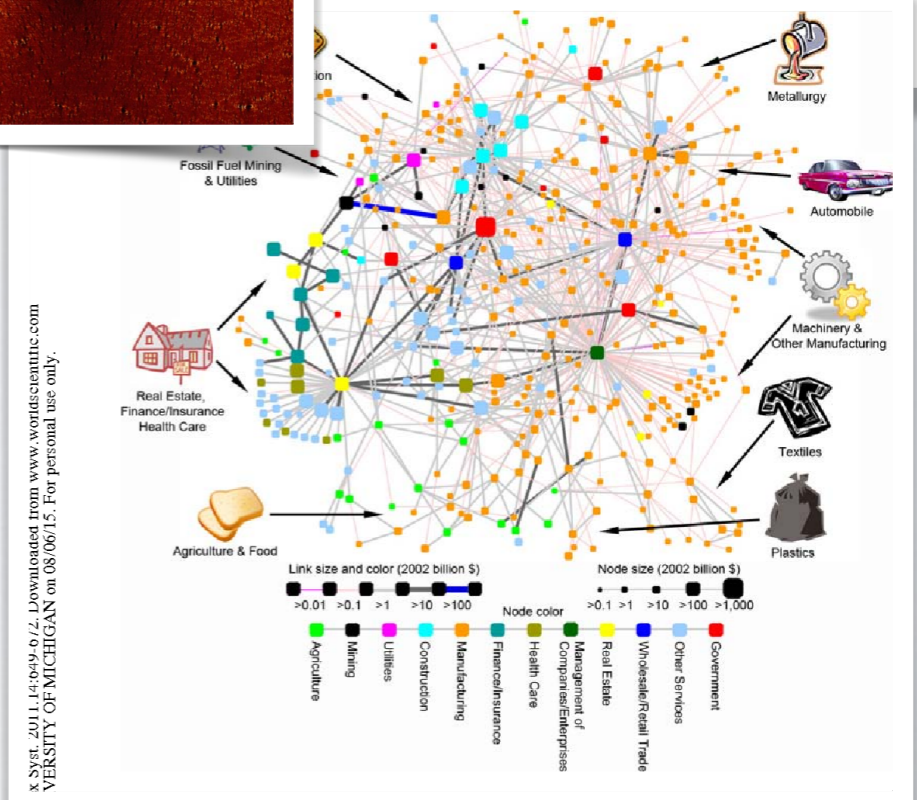
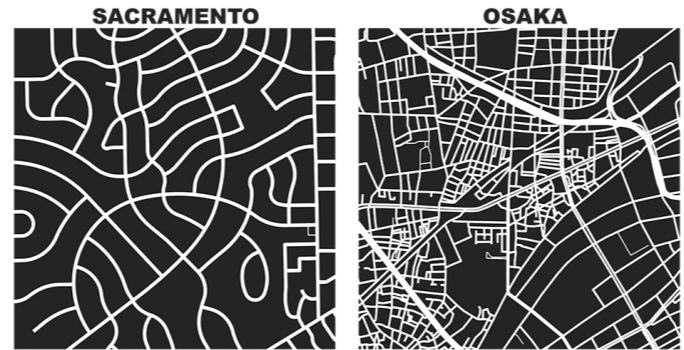
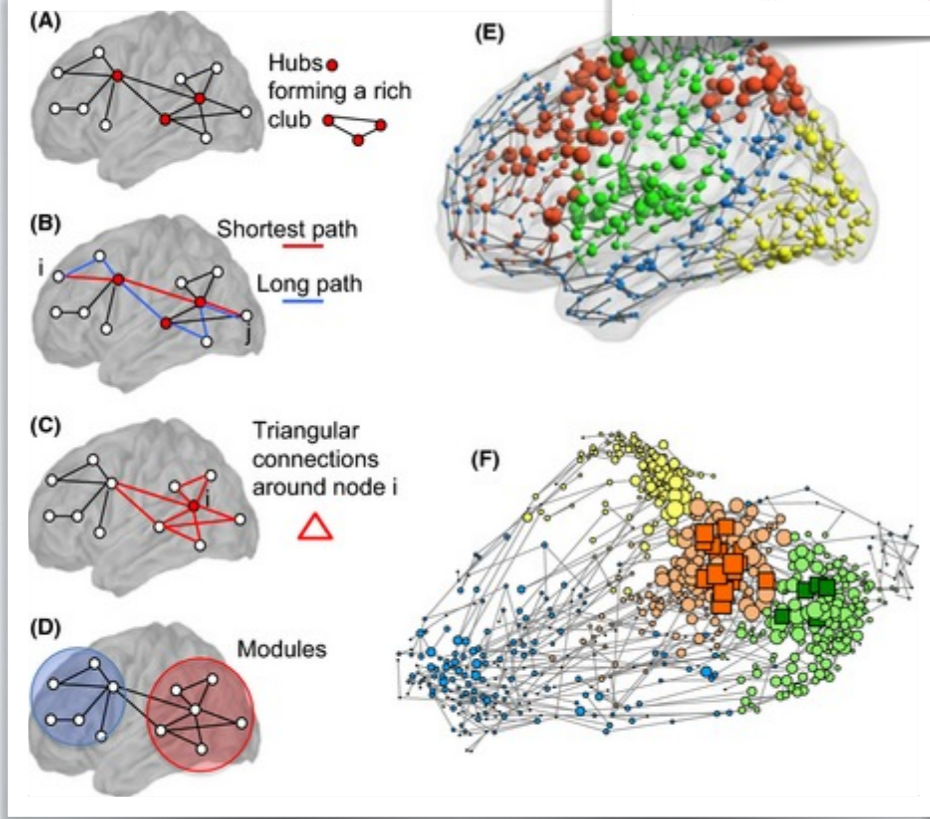
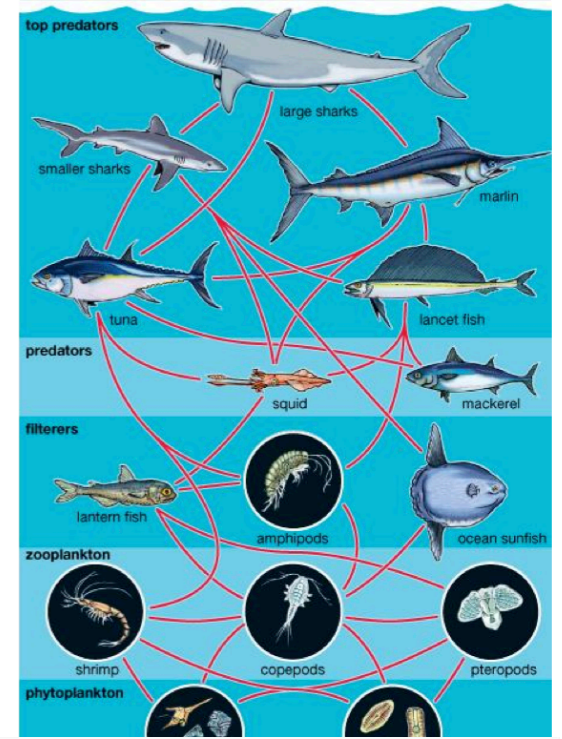
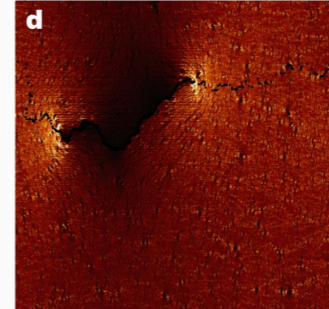
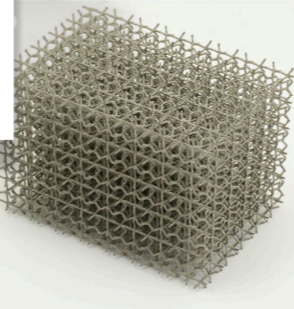
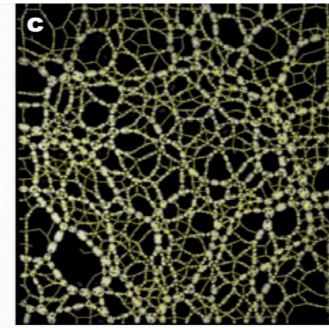
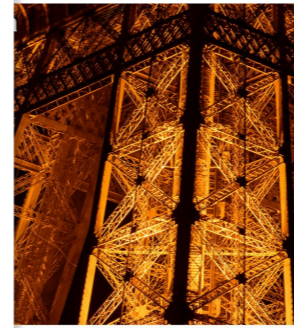
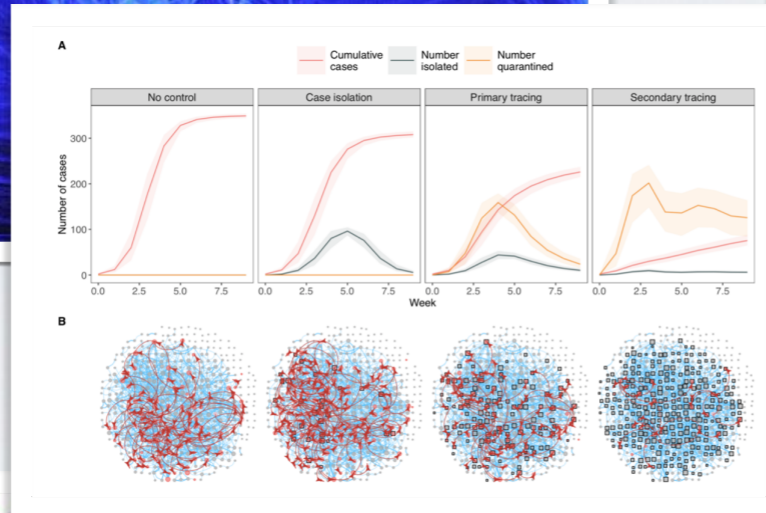
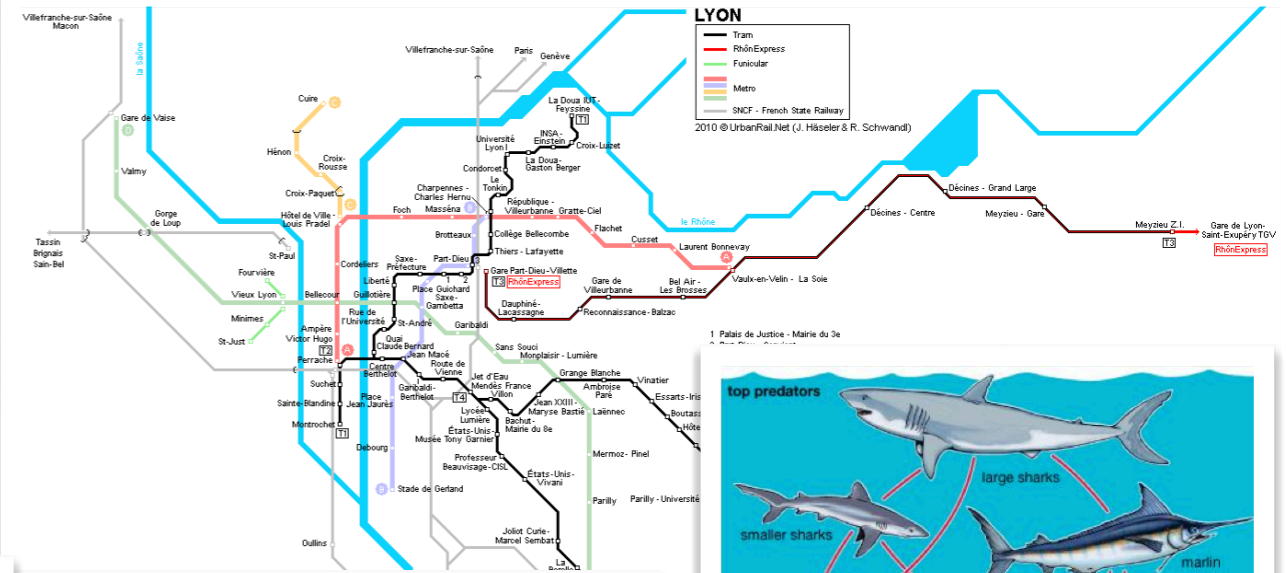


NETWORK DATA MINING

NETWORKS/GRAPHS

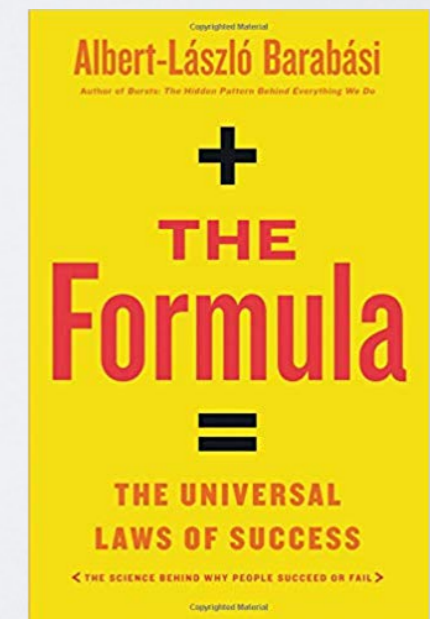
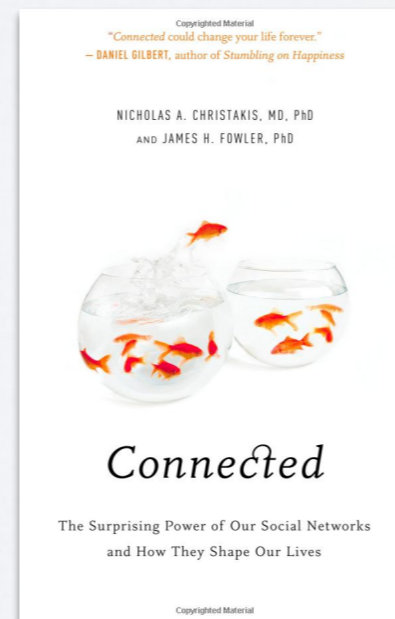
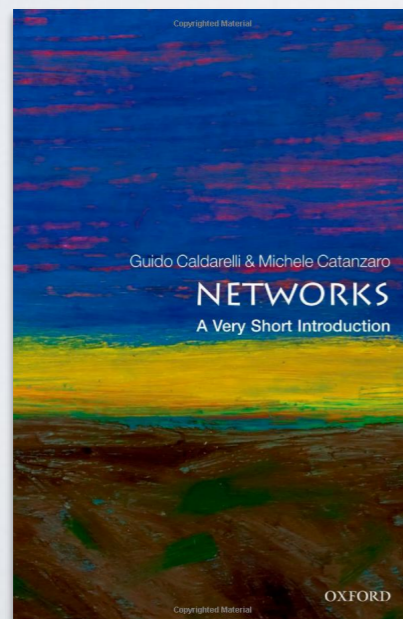
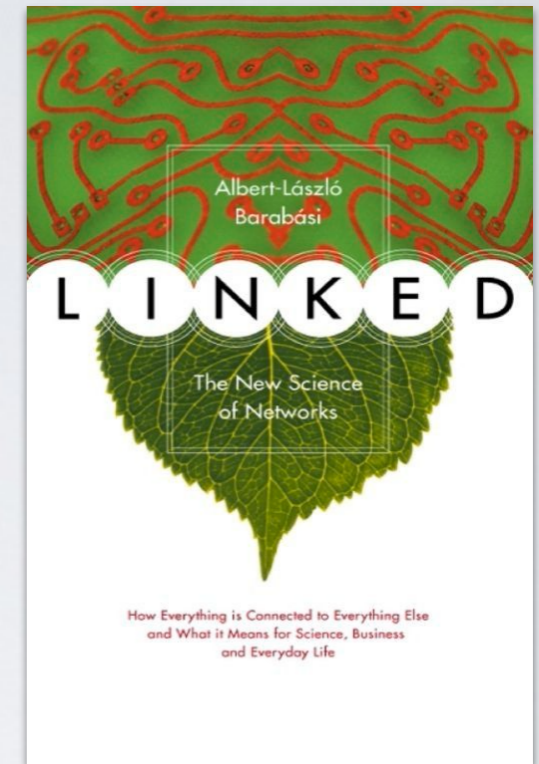
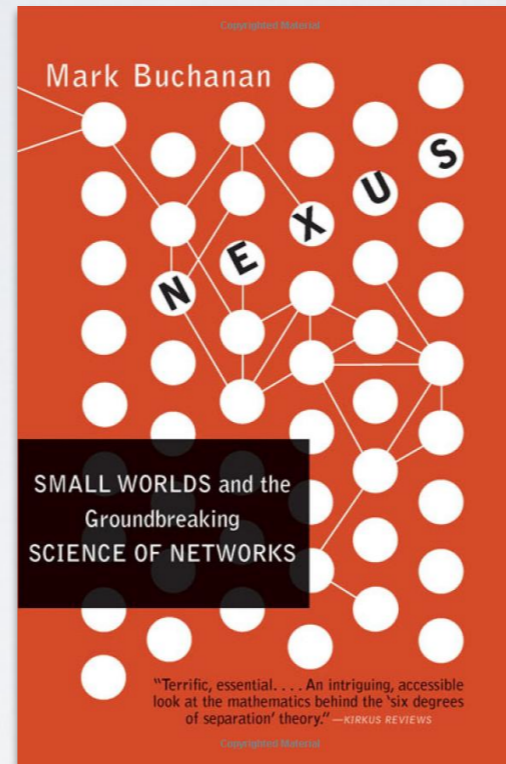
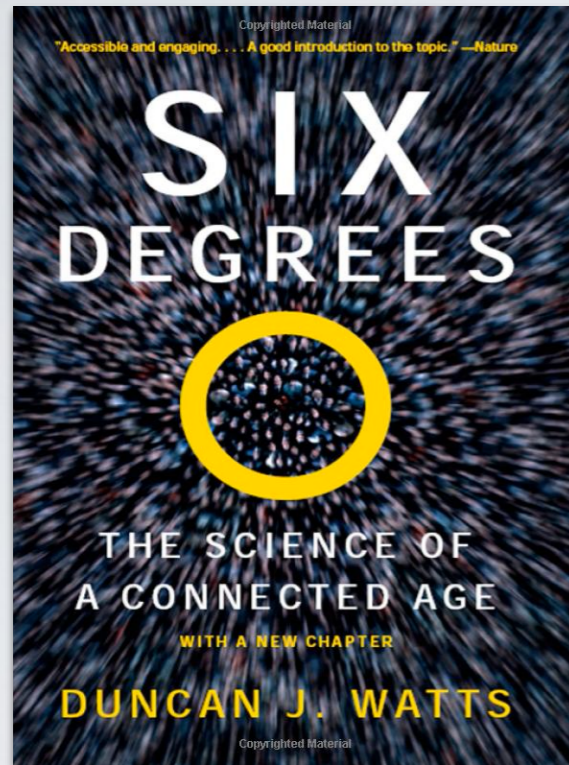
- Some data are not (only) “tabular”.
 - ▶ Items can have features that characterize them
 - Person: age, gender, etc.
 - ▶ Items also are linked/related to other items
 - This cannot be represented as a feature



Downloaded from www.worldscientific.com by UNIVERSITY OF MICHIGAN on 08/06/15. For personal use only.

Materials

Pop-science books



I have a copy I can lend

GRAPHS & NETWORKS

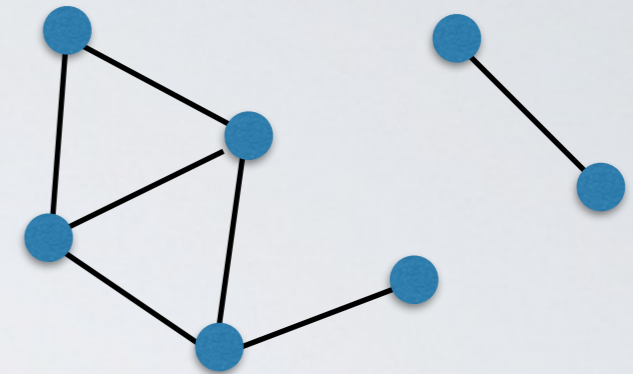
Networks often refers to real systems

- www,
- social network
- metabolic network.
- Language: (Network, node, link)

Graph is the mathematical representation of a network

- Language: (Graph, vertex, edge)

In most cases we will use the two terms interchangeably.



Vertex	Edge
person	friendship
neuron	synapse
Website	hyperlink
company	ownership
gene	regulation

NETWORK REPRESENTATIONS

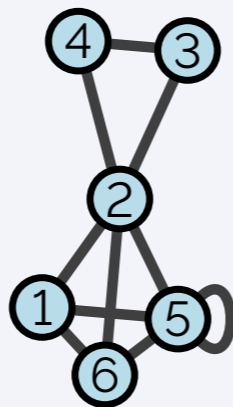
Networks: Graph notation

Graph notation : $G = (V, E)$

V	set of vertices/nodes.
E	set of edges/links.
$u \in V$	a node.
$(u, v) \in E$	an edge.

Network - Graph notation

Graph



Graph notation

$G = (V, E)$
 $V = \{1, 2, 3, 4, 5, 6\}$
 $E = \{(1, 2), (1, 6),$
 $(1, 5), (2, 4), (2, 3), (2, 5),$
 $(2, 6), (6, 5), (5, 5), (4, 3)\}$

GRAPH REPRESENTATION

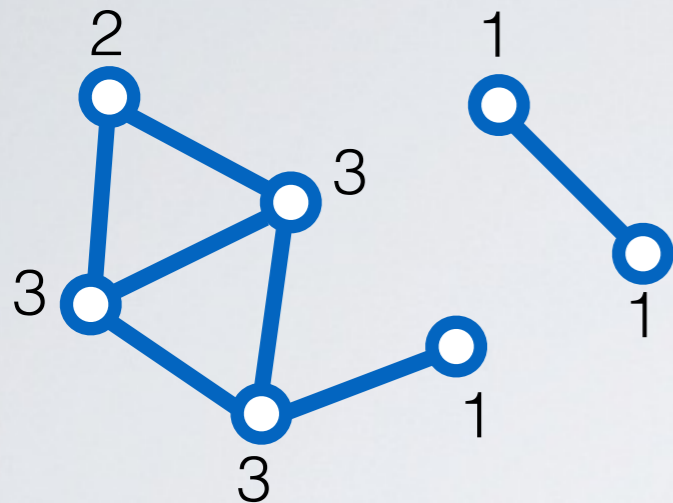
Node-Edge description

N_u	Neighbourhood of u , nodes sharing a link with u .
k_u	Degree of u , number of neighbors $ N_u $.
N_u^{out}	Successors of u , nodes such as $(u, v) \in E$ in a directed graph
N_u^{in}	Predecessors of u , nodes such as $(v, u) \in E$ in a directed graph
k_u^{out}	Out-degree of u , number of outgoing edges $ N_u^{out} $.
k_u^{in}	In-degree of u , number of incoming edges $ N_u^{in} $
$w_{u,v}$	Weight of edge (u, v) .
s_u	Strength of u , sum of weights of adjacent edges, $s_u = \sum_v w_{uv}$.

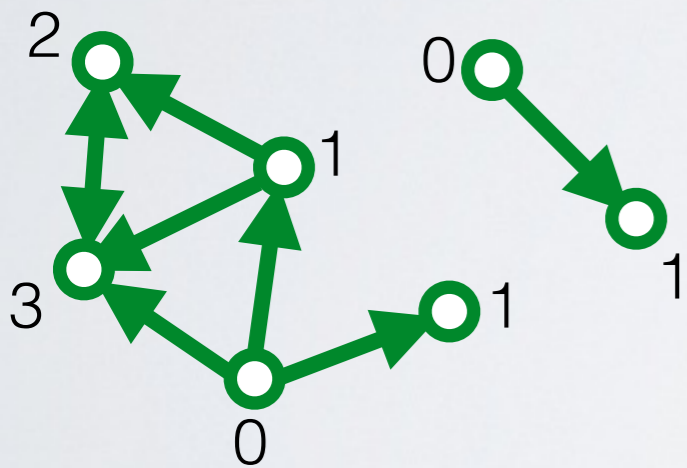
Node degree

Number of connections of a node

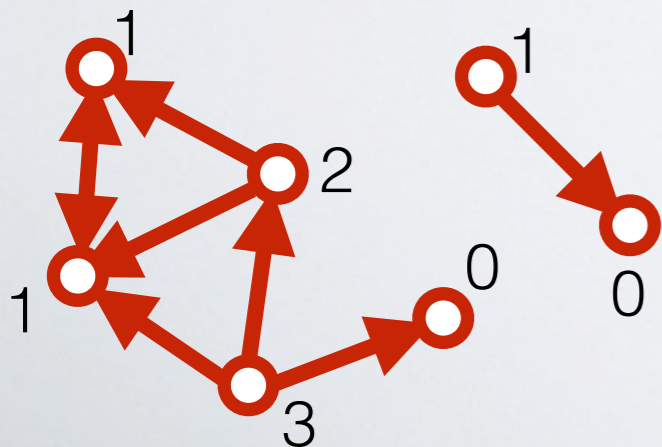
- Undirected network



- Directed network



In degree



Out degree

SIZE

Counting nodes and edges

N/n

L/m

L_{max}

size: number of nodes $|V|$.

number of edges $|E|$

Maximum number of links

Undirected network: $\binom{N}{2} = N(N - 1)/2$

Directed network: $\binom{N}{2} = N(N - 1)$

DENSITY

Network descriptors 1 - Nodes/Edges

$\langle k \rangle$

Average degree: Real networks are sparse, i.e., typically $\langle k \rangle \ll n$. Increases slowly with network size, e.g., $d \sim \log(m)$

$$\langle k \rangle = \frac{2m}{n}$$

$d/d(G)$

Density: Fraction of pairs of nodes connected by an edge in G .

$$d = L/L_{\max}$$

	#nodes	#edges	Densité	Deg. Moyen
Wikipedia	2M	30M	1.5×10^{-5}	30
Twitter 2015	288M	60B	1.4×10^{-6}	416
Facebook	1.4B	400B	4×10^{-9}	570
Brain c.	280	6393	0,16	46
Roads Calif.	2M	2.7M	6×10^{-7}	2,7
Airport	3k	31k	0,007	21

Attention: Densité difficile à comparer entre des graphes de taille différente

SUBGRAPHS

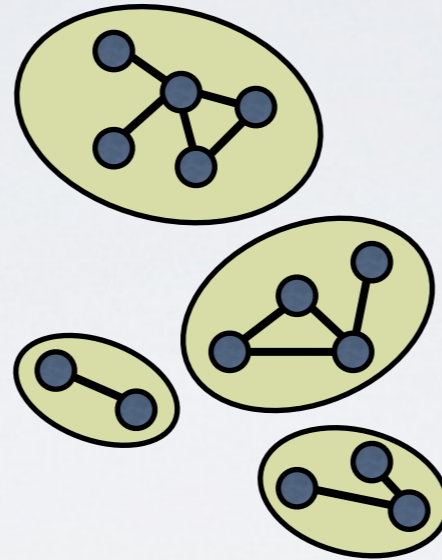


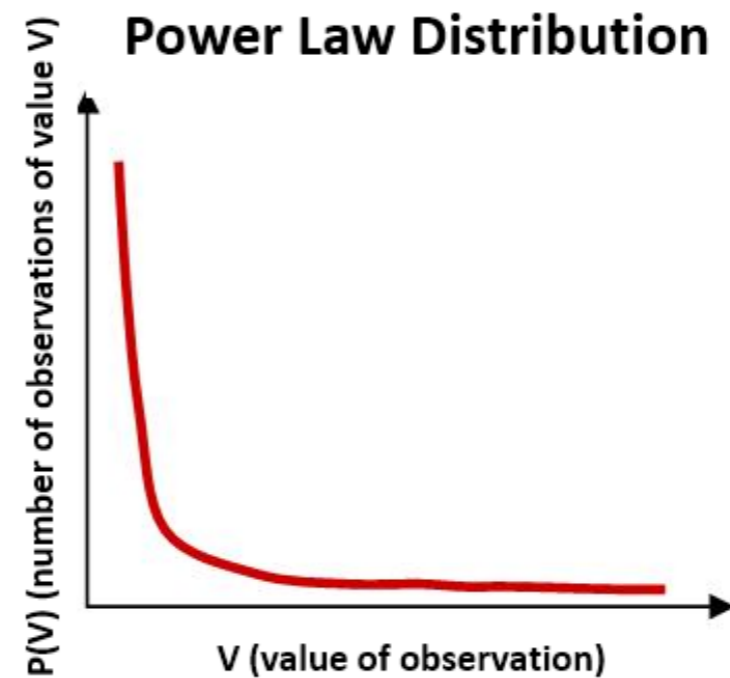
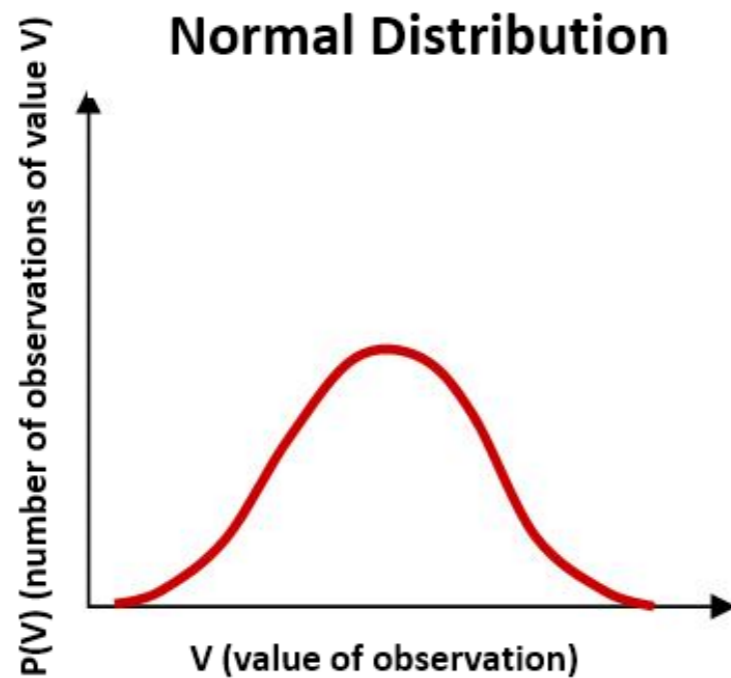
Figure after Newman, 2010

Clique: subgraph with $d = 1$

Triangle: clique of size 3

Connected component: a subgraph in which any two vertices are connected to each other by paths, and which is connected to no additional vertices in the supergraph

DEGREE DISTRIBUTION



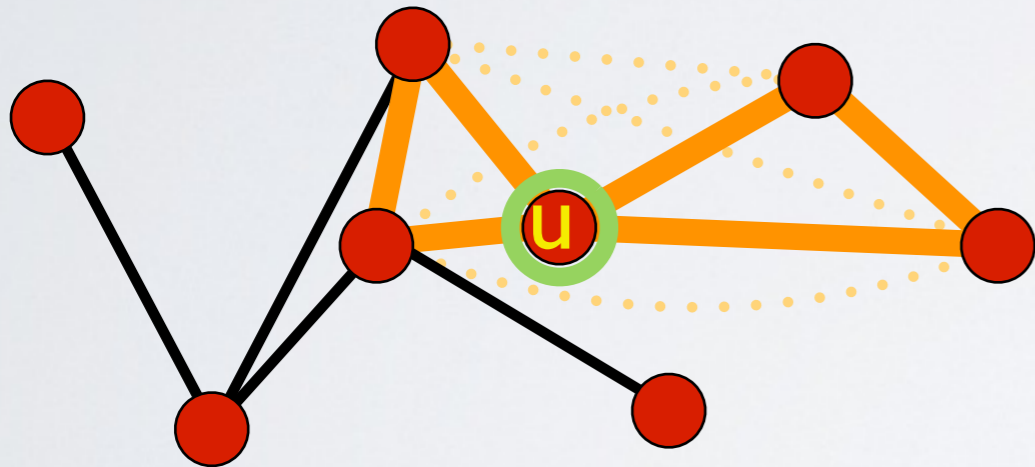
PDF (Probability Distribution Function)

CLUSTERING COEFFICIENT

- **Clustering coefficient** or **triadic closure**
- Triangles are considered important in real networks
 - ▶ Think of social networks: *friends of friends are my friends*
 - ▶ # triangles is a big difference between real and random networks

CLUSTERING COEFFICIENT

C_u - **Node clustering coefficient**: density of the subgraph induced by the neighborhood of u , $C_u = \frac{d(H(N_u))}{\binom{\delta_u}{2}}$. Also interpreted as the fraction of all possible triangles in N_u that exist, $\frac{\delta_u}{\delta_u^{\max}}$



Edges: 2
Max edges: $4 \cdot 3 / 2 = 6$
 $C_u = 2 / 6 = 1 / 3$

Triangles=2
Possible triangles = $\binom{4}{2} = 6$
 $C_u = 2 / 6 = 1 / 3$

PATH RELATED SCORES

Paths - Walks - Distance

Walk: Sequences of adjacent edges or nodes (e.g., **1.2.1.6.5** is a valid walk)

Path: a walk in which each node is distinct.

Path length: number of edges encountered in a path

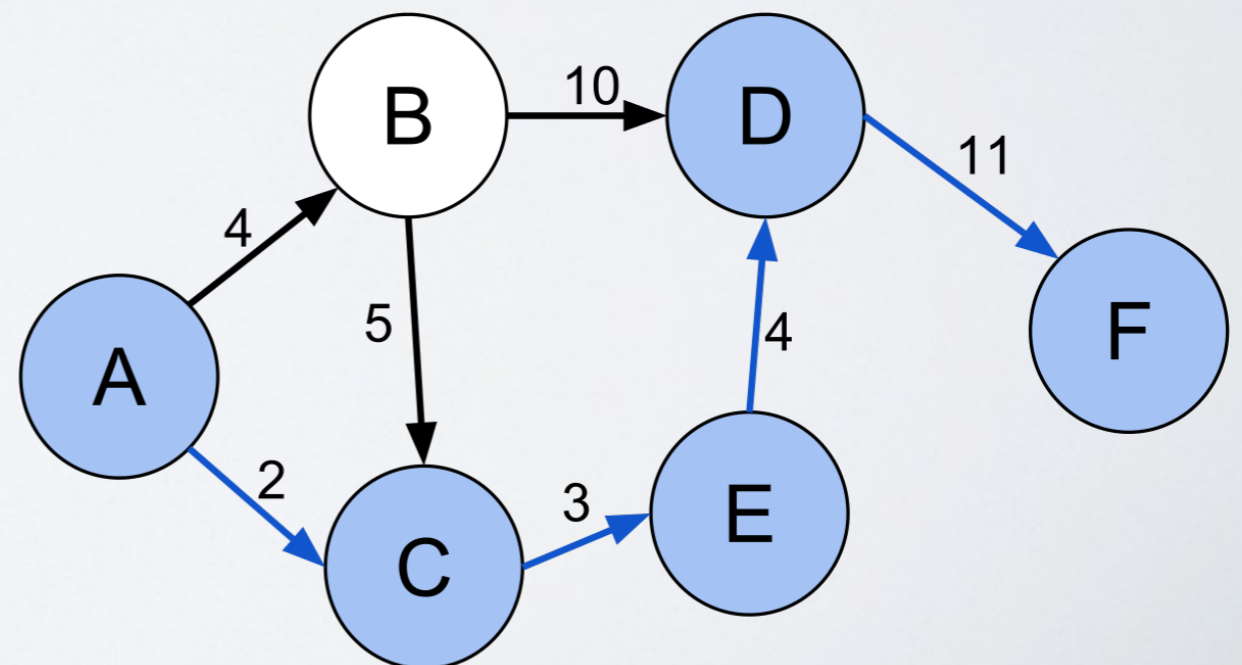
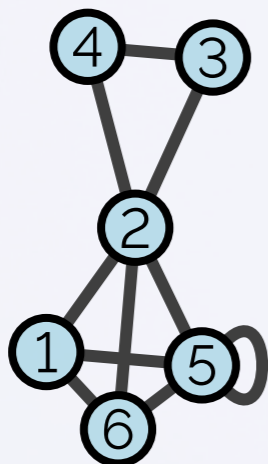
Weighted Path length: Sum of the weights of edges on a path

Shortest path: The shortest path between nodes u, v is a path of minimal *path length*. Often it is not unique.

Weighted Shortest path: path of minimal *weighted path length*.

$l_{u,v}$: **Distance:** The distance between nodes u, v is the length of the shortest path

Graph



PATH RELATED SCORES

Network descriptors 2 - Paths

l_{\max}
 $\langle l \rangle$

Diameter: maximum *distance* between any pair of nodes.

Average distance:

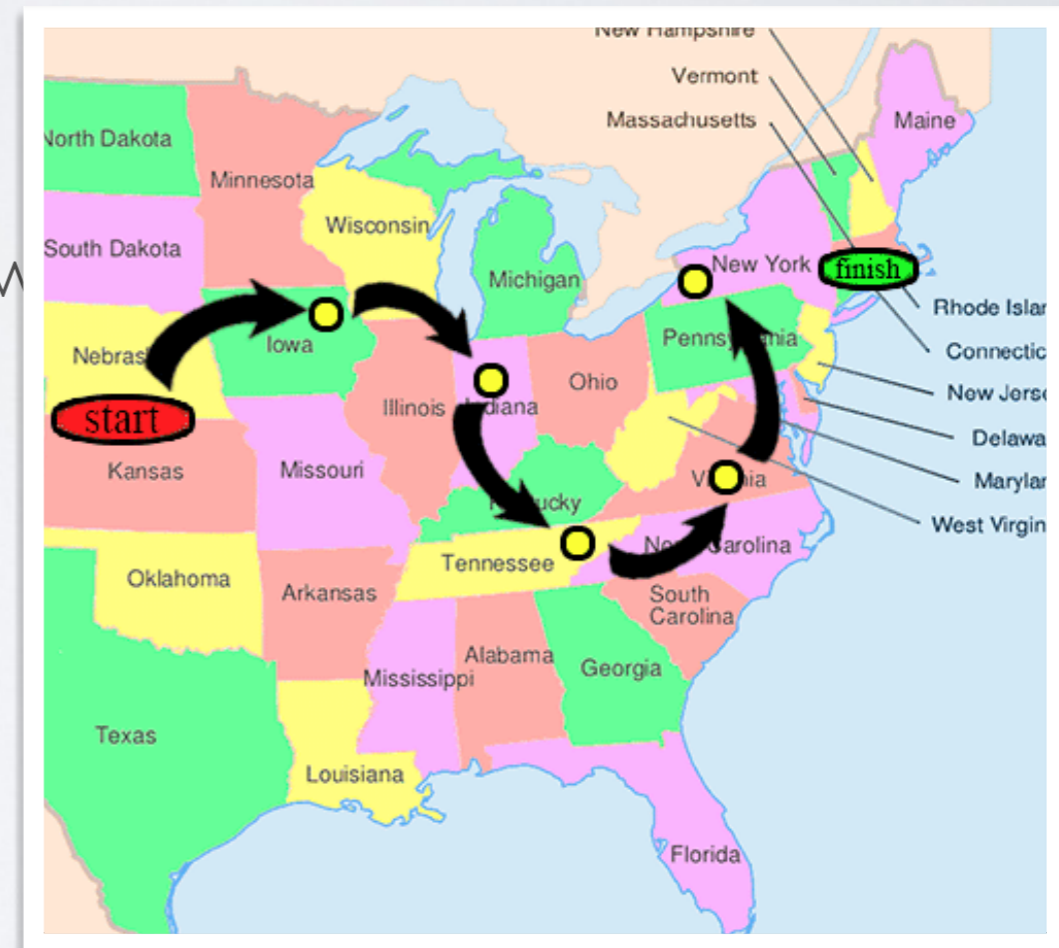
$$\langle l \rangle = \frac{1}{n(n-1)} \sum_{i \neq j} d_{ij}$$

AVERAGE PATH LENGTH

- The famous 6 degrees of separation (Milgram experiment)
 - (More on that next slide)
- Not too sensible to noise
- Tells you if the network is “stretched” or “hairball” like

SIDE-STORY: MILGRAM EXPERIMENT

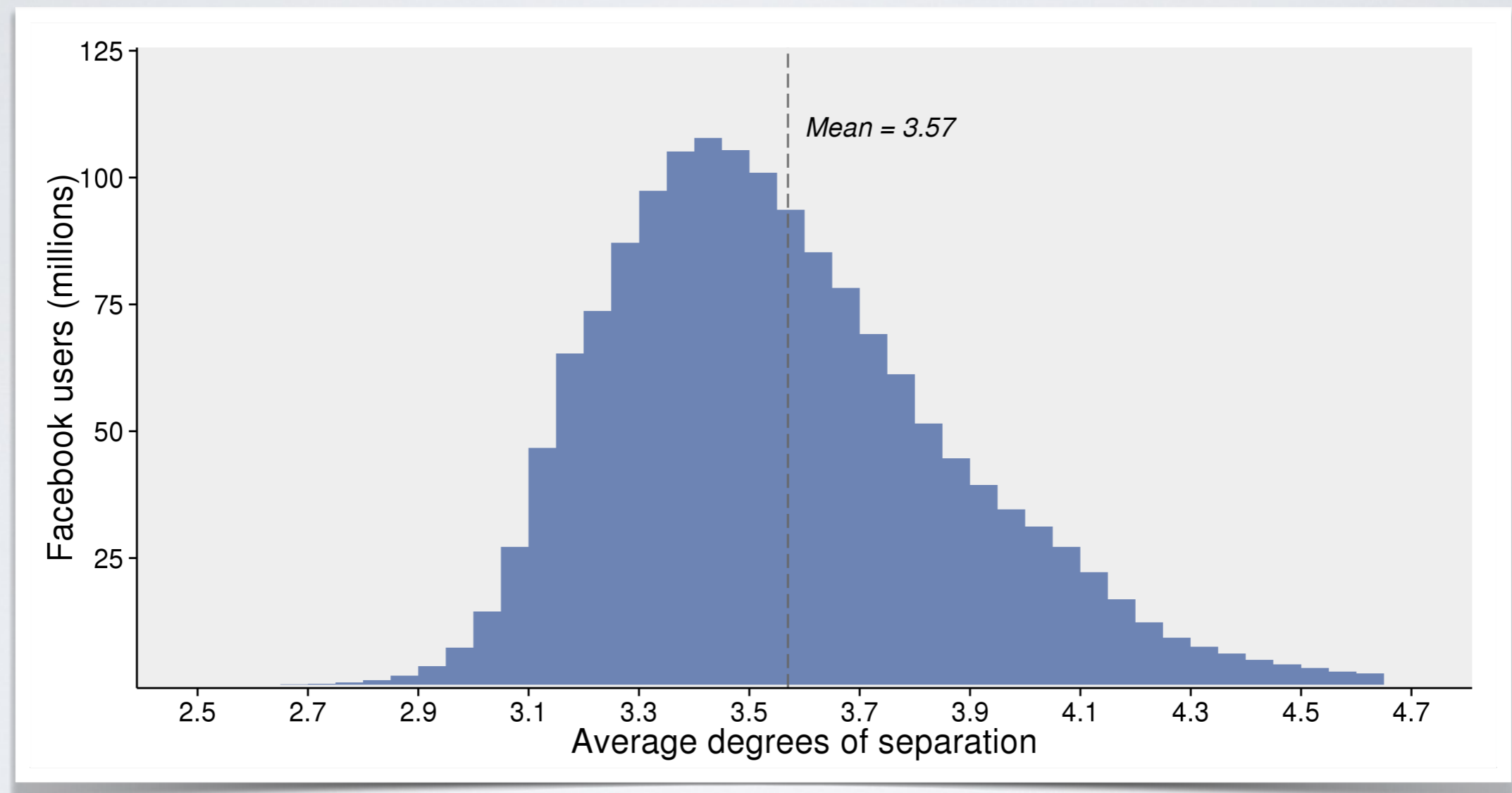
- Small world experiment (60's)
 - ▶ Give a (physical) mail to random people
 - ▶ Ask them to send to someone they don't know
 - They know his city, job
 - ▶ They send to their most relevant contact
- Results: In average, 6 hops to arrive



SIDE-STORY: MILGRAM EXPERIMENT

- Many criticism on the experiment itself:
 - ▶ Some mails did not arrive
 - ▶ Small sample
 - ▶ ...
- Checked on “real” complete graphs (giant component):
 - ▶ MSN messenger
 - ▶ Facebook
 - ▶ The world wide web
 - ▶ ...

SIDE-STORY: MILGRAM EXPERIMENT



Facebook

SMALL WORLD

Small World Network

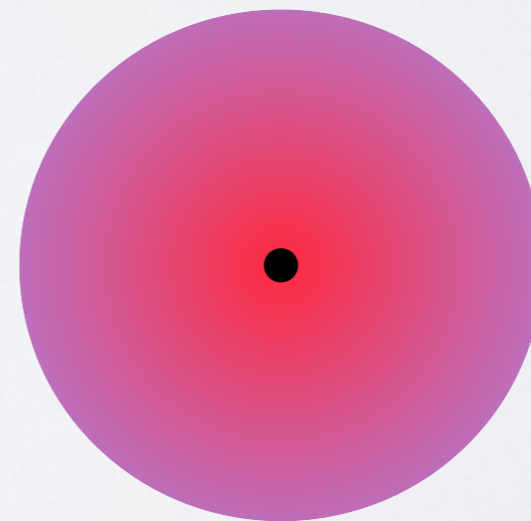
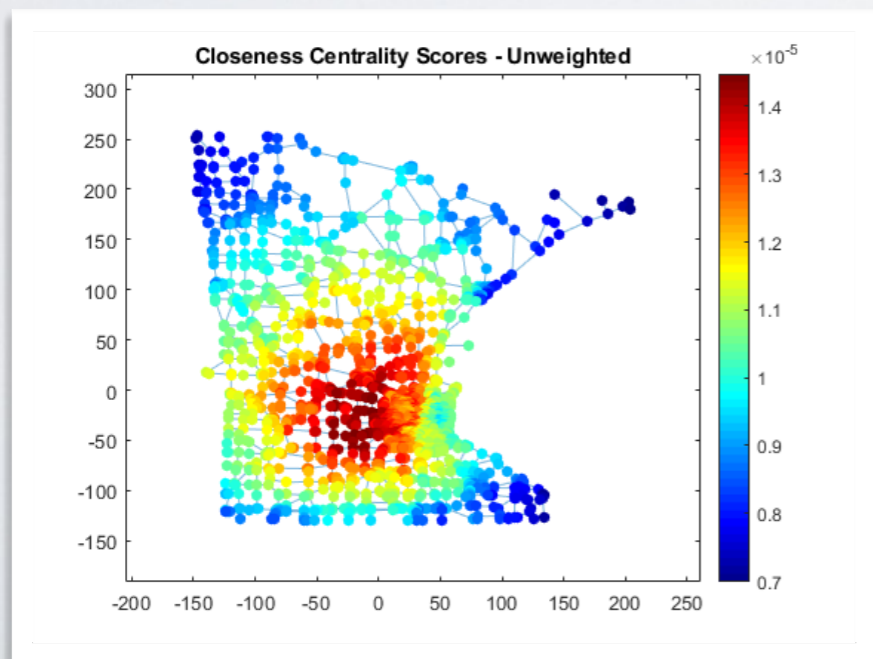
A network is said to have the **small world** property when it has some structural properties. The notion is not quantitatively defined, but two properties are required:

- Average distance must be short, i.e., $\langle \ell \rangle \approx \log(N)$
- Clustering coefficient must be high, i.e., much larger than in a random network, e.g., $C^g \gg d$, with d the network density

FARNES, CLOSENESS

FARNESS, CLOSENESS

- How close the node is to all other nodes
- Parallel with the center of a figure:
 - Center of a circle is the point of shorter average distance to any points in the circle



FARNNESS, CLOSENESS

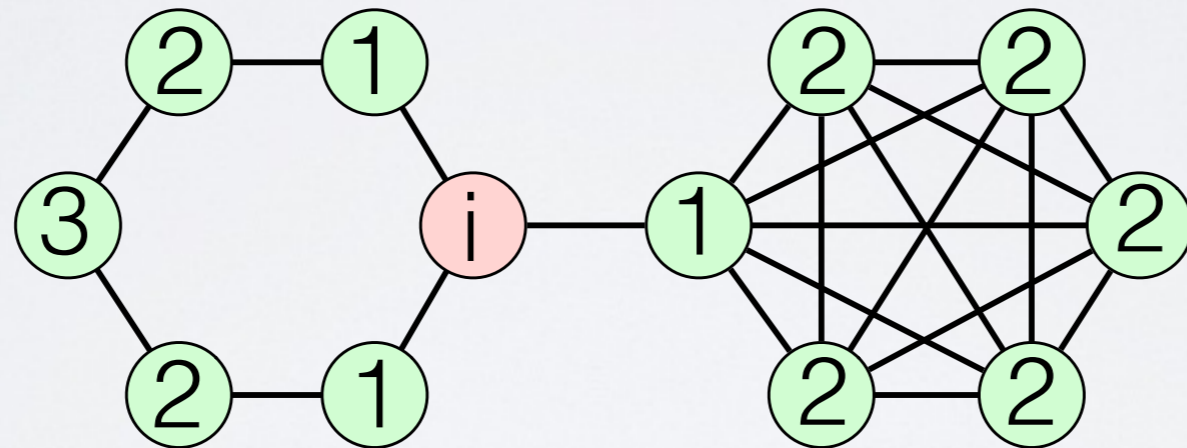
Farness: Average distance to all other nodes in the graph

$$\text{Farness}(u) = \frac{1}{N-1} \sum_{v \in V \setminus u} \ell_{u,v}$$

CLOSENESS CENTRALITY

Closeness: Inverse of the farness, i.e., how close the node is to all other nodes in term of shortest paths.

$$\text{Closeness}(u) = \frac{N - 1}{\sum_{v \in V \setminus u} l_{u,v}}$$



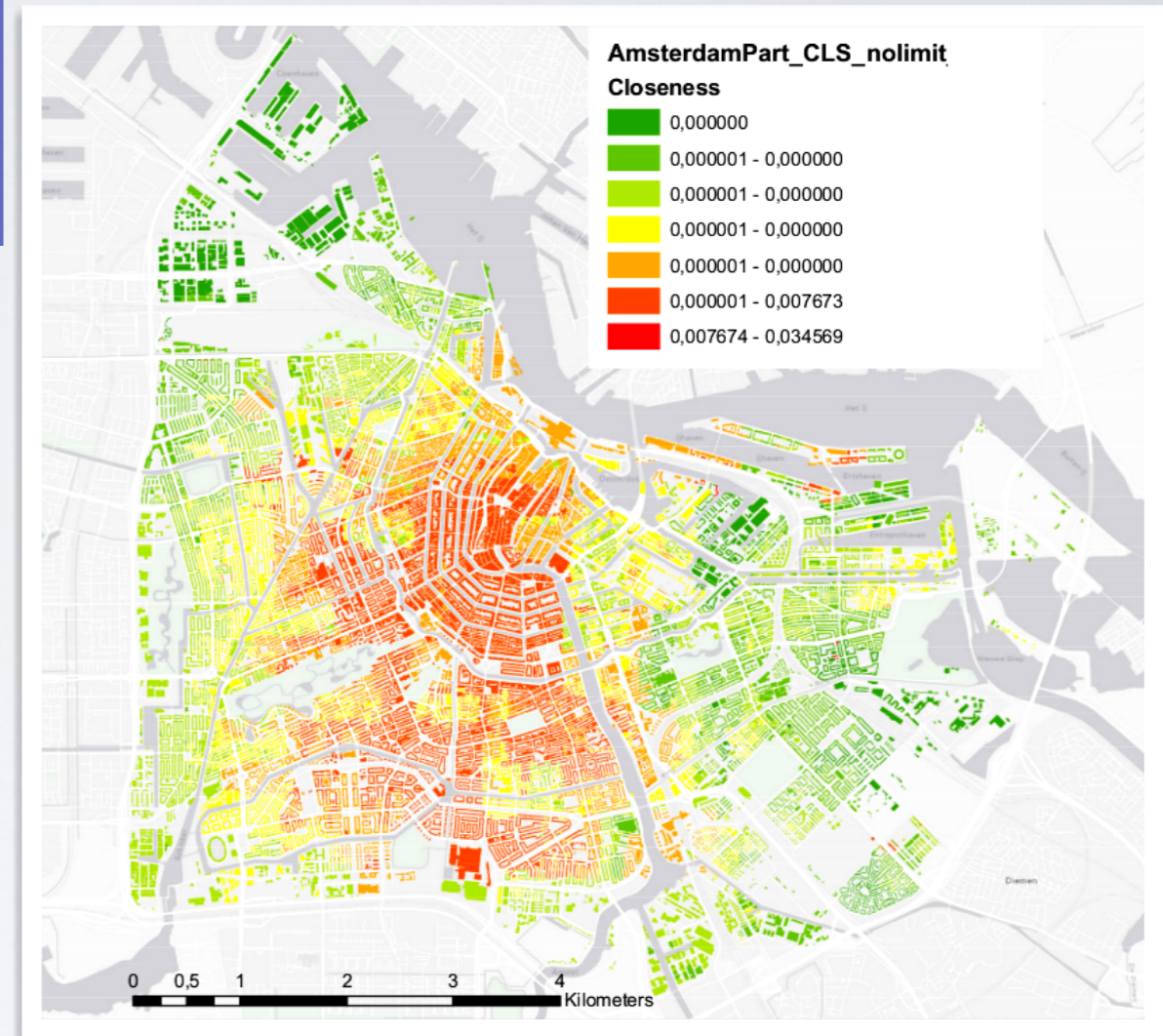
$$C_{cl}(i) = \frac{12 - 1}{(3 \times 1 + 7 \times 2 + 1 \times 3)} = \frac{11}{20} = 0.55$$

CLOSENESS CENTRALITY

Closeness: Inverse of the farness, i.e., how close the node is to all other nodes in term of shortest paths.

$$\text{Closeness}(u) = \frac{N - 1}{\sum_{v \in V \setminus u} \ell_{u,v}}$$

1 = all nodes are at distance one



BETWEENNESS CENTRALITY

- Measure how much the node plays the role of a bridge
- Betweenness of u : fraction of all the shortest paths between all the pairs of nodes going through u .

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

with σ_{st} the number of shortest paths between nodes s and t and $\sigma_{st}(v)$ the number of those paths passing through v .

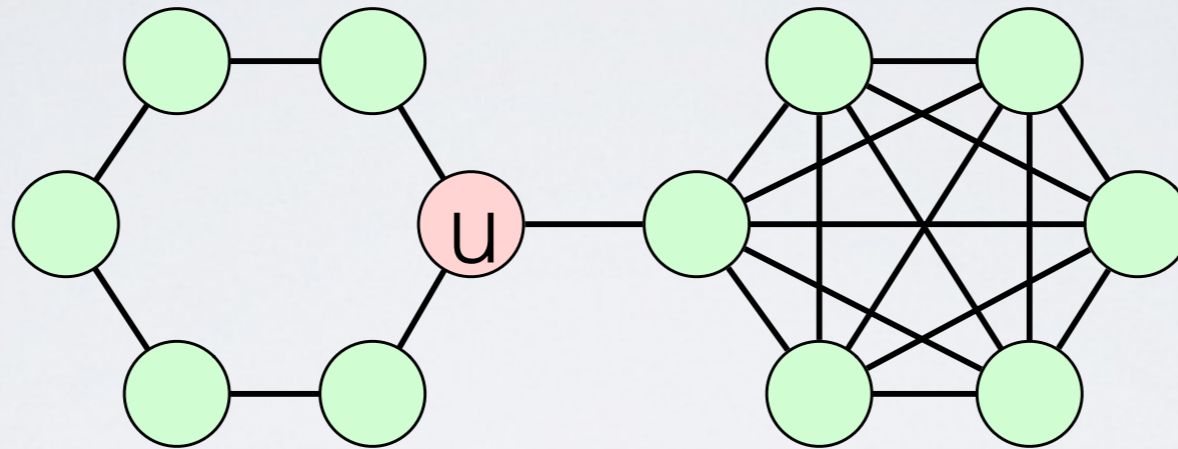
The betweenness tends to grow with the network size. A normalized version can be obtained by dividing by the number of pairs of nodes, i.e., for a

directed graph: $C_B^{\text{norm}}(v) = \frac{C_B(v)}{(N-1)(N-2)}$.

Betweenness Centrality

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

directed graph: $C_B^{\text{norm}}(v) = \frac{C_B(v)}{(N-1)(N-2)}$.



$$C_B(u) = 2 \frac{5 * 6 + 1 + \frac{1}{2} + \frac{1}{2}}{11 * 10} = \frac{64}{110}$$

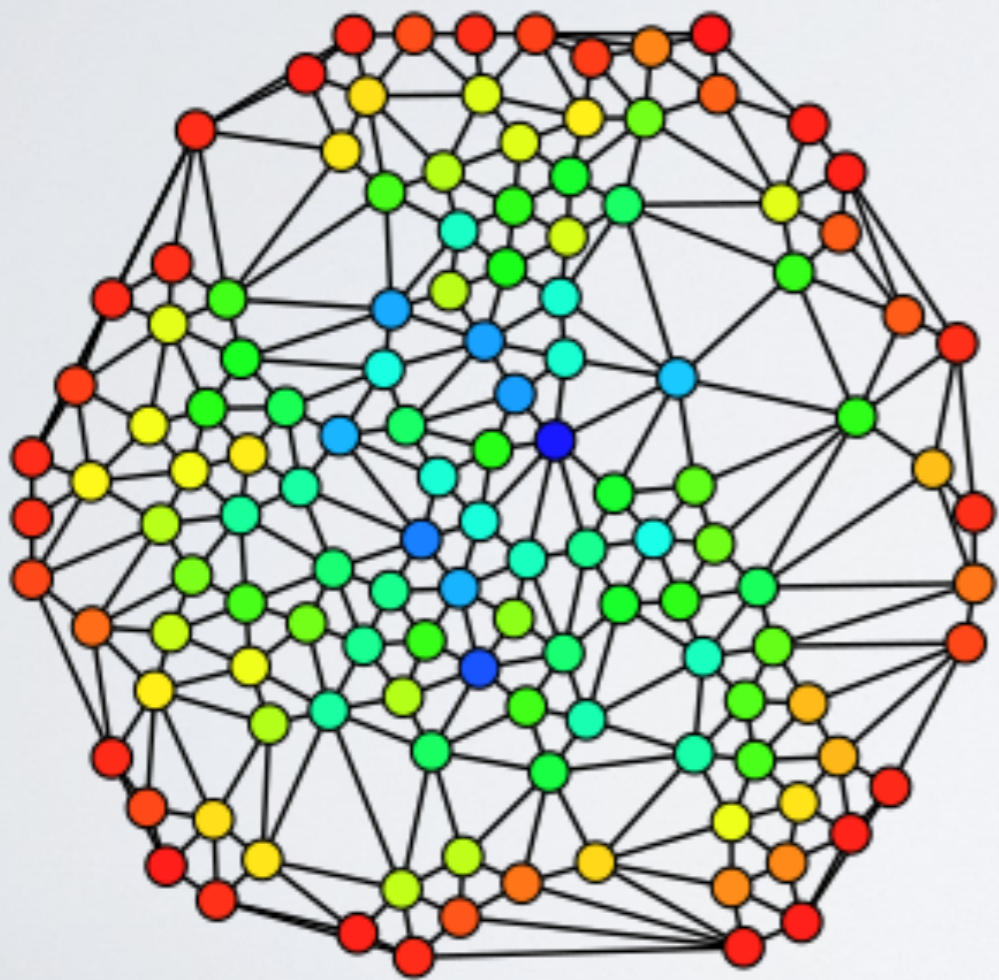
Exact computation:

Floyd-Warshall: $O(n^3)$ time complexity
 $O(n^2)$ space complexity

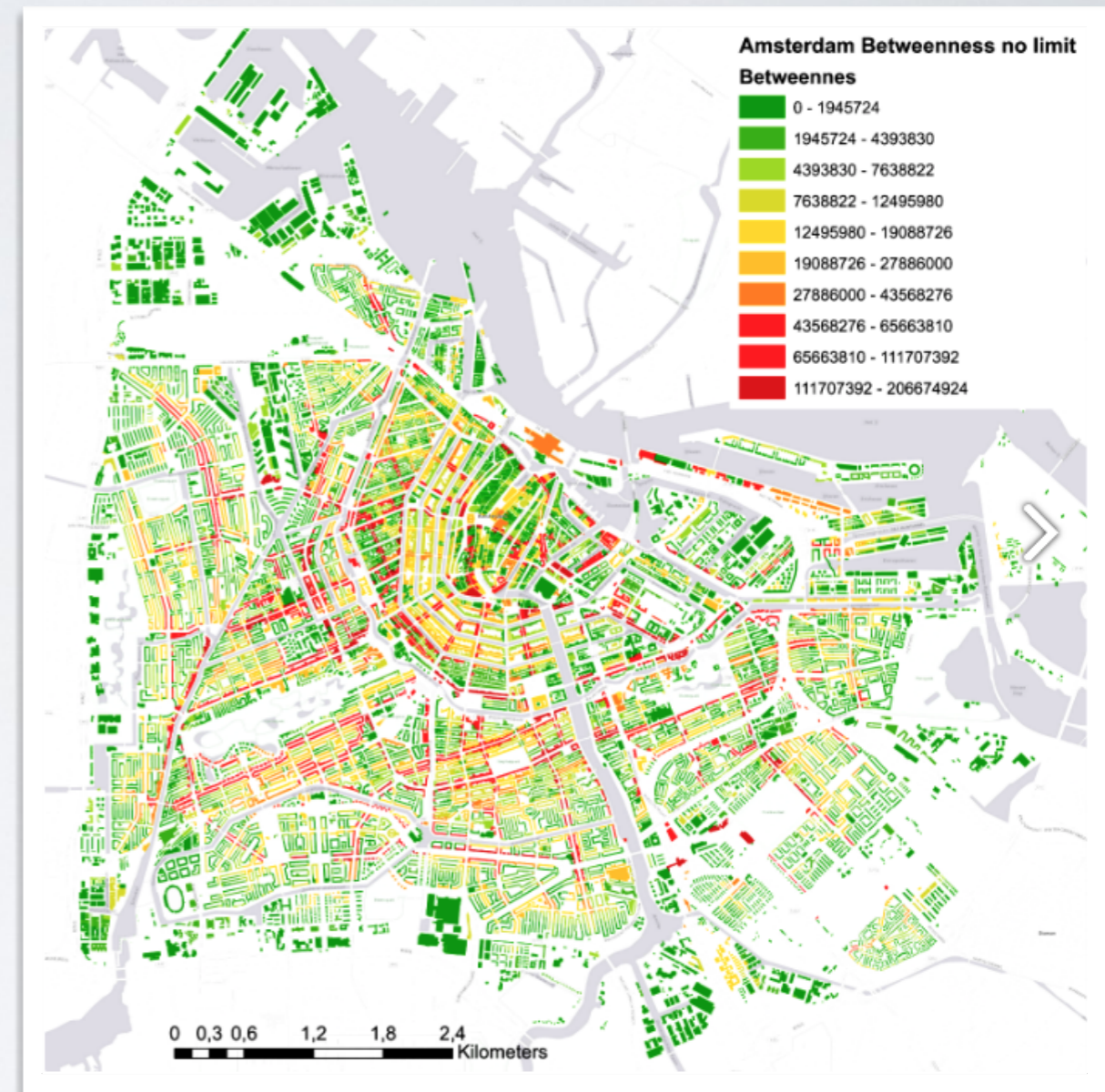
Approximate computation

Dijkstra: $O(n(m+n \log n))$ time complexity

BETWEENNESS CENTRALITY



(blue higher)



(red higher)

EDGE - BETWEENNESS

Same definition as for nodes

Can you guess the edge of highest betweenness in the European rail network?



RECURSIVE DEFINITIONS

RECURSIVE DEFINITIONS

- Recursive importance:
 - **Important nodes** are those connected **to important nodes**
- Several centralities based on this idea:
 - Eigenvector centrality
 - PageRank
 - ...

RECURSIVE DEFINITION

- We would like scores such as :
 - Each node has a score (centrality),
 - If every node “sends” its score to its neighbors, the sum of all scores received by each node will be equal to its original score

$$C_u^{t+1} = \frac{1}{\lambda} \sum_{v \in N_u^{in}} C_v^t \quad (1)$$

- With λ a normalisation constant

RECURSIVE DEFINITION

- This problem can be solved by what is called the *power method*:
 - 1) We initialize all scores to random values
 - 2) Each score is updated according to the desired rule, until reaching a stable point (after normalization)
- Why does it converge?
 - Perron-Frobenius theorem (see next slide)
 - => True for undirected graphs with a single connected component

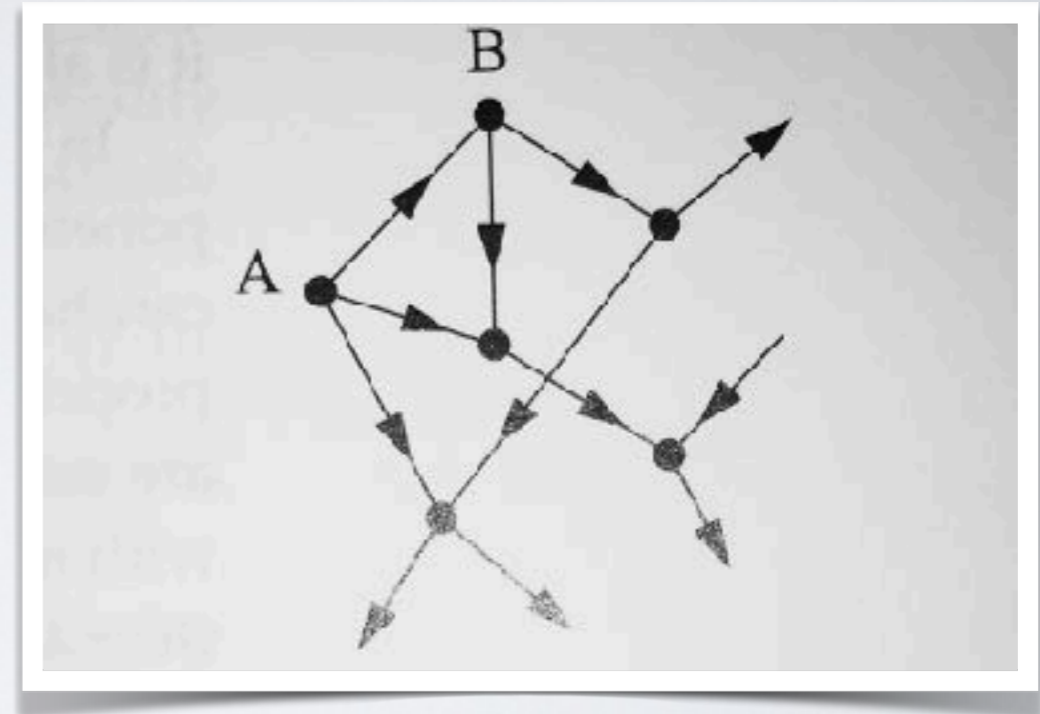
EIGENVECTOR CENTRALITY

- What we just described is called the Eigenvector centrality
- A couple eigenvector (x) and eigenvalue (λ) is defined by the following relation: $Ax = \lambda x$
 - x is a column vector of size n , which can be interpreted as the scores of nodes
- What Perron-Frobenius algorithm says is that the power method will always converge to the *leading eigenvector*, i.e., the eigenvector associated with the highest eigenvalue

Eigenvector Centrality

Some problems in case of **directed network**:

- Adjacency matrix is asymmetric
- 2 sets of eigenvectors (Left & Right)
- 2 leading eigenvectors
 - Use right eigenvectors : consider nodes that are pointing towards you



But problem with source nodes (0 in-degree)

- Vertex A is connected but has only outgoing link = Its centrality will be 0
- Vertex B has outgoing and an incoming link, but incoming link comes from A = Its centrality will be 0
- etc.

Solution: Only in strongly connected component

Note: Acyclic networks (citation network) do not have strongly connected component

PageRank Centrality

- Eigenvector centrality generalised for directed networks

PageRank

The Anatomy of a Large-Scale Hypertextual Web Search Engine

Brin, S. and Page, L. (1998) The Anatomy of a Large-Scale Hypertextual Web Search Engine. In: Seventh International World-Wide Web Conference (WWW 1998), April 14-18, 1998, Brisbane, Australia.

Sergey Brin and Lawrence Page

*Computer Science Department,
Stanford University, Stanford, CA 94305, USA
sergey@cs.stanford.edu and page@cs.stanford.edu*

PageRank Centrality

- Eigenvector centrality generalised for directed networks

PageRank

The Anatomy of a Large-Scale Hypertextual Web Search Engine

Brin, S. and Page, L. (1998) The Anatomy of a Large-Scale Hypertextual Web Search Engine. In: Seventh International World-Wide Web Conference (WWW 1998), April 14-18, 1998, Brisbane, Australia.

Sergey Brin and Lawrence Page

*Computer Science Department,
Stanford University, Stanford, CA 94305, USA
sergey@cs.stanford.edu and page@cs.stanford.edu*

Abstract

In this paper, we present Google, a prototype of a large-scale search engine which makes heavy use of the structure present in hypertext. Google is designed to crawl and index the Web efficiently and produce much more satisfying search results than existing systems. The prototype with a full text and hyperlink database of at least 24 million pages is available at <http://google.stanford.edu/>

PAGERANK

- 2 main improvements over eigenvector centrality:
 - ▶ In directed networks, problem of source nodes
 - => Add a constant centrality gain for every node
 - ▶ Nodes with very high centralities give very high centralities to all their neighbors (even if that is their only in-coming link)
 - => What each node “is worth” is divided equally among its neighbors (normalization by the degree)

$$C_u^{t+1} = \frac{1}{\lambda} \sum_{v \in N_u^{in}} C_v^t$$

=>

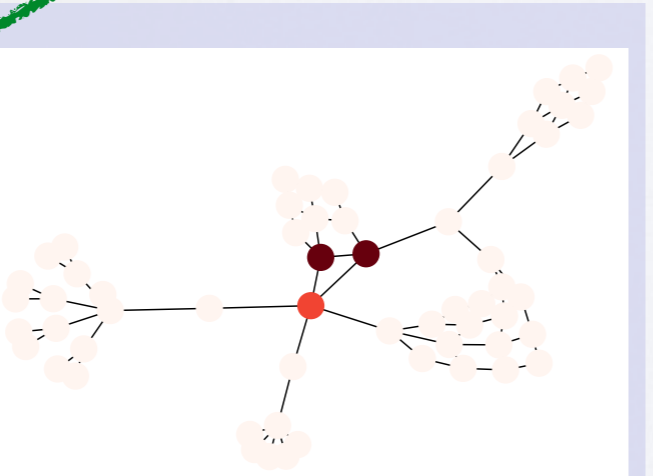
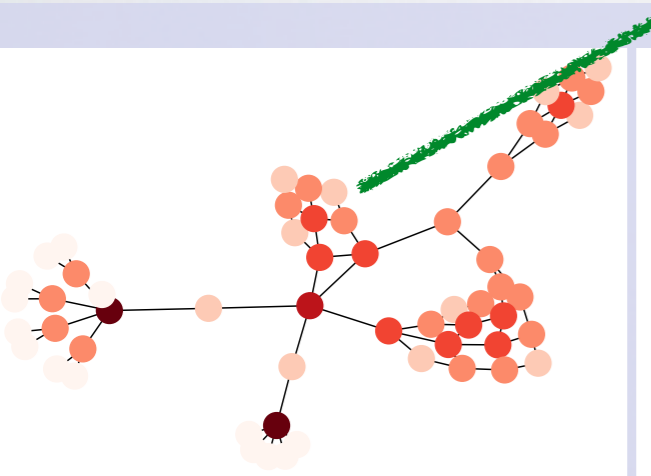
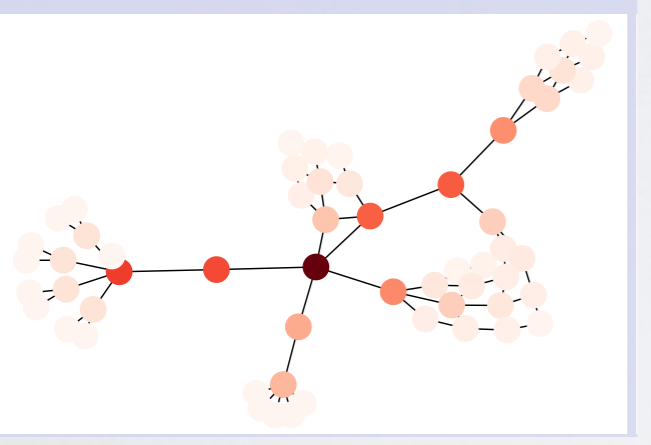
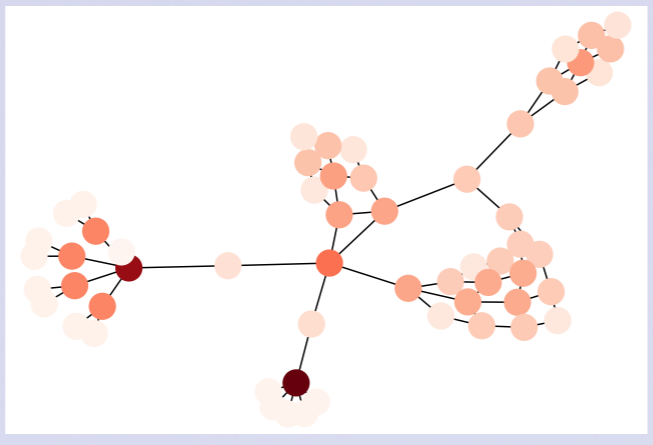
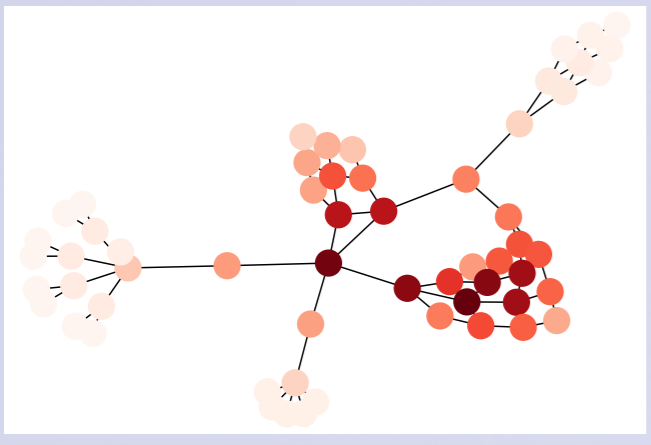
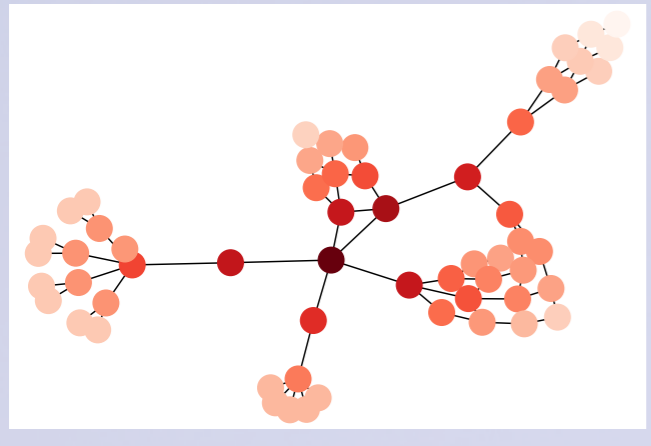
$$C_u^{t+1} = \alpha \sum_{v \in N_u^{in}} \frac{C_v^t}{k_v^{out}} + \beta$$

With by convention $\beta=1$ and α a parameter (usually 0.85) controlling the relative importance of β

PAGERANK

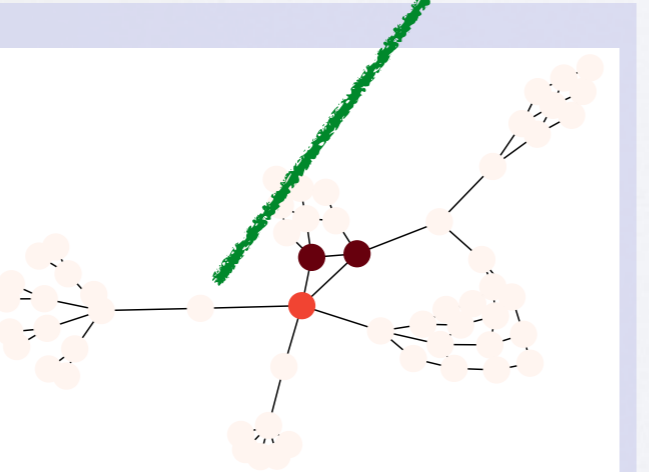
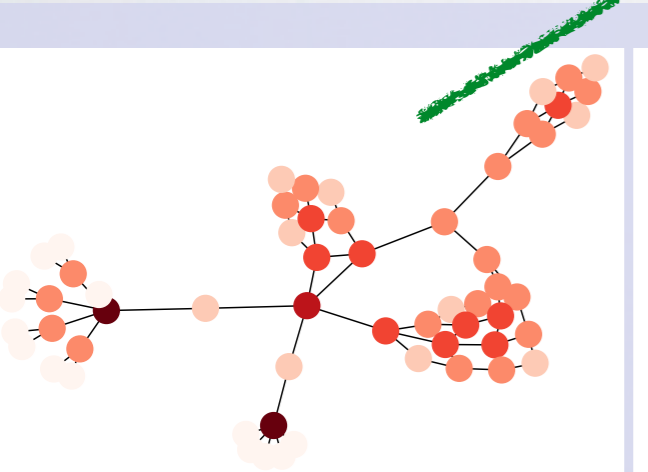
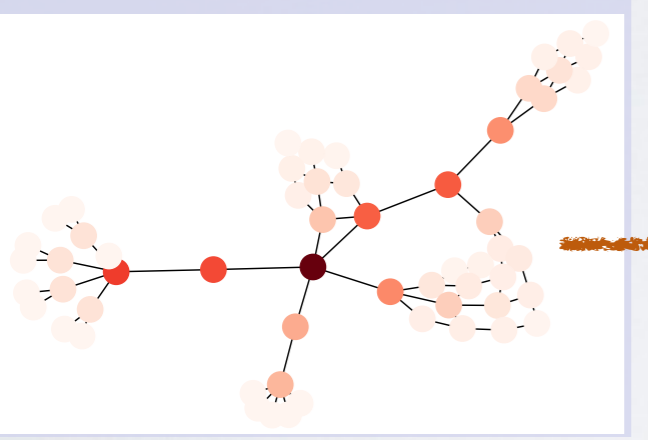
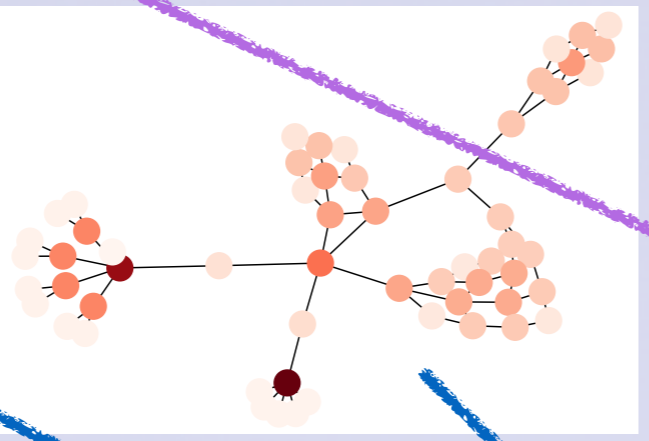
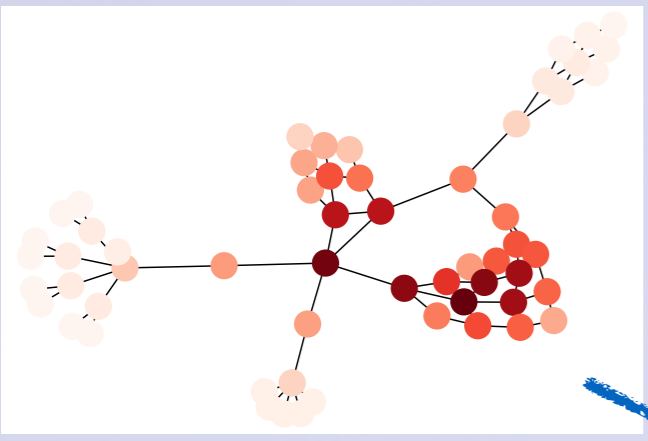
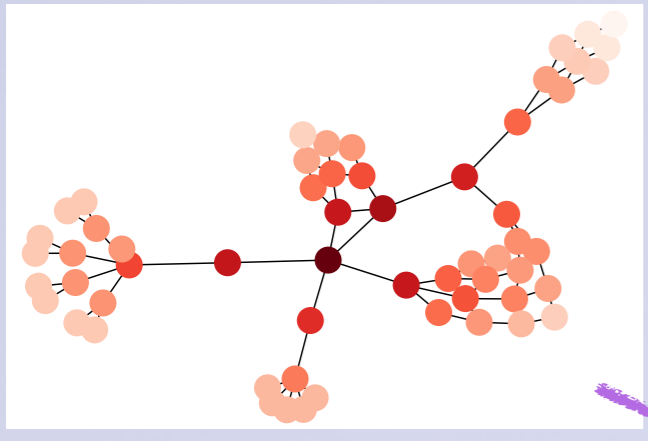
- Then how do Google rank when we do a research?
- Compute Pagerank (using the power method for scalability)
- Create a subgraph of documents related to our topic
- Of course now it is certainly much more complex, but we don't really know:
“Most search engine development has gone on at companies with little publication of technical details. This causes search engine technology to remain largely a black art” [Page, Brin, 1997]

Which is which ?

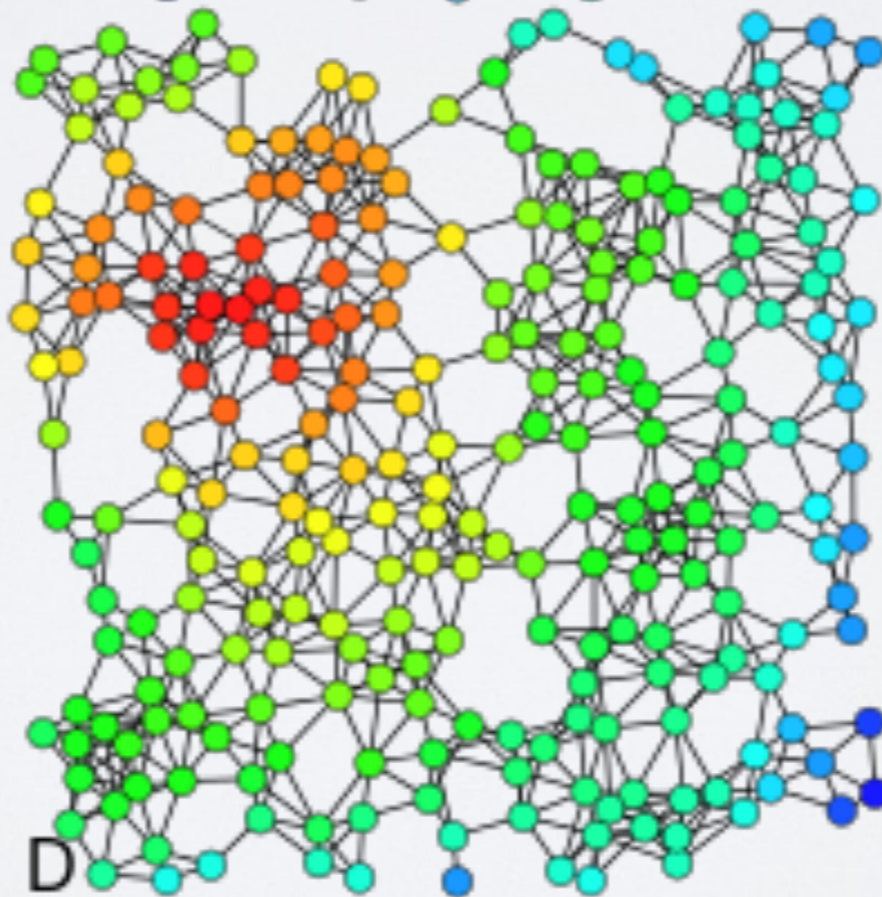
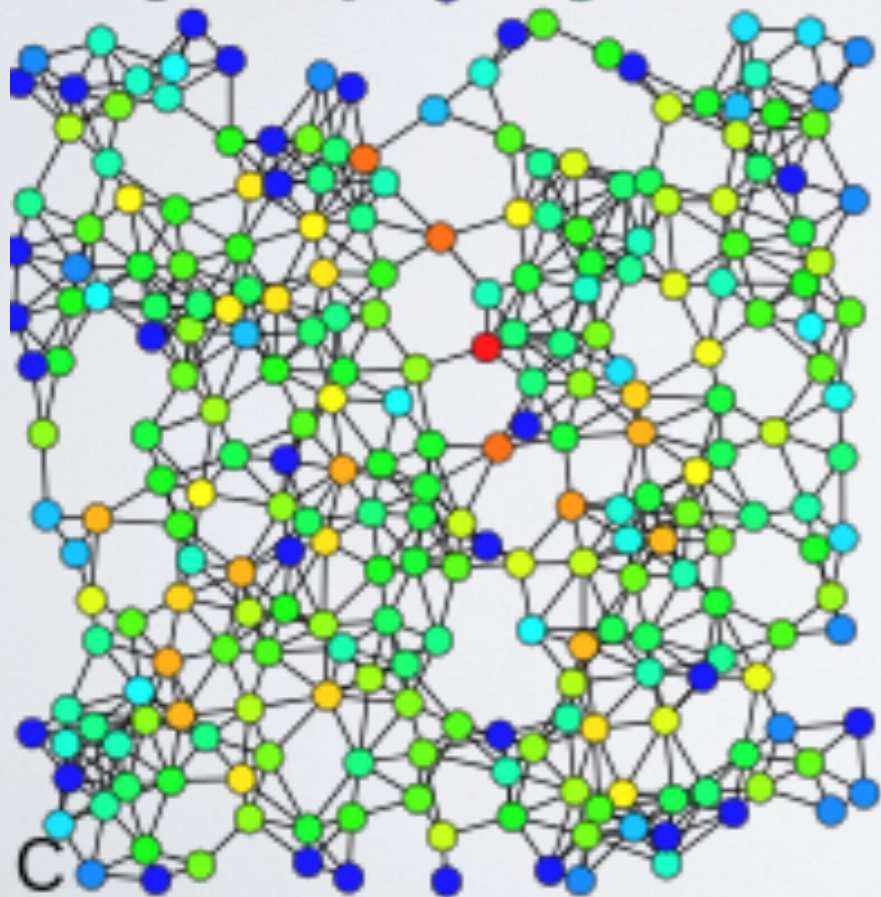
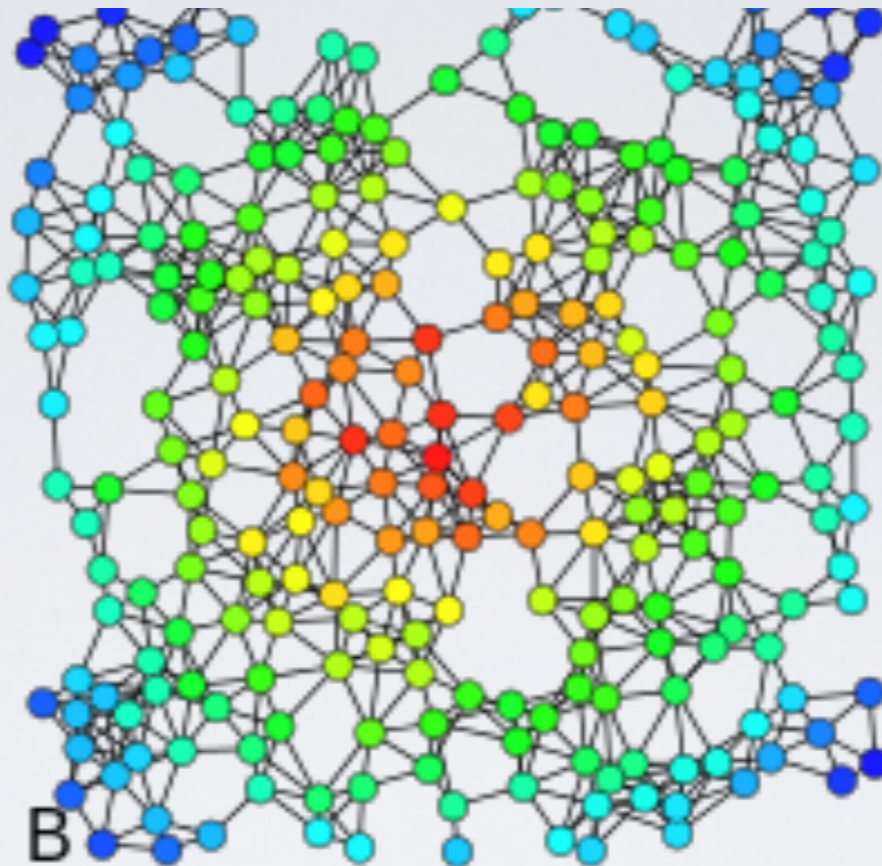
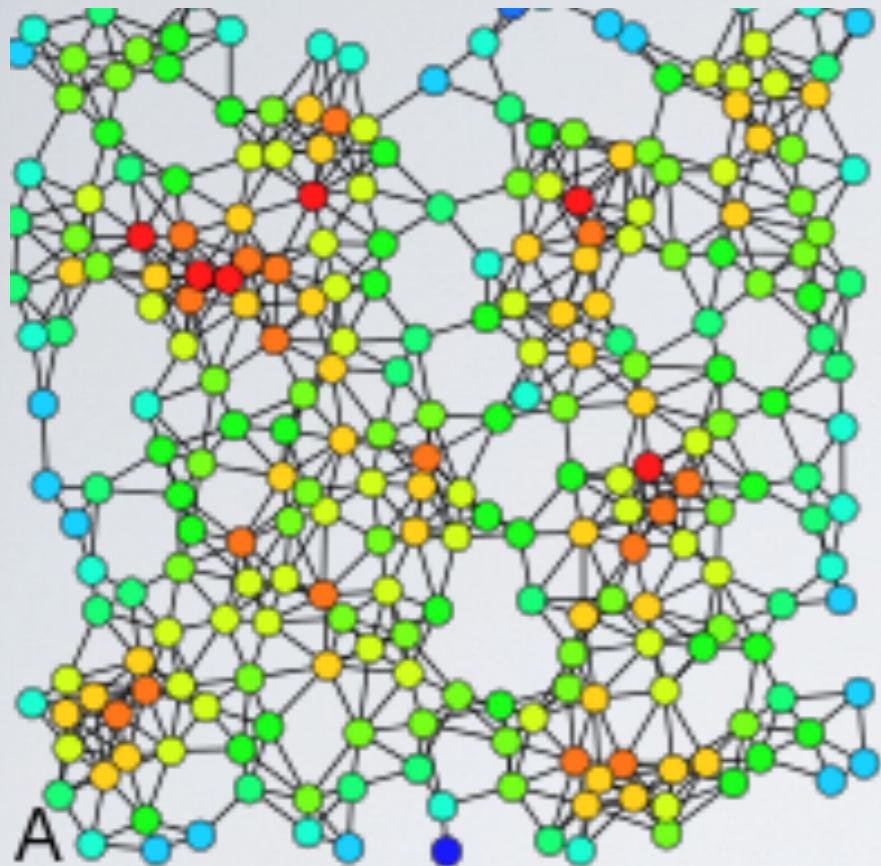


Degree
Clustering coefficient
Closeness
Betweenness
Eigenvector
PageRank

Which is which ?

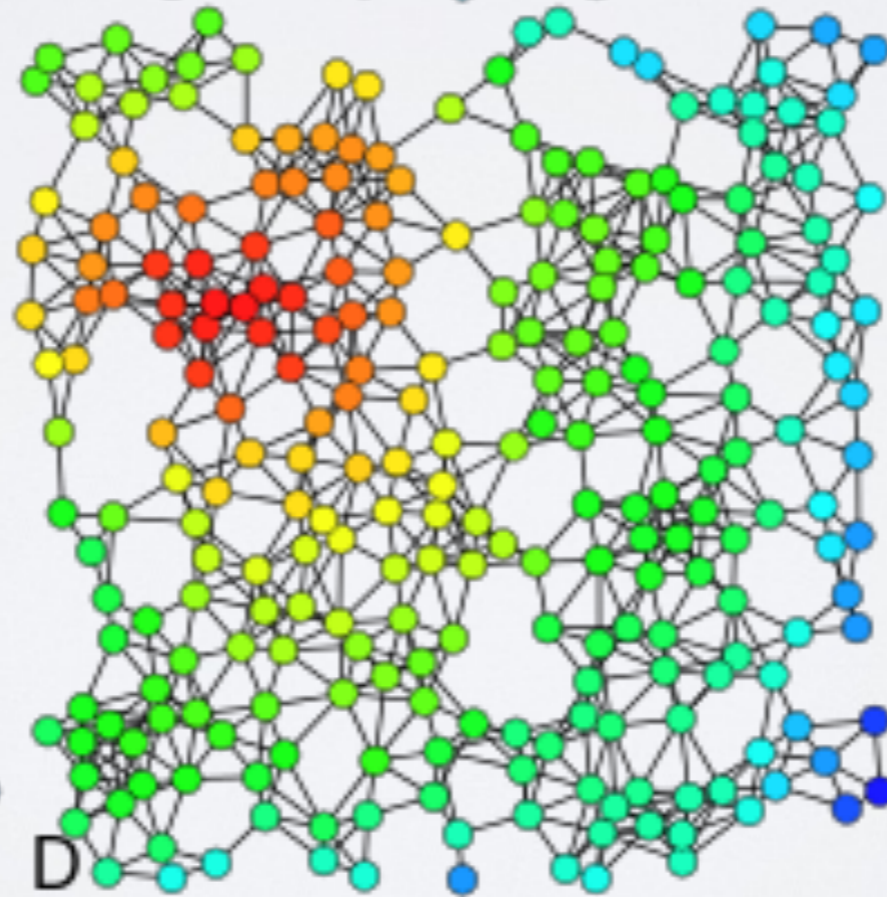
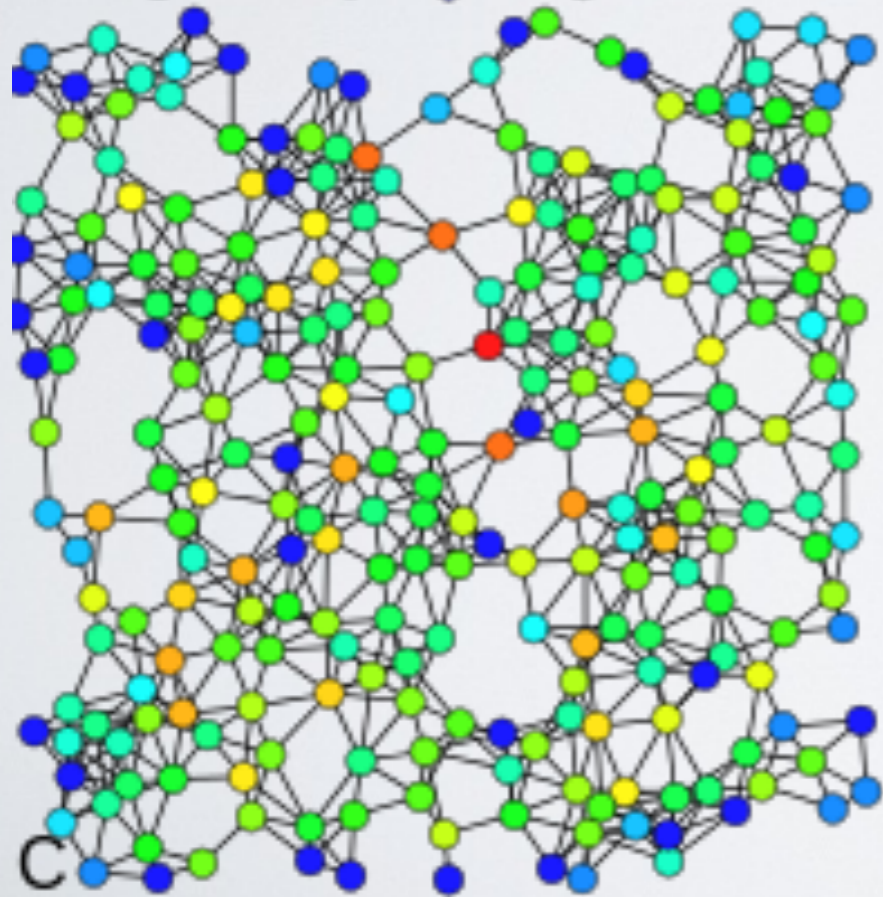
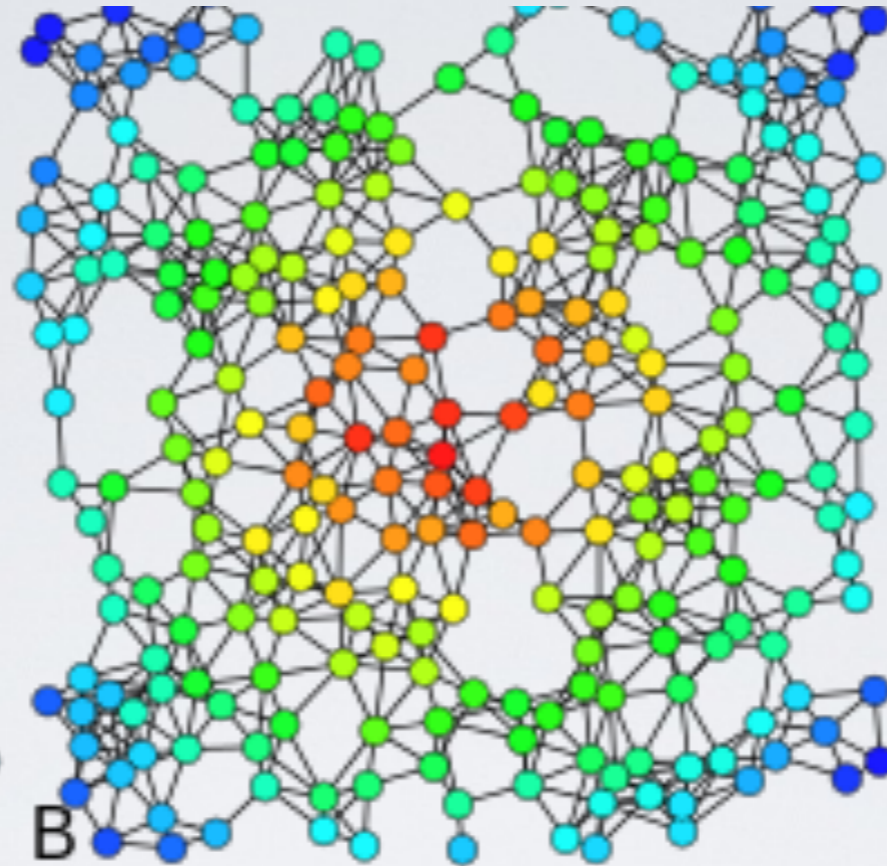
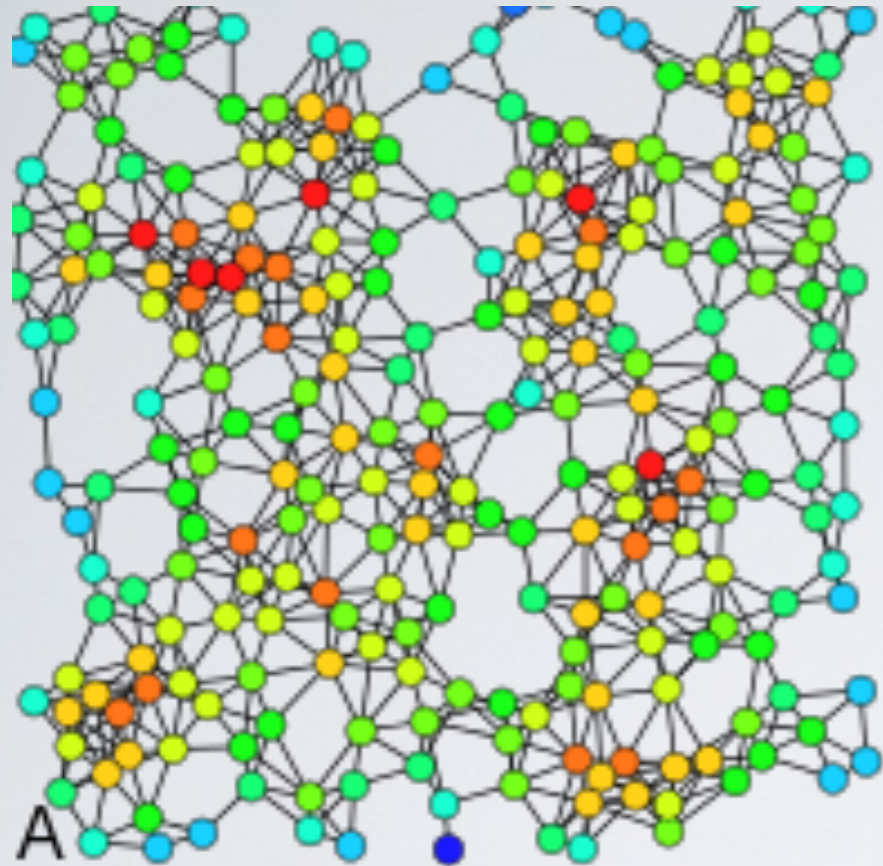


- Degree
- Clustering coefficient
- Closeness
- Betweenness
- Eigenvector
- PageRank



Try again :)

Degree
Betweenness
Closeness
Eigenvector



Try again :)

A: Degree

B: Closeness

C: Betweenness

D: Eigenvector

COMMUNITY DETECTION (GRAPH CLUSTERING)

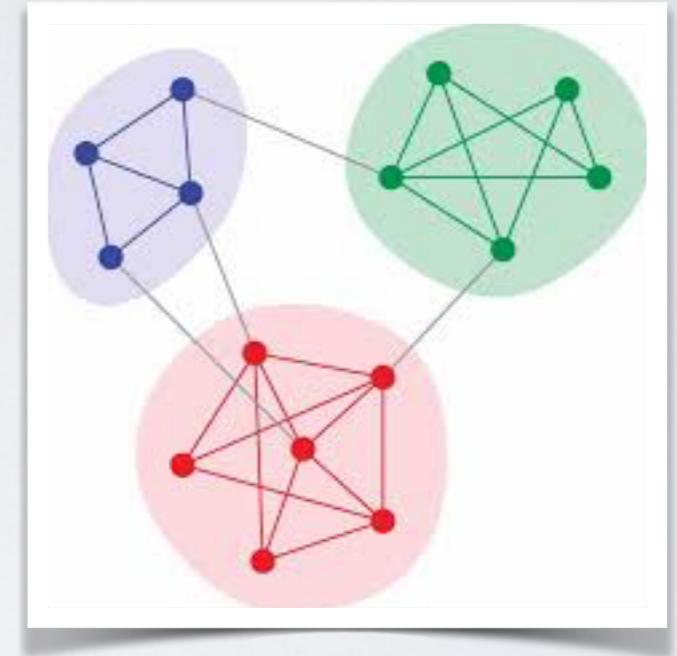
COMMUNITY DETECTION

- Community detection is equivalent to “clustering” in unstructured data
- Clustering: unsupervised machine learning
 - ▶ Find groups of elements that are similar to each other
 - People based on DNA, apartments based on characteristics, etc.
 - ▶ Hundreds of methods published since 1950 (k-means)
 - ▶ Problem: what does “similar to each other” means ?

COMMUNITY DETECTION

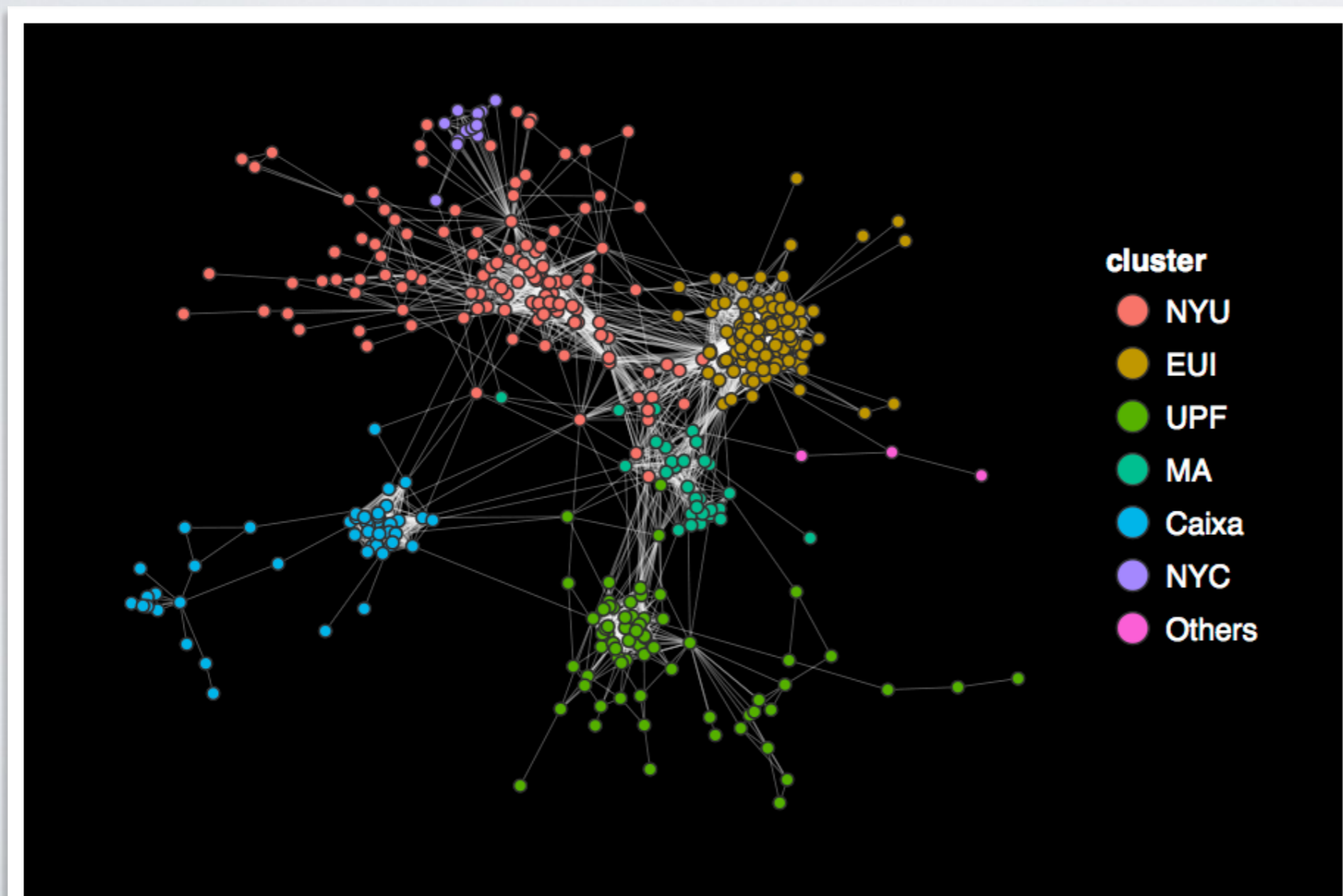
- Community detection:

- ▶ Find groups of nodes that are:
 - Strongly connected to each other
 - Weakly connected to the rest of the network
 - Ideal form: each community is 1) A clique, 2) A separate connected component
- ▶ No formal definition
- ▶ Hundreds of methods published since 2003



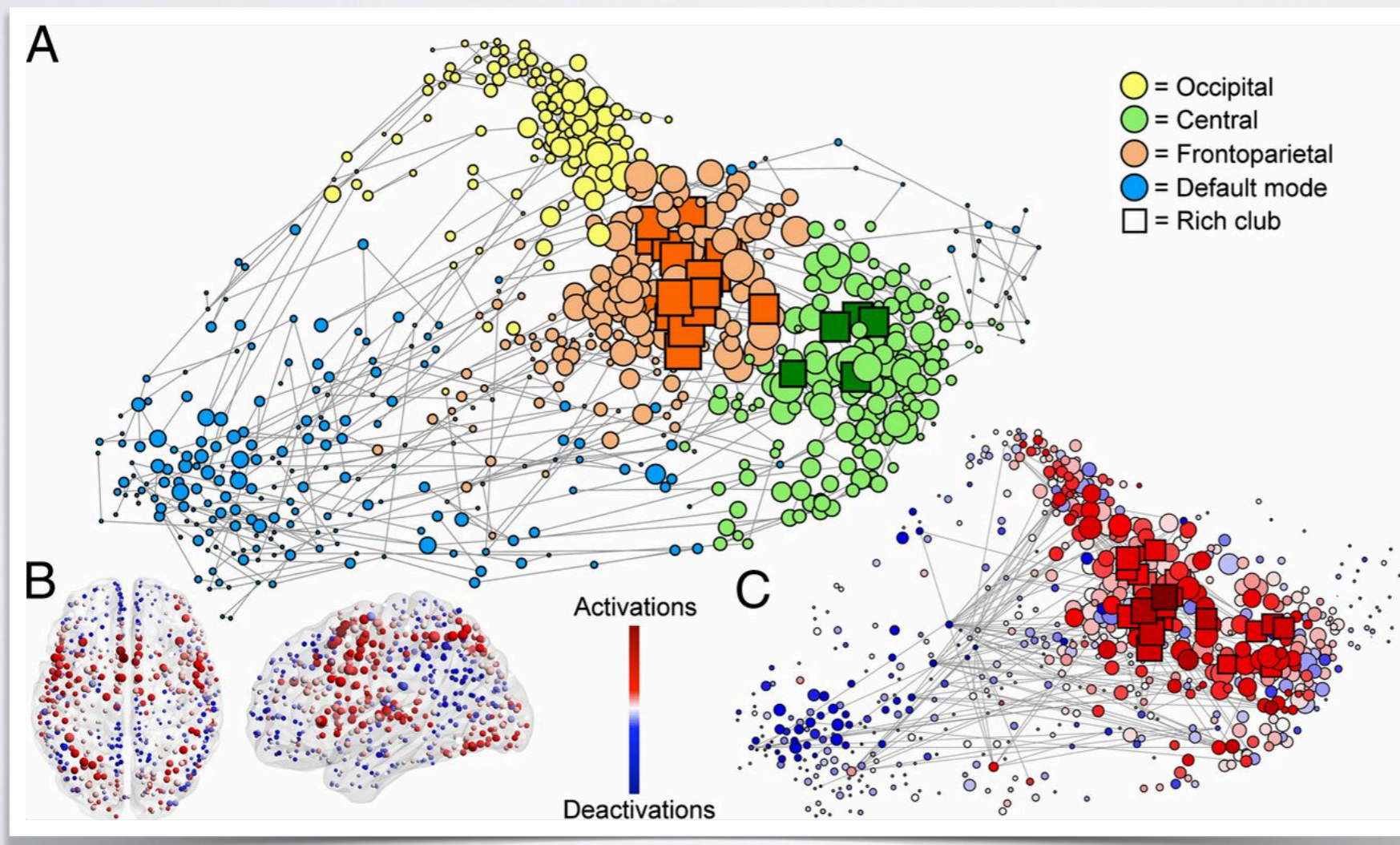
COMMUNITY STRUCTURE IN REAL GRAPHS

- If you plot the graph of your facebook friends, it looks like this



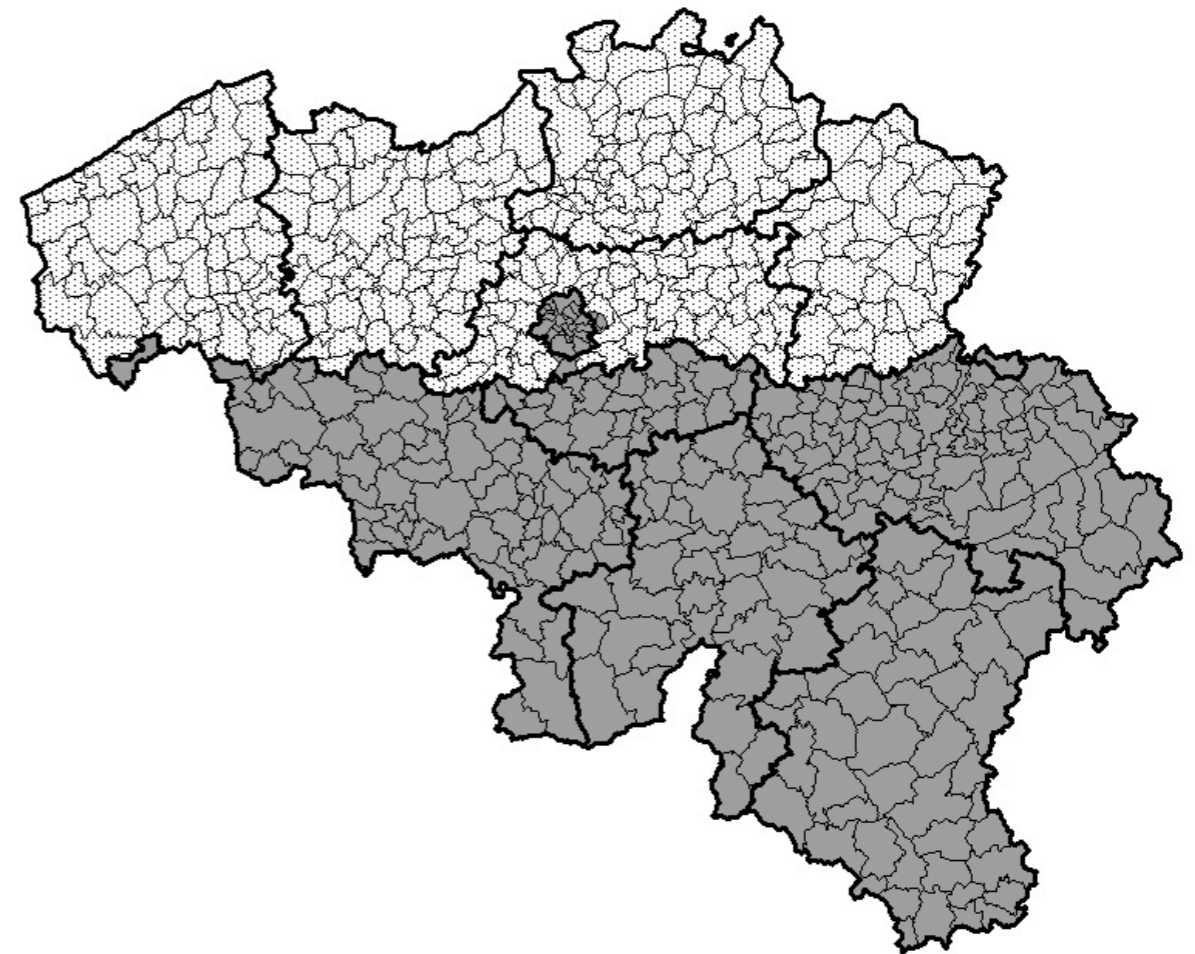
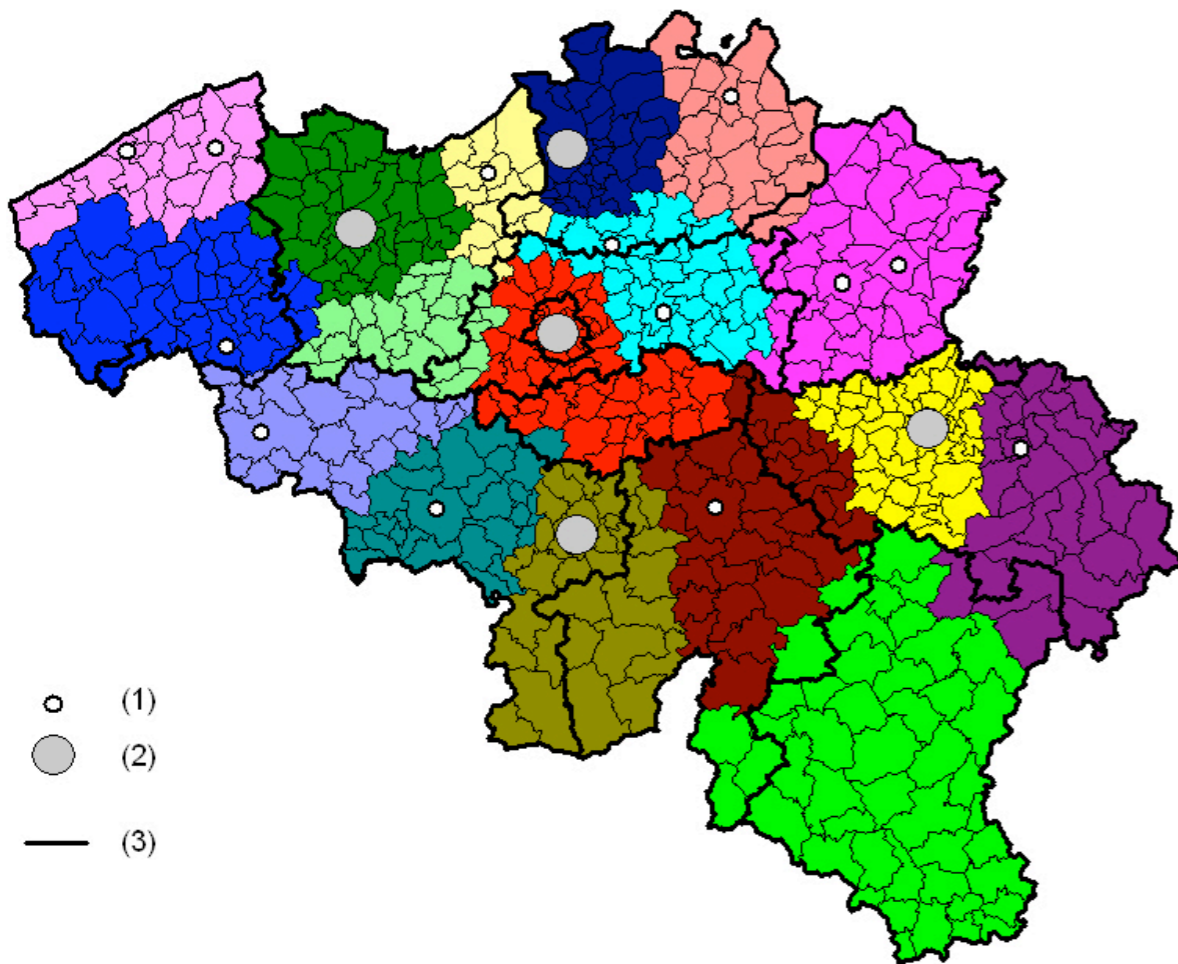
COMMUNITY STRUCTURE IN REAL GRAPHS

- Connections in the brain ?



COMMUNITY STRUCTURE IN REAL GRAPHS

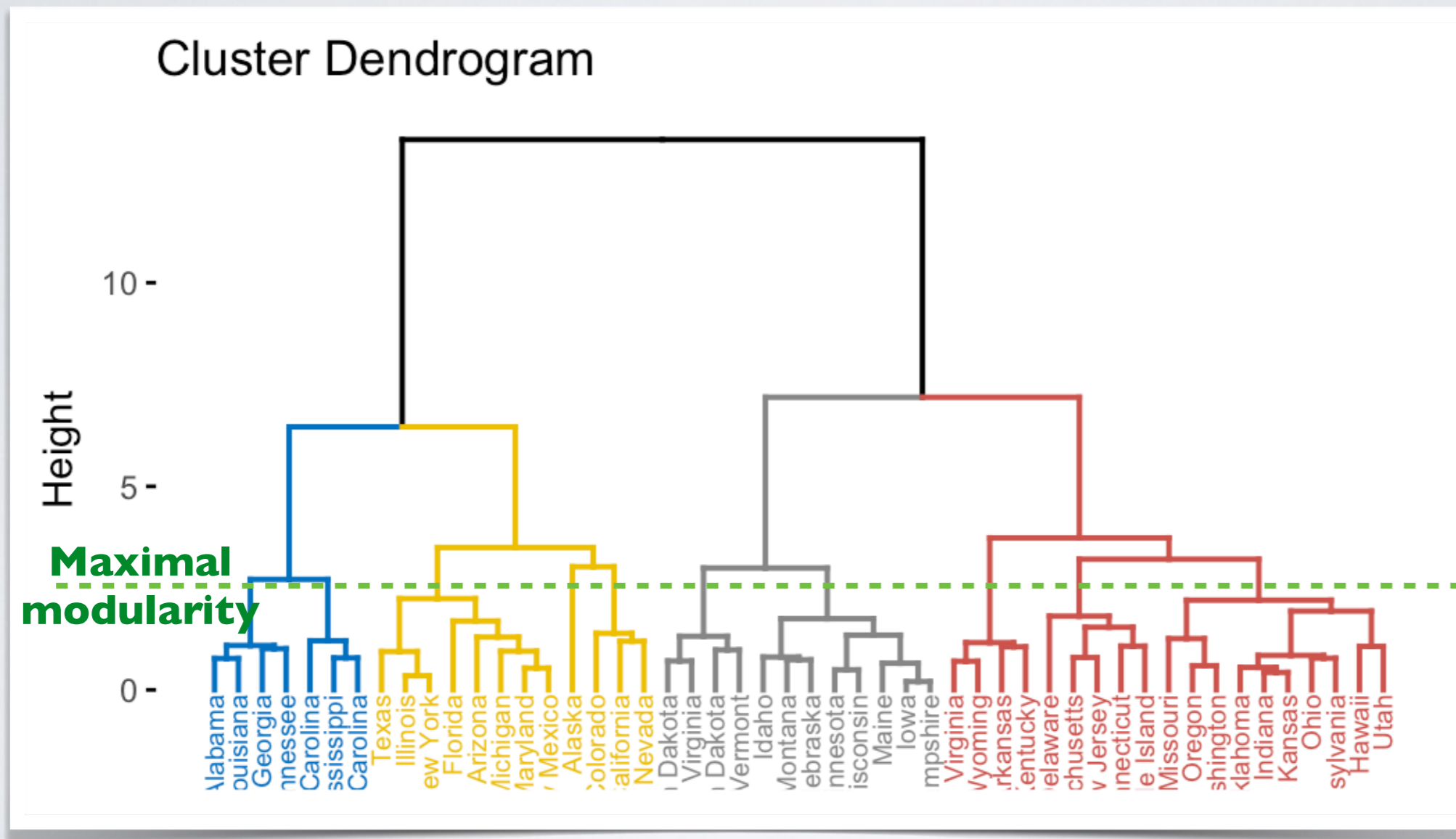
- Phone call communications in Belgium ?



FIRST METHOD BY GIRVAN & NEWMAN

- 1) Compute the betweenness of all edges
- 2) Remove the edge of highest betweenness
- 3) Repeat until all edges have been removed
 - Connected components are communities
- => It is called a *divisive* method
- => What you obtain is a dendrogram
- How to cut this dendrogram at the *best* level ?

FIRST METHOD BY GIRVAN & NEWMAN



FIRST METHOD BY GIRVAN & NEWMAN

- Introduction of the **Modularity**
- The modularity is computed for a partition of a graph
 - (each node belongs to one and only one community)
- It compares :
 - The **observed** *fraction of edges inside communities*
 - To the **expected** *fraction of edges inside communities* in a random network

MODULARITY

$$Q = \frac{1}{(2m)} \sum_{vw} \left[A_{vw} - \frac{k_v k_w}{(2m)} \right] \delta(c_v, c_w)$$

Original formulation

MODULARITY

$$Q = \frac{1}{(2m)} \sum_{vw} \left[A_{vw} - \frac{k_v k_w}{(2m)} \right] \delta(c_v, c_w)$$

Sum over all pairs of nodes

MODULARITY

$$Q = \frac{1}{(2m)} \sum_{vw} \left[A_{vw} - \frac{k_v k_w}{(2m)} \right] \delta(c_v, c_w)$$

| if in same community

MODULARITY

$$Q = \frac{1}{(2m)} \sum_{vw} \left[A_{vw} - \frac{k_v k_w}{(2m)} \right] \delta(c_v, c_w)$$

| if there is an edge between them

MODULARITY

$$Q = \frac{1}{(2m)} \sum_{vw} \left[A_{vw} - \frac{k_v k_w}{(2m)} \right] \delta(c_v, c_w)$$

Probability of an edge in
a configuration model
(Edges at random, keeping degrees)

MODULARITY

- Modularity compares the observed network to a **null model**
 - Usually the configuration model (degree preserving random graphs)
 - Multi-edges and loops are allowed
 - Other models could be used, such as ER random graphs (fully random)
- Natural extension to weighted/multi-edge networks

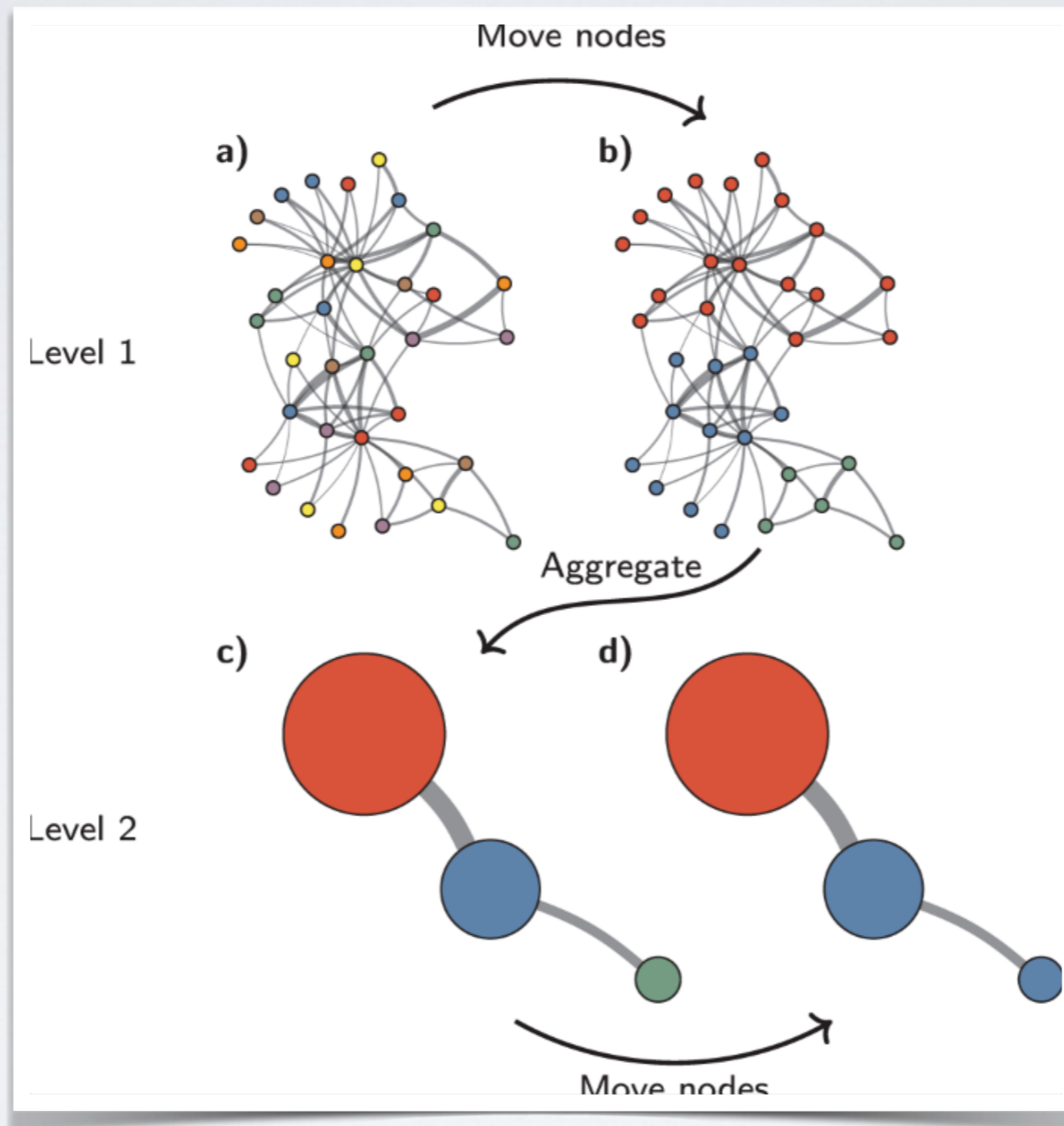
FIRST METHOD BY GIRVAN & NEWMAN

- Back to the method:
 - Create a dendrogram by removing edges
 - Cut the dendrogram at the best level using modularity
- => In the end, your objective is... to optimize the Modularity, right ?
- Why not optimizing it directly !

LOUVAIN ALGORITHM

- Greedy approach
- Each node start in its own community
- Repeat until convergence
 - FOR each node:
 - FOR each neighbor:
 - if adding node to its community increase modularity, do it
- When converged, create an *induced network*
 - Each community becomes a node
 - Edge weight is the sum of weights of edges between them
- Trick: Modularity is computed *by community*

LOUVAIN ALGORITHM

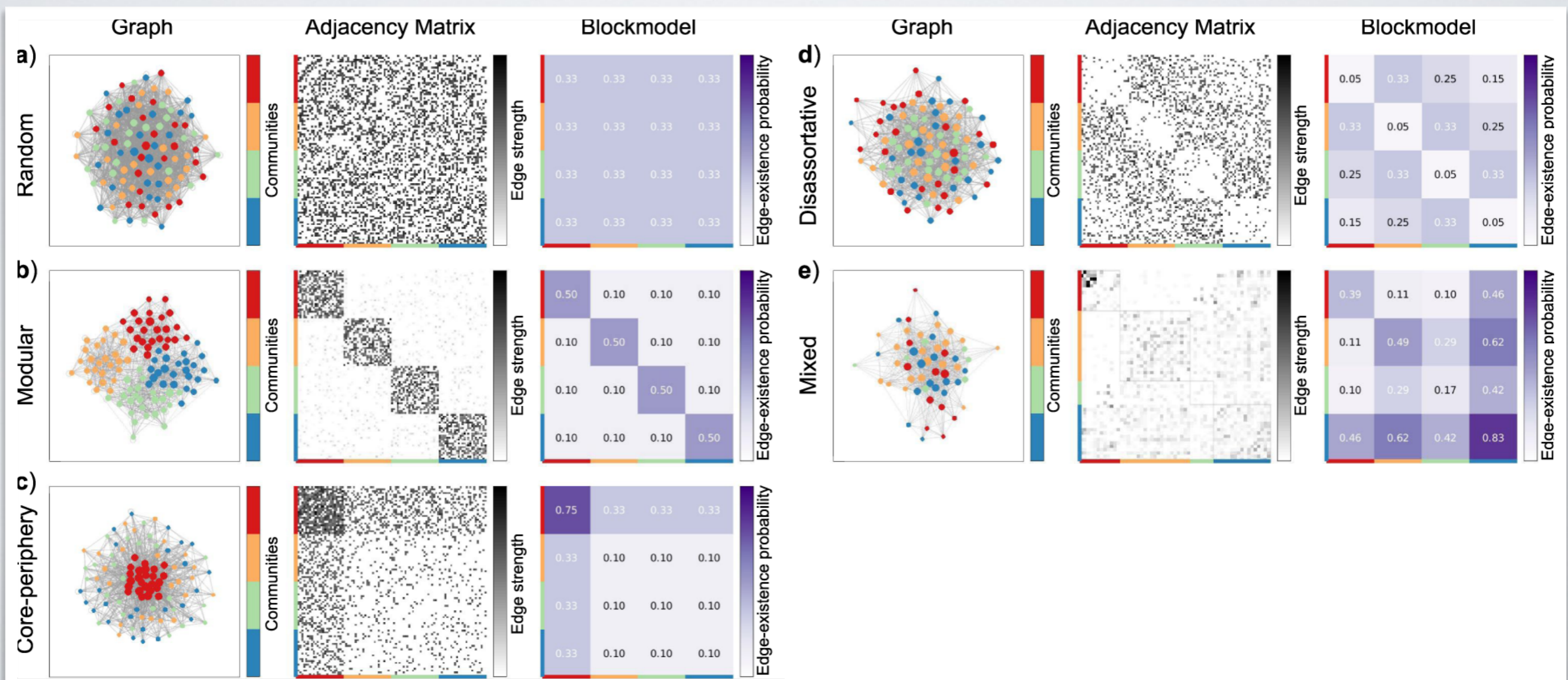


STOCHASTIC BLOCK MODELS

- Stochastic Block Models (SBM) are based on statistical models of networks
 - Likelihood maximization of observing the graph given a model
- The model is:
 - Each node belongs to 1 and only 1 community (blocks)
 - To each pair of communities, there is an associated density (probability of each edge to exist)

STOCHASTIC BLOCK MODELS

- SBM can represent different things:
 - Associative SBM: density inside nodes of a same communities \gg density of pairs belonging to different communities.



STOCHASTIC BLOCK MODELS

- General idea of SBM community detection:
 - ▶ Specify the desired number of cluster
 - ▶ Find parameters to optimize the maximum likelihood
 - Principle: The best parameters are those that allow to generate the observed network with the highest probability
- Main weakness of this approach
 - ▶ Number of clusters k must be specified (avoid trivial solution)
 - ▶ MDL (Minimum Description Length) approaches exist to find automatically k

EVALUATION OF COMMUNITY STRUCTURE

EVALUATION

- Similar to clustering:
 - ▶ Intrinsic/Internal evaluation
 - Partition quality function
 - Individual Community quality function
 - ▶ Comparison of observed communities and expected communities
 - Synthetic networks with community structure
 - Real networks with Ground Truth

INTRINSIC EVALUATION

- Partition quality function
 - Already defined: **Modularity**, **graph compression**, etc.

- Quality function for individual community

- Internal Clustering Coefficient

- Conductance: $\frac{|E_{out}|}{|E_{out}| + |E_{in}|}$

- Fraction of external edges

$|E_{in}|, |E_{out}|$:
of links to nodes inside
(respectively, outside) the
community