Network Science

Communities Part 1

Sean P. Cornelius

With

Emma K. Towlson and Albert-László Barabási

www.BarabasiLab.com

Questions

1) What is a community (intuitively)? Examples from the real world. Zachary's Karate Club.
2) Fundamental hypotheses H1 and H2. Basic definitions (strong, weak, cliques). Clearly define "community" vs. "partition".
3) Graph partitioning and its computational complexity. The Bell number. Why is delineating communities hard?
4) Hierarchical clustering: the Ravasz algorithm and its computational complexity.
5) Hierarchical clustering: the Girvan-Newman algorithm and its complexity.
6) Hierarchy in real networks.
7) Modularity. Hypotheses H3 and H4. The greedy algorithm and its complexity.

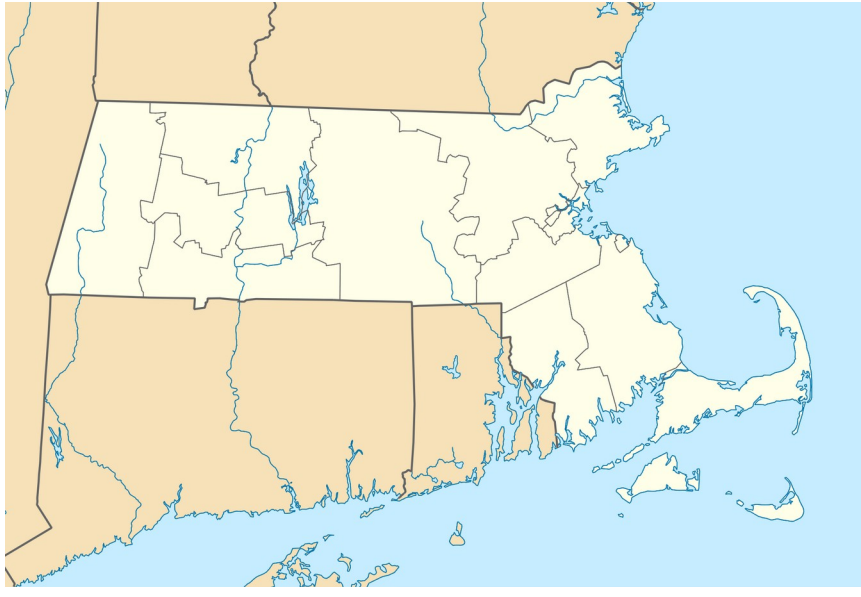
Introduction

Section 1

Introduction: Belgium

Section 1

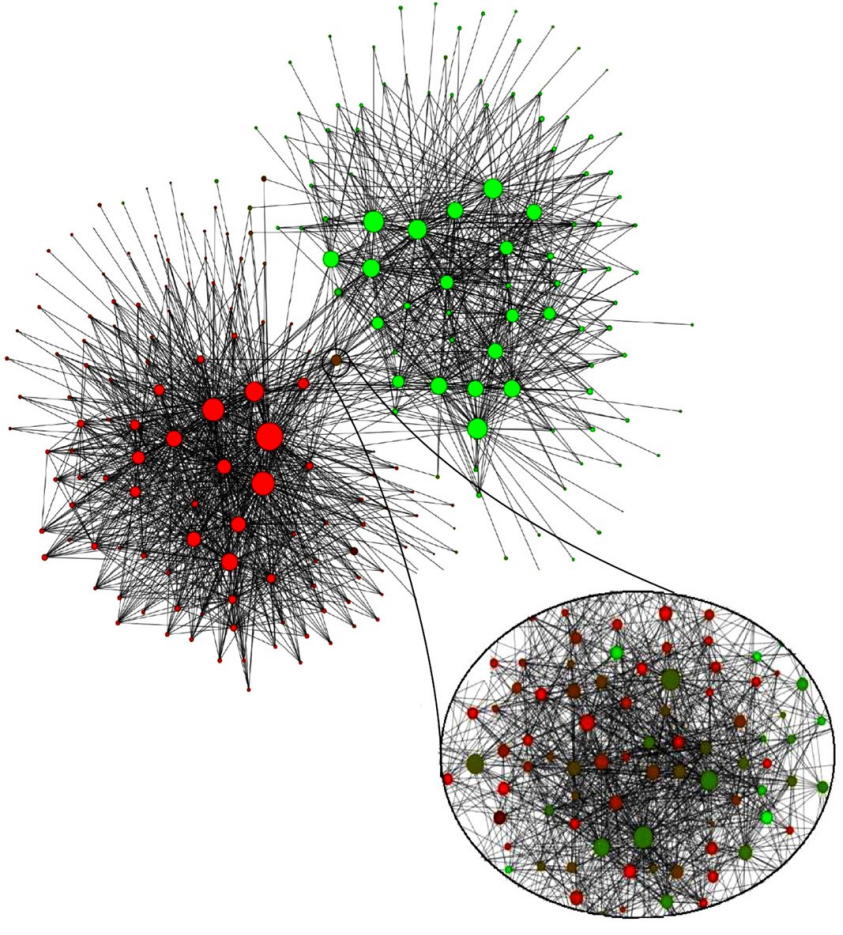
Introduction: Belgium



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

Section 1

Introduction: Belgium



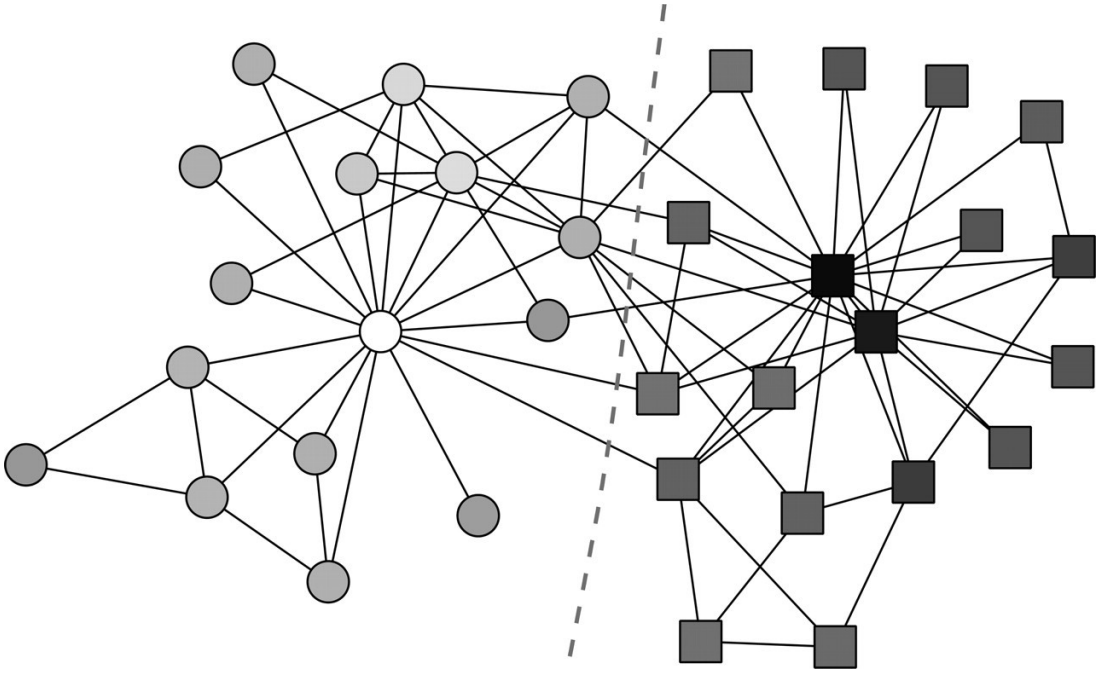
V.D. Blondel et al, *J. Stat. Mech*. P10008 (2008).

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

Examples of communities

Section 2

Zachary's Karate Club

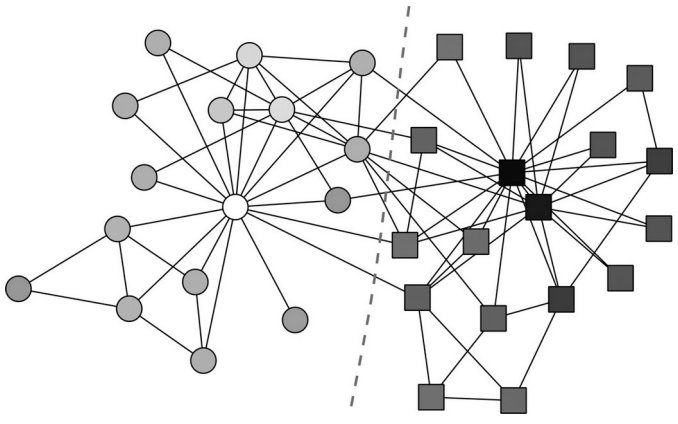


W.W. Zachary, *J. Anthropol. Res.* 33:452-473 (1977).

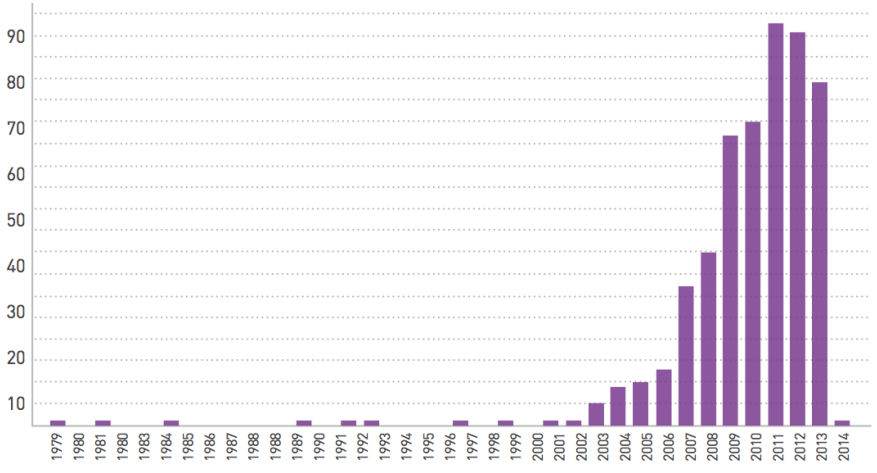
A.-L. Barabási, *Network Science: Communities*.

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

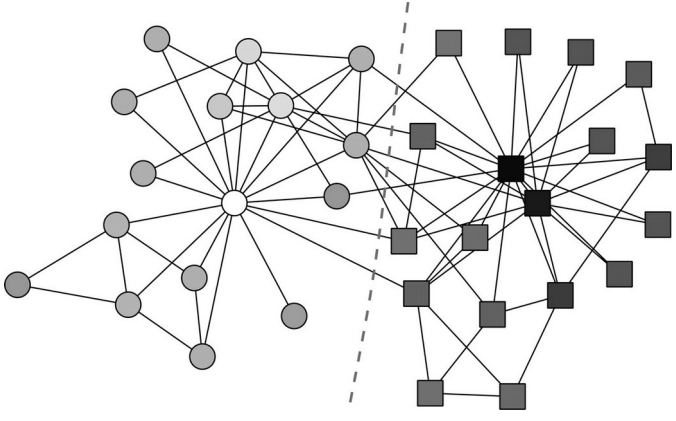
The first scientist at any conference on networks who uses Zachary's karate club as an example is inducted into the Zachary Karate Club Club, and awarded a prize.

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)

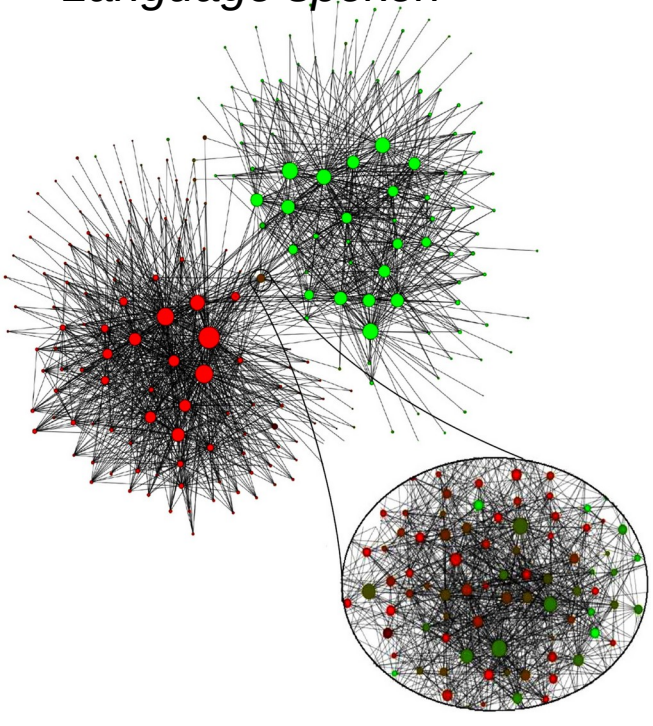
http://networkkarate.tumblr.com/)

Section 2 Auxiliary information

→ *Karate Club:*
Breakup of the club

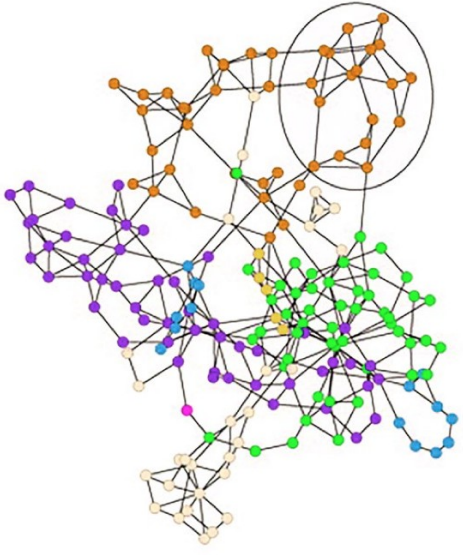


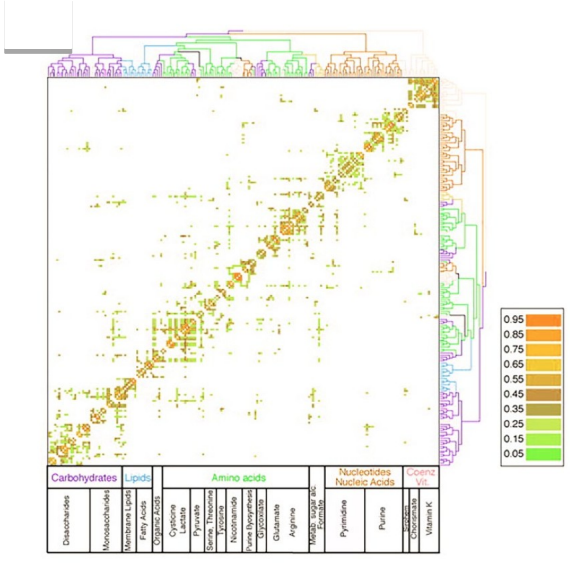
→ *Belgian Phone Data:*
Language spoken

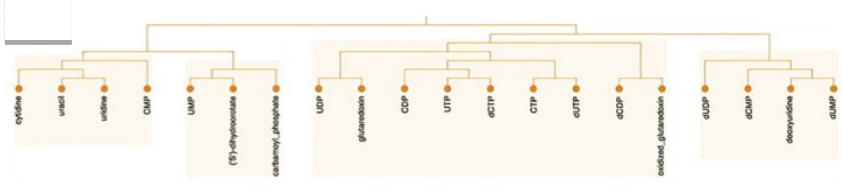


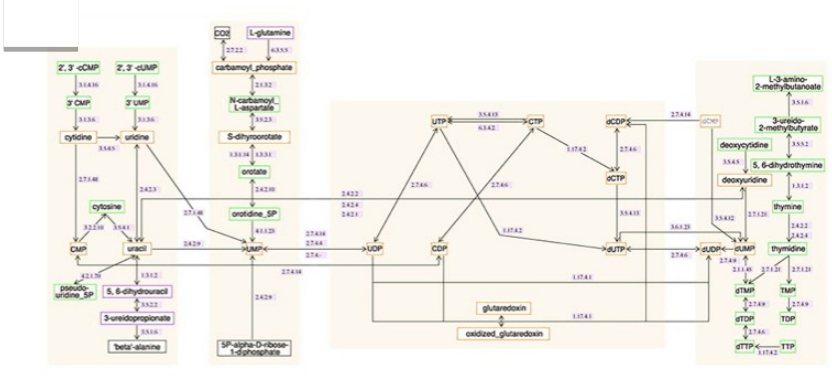
Section 2

Biological Modules







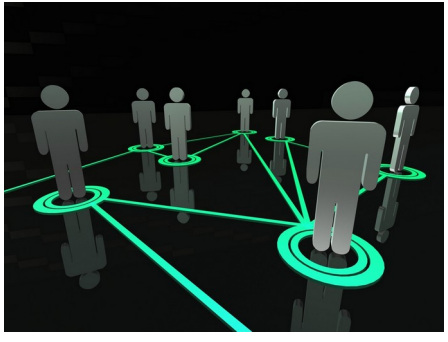


E. Ravasz et al., *Science* 297 (2002).

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

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*

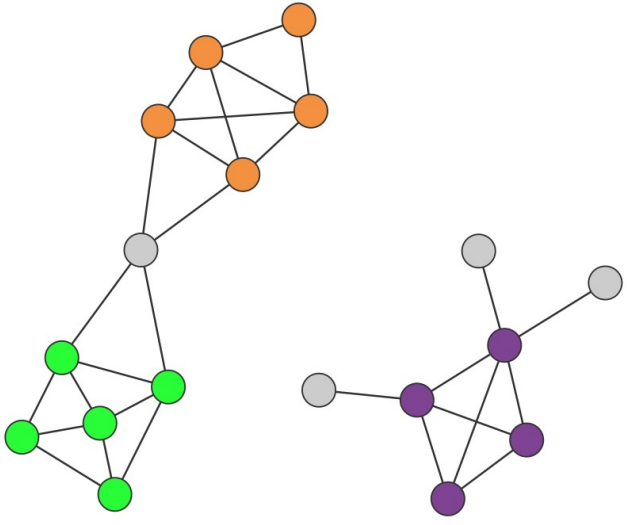
Communities correspond to locally dense neighborhoods of a network.

H2: *Connectedness Hypothesis*

A community corresponds to a connected subgraph.

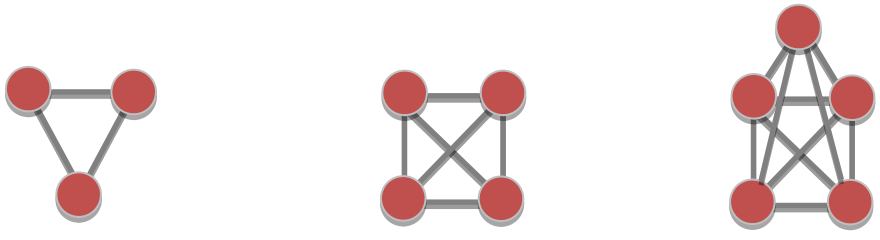
H3: *Density Hypothesis*

Communities correspond to locally dense neighborhoods of a network.



Cliques as communities

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

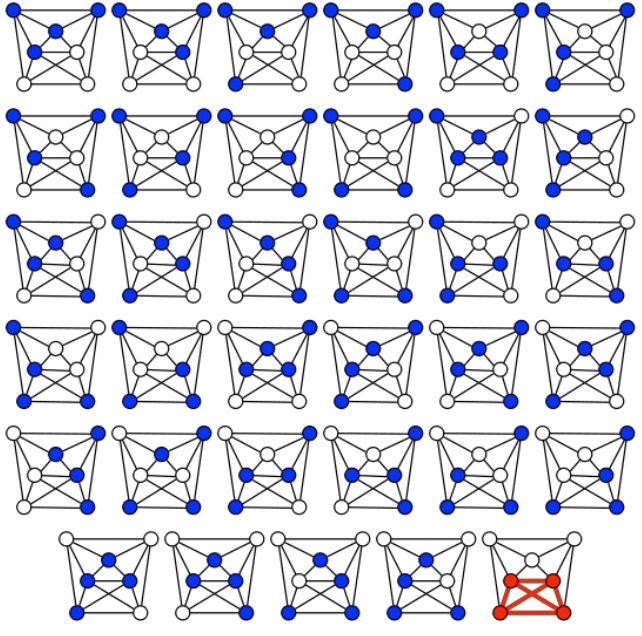


R.D. Luce & A.D. Perry, *Psychometrika* 14 (1949)

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

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

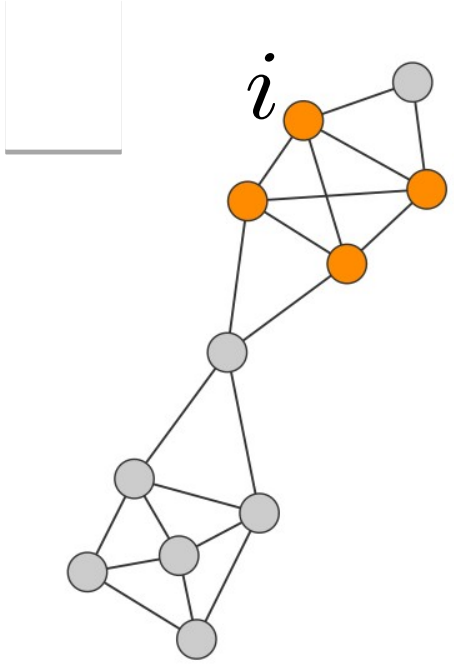
Consider a connected subgraph *C* of N_c nodes

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

External degree k_i^{ext}*:* number of links of node *i* that connect 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.

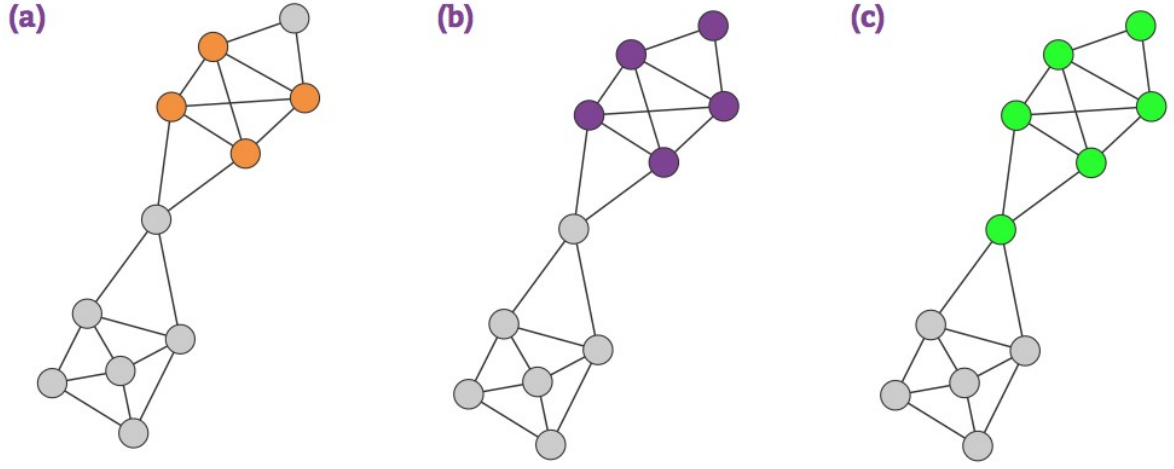


Strong community:
Each node of *C* has more links within the community than with the rest of the graph.

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

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

$$\sum_{i \in C} k_i^{\text{int}} > \sum_{i \in C} k_i^{\text{ext}}$$



Clique *Strong* *Weak*

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: $\frac{N!}{n_1!n_2!}$

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

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

$$n_1 = n_2 = N/2$$

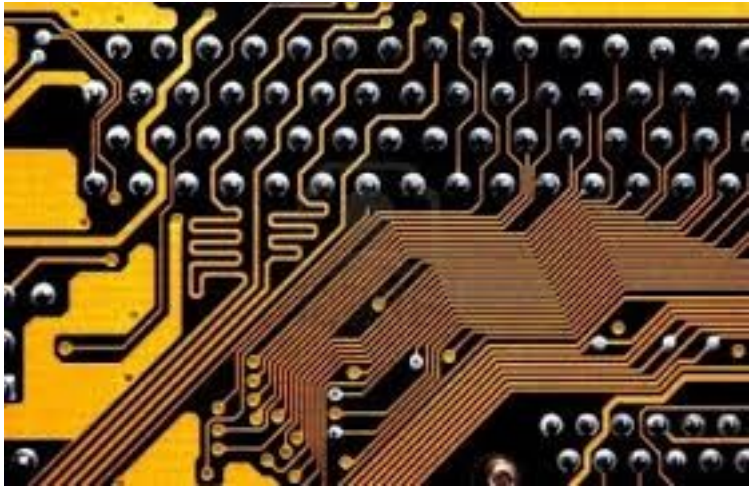
$$\frac{2^{N+1}}{\sqrt{N}} = e^{(N+1)\ln 2/\sqrt{N}}$$

N=10 → 256 partitions (1 ms)

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

Graph Partitioning

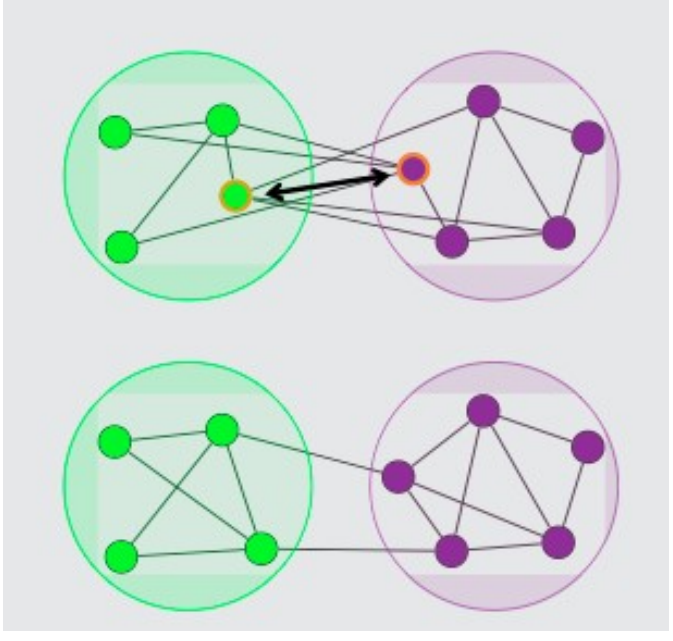
partition the full wiring diagram of an integrated circuit into smaller subgraphs, so that they minimize the number of connections between them.



2.5 billion transistors

Kernighan-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 reduces 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

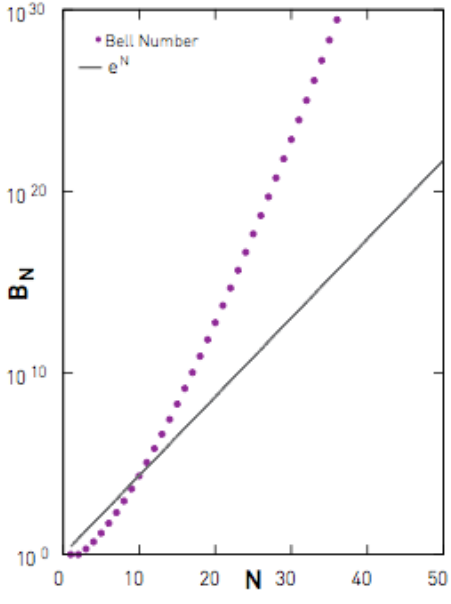
The *number* and *size* of the communities are unknown at the beginning.

Partition

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

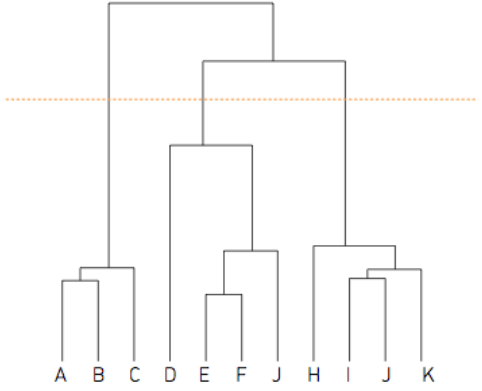
$$B_N = \frac{1}{e}\sum_{j=0}^{\infty}\frac{j^N}{j!} \quad .$$

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



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

Hierarchical Clustering



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.

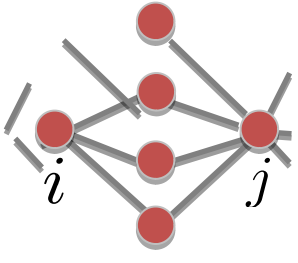
Agglomerative algorithms merge nodes and communities with high similarity.

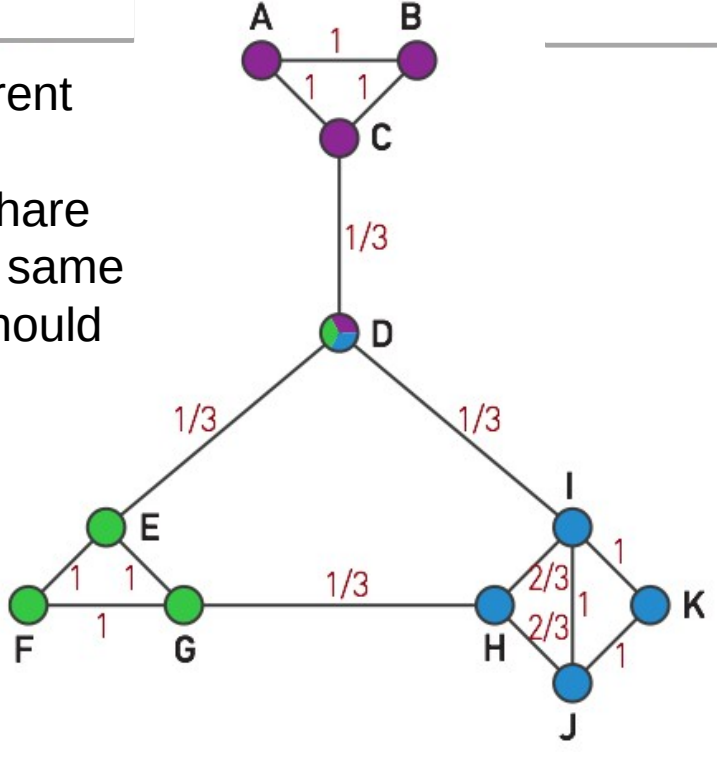
Step 1: Define the Similarity Matrix (Ravasz algorithm)

- 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;





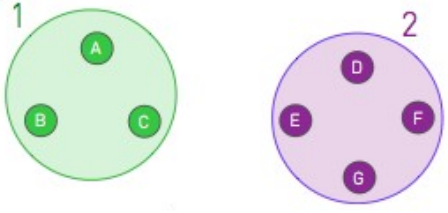
E. Ravasz et al., *Science* 297 (2002).

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*

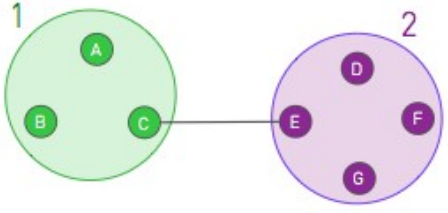
(a)



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

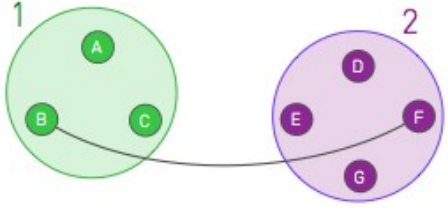
	D	E	F	G
A	2.75	2.22	3.46	3.08
B	3.38	2.68	3.97	3.40
C	2.31	1.59	2.88	2.34

(b)



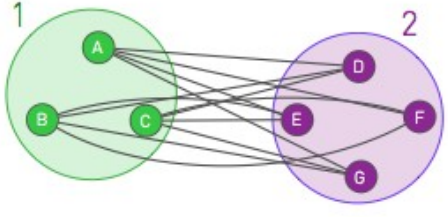
Single Linkage: $r_{12} = 1.59$

(c)



Complete Linkage: $r_{12} = 3.97$

(d)



Average Linkage: $r_{12} = 2.84$

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

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

Step 3: Apply Hierarchical Clustering

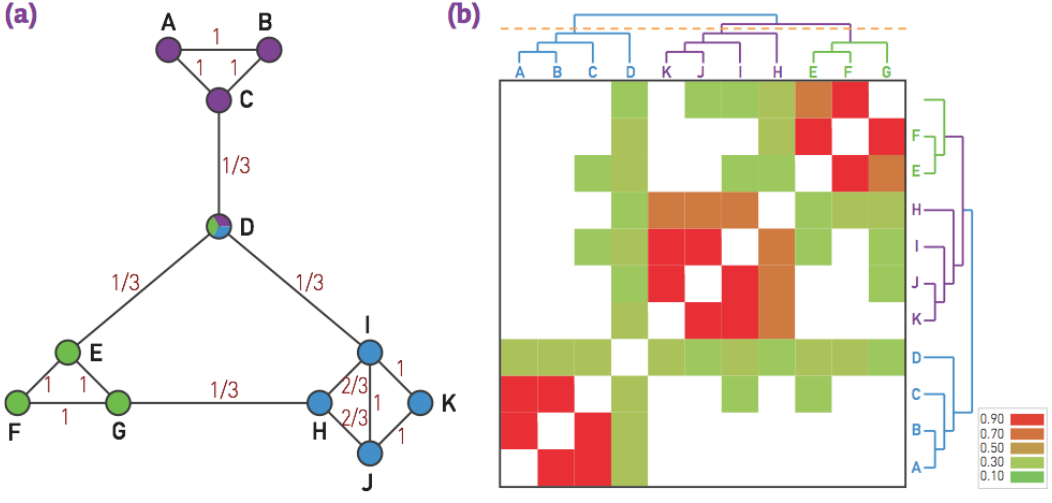
- Assign each node to a community of its own and evaluate the similarity for all node pairs. The initial similarities between these “communities” are simply the node similarities.
- Find the community pair with the highest similarity and merge them to form a single community.
- Calculate the similarity 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*.



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.*

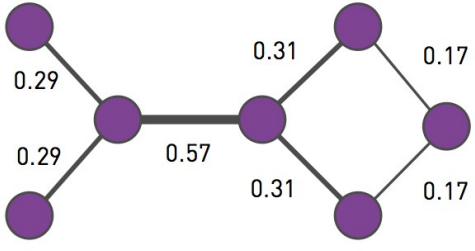
Divisive algorithms split communities by removing links that connect nodes with low similarity.

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

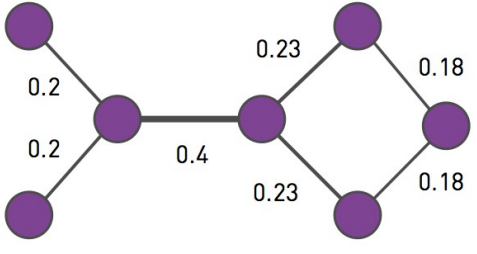
Examples of centrality measures:

- *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

(a)



(b)

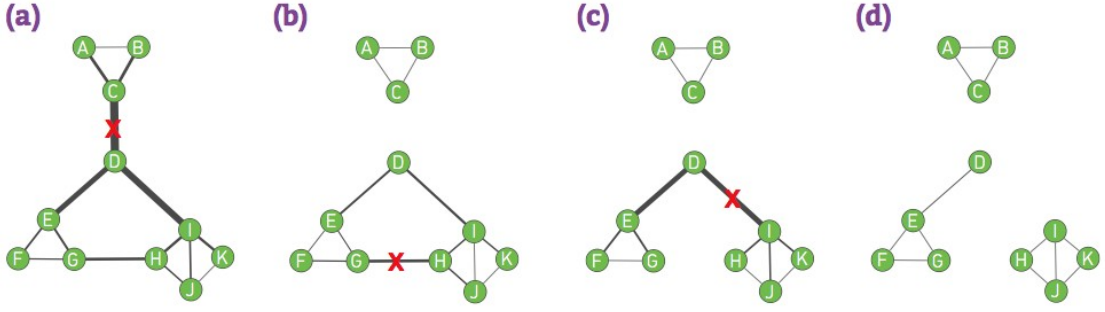


M. Girvan & M.E.J. Newman, *PNAS* 99 (2002).

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).

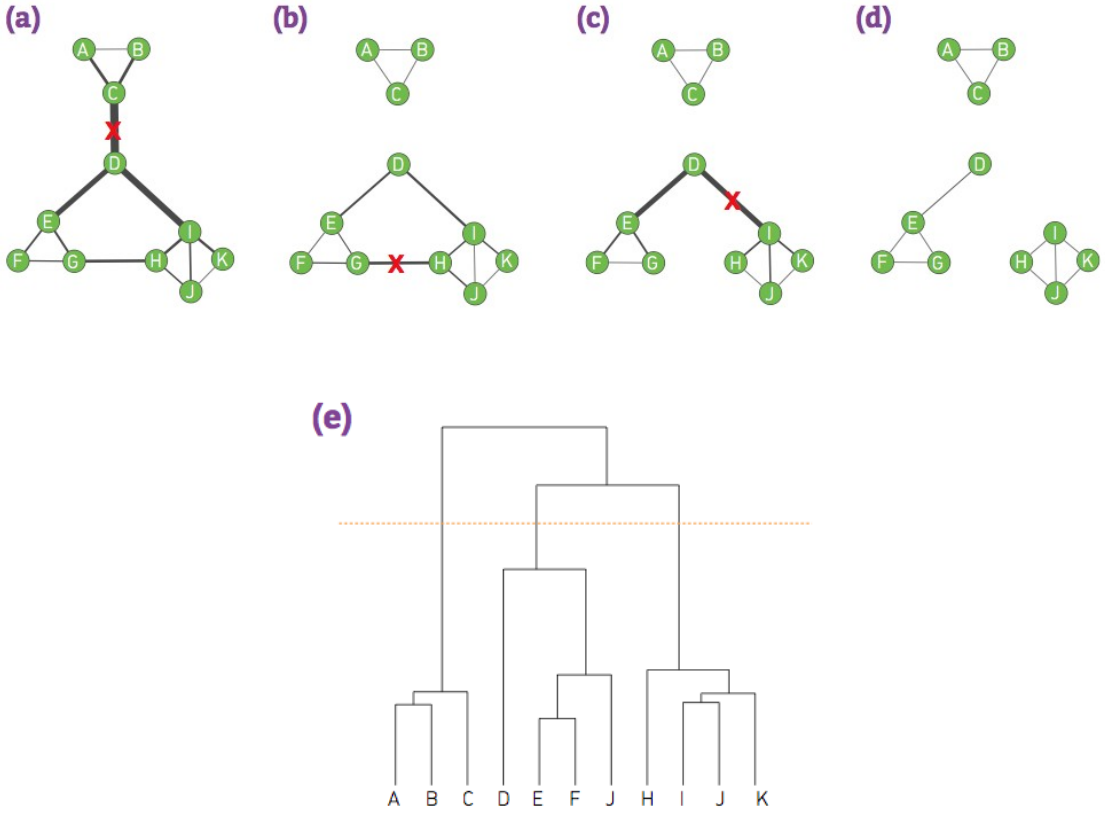


M. Girvan & M.E.J. Newman, *PNAS* 99 (2002).

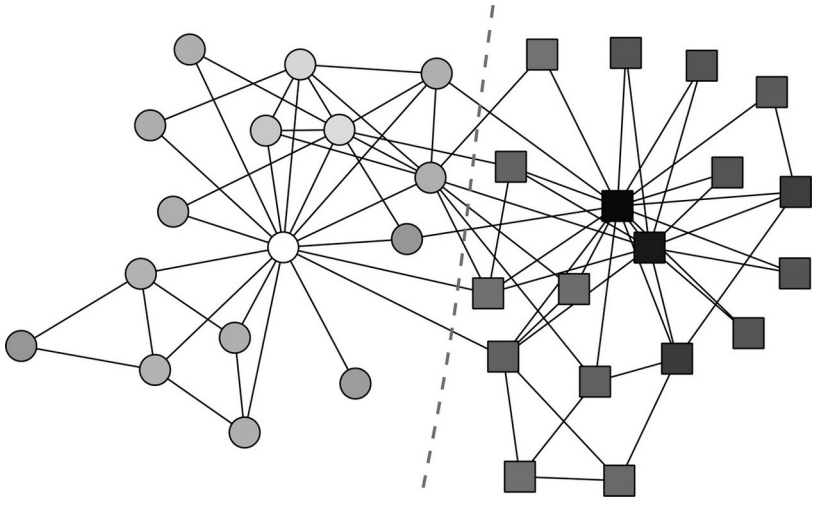
A.-L. Barabási, *Network Science: Communities*.

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*.



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*.