired networks are not expected to have a community structure.

Imagine a partition in n_c communities $\{C_c,\ c = 1,\ n_c\}$

Modularity $$M\left(C_c\right) = \frac{1}{2L}\sum_{i,j=1}^{N}\left(A_{ij} - P_{ij}\right)\delta(C_i - C_j)$$

Original data

Expected connections, a model

Relative to a specific partition

MEJ Newman, *PNAS* 103 (2006).

A.-L. Barabási, *Network Science: Communities.*

H4: *Random Hypothesis*

Randomly wired networks are not expected to have a community structure.

Imagine a partition in n_c communities $\{C_c,\ c = 1,\ n_c\}$

Modularity $$M\left(C_c\right) = \frac{1}{2L}\sum_{i,j=1}^{N}\left(A_{ij} - P_{ij}\right)\delta(C_i - C_j)$$

- A_{ij}: Original data
- P_{ij}: Expected connections, a model
- $\delta(C_i - C_j)$: Relative to a specific partition

→ Random network $P_{ij} = 2Lp_ip_j = \frac{k_ik_j}{2L}$

→ Modularity is a measure associated to a partition

MEJ Newman, *PNAS* 103 (2006).

A.-L. Barabási, *Network Science: Communities.*

Another way of writing *M*

$$M(C_c) = \frac{1}{2L}\sum_{i,j=1}^{N}(A_{ij} - P_{ij})\delta(C_i - C_j) \qquad P_{ij} = 2Lp_ip_j = \frac{k_ik_j}{2L}$$

We can rewrite the first term as

$$\frac{1}{2L}\sum_{i,j=1}^{N}A_{ij}\delta(C_i - C_j) = \sum_{c=1}^{n_c}\frac{1}{2L}\sum_{i,j\in C_c}A_{ij} = \sum_{c=1}^{n_c}\frac{l_c}{L}$$

where L_C is the number of links within C. In a similar fashion, the second term becomes

$$\frac{1}{2L}\sum_{i,j}\frac{k_ik_j}{2L}\delta(C_i - C_j) = \sum_{c=1}^{n_c}\frac{1}{(2L)^2}\sum_{i,j\in C_c}k_ik_j = \sum_{c=1}^{n_c}\frac{k_c^2}{4L^2}$$

Finally we get:

$$M(C_c) = \sum_{c=1}^{n_c}\left[\frac{l_c}{L} - \left(\frac{k_c}{2L}\right)^2\right]$$

MEJ Newman, *PNAS* 103 (2006).

A.-L. Barabási, *Network Science: Communities.*

H5: *Maximal Modularity Hypothesis*

The partition with the maximum modularity *M* for a given network offers the optimal community structure

MEJ Newman, *PNAS* 103 (2006).

A.-L. Barabási, *Network Science: Communities.*

H5: *Maximal Modularity Hypothesis*

The partition with the maximum modularity *M* for a given network offers the optimal community structure

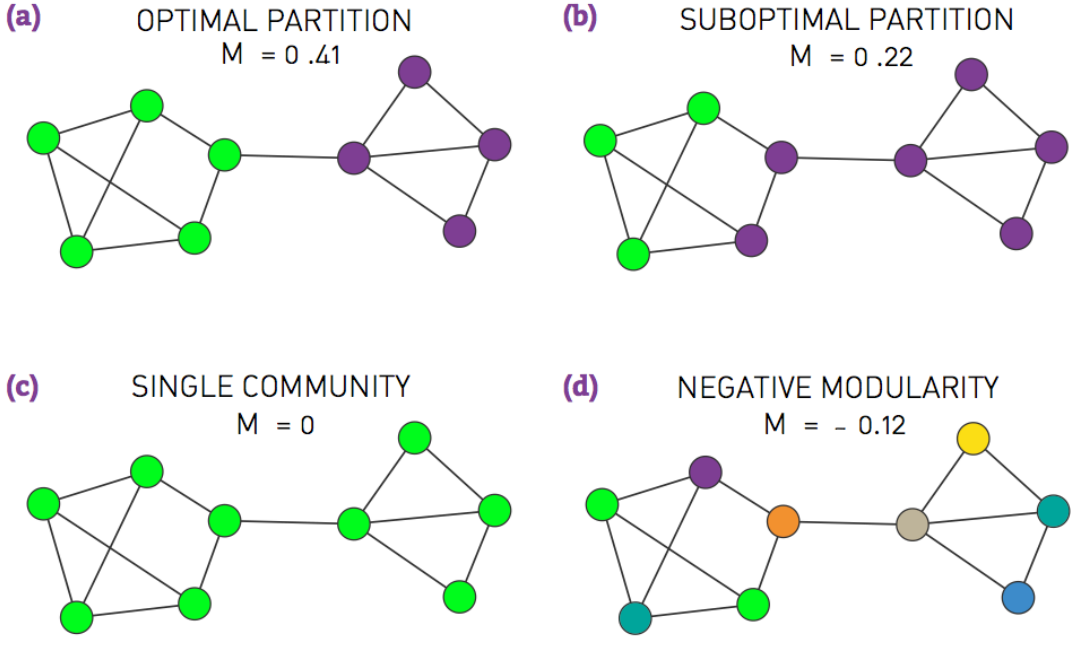
Goal

Find $\{C_c,\ c = 1,\ n_c\}$ that maximizes M

MEJ Newman, *PNAS* 103 (2006).

A.-L. Barabási, *Network Science: Communities.*

Which partition $\{C_c,\ c = 1,\ n_c\}$?



- *Optimal partition*, that maximizes the modularity.
- *Sub-optimal* but positive modularity.
- *Negative Modularity*: If we assign each node to a different community.
- *Zero modularity:* Assigning all nodes to the same community, independent of the network structure.
- *Modularity is size dependent*

Section 4 Modularity based community identification

A *greedy algorithm*, which iteratively joins nodes if the move increases the new partition's modularity.

Step 1. Assign each node to a community of its own. Hence we start with N communities.

Step 2. Inspect each pair of communities connected by at least one link and compute the modularity variation obtained if we merge these two communities.

Step 3. Identify the community pairs for which ΔM is the largest and merge them. Note that modularity of a particular partition is always calculated from the full topology of the network.

Step 4. Repeat step 2 until all nodes are merged into a single community.

Step 5. Record for each step and select the partition for which the modularity is maximal.

MEJ Newman, *PRE 69* (2004).

A.-L. Barabási, *Network Science: Communities.*

Which partition $\{C_c,\ c = 1,\ n_c\}$ **?**

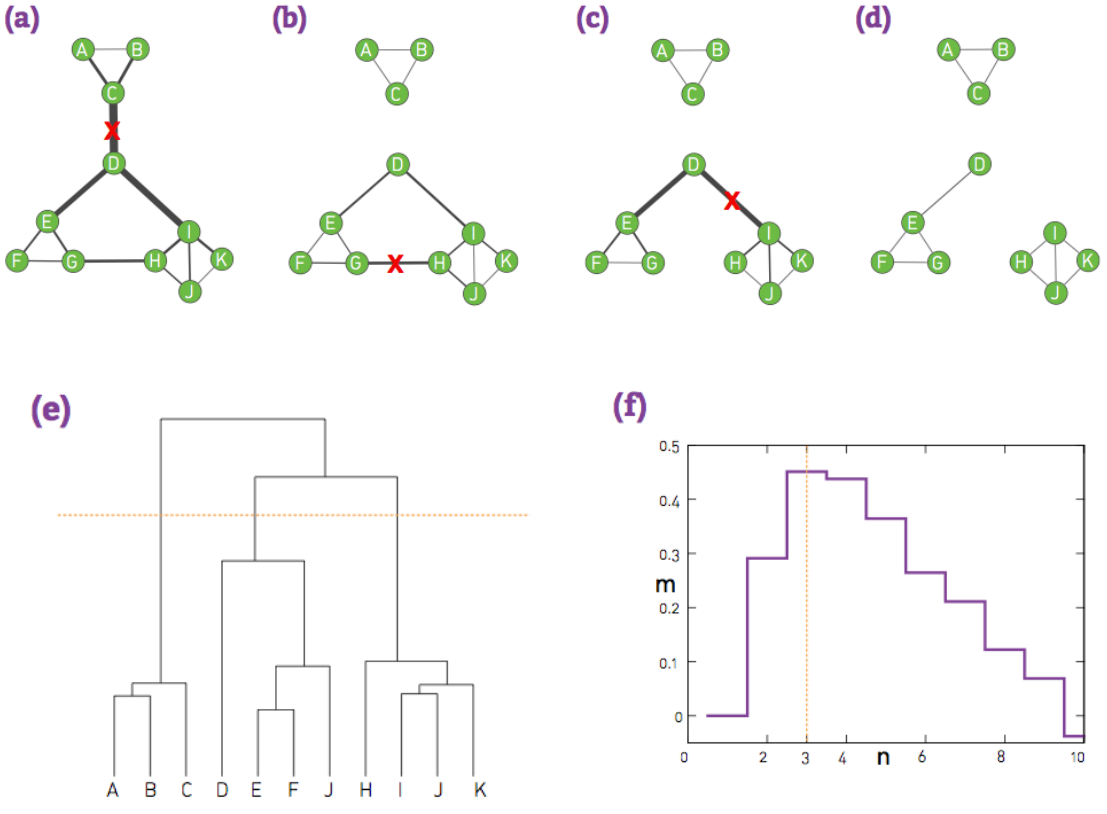
→ It can be used to design new algorithms, **aiming at maximizing M**

→ Modularity can be used to compare different partitions provided by other algorithms, like hierarchical clustering

Modularity for the Girvan-Newman

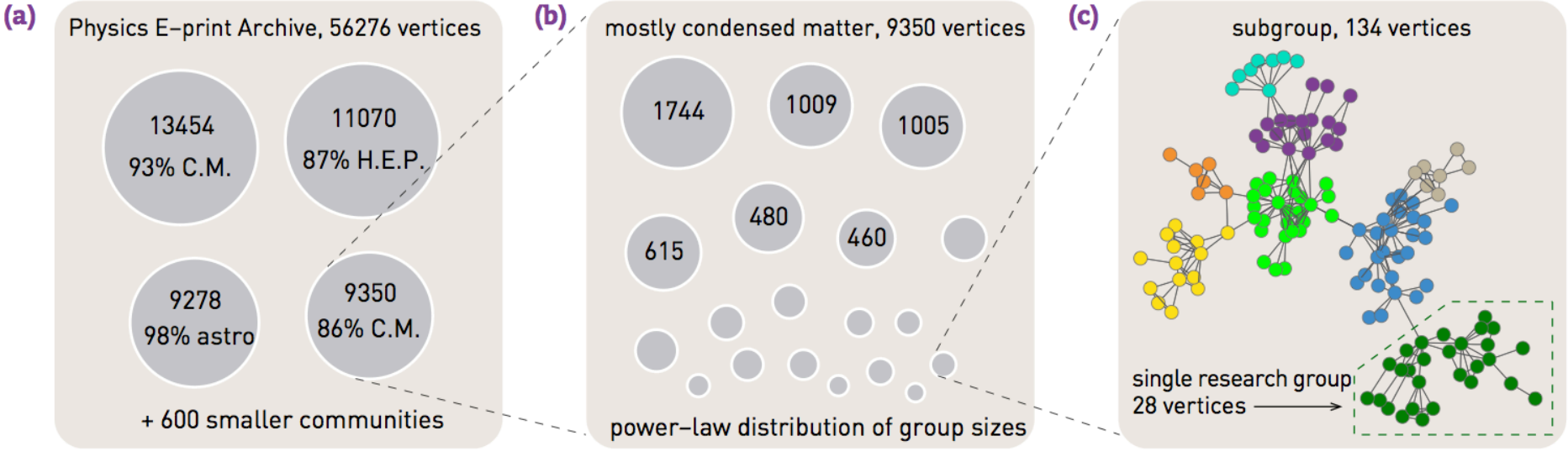
Which partition $\{C_c,\ c=1,\ n_c\}$ **?**

$$M(C_c) = \sum_{c=1}^{n_c} \left[\frac{l_c}{L} - \left(\frac{k_c}{2L} \right)^2 \right]$$



A.-L. Barabási, *Network Science: Communities.*

Section 4 Modularity based community identification



Computational complexity:

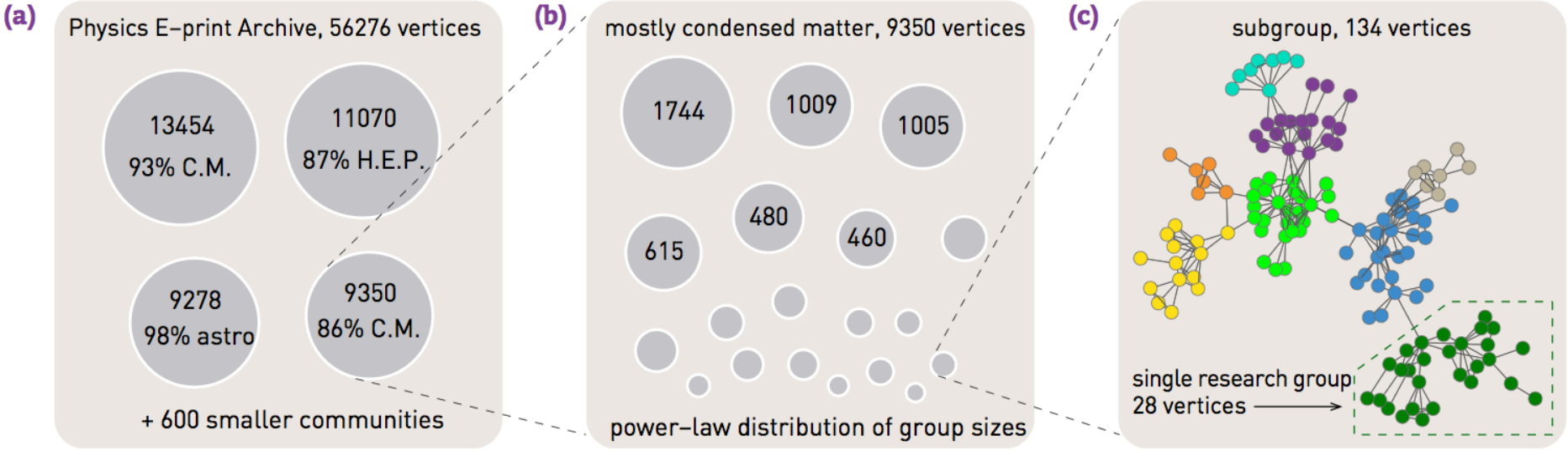
- Step 1-2 (calculation of ΔM for L links): $O(L)$
- Step 3 (matrix update): $O(N)$
- Step 4 (N-1 community merges): $O((L+N)N)$

for sparse networks

MEJ Newman, *PRE 69* (2004).

A.-L. Barabási, *Network Science: Communities.*

Section 4 Modularity based community identification



Computational complexity:

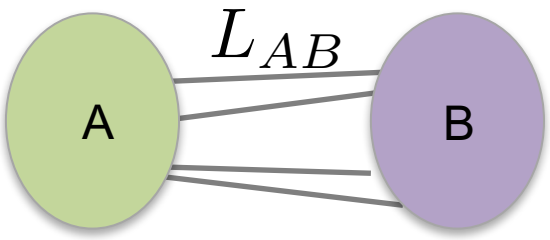
- Step 1-2 (calculation of ΔM for L links): $O(L)$
- Step 3 (matrix update): $O(N)$
- Step 4 (N-1 community merges): $O((L+N)N)$

for sparse networks → $O(N^2)$

MEJ Newman, *PRE 69* (2004).

A.-L. Barabási, *Network Science: Communities.*

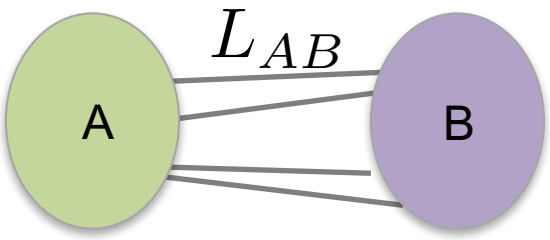
Resolution limit



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

k_A and k_B total degree in A and B

Resolution limit

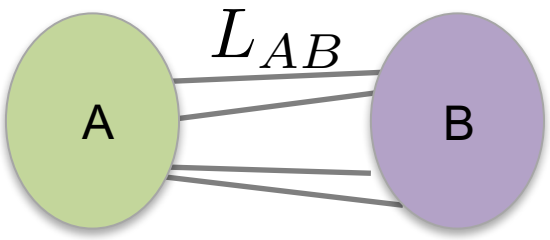


$$\Delta M_{AB} = \frac{L_{AB}}{L} - \frac{k_A k_B}{2L^2},$$

k_A and k_B total degree in A and B

If $\frac{k_A k_B}{2L} < 1$ and $L_{AB} \geq 1$

Resolution limit

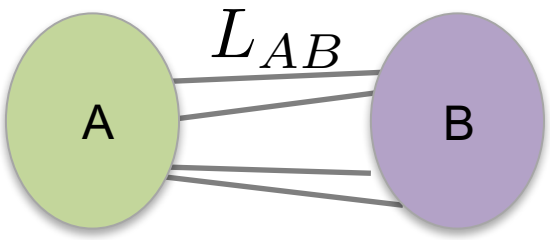


$$\Delta M_{AB} = \frac{L_{AB}}{L} - \frac{k_A k_B}{2L^2},$$

k_A and k_B total degree in A and B

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

Resolution limit



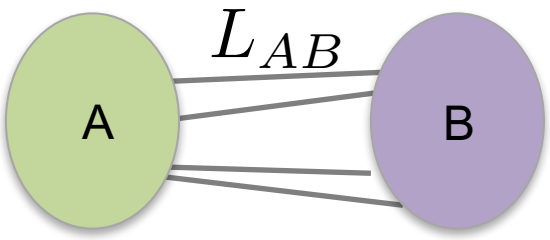
$$\Delta M_{AB} = \frac{L_{AB}}{L} - \frac{k_A k_B}{2L^2},$$

k_A and k_B total degree in A and B

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

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

Resolution limit



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

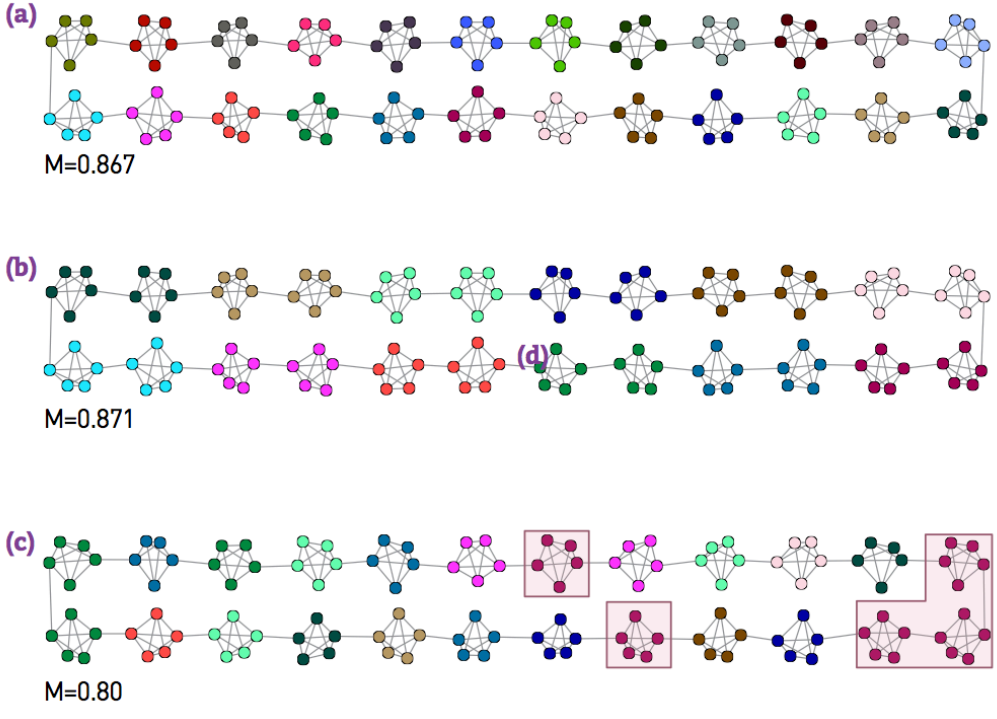
k_A and k_B total degree in A and B

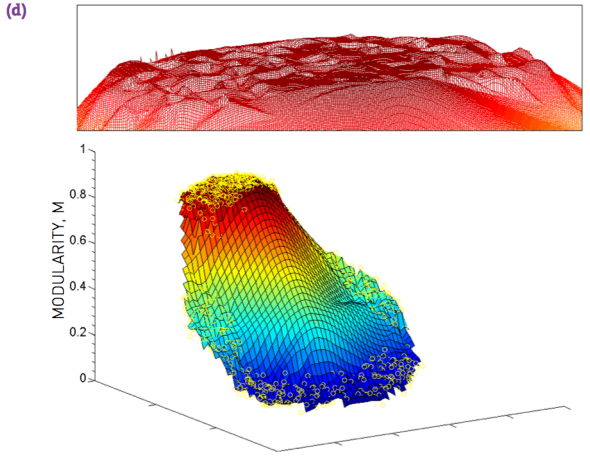
If $\frac{k_A k_B}{2L} < 1$ and $L_{AB} \geq 1$ ➡ $\Delta M_{AB} > 0$ We merge A and B to maximize modularity.

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

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

One maximum?





A.-L. Barabási, *Network Science: Communities.*

Null models

$$M(C_c) = \frac{1}{2L}\sum_{i,j=1}^{N}(A_{ij} - P_{ij})\delta(C_i - C_j)$$

Expected connections, a model

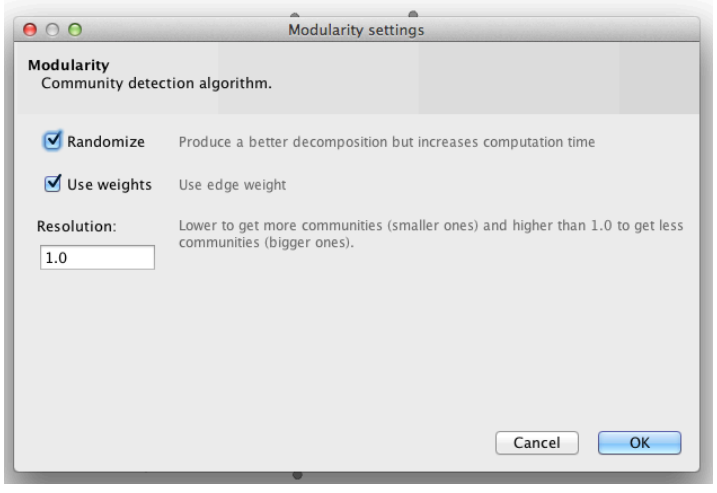
→ P_{ij} can take into account weights — S. Fortunato, *Phys. Rep.* 486 (2010)

→ P_{ij} can take into account directions — S. Fortunato, *Phys. Rep.* 486 (2010)

→ P_{ij} can take into account attributes or space — P. Expert el al., *PNAS* 108 (2011)

Section 5 Online Resources (Modularity)

Gephi



R assigns self-loops to nodes to increase or decrease the aversion of nodes to form communities

NetworkX

`community.best_partition(graph, partition=None)` ¶

Compute the partition of the graph nodes which maximises the modularity (or try..) using the Louvain heuristices

This is the partition of highest modularity, i.e. the highest partition of the dendogram generated by the Louvain algorithm.

Finds the partition that maximizes modularity (considers weights and direction)

`community.modularity(partition, graph)`

Compute the modularity of a partition of a graph

Calculates the modularity of the partition you provide

Online Resources (1)

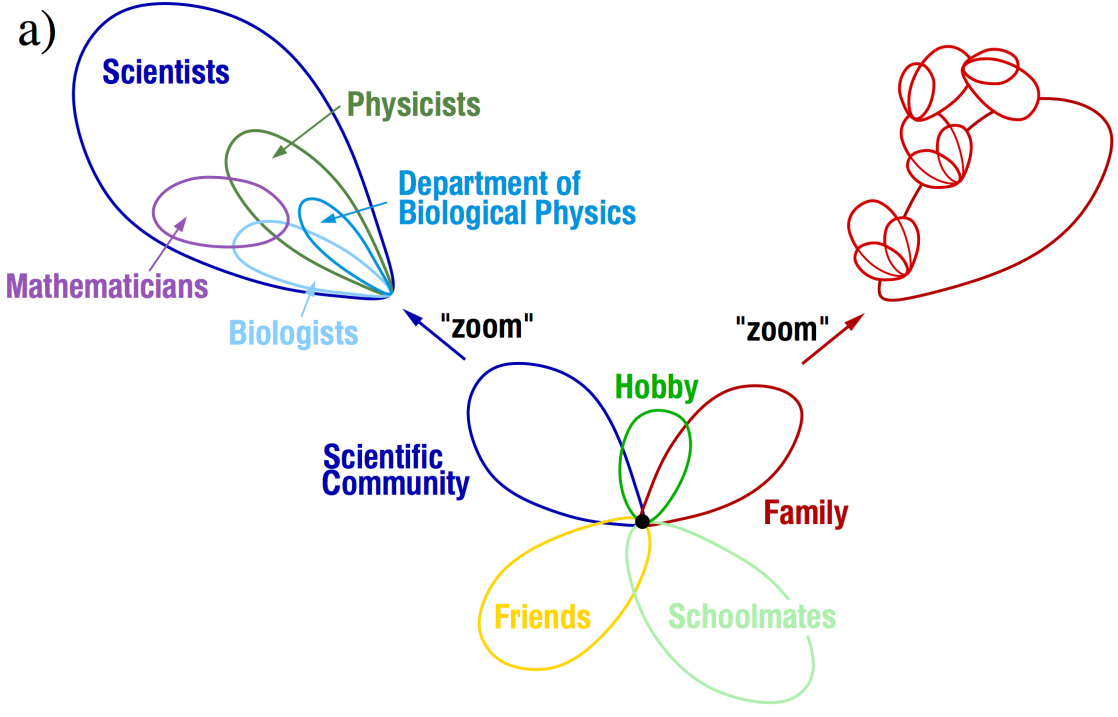
The greedy algorithm is neither particularly fast nor particularly successful at maximizing M.

Scalability: Due to the sparsity of the adjacency matrix, the update of the matrix involves a large number of useless operations. The use of data structures for sparse matrices can decrease the complexity of the computational algorithm to , which allows us to analyze is of networks up to nodes. See
"Fast Modularity" Community Structure Inference Algorithm
http://cs.unm.edu/~aaron/research/fastmodularity.htm for the code.

A fast greedy algorithm was proposed by Blondel and collaborators, that can process networks with millions of nodes. For the description of the algorithm see
Louvain method: Finding communities in large networks
https://sites.google.com/site/findcommunities/ for the code.

Overlapping Communities

Section 5 Overlapping Communities

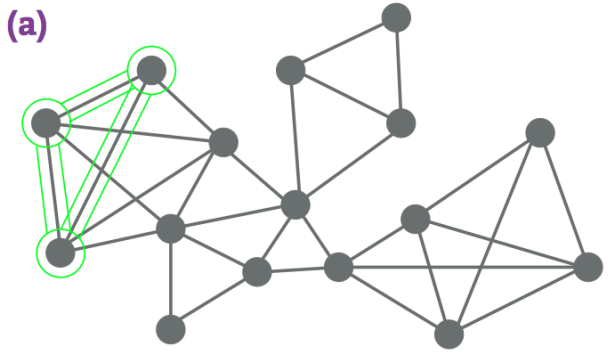


G. Palla et al., *Nature* 435 (2005).

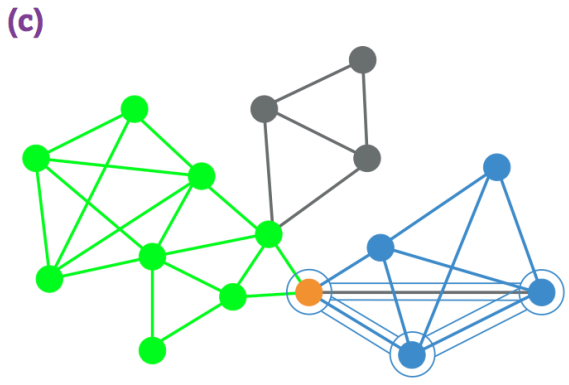
A.-L. Barabási, *Network Science: Communities.*

Section 5

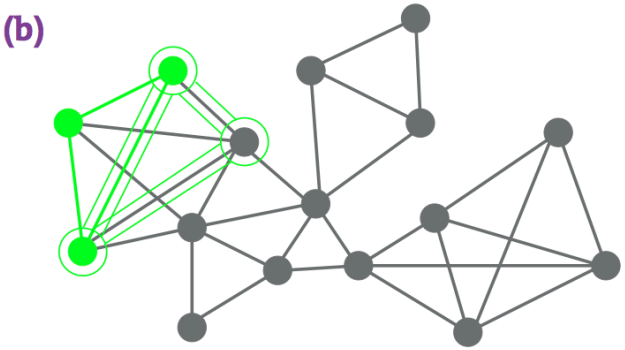
Clique Percolation (CFinder)



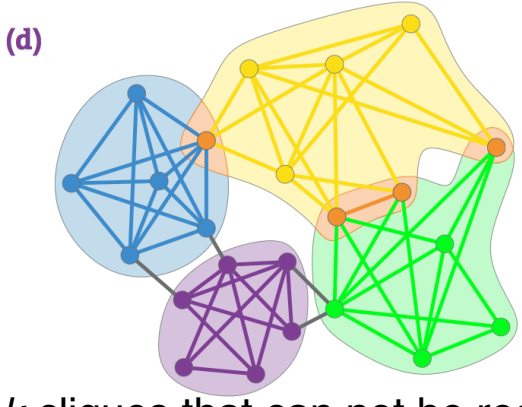
Start with a *k*-clique (complete subgraphs of *k* nodes), a 3-clique for example



A *k*-clique community is the largest connected subgraph obtained by the union of all adjacent *k*–cliques



Start "rolling" the clique over adjacent cliques. Two *k*-cliques are considered adjacent if they share *k-1* nodes



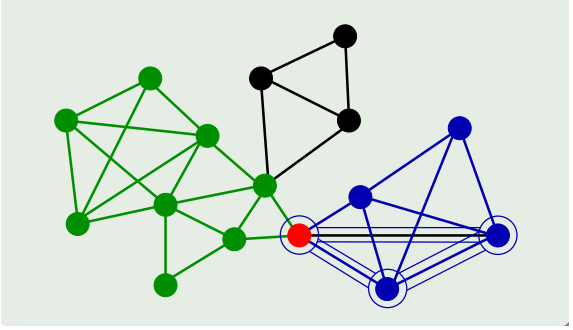
Other *k*-cliques that can not be reached from a particular clique correspond to other clique-communities

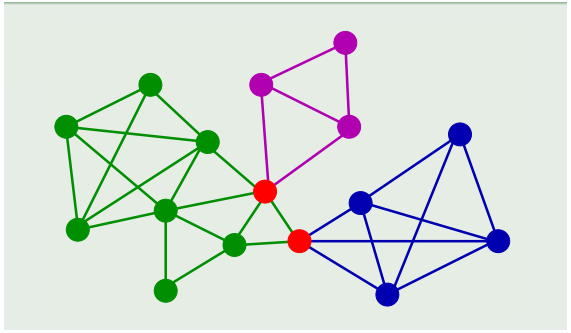
G. Palla et al., *Nature* 435 (2005).

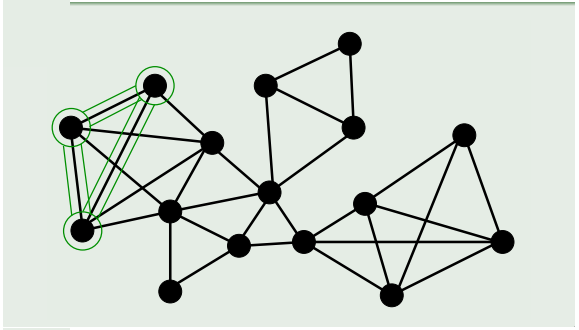
A.-L. Barabási, *Network Science: Communities.*

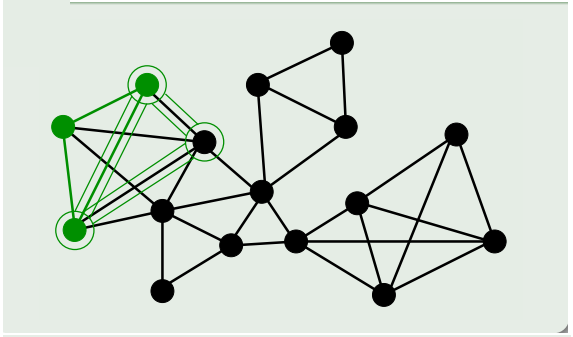
Clique Percolation

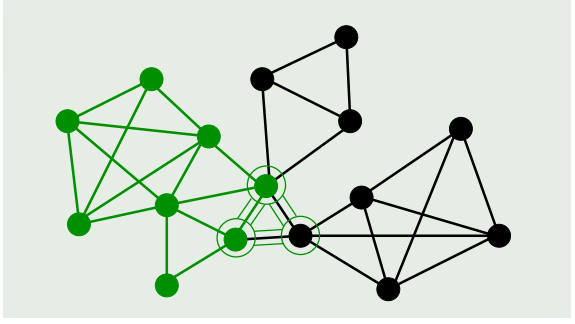
- Other *k*-cliques that can not be reached from a particular -clique correspond to other -clique-communities





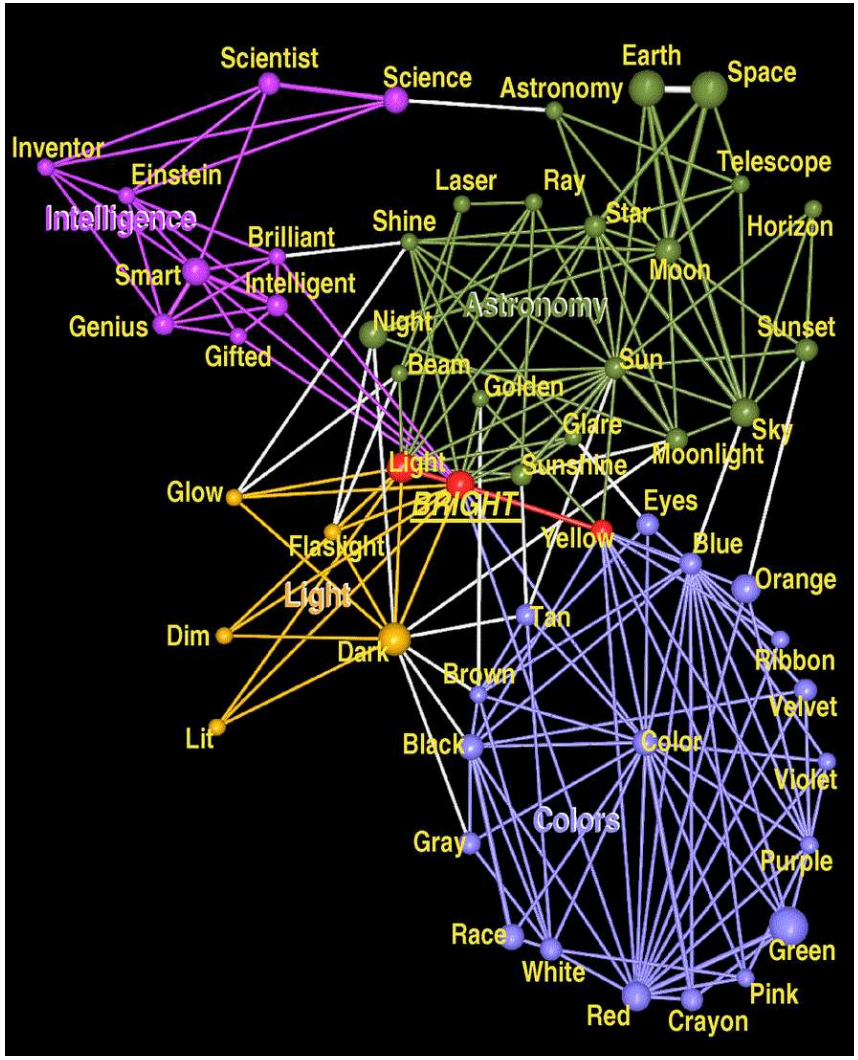






Palla, Derenyi, Farkas, Vicsek. Nature (2005).

Overlapping Communities



Bright:

- community containing light-related words (*glow* or *dark*);
- community capturing different colors (*yellow*, *brown*)
- community consisting of astronomical terms (*sun*, *ray*).
- community linked to intelligence (*gifted*, *brilliant*).

Could CP communities emerge by chance?

Notice that if a random network is sufficiently dense, there are cliques of varying order.

I. Derényi et al., *PRL* 94 (2005).

A.-L. Barabási, *Network Science: Communities.*

Could CP communities emerge by chance?

Notice that if a random network is sufficiently dense, there are cliques of varying order.

A *k*-clique community emerges in a random graph only if the connection probability exceeds the threshold:

$$k = 2 \qquad p_c(k) \sim 1/N$$

$$k = 3 \qquad p_c(k) = 1/\sqrt{2N}$$

$$p_c(k) = \frac{1}{[(k-1)N]^{1/(k-1)}}$$

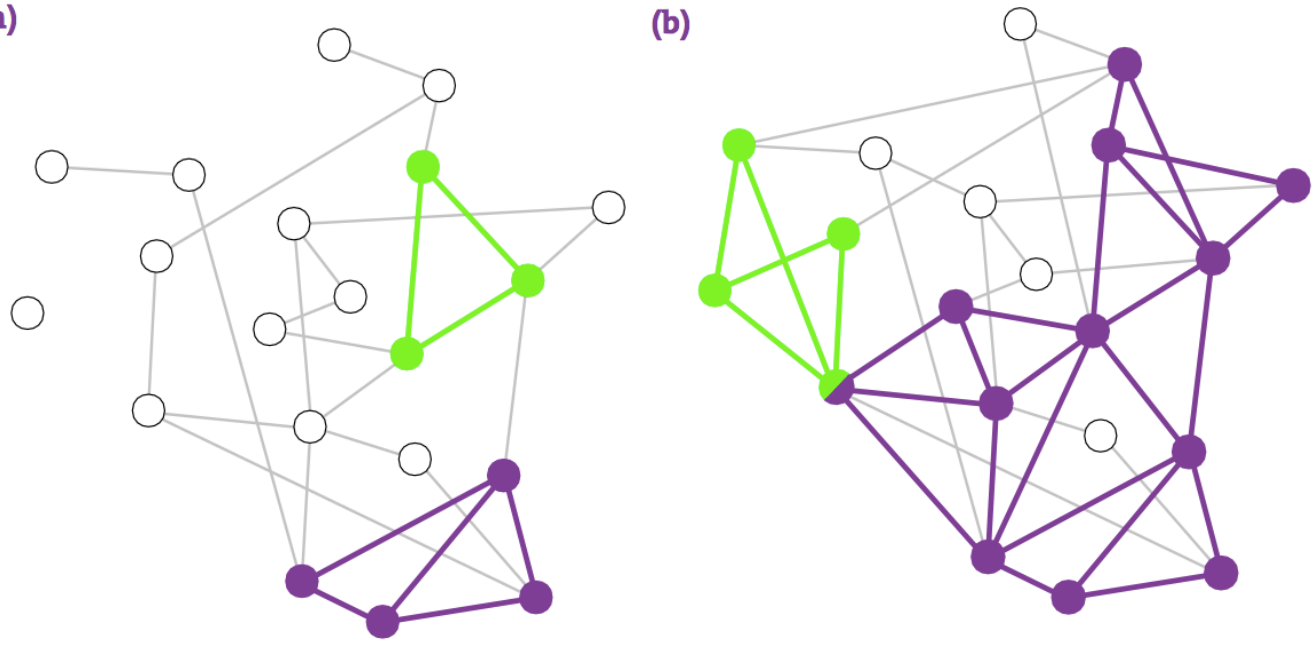
I. Derényi et al., *PRL* 94 (2005).

A.-L. Barabási, *Network Science: Communities.*

Section 5 Could CP communities emerge by chance?

$k = 3 \quad p_c(k) = 1/\sqrt{2N}$

$N=20 \qquad p_c=0.16$



Random networks with $p=0.13$ ($<p_c$) $p=0.22$ ($>p_c$)

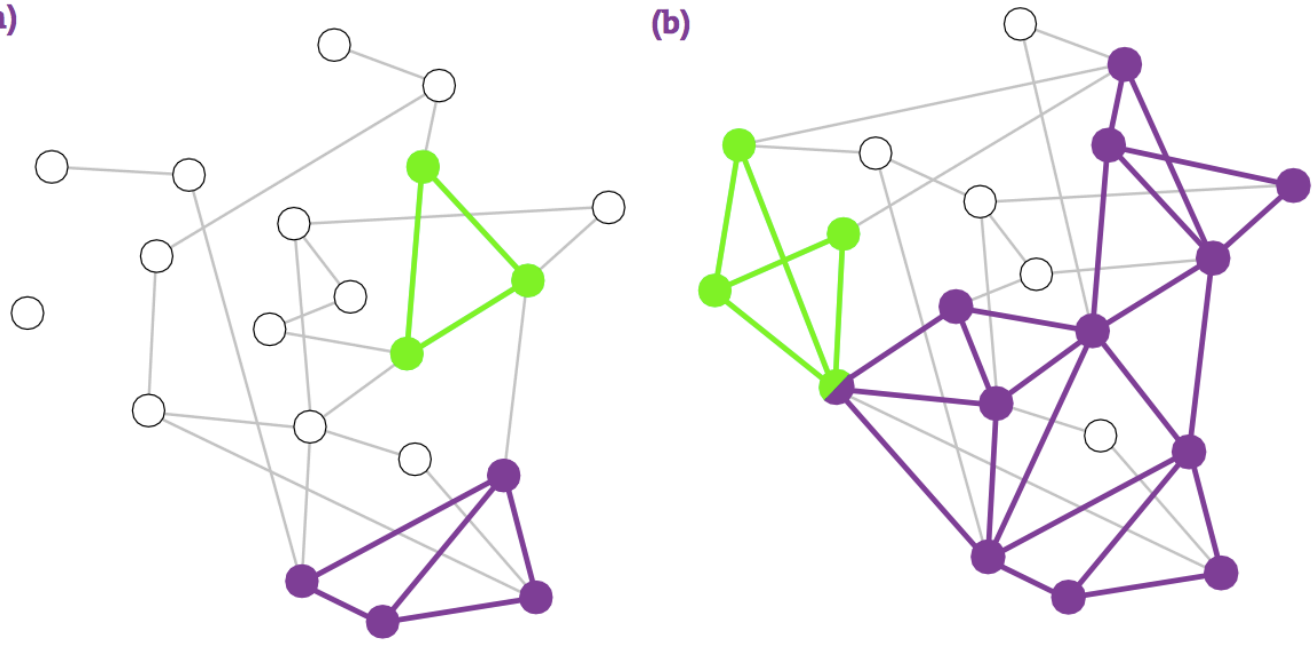
I. Derényi et al., *PRL* 94 (2005).

A.-L. Barabási, *Network Science: Communities.*

Section 5 Could CP communities emerge by chance?

$k = 3 \quad p_c(k) = 1/\sqrt{2N}$

$N=20 \qquad p_c=0.16$



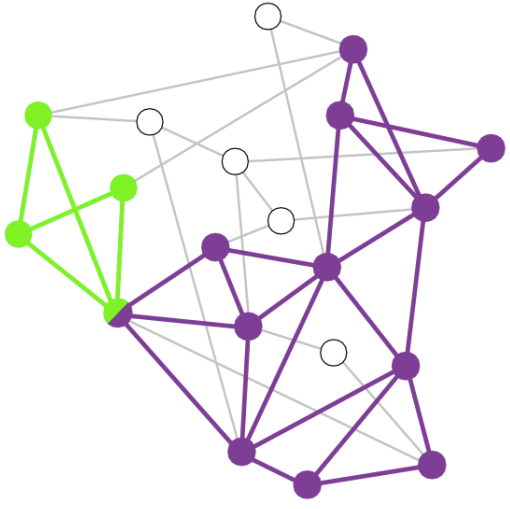
Random networks with $p=0.13$ ($<p_c$) $p=0.22$ ($>p_c$)

Compare your Cfinder output with that you obtain in a random graph with same N and p!

I. Derényi et al., *PRL* 94 (2005).

A.-L. Barabási, *Network Science: Communities.*

Clique percolation



Computational complexity:

- Finding maximal cliques require exponential time.
- However the algorithm has to find only *k*-cliques, which can be done in polynomial time

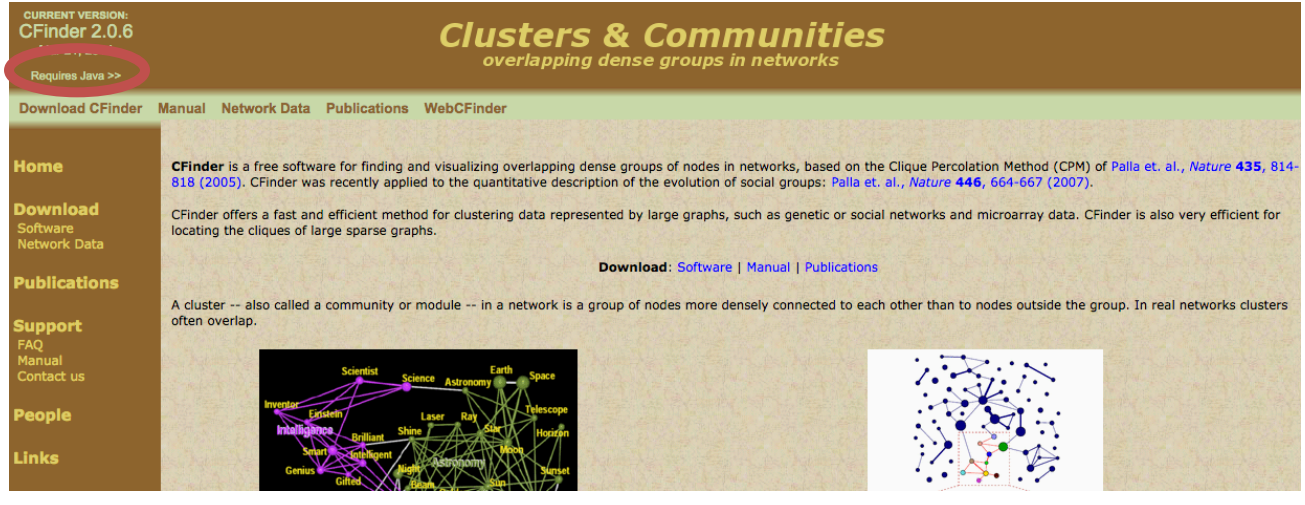
I. Derényi et al., *PRL* 94 (2005).

A.-L. Barabási, *Network Science: Communities.*

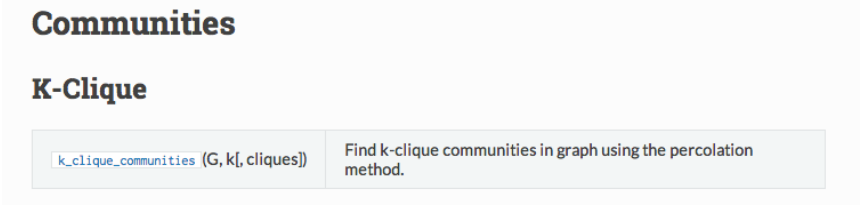
Section 5 Online Resources (CFinder)

→ The **CFinder** software package that implements the Clique Percolation Method can be downloaded at

www.cfinder.org



→ **NetworkX**



Section 5 Link Clustering

→ 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 <u>similarity of links</u>

Ahn, Bragow and Lehmann, *Nature* 466 (2010).

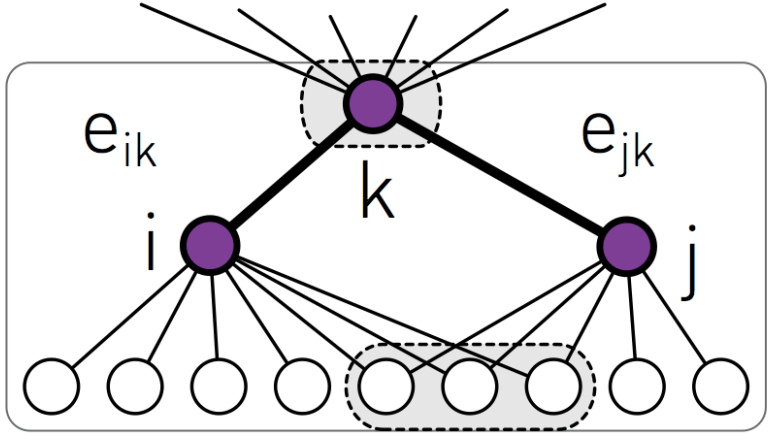
A.-L. Barabási, *Network Science: Communities.*

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.

(a)



$$\left|n_+(i) \cap n_+(j)\right| = 4$$
$$\left|n_+(i) \cup n_+(j)\right| = 12$$

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

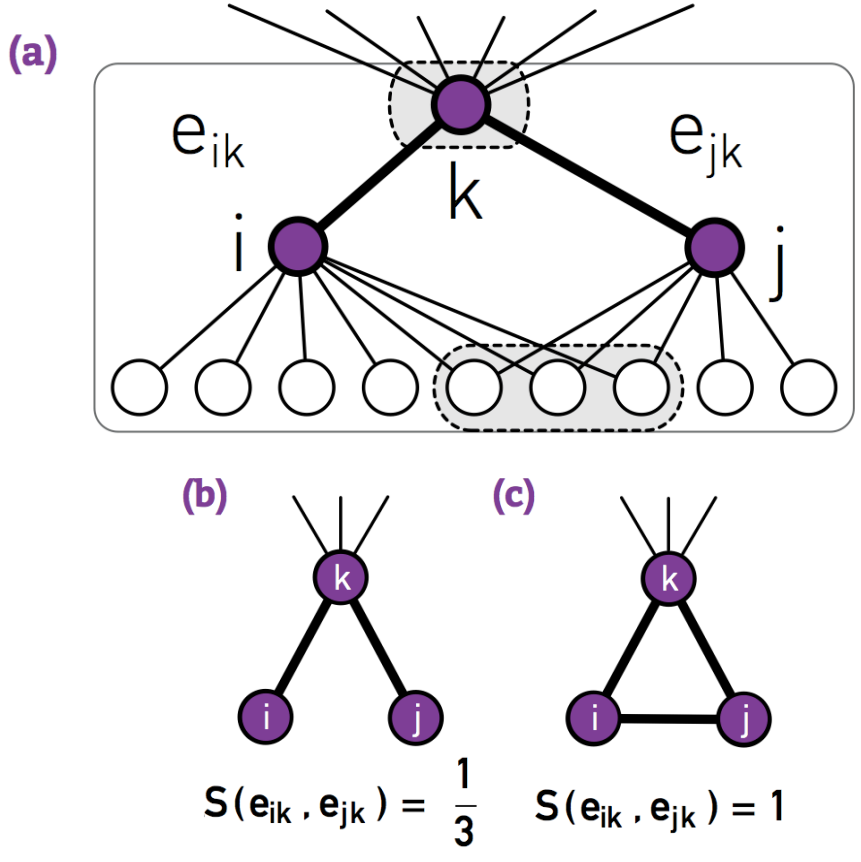
Ahn, Bragow and Lehmann, *Nature* 466 (2010).

A.-L. Barabási, *Network Science: 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.



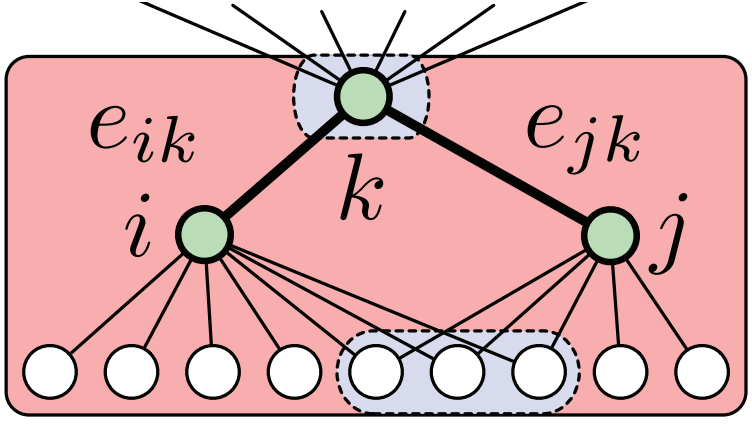
Ahn, Bragow and Lehmann, *Nature* 466 (2010).

A.-L. Barabási, *Network Science: Communities.*

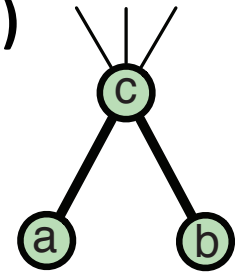
$$S(e_{ik}, e_{jk}) = \frac{|n_+(i) \cap n_+(j)|}{|n_+(i) \cup n_+(j)|}$$

$n_+(i)$: the list of the neighbors of node *i*, including itself.
S measures the relative number of common neighbors *i* and *j* have.

(a)



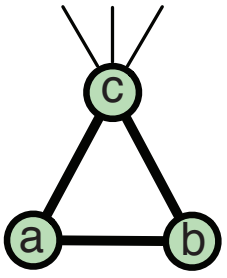
(b)



$$S(e_{ac}, e_{bc}) = \frac{1}{3}$$

$$\left|n_+(i) \cap n_+(j)\right| = 4$$
$$\left|n_+(i) \cup n_+(j)\right| = 12$$

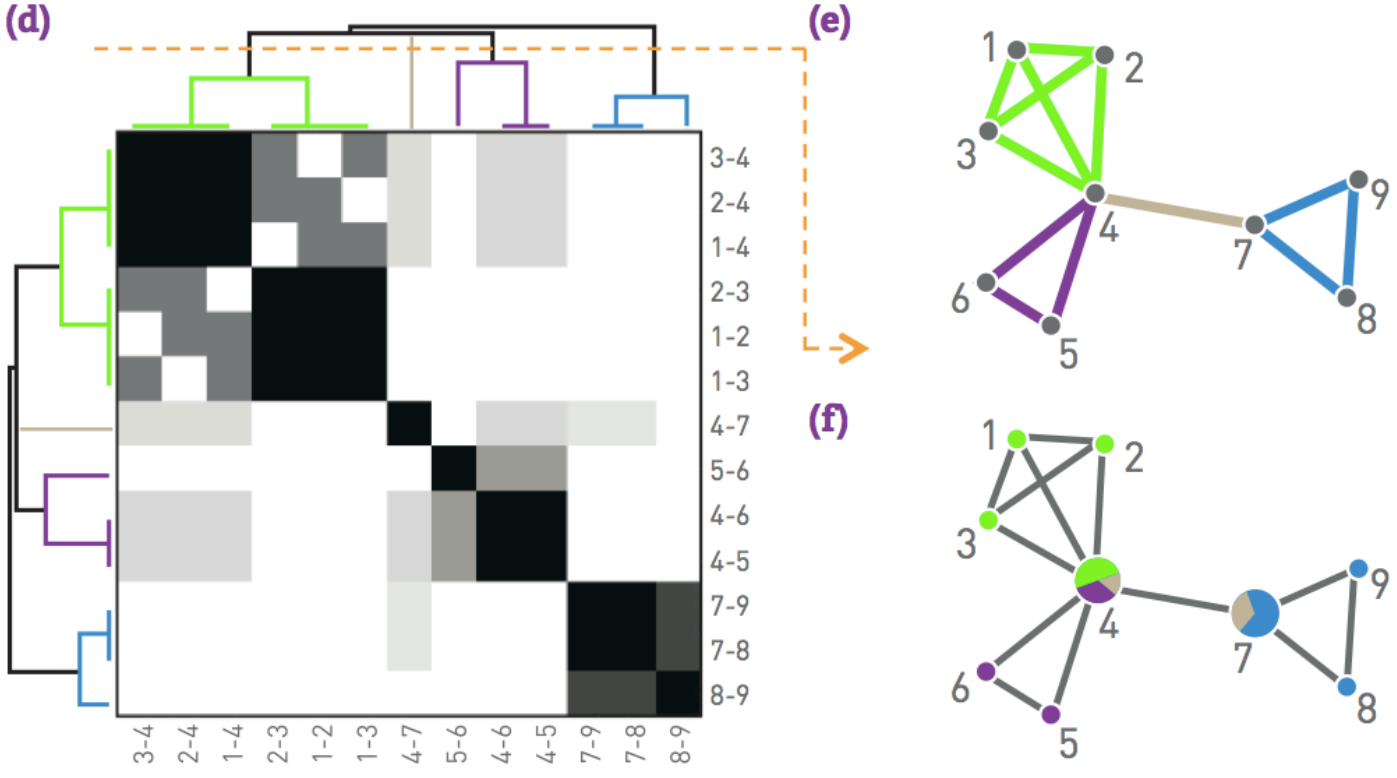
(c)



$$S(e_{ac}, e_{bc}) = 1$$

Ahn, Bargow, Lehmann, Nature 2010

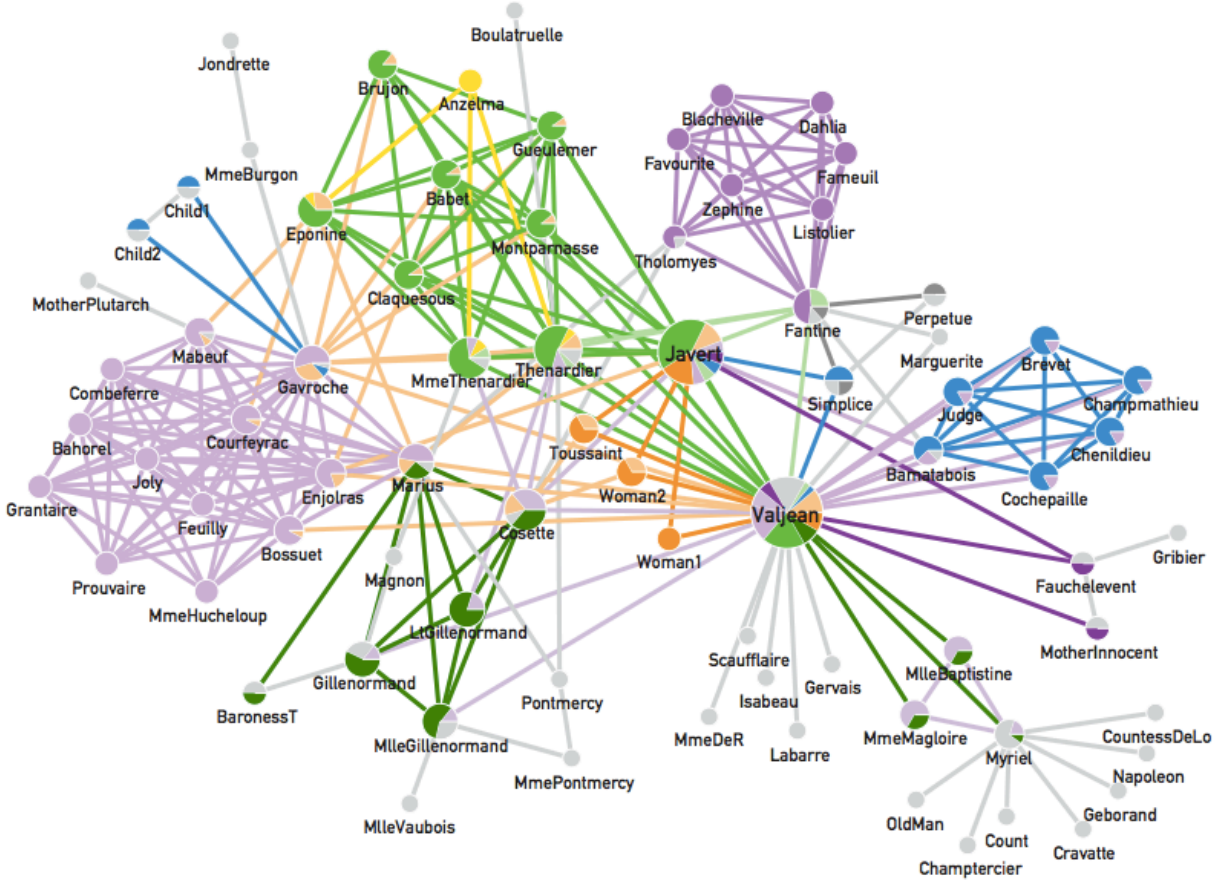
2. Apply hierarchical clustering (agglomerative, single linkage)



Ahn, Bragow and Lehmann, *Nature* 466 (2010).

A.-L. Barabási, *Network Science: Communities.*

Section 5 Link Clustering

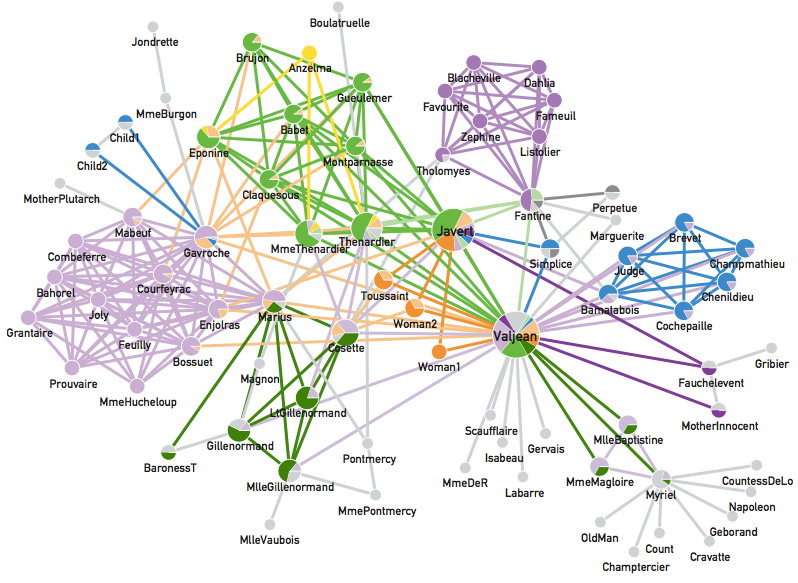


The network of characters in Victor Hugo's 1862 novel *Les Miserables*. Two characters are connected if they interact directly with each other in the story. The link colors indicate the clusters, grey nodes corresponding to single-link clusters. Each node is depicted as a pie-chart, illustrating its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

membership

Ahn, Bragow and Lehmann, *Nature* 466 (2010).

A.-L. Barabási, *Network Science: Communities.*



Computational complexity:

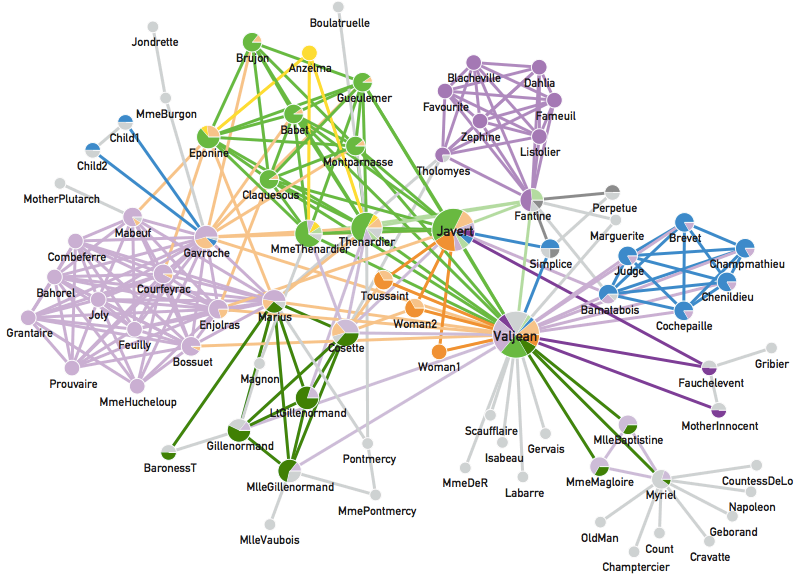
- Step 1: Comparison between two links requires $\max(k_1,k_2)$ steps. For scale free networks the step has complexity

- Step 2: hierarchical clustering requires $O\left(L^2\right)$

Ahn, Bragow and Lehmann, *Nature* 466 (2010).

A.-L. Barabási, *Network Science: Communities.*

Section 5 Link Clustering

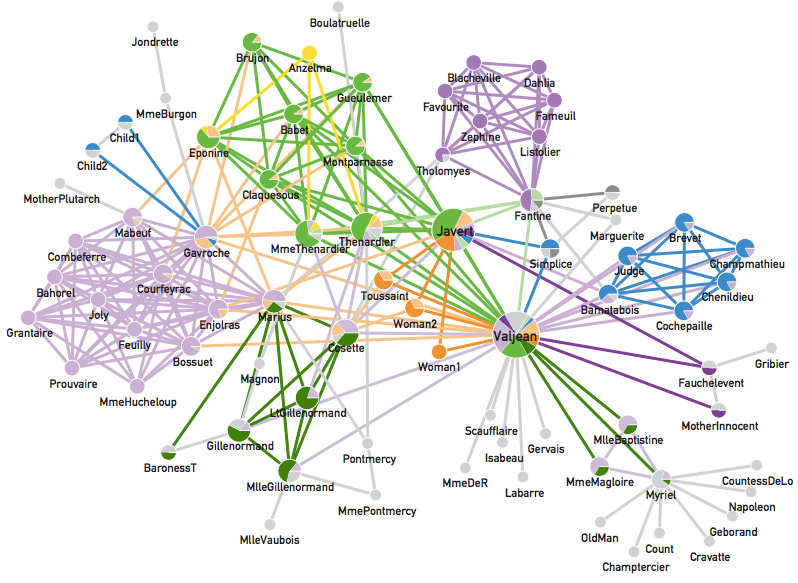


Computational complexity:

- Step 1: Comparison between two links requires $\max(k_1,k_2)$ steps. For scale free networks the step has complexity $O\left(N^{\frac{2}{\gamma-1}}\right)$
- Step 2: hierarchical clustering requires $O\left(L^2\right)$

Ahn, Bragow and Lehmann, *Nature* 466 (2010).

A.-L. Barabási, *Network Science: Communities.*



Computational complexity:

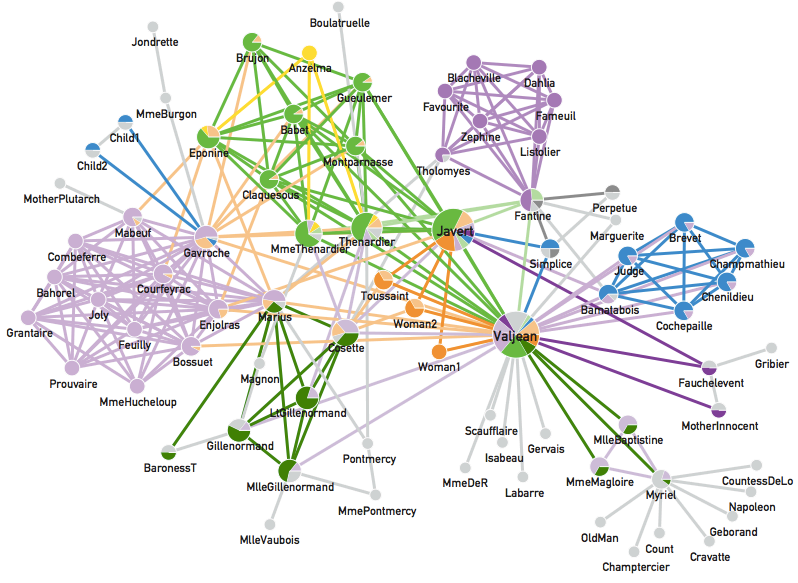
- Step 1: Comparison between two links requires $\max(k_1,k_2)$ steps. For scale free networks the step has complexity $O\left(N^{\frac{2}{\gamma-1}}\right)$
- Step 2: hierarchical clustering requires $O\left(L^2\right)$

for sparse networks

Ahn, Bragow and Lehmann, *Nature* 466 (2010).

A.-L. Barabási, *Network Science: Communities.*

Link Clustering



Computational complexity:

- Step 1: Comparison between two links requires max(k_1,k_2) steps. For scale free networks the step has complexity $O\left(N^{\frac{2}{\gamma-1}}\right)$

for sparse networks → $O\left(N^2\right)$

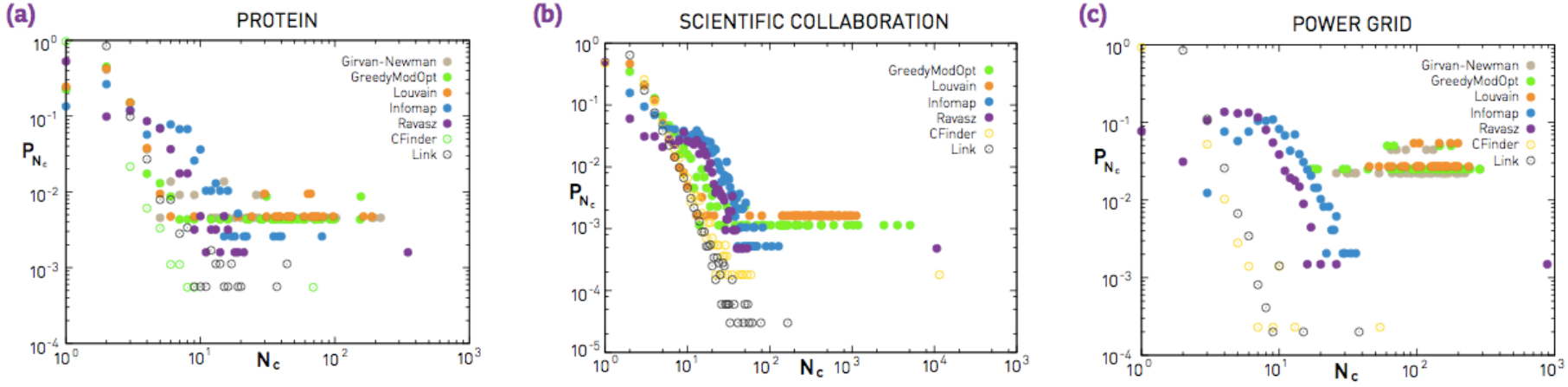
- Step 2: hierarchical clustering requires $O\left(L^2\right)$

Ahn, Bragow and Lehmann, *Nature* 466 (2010).

A.-L. Barabási, *Network Science: Communities.*

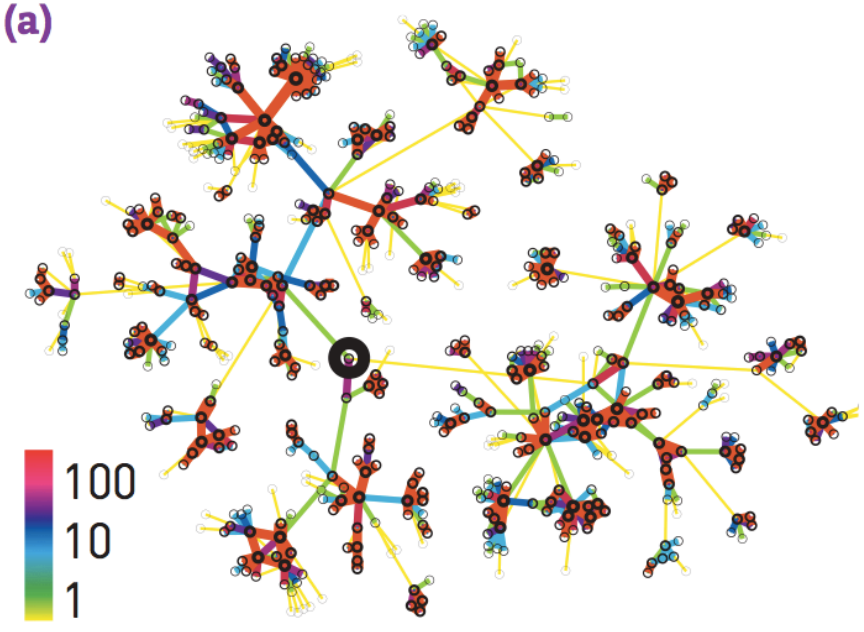
Characterizing Communities

Community size is distributed according to a power-law

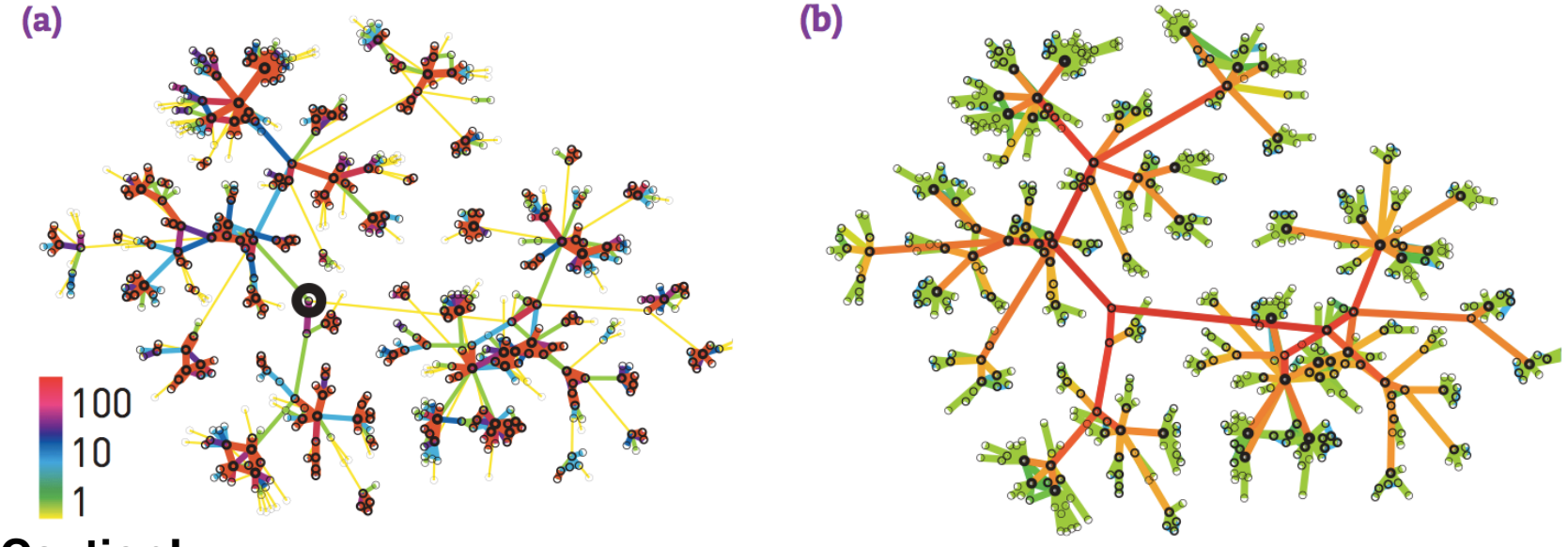


Not a subproduct of algorithms

Weights can help identifying communities



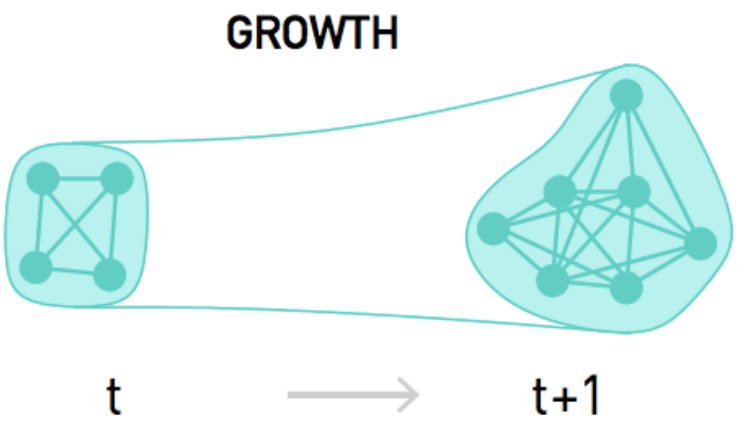
Weights can help identifying communities



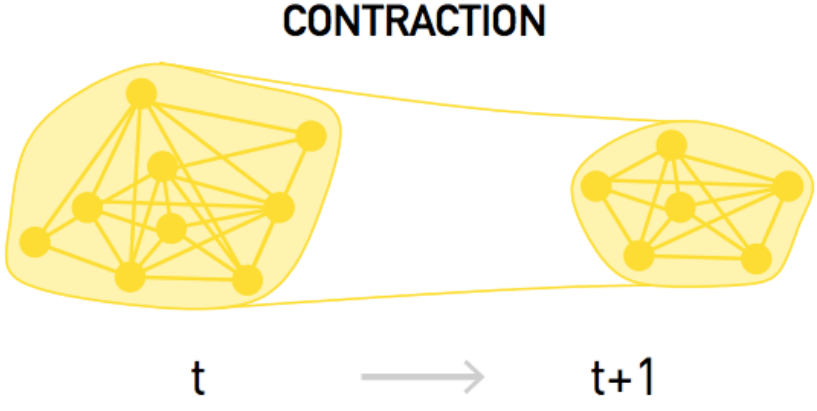
Caution!

- In social networks (a), the weak tie hypothesis says that communities are connected by links with smaller weight
- In transport systems, high betweenness links (correlates with weight, (b)) connect different communities

Community evolution

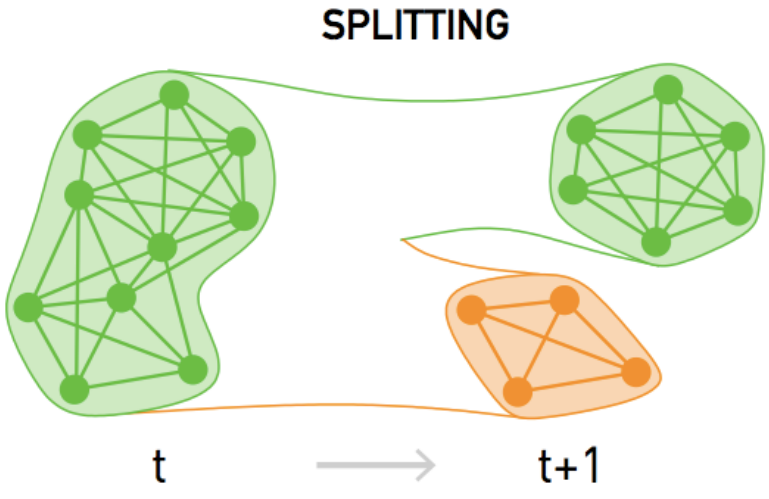


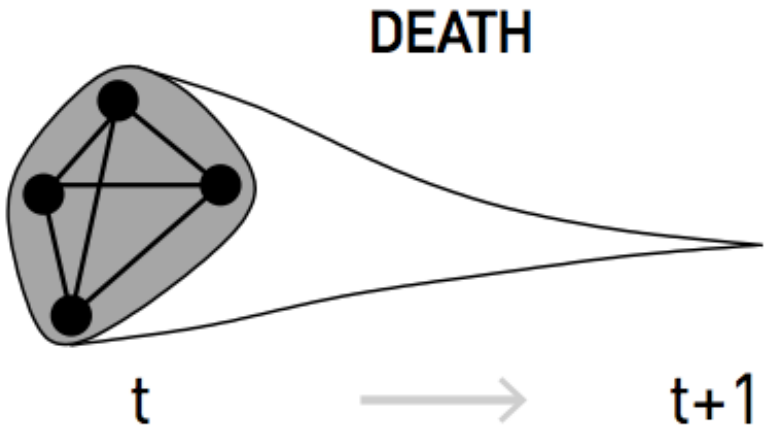
Probability that a node joins a community grows with the # of links the node has with the community



Node with fewer links within the community are more like to leave than those with more links

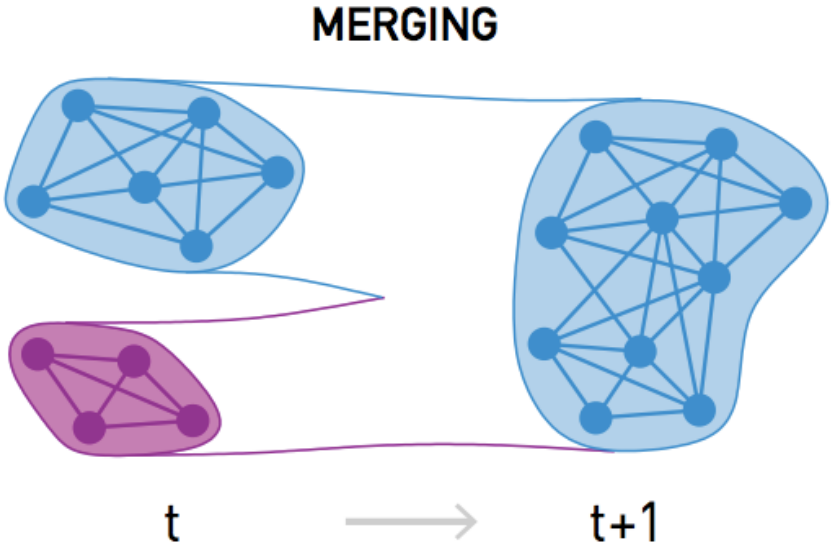
Community evolution

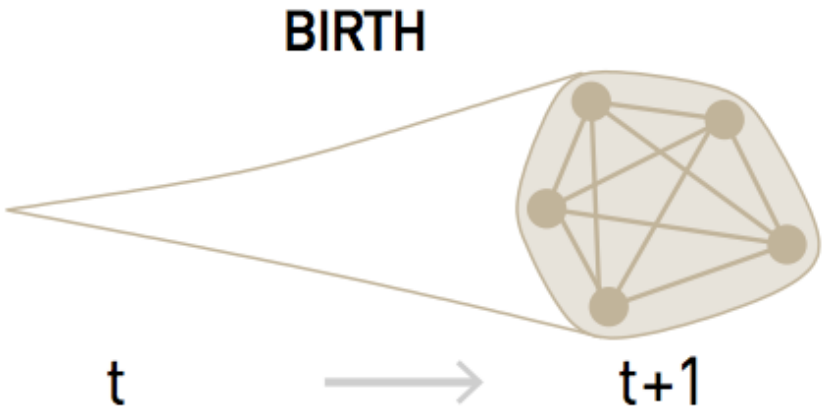




Probability that a community disintegrates increases with the aggregate link weights to nodes outside the community

Community evolution





Older communities tend to be larger

Newman-Girvan (NG) Benchmark

- N nodes in $n_c=4$ communities
- Nodes in the same community connected with probability p^{int}, otherwise p^{ext}

S. Fortunato, *Phys. Rep.* 486 (2010)

A.-L. Barabási, *Network Science: Communities.*

Newman-Girvan (NG) Benchmark

- N nodes in n_c=4 communities
- Nodes in the same community connected with probability p^{int}, otherwise p^{ext}
- Control parameter $\mu = \frac{k^{ext}}{k^{ext} + k^{int}}$

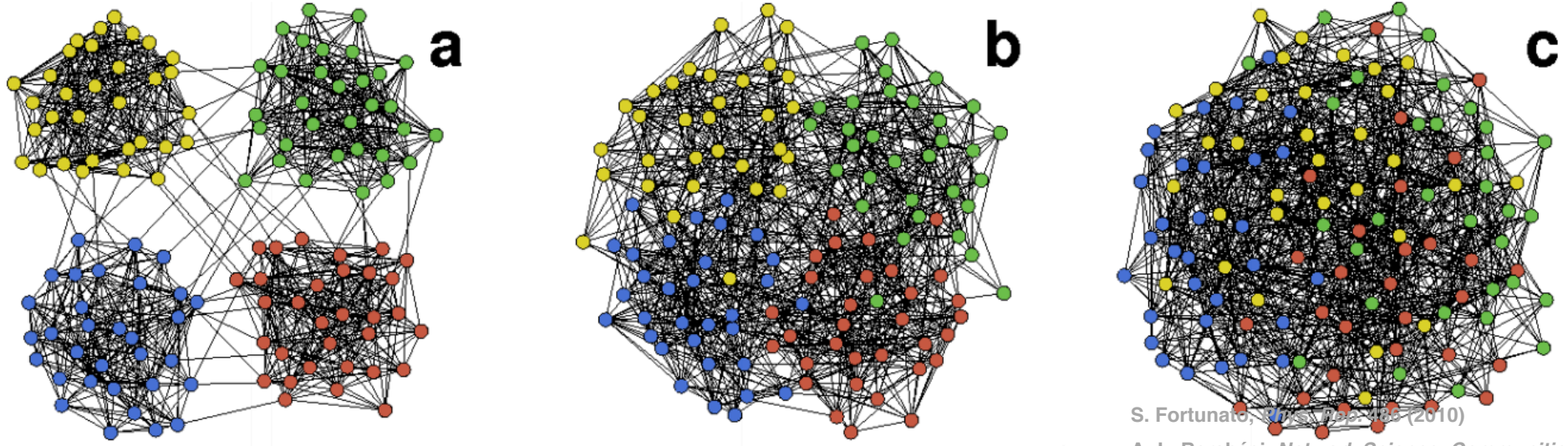
S. Fortunato, *Phys. Rep.* 486 (2010)

A.-L. Barabási, *Network Science: Communities*

Newman-Girvan (NG) Benchmark

- N nodes in n_c=4 communities
- Nodes in the same community connected with probability p^{int}, otherwise p^{ext}
- Control parameter $\mu = \dfrac{k^{ext}}{k^{ext} + k^{int}}$

$k^{ext} \ll k^{int}$ $k^{ext} < k^{int}$ $k^{ext} = k^{int}$



S. Fortunato, *Phys. Rep.* 486 (2010)

A.-L. Barabási, *Network Science: Communities.*

Measuring Accuracy

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**

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

L. Danon et al, *JStat.* P09008 (2005)
S. Fortunato, *Phys. Rep.* 486 (2010)

Measuring Accuracy

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**

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

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

$$p\left(C_1,C_2\right) = \frac{\text{how many nodes that are in } C_1 \text{ are also in } C_2}{\text{sum over all possible pairs } C_1 \text{ and } C_2} = \frac{N_{C_1C_2}}{\sum_{C_1,C_2} N_{C_1C_2}}$$

L. Danon et al, *JStat.* P09008 (2005)
S. Fortunato, *Phys. Rep.* 486 (2010)

Measuring Accuracy

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**

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

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

$$p\left(C_1,C_2\right) = \frac{\text{how many nodes that are in } C_1 \text{ are also in } C_2}{\text{sum over all possible pairs } C_1 \text{ and } C_2} = \frac{N_{C_1C_2}}{\sum_{C_1,C_2} N_{C_1C_2}}$$

Probability that a randomly chosen node belongs to community C_1

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

L. Danon et al, *JStat.* P09008 (2005)
S. Fortunato, *Phys. Rep.* 486 (2010)

Measuring Accuracy

The mutual information is not ideal as a similarity measure as it is: in fact, given a partition C, all partitions derived from C by further partitioning its clusters would all have the same mutual information with C, even though they could be very different from each other. In this case the mutual information would simply equal the entropy H(X), because the conditional entropy would be systematically zero. Also the measure is not normalized.

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

Measuring Accuracy

The mutual information is not ideal as a similarity measure as it is: in fact, given a partition C, all partitions derived from C by further partitioning its clusters would all have the same mutual information with C, even though they could be very different from each other. In this case the mutual information would simply equal the entropy H(X), because the conditional entropy would be systematically zero. Also the measure is not normalized.

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

Shannon entropy

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

Measuring Accuracy

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

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

Measuring Accuracy

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

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

→ $I_n = 1$ If the communities are identical

→ $I_n = 0$ If the communities are independent

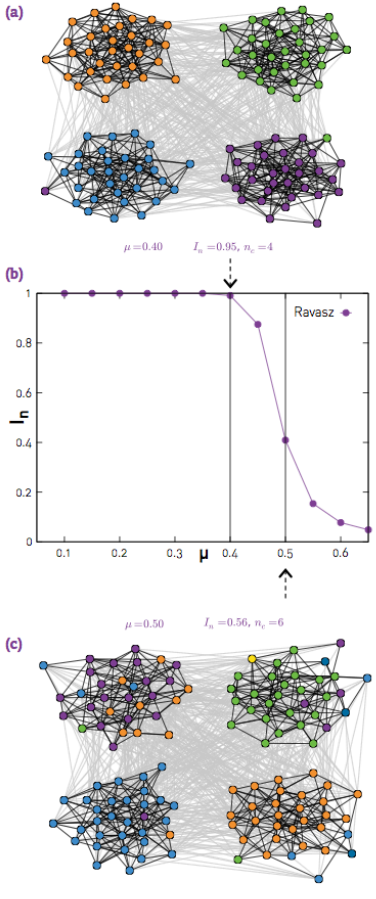
Measuring Accuracy

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

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

→ $I_n = 1$ If the communities are identical

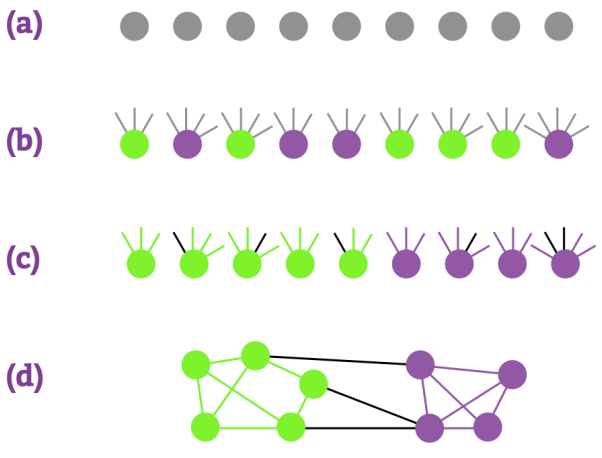
→ $I_n = 0$ If the communities are independent

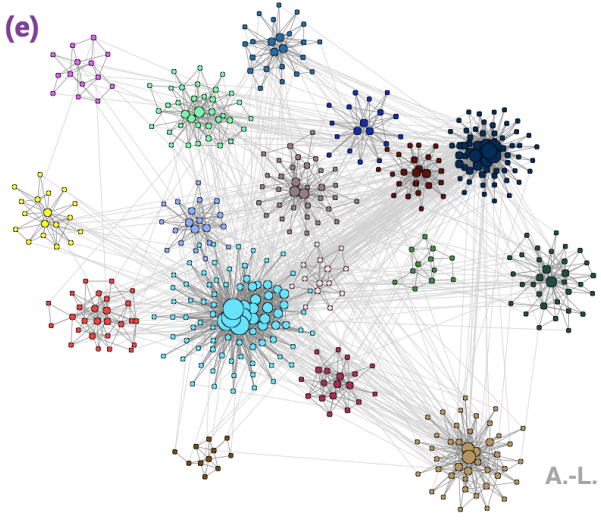


Lancichinetti-Fortunato-Radicchi (LFR) Benchmark

Lancichinetti-Fortunato-Radicchi (LFR) Benchmark

- N nodes in N_C communities, taken from $P_{Nc} \sim N_C^{-\xi}$
- Each node is assigned a degree k from $P_k \sim k^{-\gamma}$
- Each node i is assigned to a community, and receives an internal degree $(1-\mu)k_i$ and an external degree μk_i.
- Connect randomly nodes, according to the constraints above, until there are no more free stubs





A.-L. Barabási, *Network Science: Communities.*

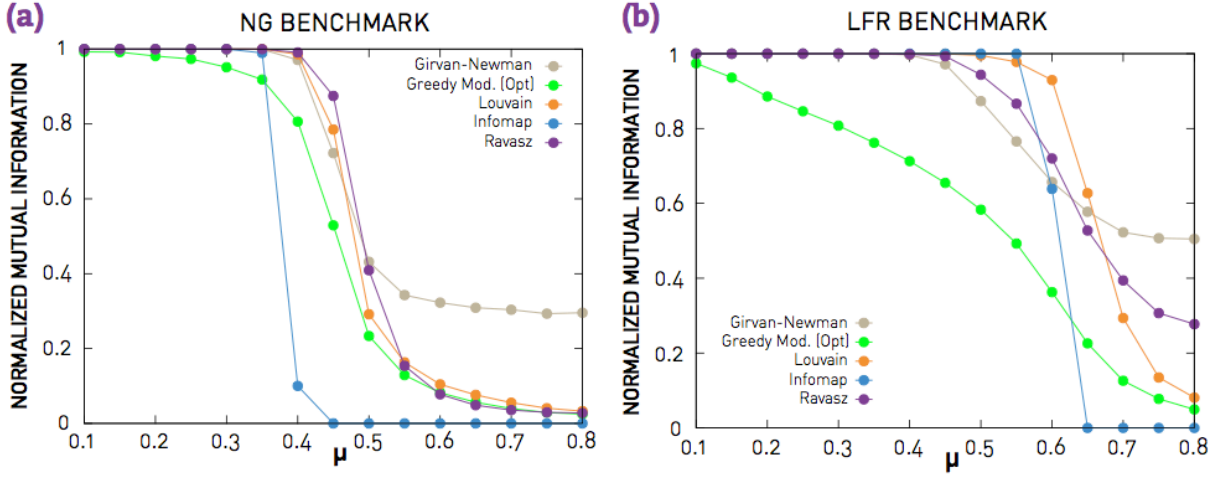
Measuring Accuracy

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

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

→ $I_n = 1$ If the communities are identical

→ $I_n = 0$ If the communities are independent

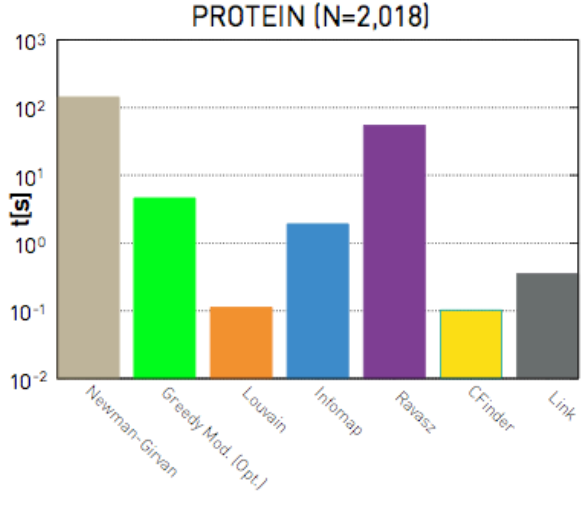


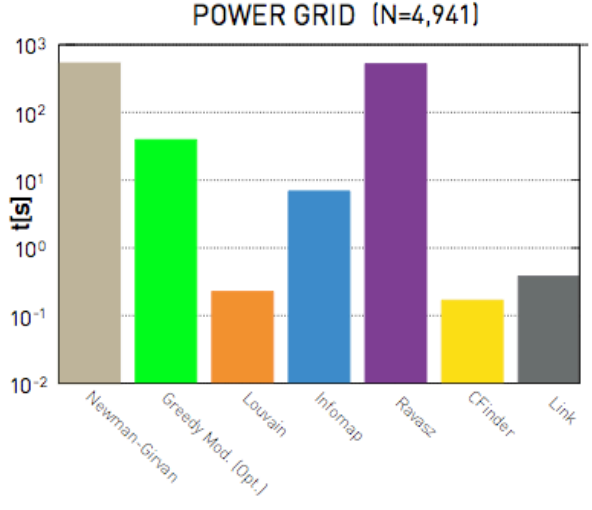
A.-L. Barabási, *Network Science: Communities.*

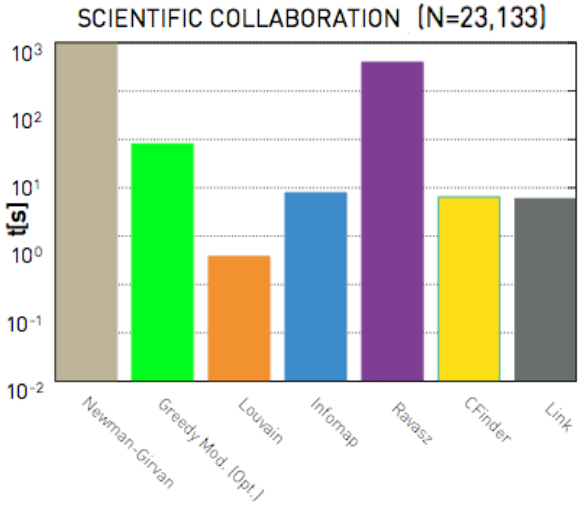
Section 7 Testing Communities

Speed

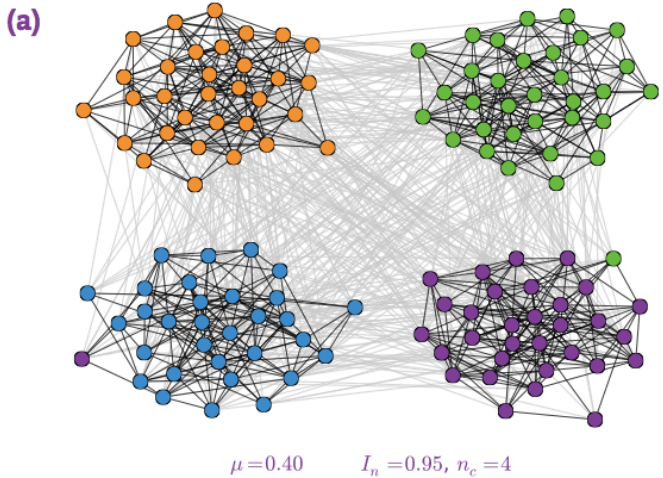
NAME	NATURE	COMP.
Ravasz	Hierarchical Agglomerative	$O(N^3)$
Girvan-Newman	Hierarchical Divisive	$O(N^3)$
Greedy Modularity	Modularity Optimization	$O(N^2)$
Greedy Modularity	Modularity Optimization	$O(N\log^2 N)$
Louvain	Modularity Optimization	$O(N\log N)$
Infomap	Flow Optimization	$O(N\log N)$
Clique Percolation	Overlapping Communities	$Exp(N)$
Link Clustering CFinder	Hierarchical Agglomerative; Overlapping Communities	$O(N^2)$

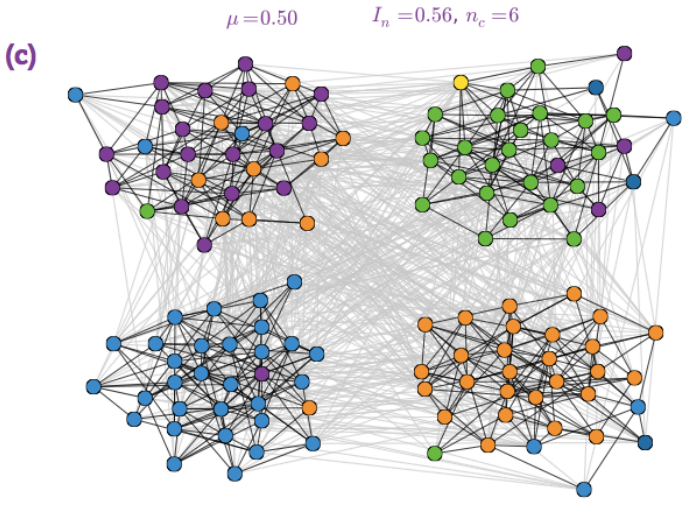




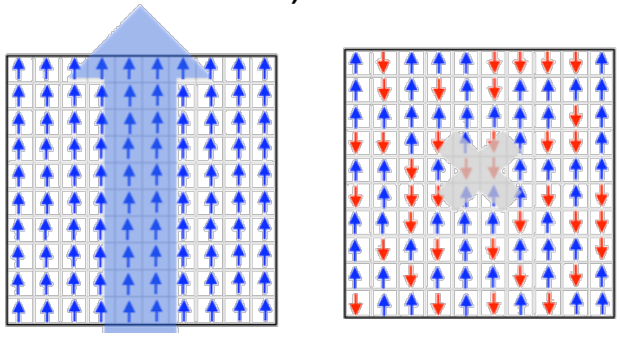


Ravasz algorithm on NG Benchmark





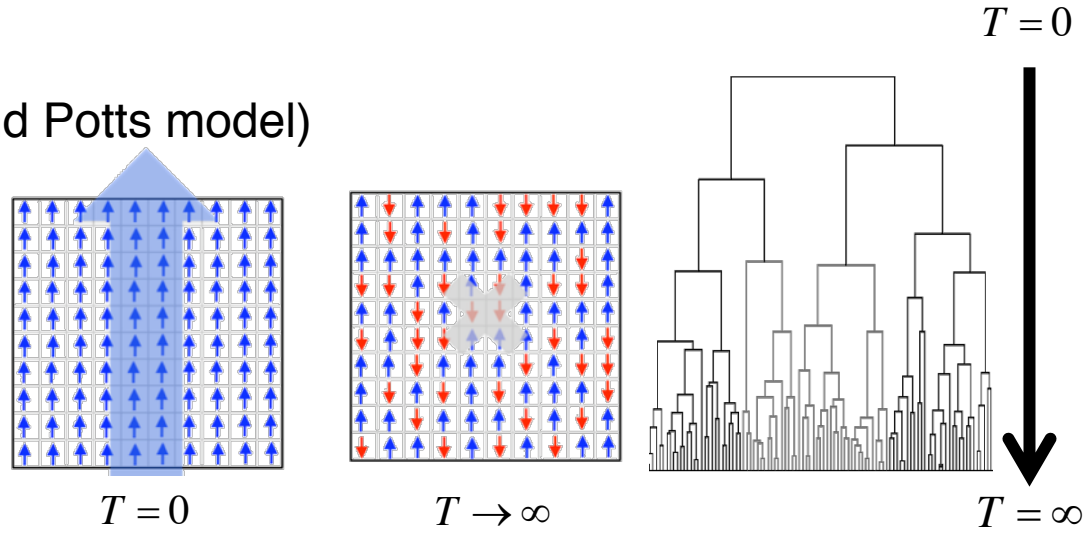
→ **Spin models** (Ising and Potts model)



$T = 0$ $T \to \infty$

S. Fortunato, *Phys. Rep.* 486 (2010)

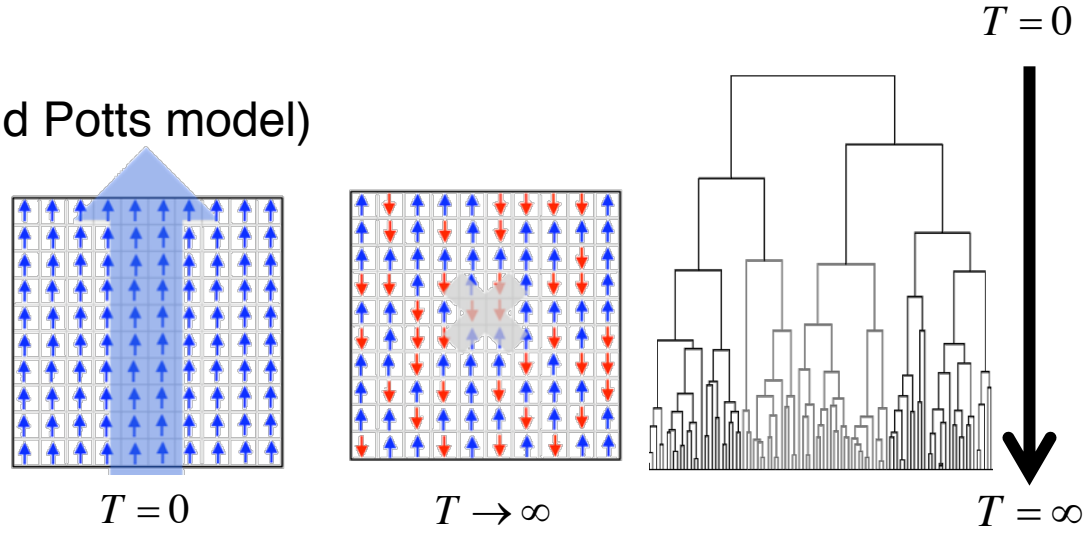
→ **Spin models** (Ising and Potts model)



S. Fortunato, *Phys. Rep.* 486 (2010)

Section 6 “Dynamic” algorithms

→ **Spin models** (Ising and Potts model)

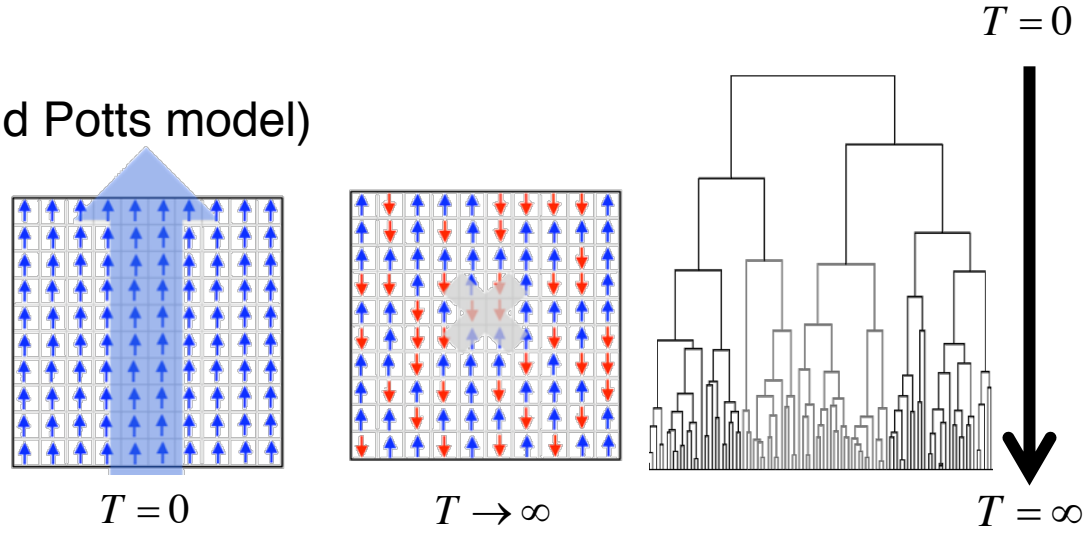


→ **Synchronization**. Each node is an oscillator (for example, Kuramoto). Nodes that are in the same community synchronize with each other sooner

S. Fortunato, *Phys. Rep.* 486 (2010)

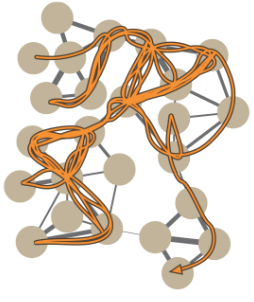
Section 6 "Dynamic" algorithms

→ **Spin models** (Ising and Potts model)



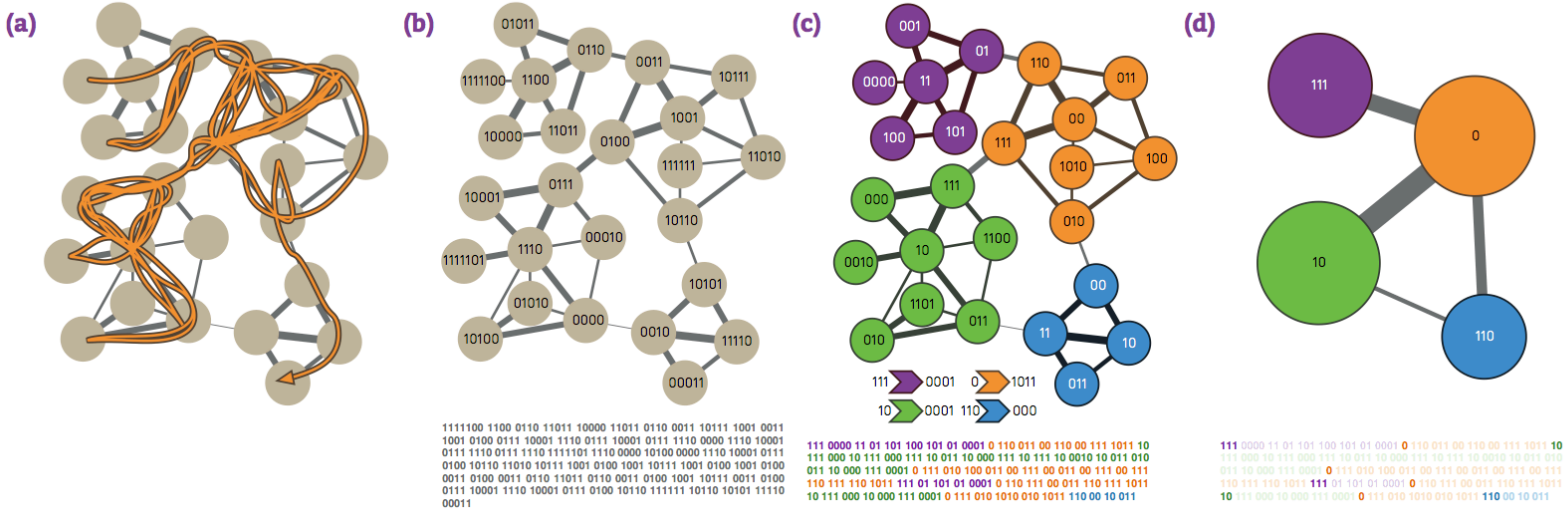
→ **Synchronization**. Each node is an oscillator (for example, Kuramoto). Nodes that are in the same community synchronize with each other sooner

→ **Random Walks** (Infomap)



S. Fortunato, *Phys. Rep.* 486 (2010)

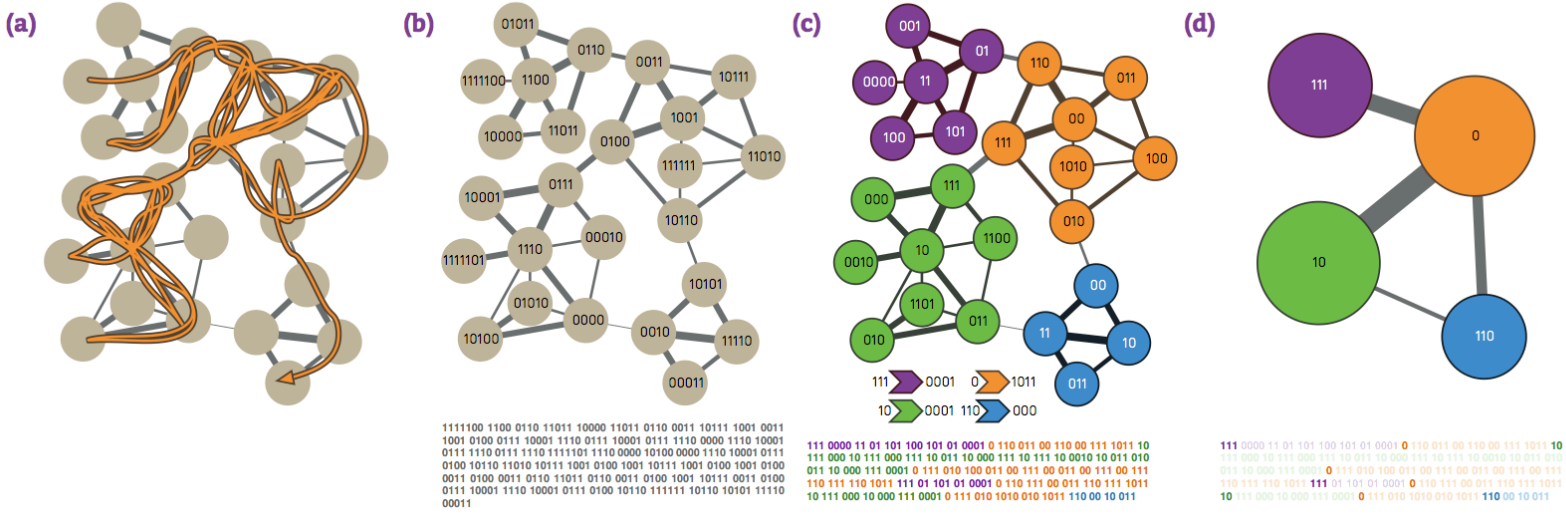
Section 6 Infomap



S. Fortunato, *Phys. Rep.* 486 (2010)

A.-L. Barabási, *Network Science: Communities.*

Section 6 Infomap

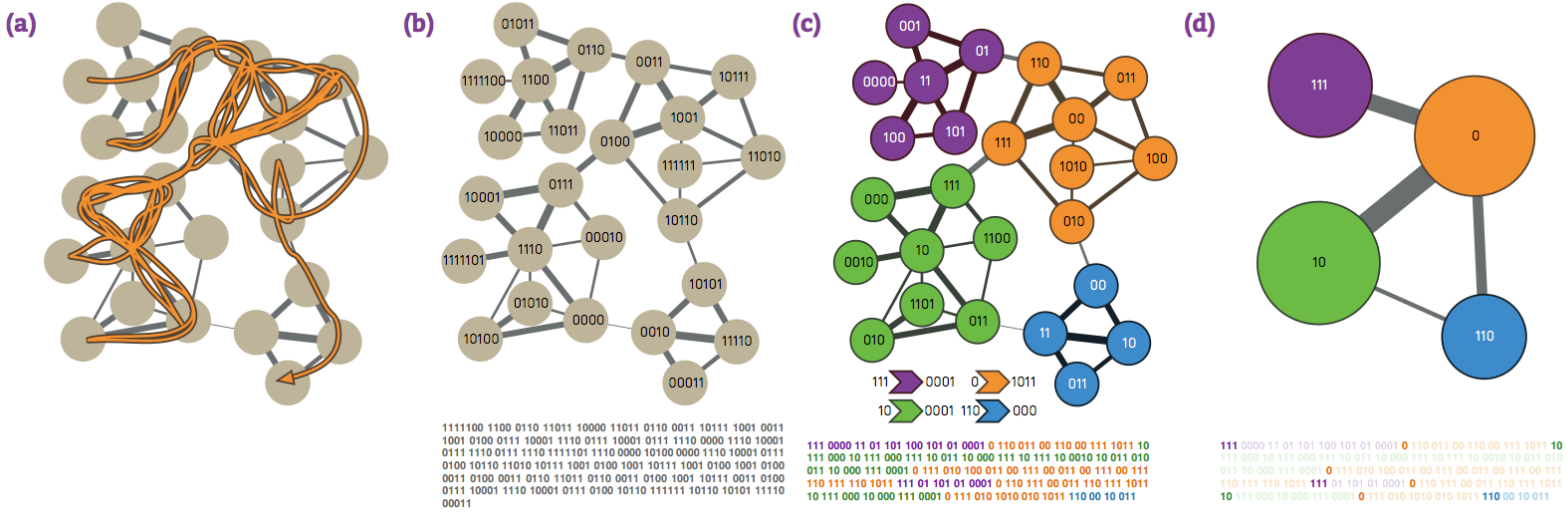


Maximize the map equation: $$L(C) = qH(Q) + \sum_{i=1}^{n_c} p^i H(p^i).$$

S. Fortunato, *Phys. Rep.* 486 (2010)

A.-L. Barabási, *Network Science: Communities.*

Section 6 Infomap



Maximize the map equation:

$$L(C) = qH(Q) + \sum_{i=1}^{n_c} p^i H(p^i).$$

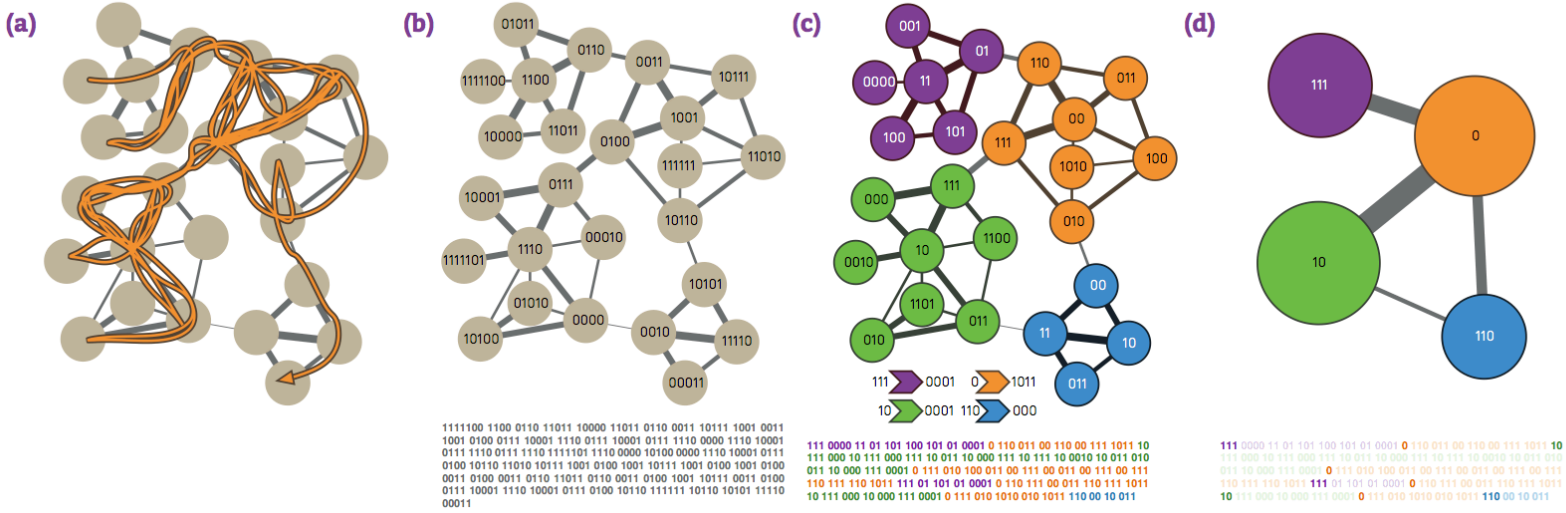
Info to describe **between module** movements

Info to describe **within module** movements

S. Fortunato, *Phys. Rep.* 486 (2010)

A.-L. Barabási, *Network Science: Communities.*

Section 6 Infomap



Maximize the map equation: $L(C) = qH(Q) + \sum_{i=1}^{n_c} p^i H(p^i)$.

Info to describe **within module** movements

Info to describe **between module** movements

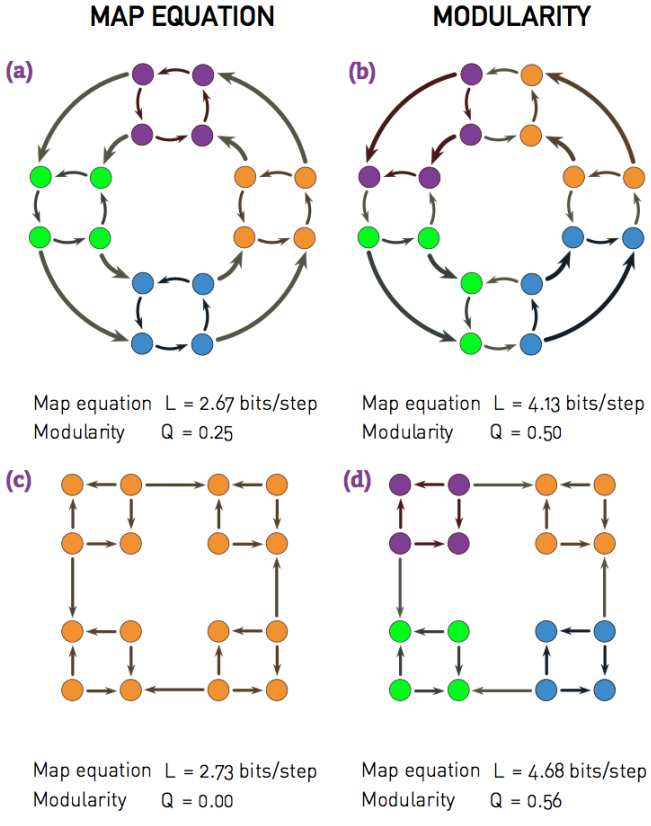
Code: http://www.mapequation.org/

S. Fortunato, *Phys. Rep.* 486 (2010)

A.-L. Barabási, *Network Science: Communities.*

Section 6

Map Equation vs Modularity



S. Fortunato, *Phys. Rep.* 486 (2010)

A.-L. Barabási, *Network Science: Communities.*

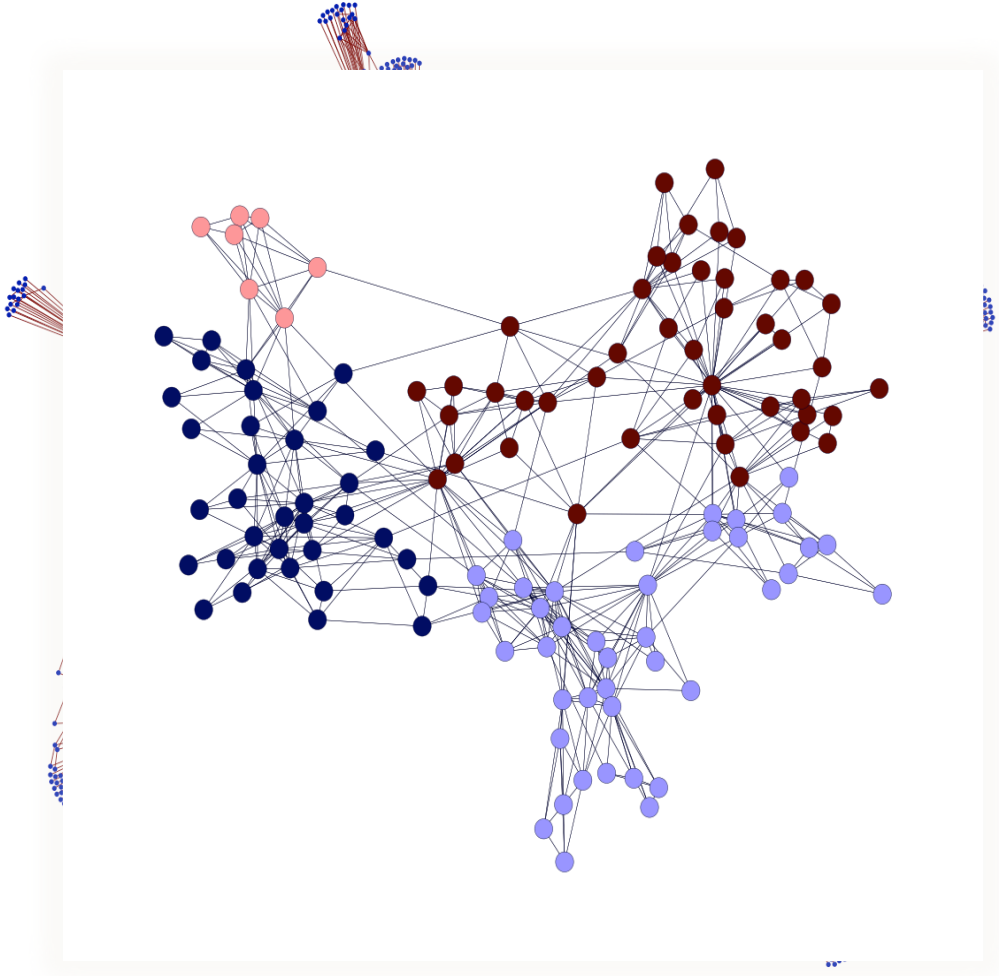
Section 6

“Dynamic” algorithms

S. Fortunato, *Phys. Rep.* 486 (2010)

Summary

ZOOMING INTO NETWORKS



Section 8

Summary

Summary

→ **How do we know that a network has communities?**

Summary

→ How do we know that a network has communities?

→ We have a number of hypothesis…

Summary

H1 (Fundamental Hypothesis): A network's community structure is uniquely encoded in its wiring diagram

H2 (Connectedness Hypothesis): A community corresponds to a connected subgraph.

H3 (Density Hypothesis): A community corresponds to locally dense neighborhood of a network.

H4 (Random Hypothesis): Randomly wired networks are not expected to have a community structure.

H5 (Maximal Modularity Hypothesis): The partition with the maximum modularity offers the best community structure, where modularity is given by

$$M = \sum_{c=1}^{n} \left[\frac{l_c}{L} - \left(\frac{d_c}{2L} \right)^2 \right].$$

Summary

→ **How do we know that a network has communities?**

→ **We have a number of hypothesis… can we prove them (hypothesis vs theorems)?**

H1 (Fundamental Hypothesis)
H2 (Connectedness Hypothesis)
H3 (Density Hypothesis)
H4 (Random Hypothesis)
H5 (Maximal Modularity Hypothesis)

→ **Force all nodes into communities?**

Summary

Community identification relies on several fundamental hypotheses, pertaining to the nature of communities:

(i) Connectedness Hypothesis: A community corresponds to a connected subgraph.

(ii) *Density Hypothesis*: A community corresponds to locally dense neighborhood of a network.

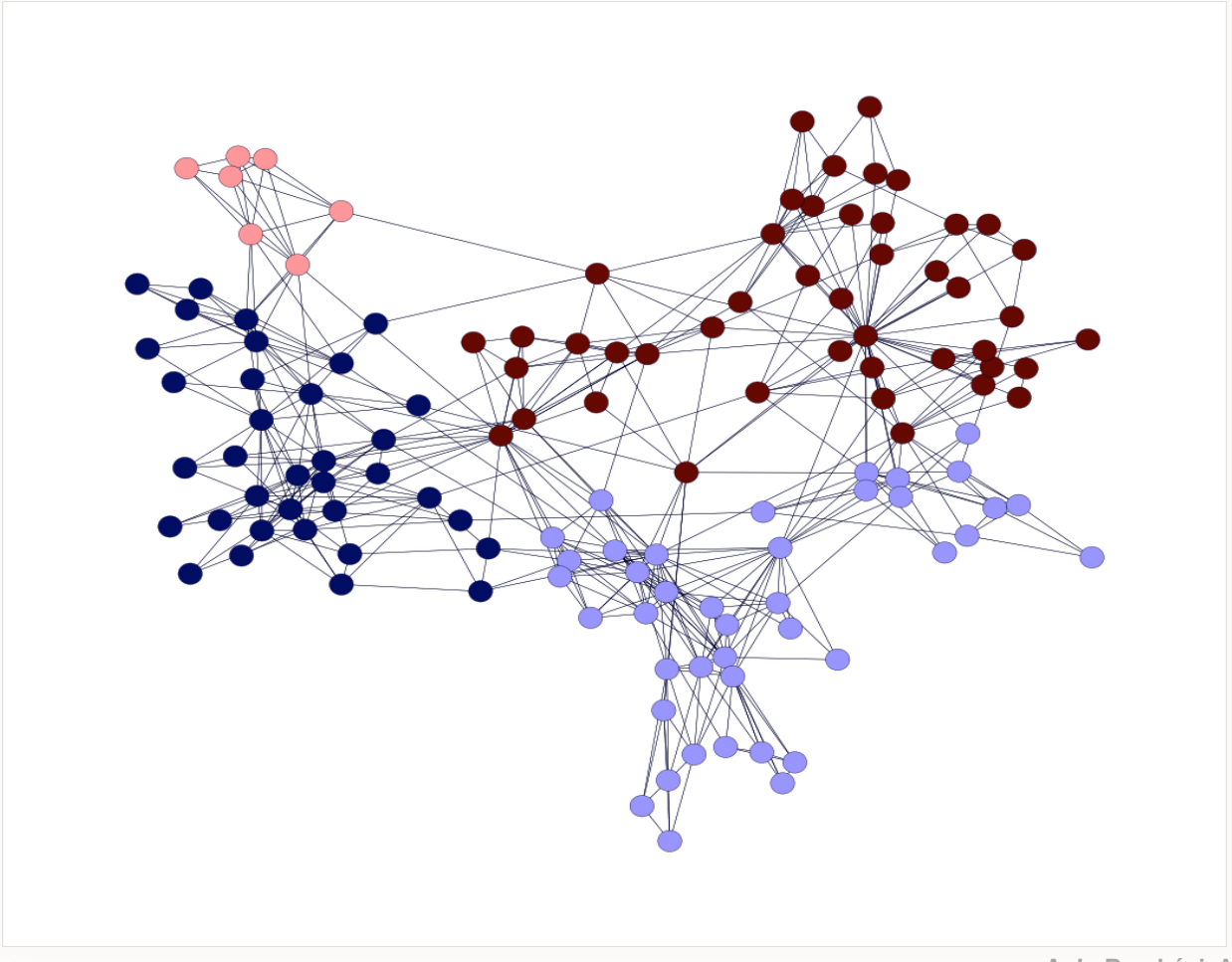
(iii) *Random Hypothesis*: Randomly wired networks are not expected to have a community structure.

(iv) Maximal Modularity Hypothesis: The partition with the maximum modularity offers the best community structure, where modularity is given by

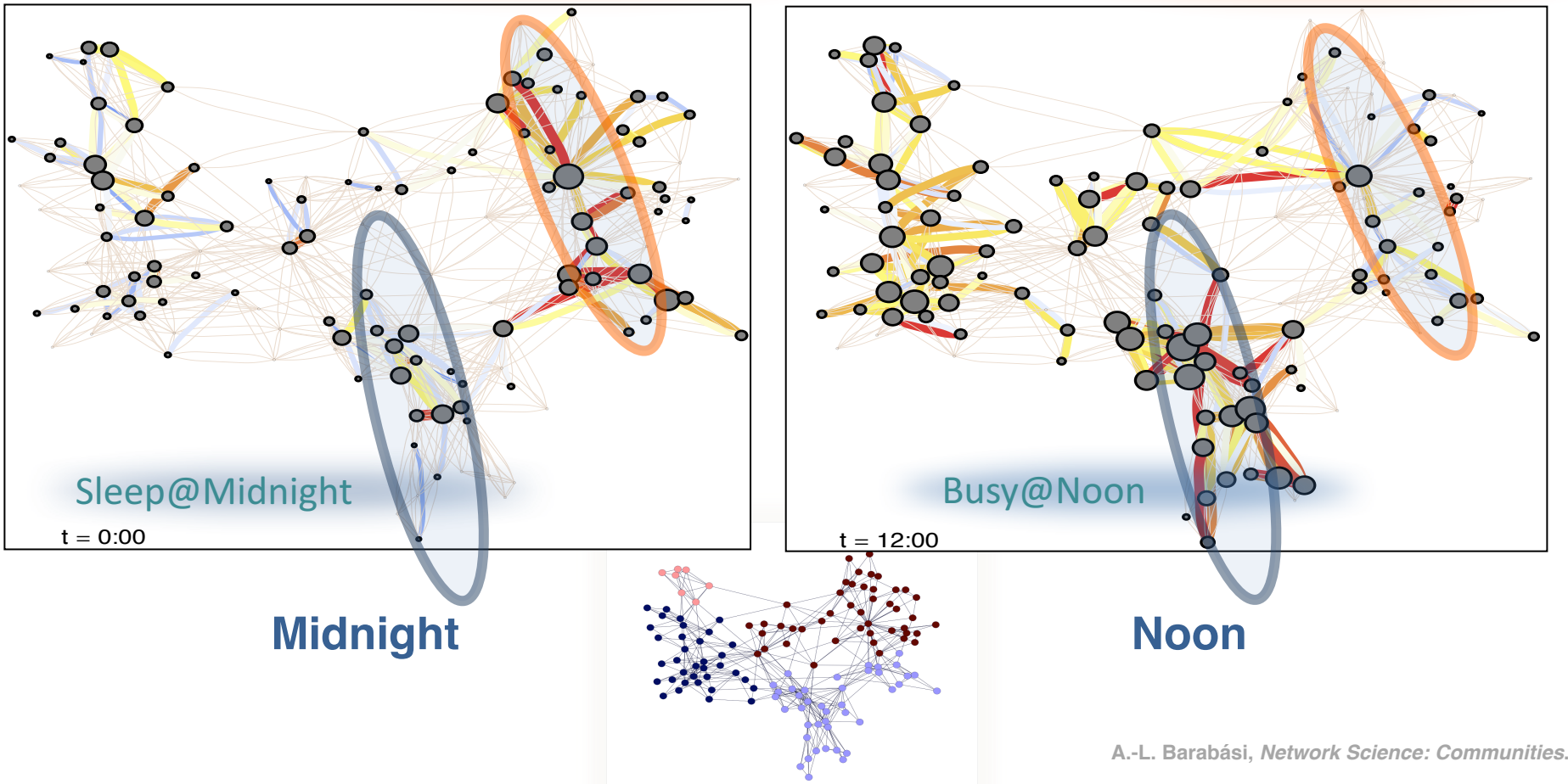
$$M = \sum_{c=1}^{n} \left[\frac{l_c}{L} - \left(\frac{d_c}{2L} \right)^2 \right].$$

Section 8

Summary



Section 8 Summary



A.-L. Barabási, *Network Science: Communities.*

CALLING PATTERNS AT MIDNIGHT AND NOON