NETWORK DATA MINING

WHO AM I

- Rémy Cazabet (remy.cazabet@univ-lyon1.fr)
- Associate Professor (Maître de conférences)
  - Université Lyon 1
  - LIRIS, DM2L Team (Data Mining & Machine Learning)
- Computer Scientist => **Network Scientist**
- Member of IXXI, Lyon's institute of **Complex Systems**
- No background in economy, finance => BITUNAM Project
  - Bitcoin User Network Analysis and Mining

CLASS OVERVIEW

- Website of the class:
  - http://cazabetremy.fr/Teaching/bitcoinClass/BitcoinNetwork.html

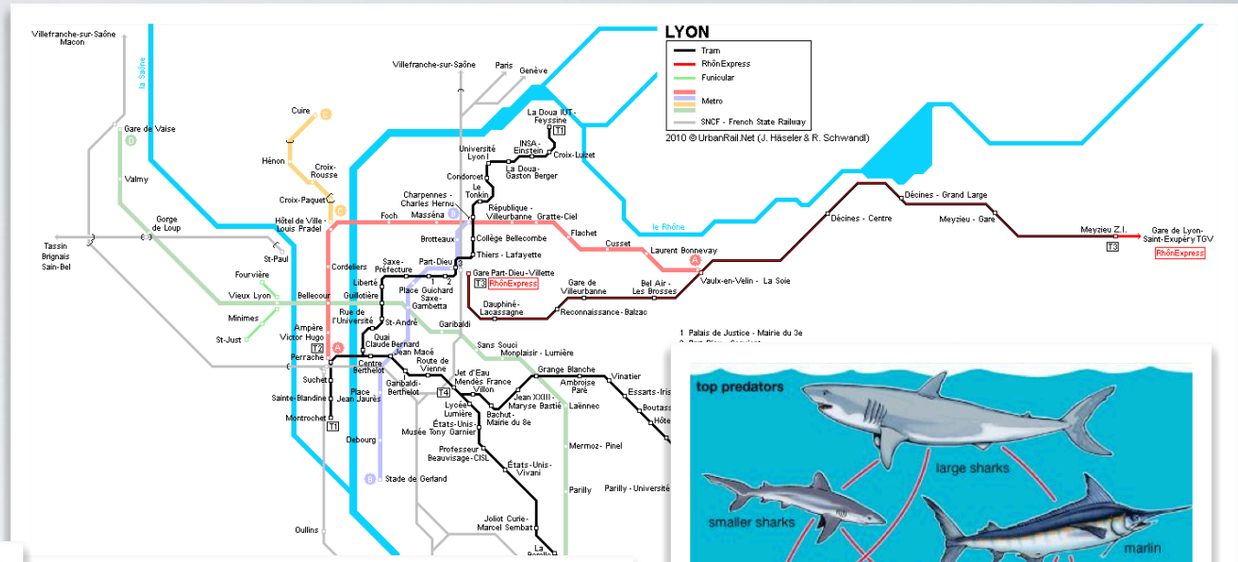
- Today: (Teacher: Rémy Cazabet)
  - Morning: Introduction to network, Gephi
  - Afternoon: Introduction to Bitcoin blockchain data, theory and python practice

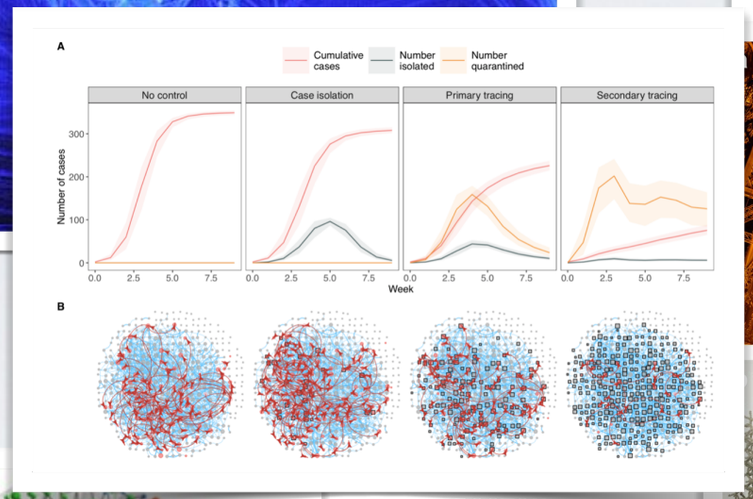
- Other days:
  - Teacher: Natkamon Tovanich
    - Network analysis with python, Ethereum Data, Defi …
    - Project: Presentation of a real dataset, supervision…

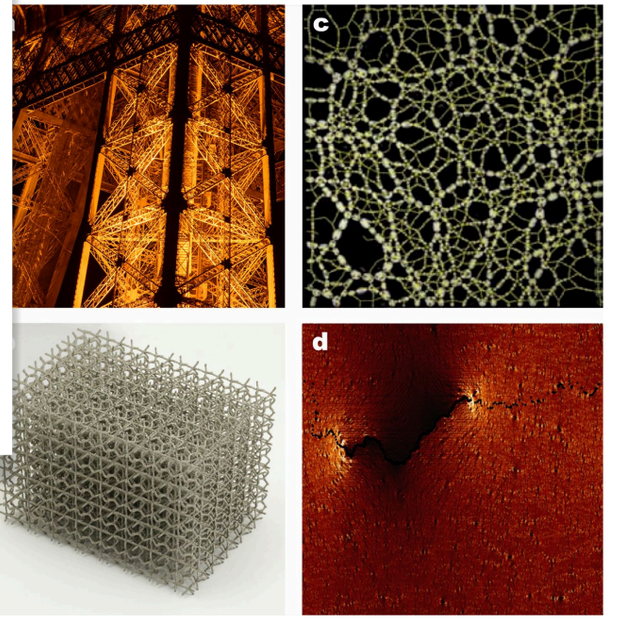
NETWORKS

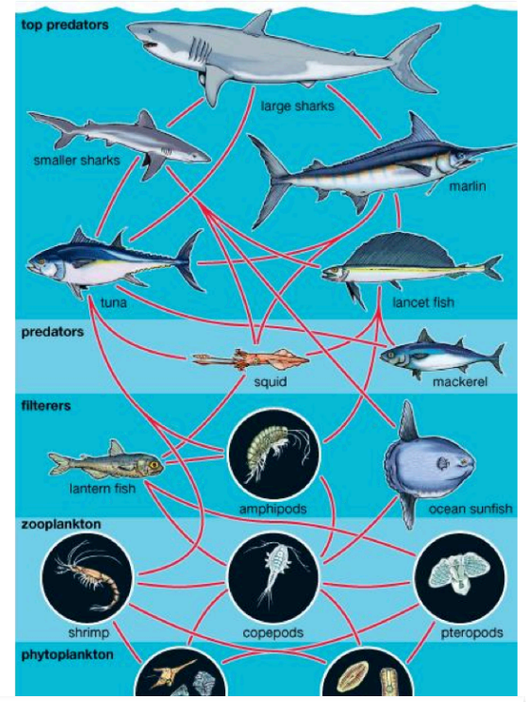
ECONOMIC NETWORKS?

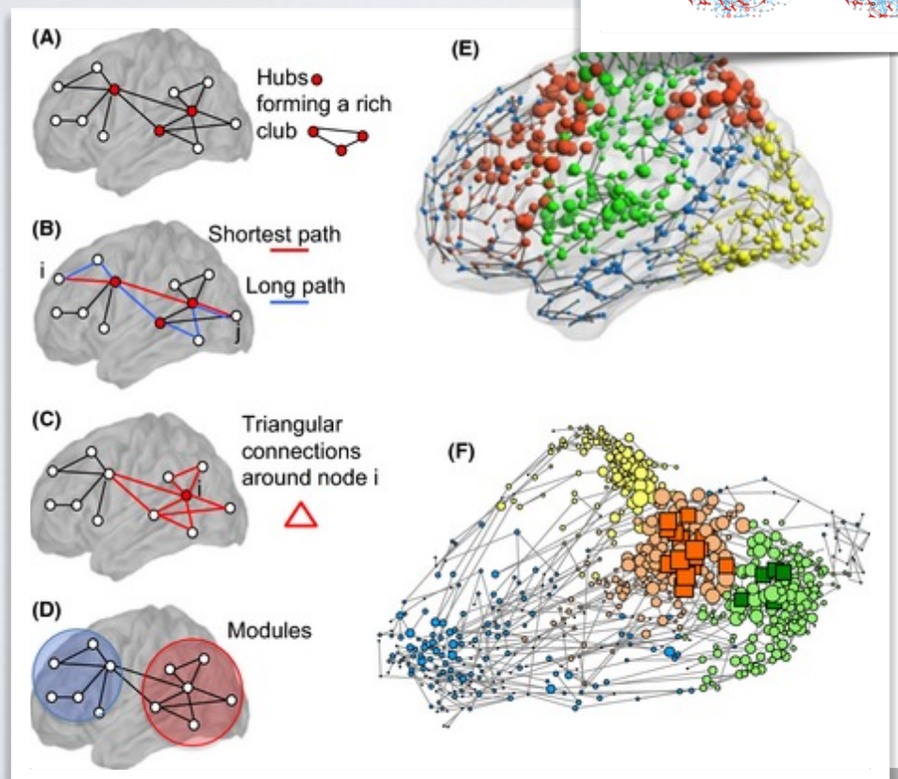
- Economic/Financial Data are composed of **transactions**
  - **Entity X1** (User/Company…) Exchange value/G-good with **entity X2**
  - **X1->X2**
- The collection of all these transactions composes a Network/ Graph
  - Understanding this network allow to better understand the economic/financial system.

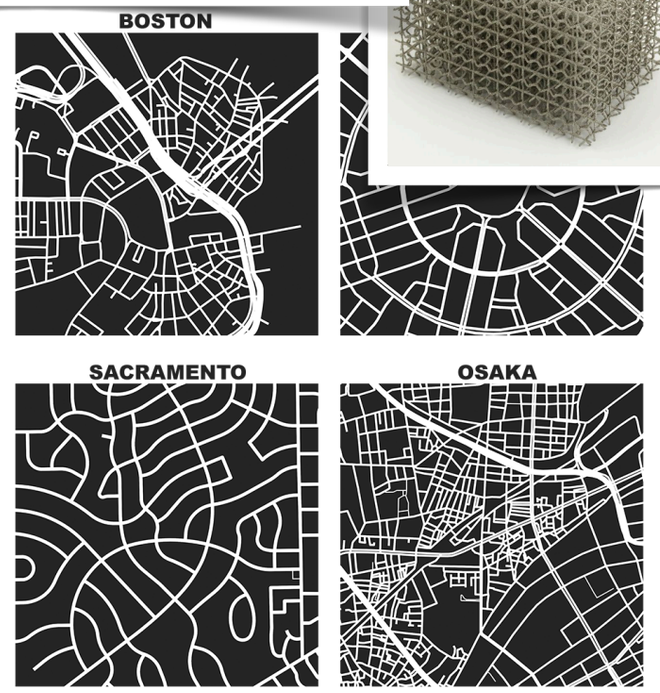


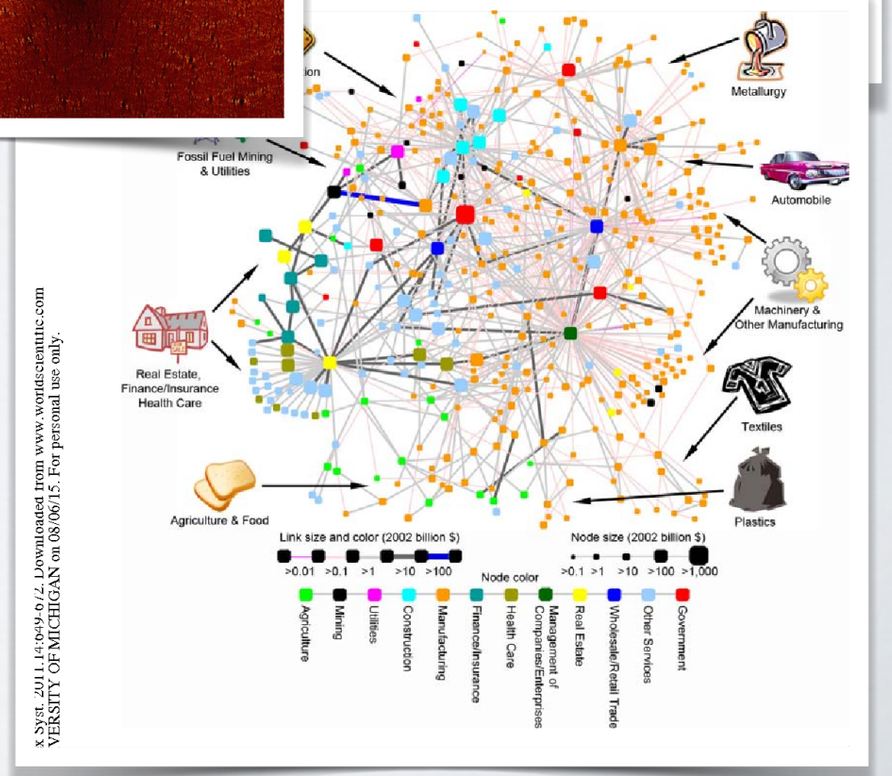


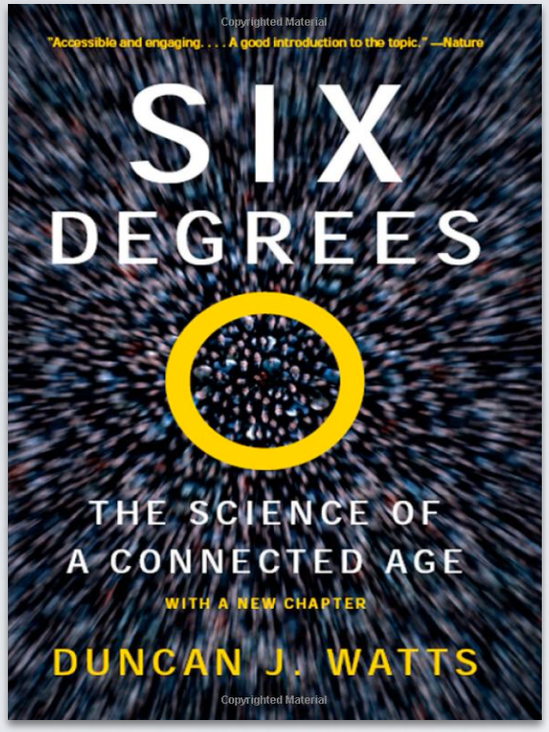


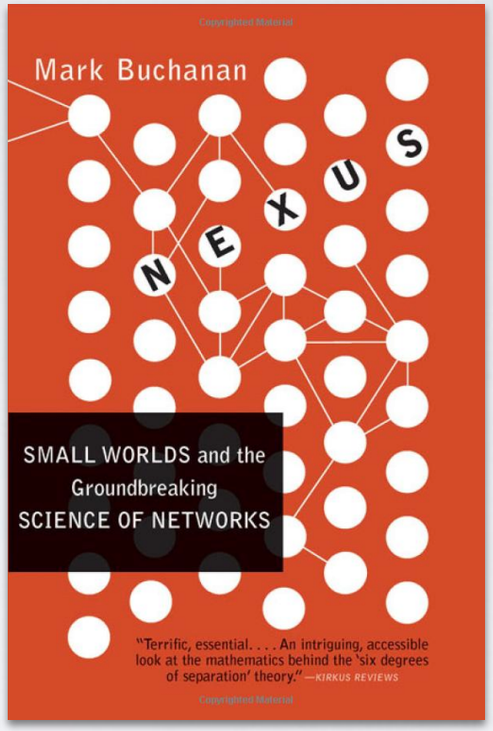


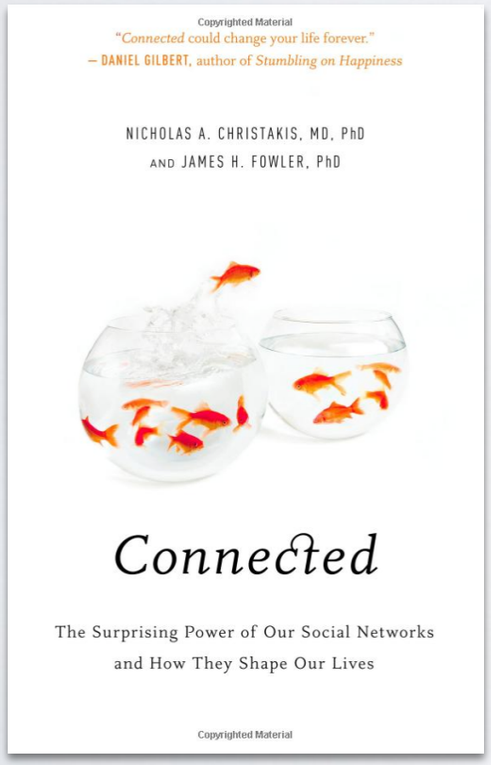


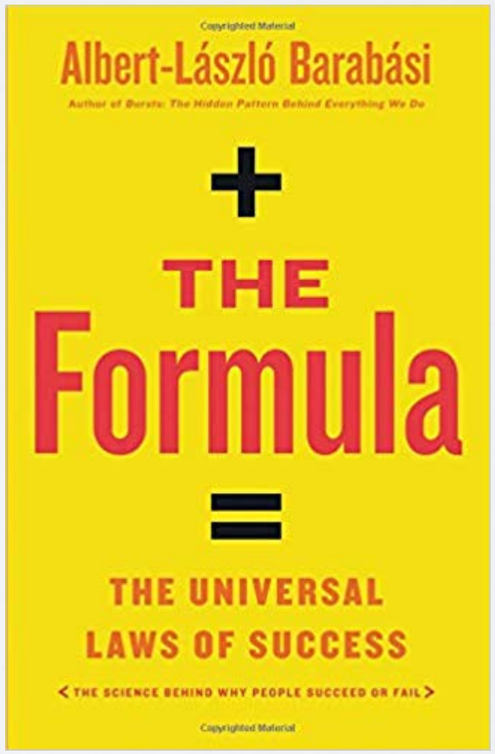






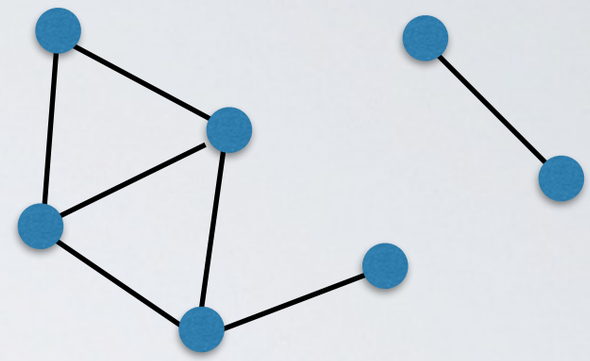






J'ai une copie que je peux prêter

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)

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

In most cases we will use the two terms interchangeably.

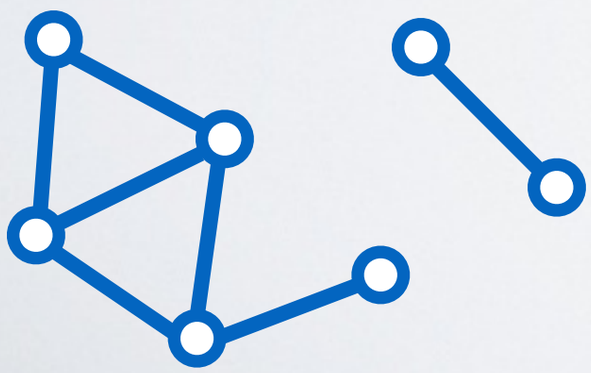
Types of Networks

Undirected networks

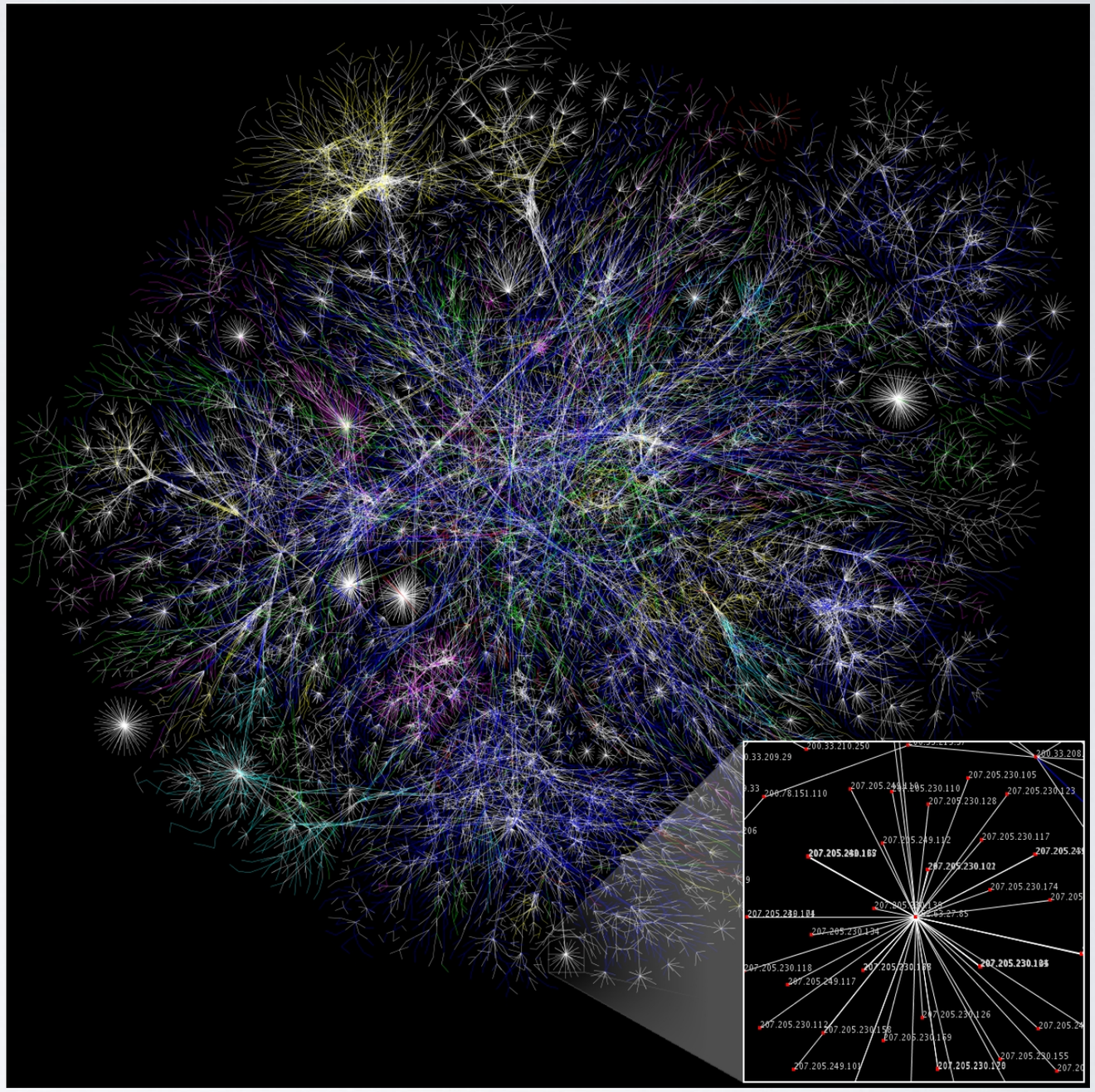
$G=(V, E)$

$(u,v) \in E \equiv (v,u) \in E$

- The directions of edges do not matter
- Interactions are possible between connected entities in both directions



Opte project



The Internet: Nodes - routers, Links - physical wires

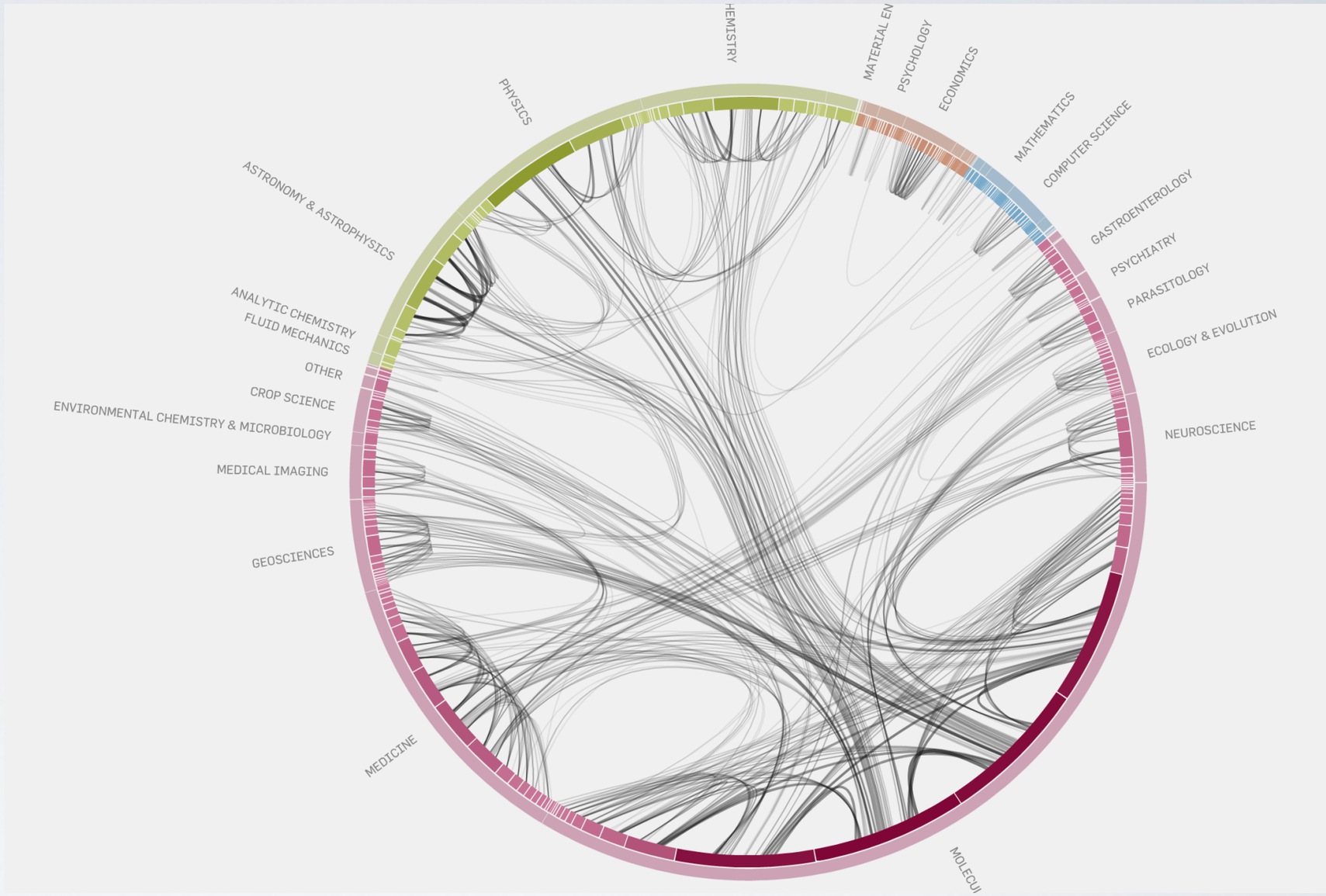
Directed networks

Moritz Stefaner, eigenfactor.com

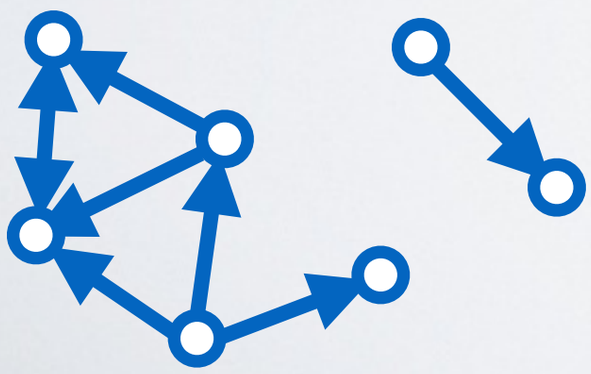
$$G=(V, E)$$

$$(u,v) \in E \not\equiv (v,u) \in E$$

- The directions of edges matter
- Interactions are possible between connected entities only in specified directions



Citation network: Nodes - publications, Links - references

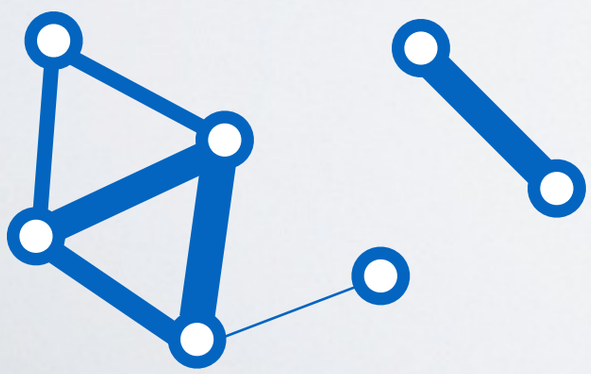


Weighted networks

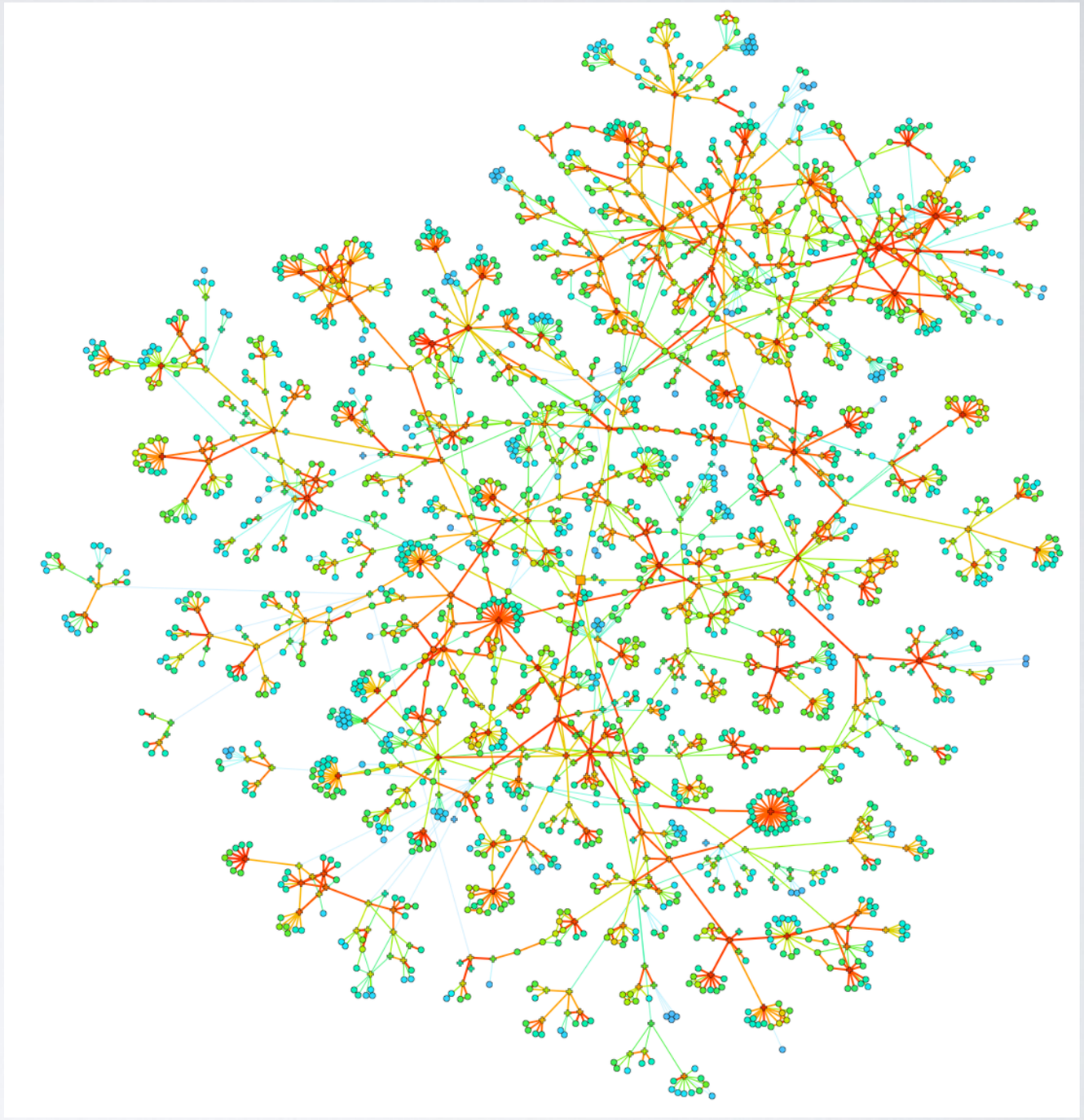
$G=(V, E, w)$

$w: (u,v) \in E \Rightarrow R$

- Strength of interactions are assigned by the weight of links

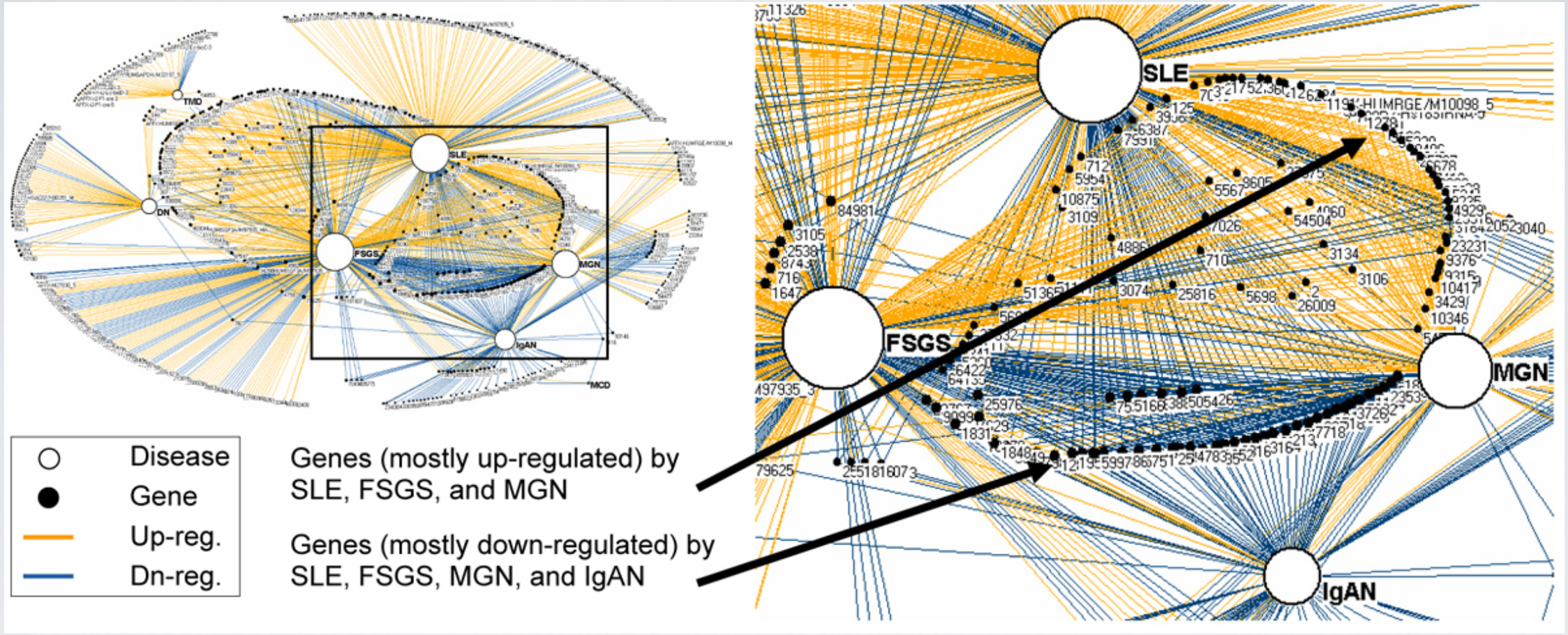


Onnela et.al. New Journal of Physics 9, 179 (2007).

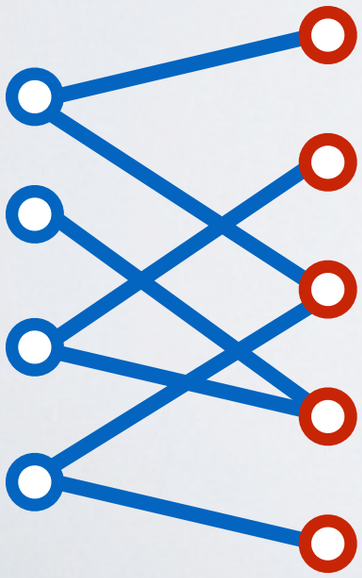


Social interaction network: Nodes - individuals
Links - social interactions

Bipartite network



Bhavnani et.al. BMC Bioinformatics 2009, **10**(Suppl 9):S3



Gene-desease network:

Nodes - Desease (7)&Genes (747)

Links - gene-desease relationship

$$G=(U, V, E)$$

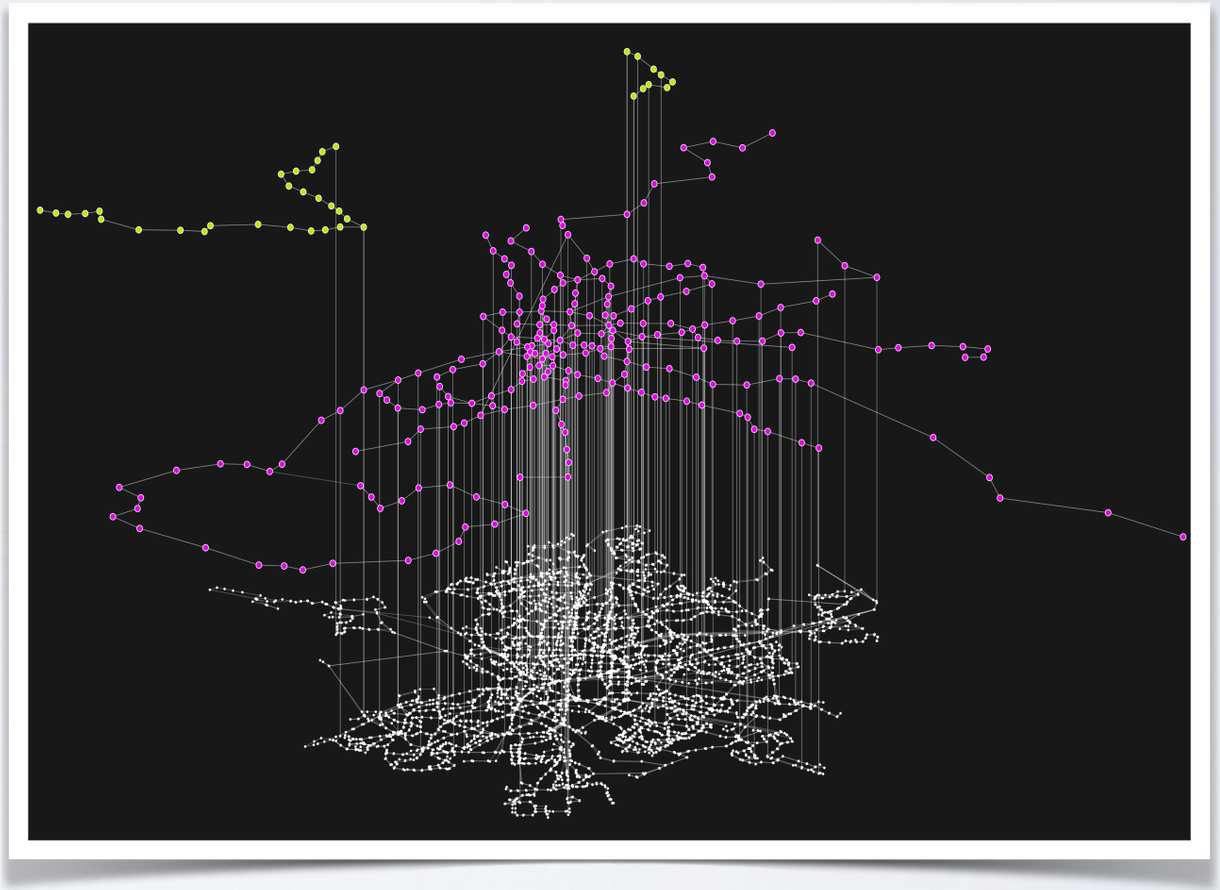
$$U \cap V = \varnothing$$

$$\forall (u,v) \in E,\ u \in U \text{ and } v \in V$$

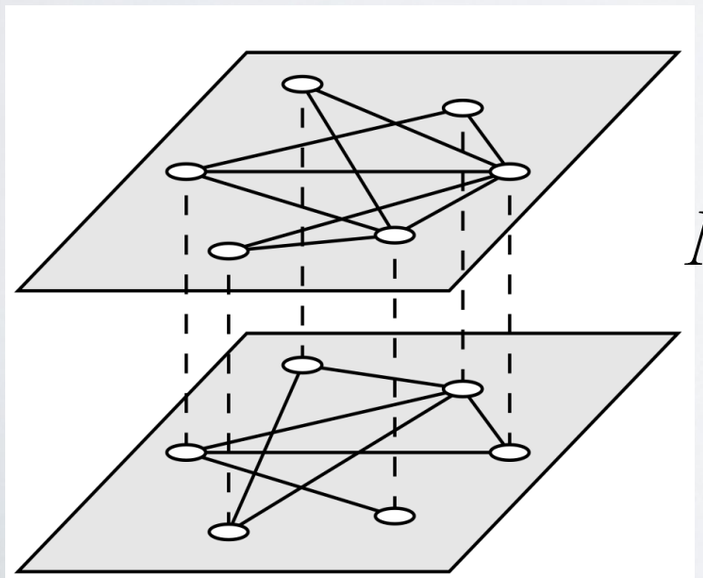
Multiplex and multilayer networks

$G=(V, E_i), i=1 \ldots M$

- Nodes can be present in multiple networks simultaneously
- These networks are connected (can influence each other) via the common nodes



[Mendez-Bermudez et al. 2017]



$M=2$

Gomes et.al. Phys. Rev. Lett. 110, 028701 (2013)

Temporal and evolving networks

$G=(V, E_t), (u,v,t,d) \in E_t$

t - time of interaction (u,v)
d - duration of interaction (u,v,t)

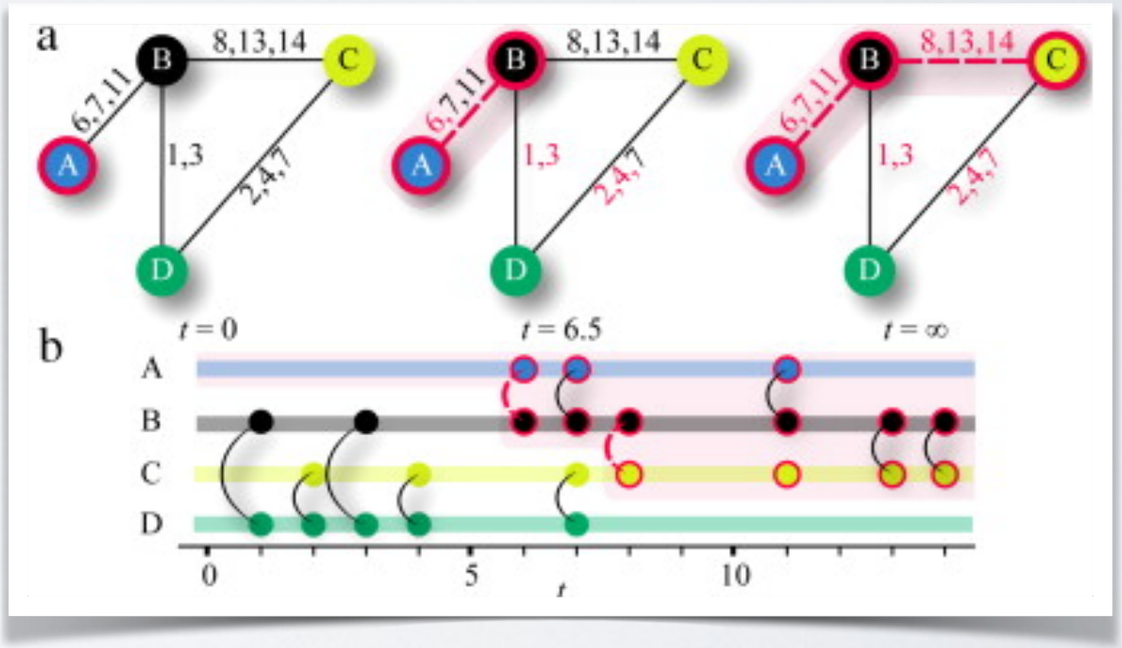
- Temporal links encode time varying interactions

$G=(V_{t'}, E_{t'})$

$v(t) \in V_{t'}$

$(u,v,t) \in E_{t'}$

- Dynamical nodes and links encode the evolution of the network



Mobile communication network
Nodes - individuals
Links - calls and SMS

NETWORK REPRESENTATIONS

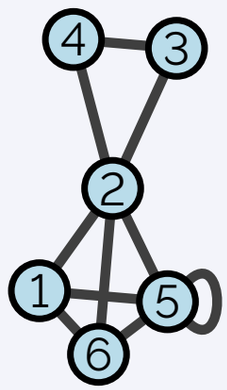
Networks: Graph notation

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

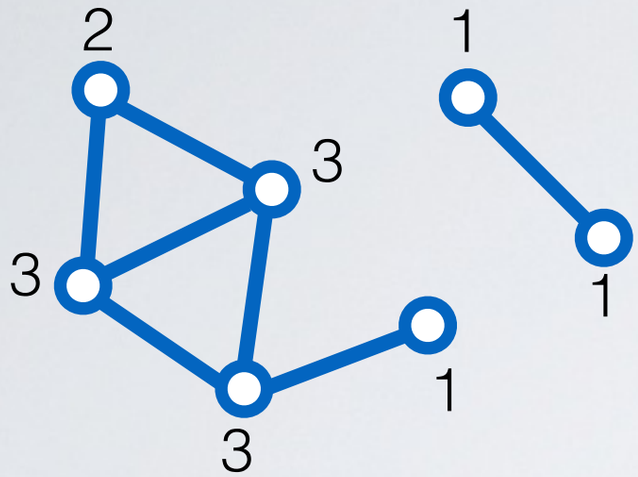
Node-Edge description

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, $s_u = \sum_v w_{uv}$.

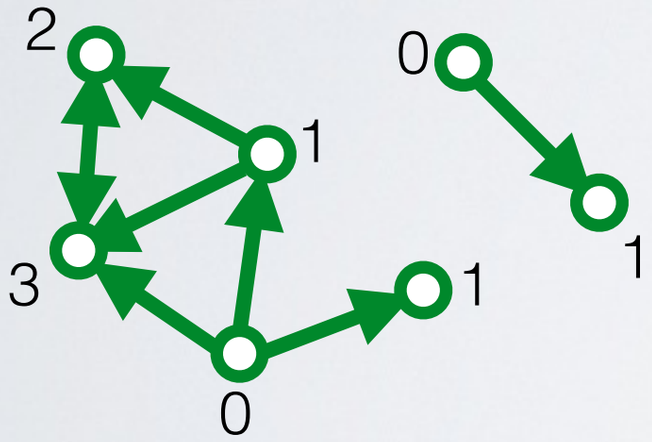
Node degree

Number of connections of a node

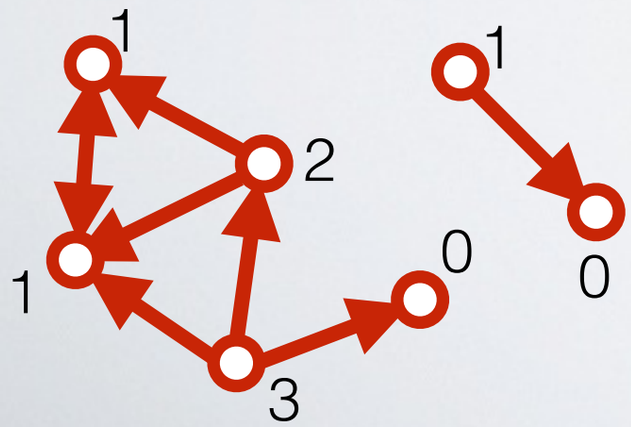
- Undirected network



- Directed network



In degree



Out degree

SIZE

Counting nodes and edges

N/n	**size**: number of nodes $\|V\|$.
L/m	number of edges $\|E\|$
L_{max}	Maximum number of links

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

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

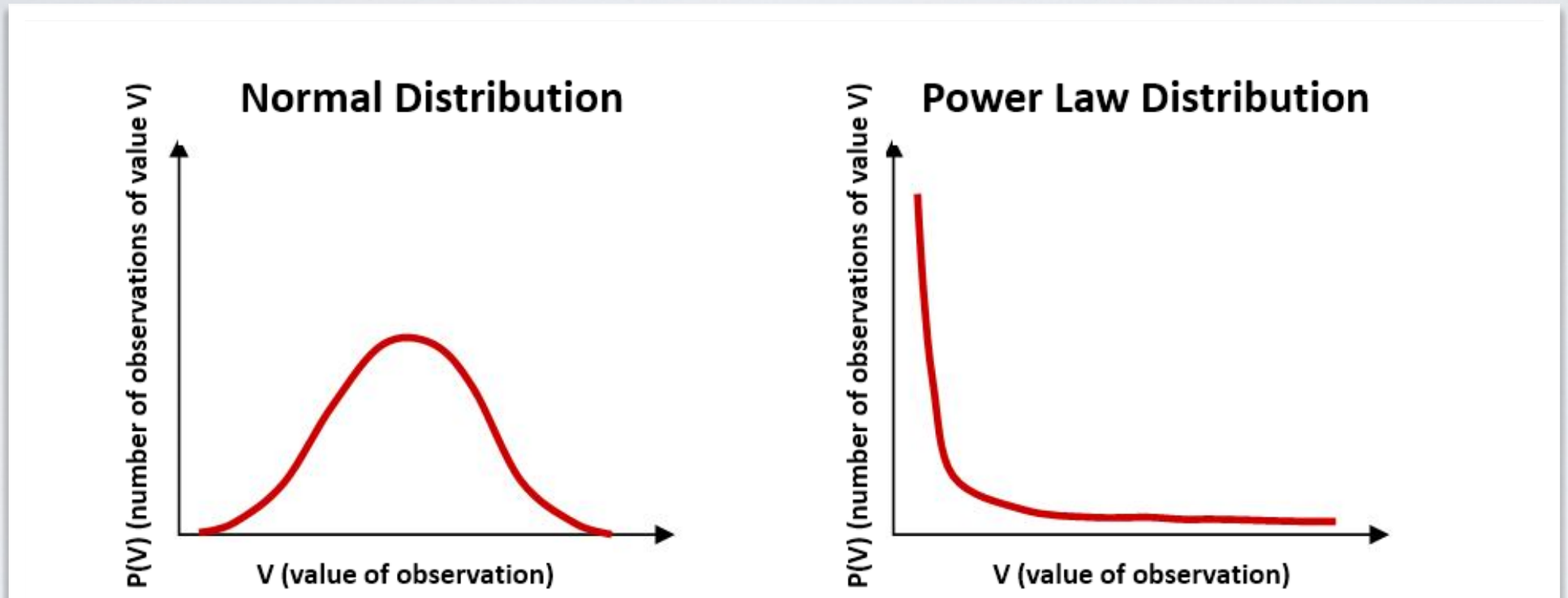
DENSITY

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	$1.5x10^{-5}$	30
Twitter 2015	288M	60B	$1.4x10^{-6}$	416
Facebook	1.4B	400B	$4x10^{-9}$	570
Brain c.	280	6393	0,16	46
Roads Calif.	2M	2.7M	$6x10^{-7}$	2,7
Airport	3k	31k	0,007	21

DEGREE DISTRIBUTION



PDF (Probability Distribution Function)

DEGREE DISTRIBUTION

- In a fully random graph (Erdos-Renyi), degree distribution is (close to) a normal distribution centered on the average degree

- In real graphs, in general, it is not the case:
  - A high majority of small degree nodes
  - A small minority of nodes with very high degree (Hubs)

- Often modeled by a **power law**
  - More details later in the course

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

SUBGRAPHS

Subgraphs

Subgraph $H(W)$ (induced subgraph): subset of nodes W of a graph $G = (V, E)$ and edges connecting them in G, i.e., subgraph $H(W) = (W, E'), W \subset V, (u, v) \in E' \iff u, v \in W \wedge (u, v) \in E$
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

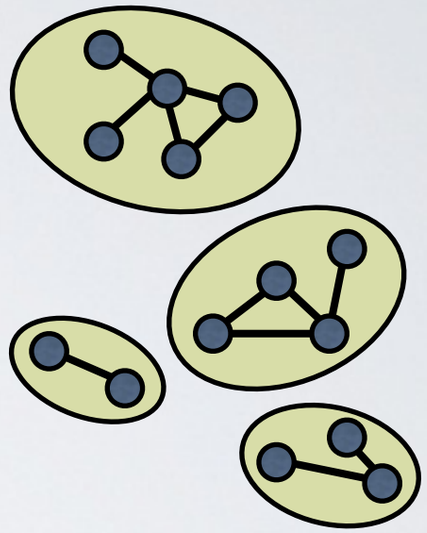
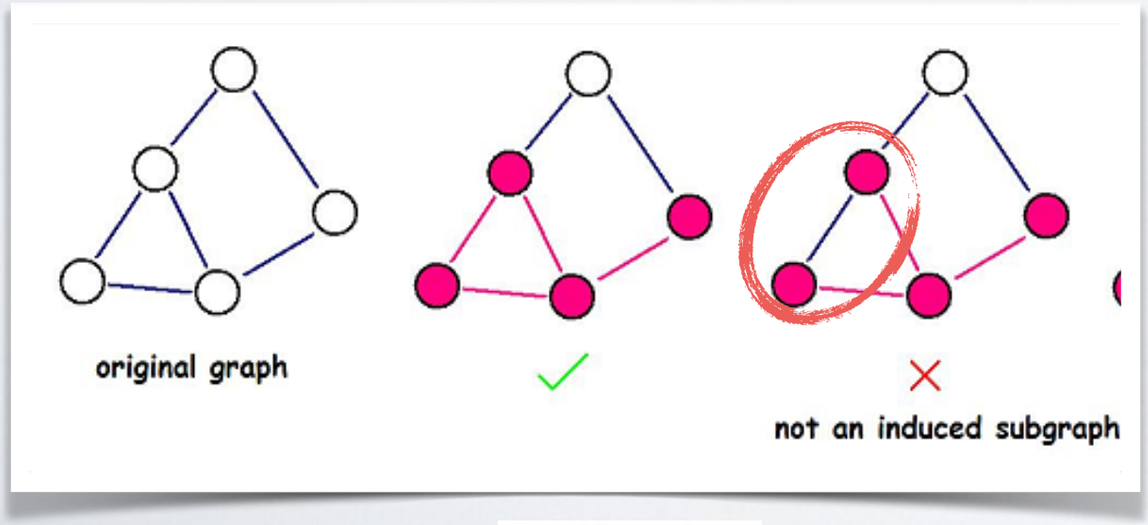


Figure after Newman, 2010

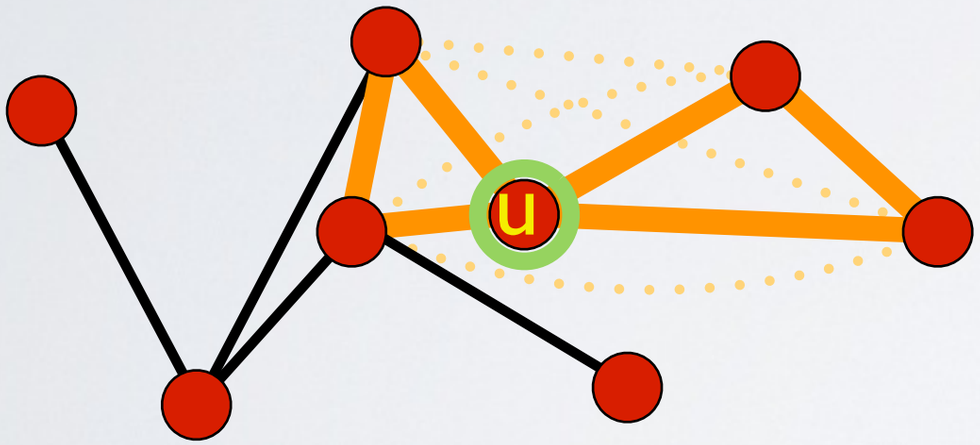


After “A. DZY Loves Physics”

Nodes/Edges
in the subgraph

CLUSTERING COEFFICIENT

C_u - **Node clustering coefficient:** density of the subgraph induced by the neighborhood of u, $C_u = d(H(N_u)$. 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*3/2=6

C_u=2/6=1/3

Triangles=2

Possible triangles= $\binom{4}{2}$ =6

C_u=2/6=1/3

CLUSTERING COEFFICIENT

$\langle C \rangle$ - **Average clustering coefficient:** Average clustering coefficient of all nodes in the graph, $\bar{C} = \frac{1}{N} \sum_{u \in V} C_u$.

Be careful when interpreting this value, since all nodes contributes equally, irrespectively of their degree, and that low degree nodes tend to be much more frequent than hubs, and their C value is very sensitive, i.e., for a node u of degree 2, $C_u \in 0, 1$, while nodes of higher degrees tend to have more contrasted scores.

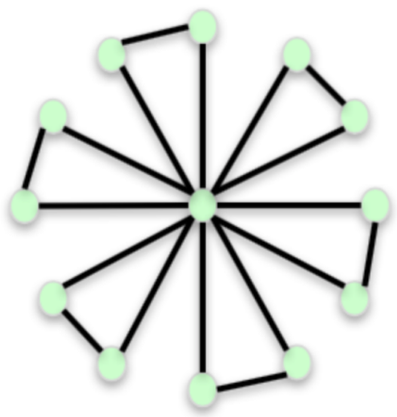
C^g - **Global clustering coefficient:** Fraction of all possible triangles in the graph that do exist, $C^g = \frac{3\Delta}{\Delta^{\max}}$

CLUSTERING COEFFICIENT

Global CC = Transitivity

Transitivity vs. Average Clustering Coefficient

Both measure the tendency for edges to form triangles.
Transitivity weights nodes with large degree higher.



- Most nodes have high LCC
- The high degree node has low LCC

Ave. clustering coeff. = 0.93
Transitivity = 0.23

- Most nodes have low LCC
- High degree node have high LCC

Ave. clustering coeff. = 0.25
Transitivity = 0.86

https://pynetwork.readthedocs.io/en/latest/connectivity.html

CLUSTERING COEFFICIENT

- Global CC:
  - In random networks, GCC = density
    - =>very small for large graphs

Network	Size	$\langle k \rangle$	C	C_{rand}	Reference
WWW, site level, undir.	153 127	35.21	0.1078	0.00023	Adamic, 1999
Internet, domain level	3015–6209	3.52–4.11	0.18–0.3	0.001	Yook *et al.*, 2001a, Pastor-Satorras *et al.*, 2001
Movie actors	225 226	61	0.79	0.00027	Watts and Strogatz, 1998
LANL co-authorship	52 909	9.7	0.43	1.8×10^{-4}	Newman, 2001a, 2001b, 2001c
MEDLINE co-authorship	1 520 251	18.1	0.066	1.1×10^{-5}	Newman, 2001a, 2001b, 2001c
SPIRES co-authorship	56 627	173	0.726	0.003	Newman, 2001a, 2001b, 2001c
NCSTRL co-authorship	11 994	3.59	0.496	3×10^{-4}	Newman, 2001a, 2001b, 2001c
Math. co-authorship	70 975	3.9	0.59	5.4×10^{-5}	Barabási *et al.*, 2001
Neurosci. co-authorship	209 293	11.5	0.76	5.5×10^{-5}	Barabási *et al.*, 2001
E. coli, substrate graph	282	7.35	0.32	0.026	Wagner and Fell, 2000
E. coli, reaction graph	315	28.3	0.59	0.09	Wagner and Fell, 2000
Ythan estuary food web	134	8.7	0.22	0.06	Montoya and Solé, 2000
Silwood Park food web	154	4.75	0.15	0.03	Montoya and Solé, 2000
Words, co-occurrence	460.902	70.13	0.437	0.0001	Ferrer i Cancho and Solé, 2001
Words, synonyms	22 311	13.48	0.7	0.0006	Yook *et al.*, 2001b
Power grid	4941	2.67	0.08	0.005	Watts and Strogatz, 1998
C. Elegans	282	14	0.28	0.05	Watts and Strogatz, 1998

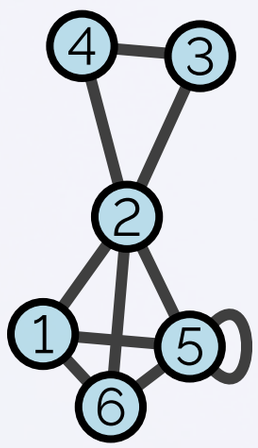
Albert, R. et.al. Rev. Mod. Phy. (2002)

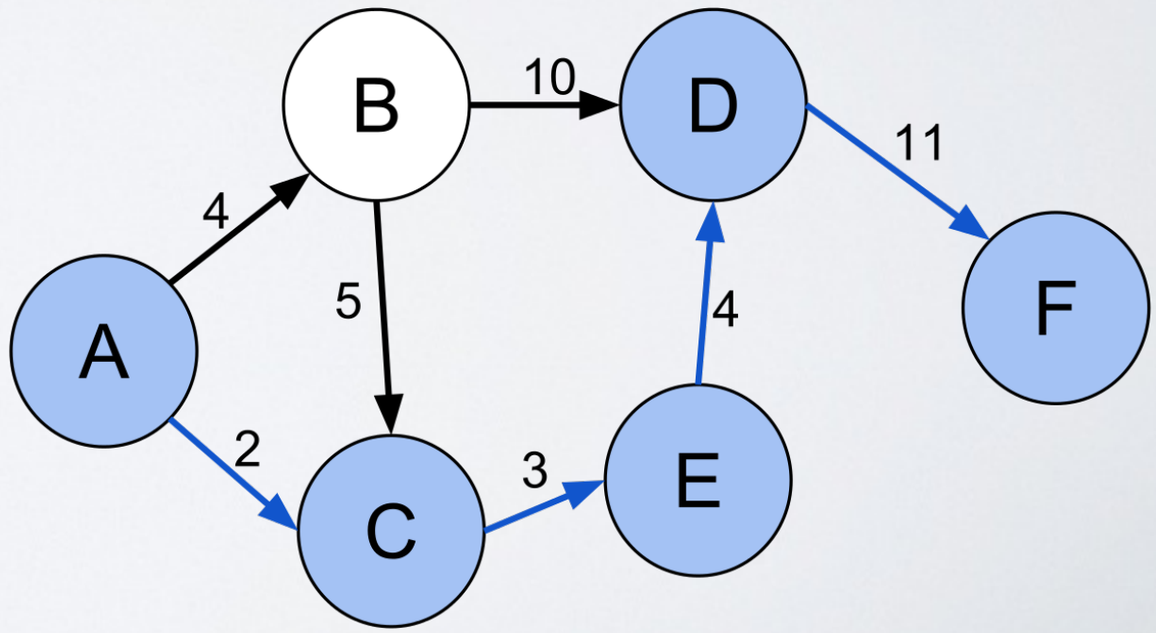
PATH RELATED SCORES

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

Graph





PATH RELATED SCORES

Network descriptors 2 - Paths

$\ell_{\max}$

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

$\langle \ell \rangle$

Average distance:

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

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

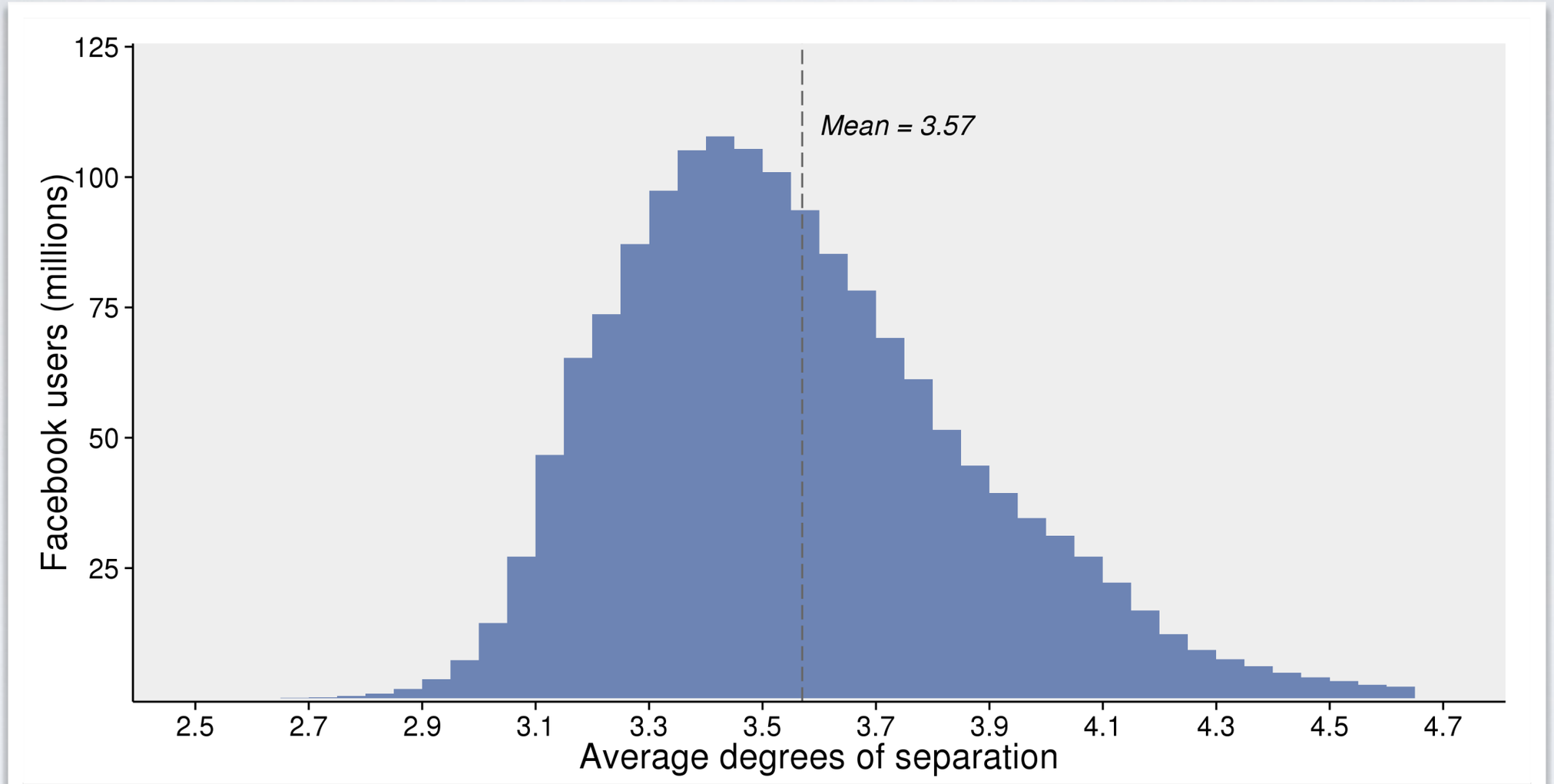
- Small world experiment (60's)
  - Give a (physical) mail to random people
  - Ask them to send to someone they don't know
    - They know his city, job
  - They send to their most relevant contact

- 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

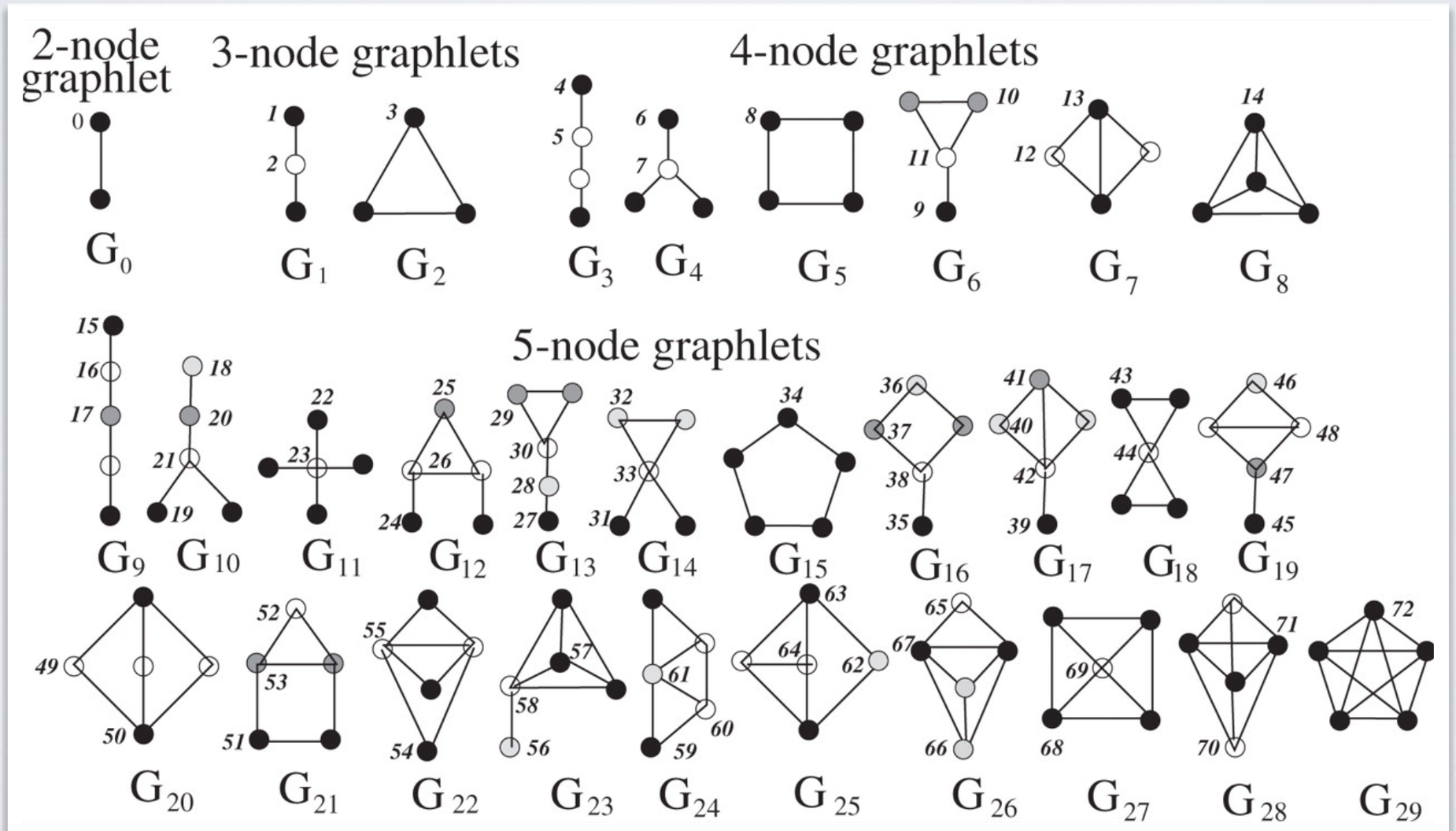
SMALL WORLD

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

GRAPHLETS


2-node graphlet
3-node graphlets
4-node graphlets
5-node graphlets
G0
G1
G2
G3
G4
G5
G6
G7
G8
G9
G10
G11
G12
G13
G14
G15
G16
G17
G18
G19
G20
G21
G22
G23
G24
G25
G26
G27
G28
G29

ADJACENCY MATRIX

Typical operations on A

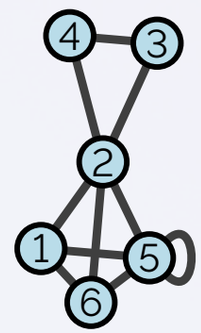
Some operations on Adjacency matrices have straightforward interpretations and are frequently used

---

Multiplying A by **itself** allows to know the number of walks of a given length that exist between any pair of nodes: A^2_{ij} corresponds to the number of walks of length 2 from node i to node j, A^3_{ij} to the number of walks of length 3, etc.

Multiplying A by a **column vector** W of length $1 \times N$ can be thought as setting the i th value of the vector to the ith node, and each node *sending* its value to its neighbors (for undirected graphs). The result is a column vector with N elements, the ith element corresponding to the sum of the values of its neighbors in W. This is convenient when working with **random walks** or **diffusion** phenomenon.

Graph



A - Adjacency Mat.

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

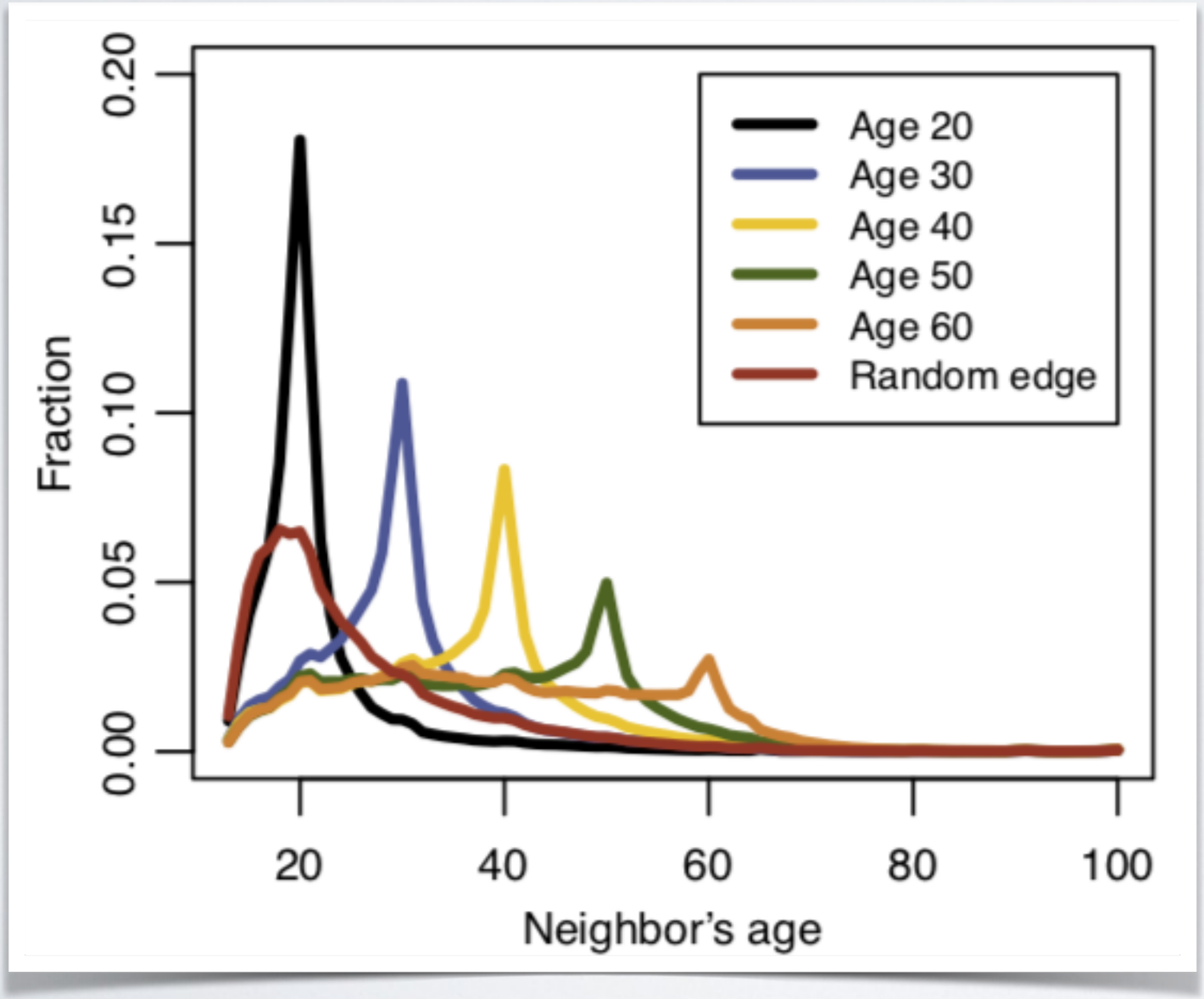
A^2

$$\begin{pmatrix} 3 & 2 & 1 & 1 & 3 & 2 \\ 2 & 5 & 1 & 1 & 3 & 2 \\ 1 & 1 & 2 & 1 & 1 & 1 \\ 1 & 1 & 1 & 2 & 1 & 1 \\ 3 & 3 & 1 & 1 & 4 & 3 \\ 2 & 2 & 1 & 1 & 3 & 3 \end{pmatrix}$$

EXEMPLE OF GRAPH ANALYSIS

- 721M users (nodes) (active in the last 28 days)
- 68B edges
- Average degree: 190 (average # friends)
- Median degree: 99
- Connected component: 99.91%

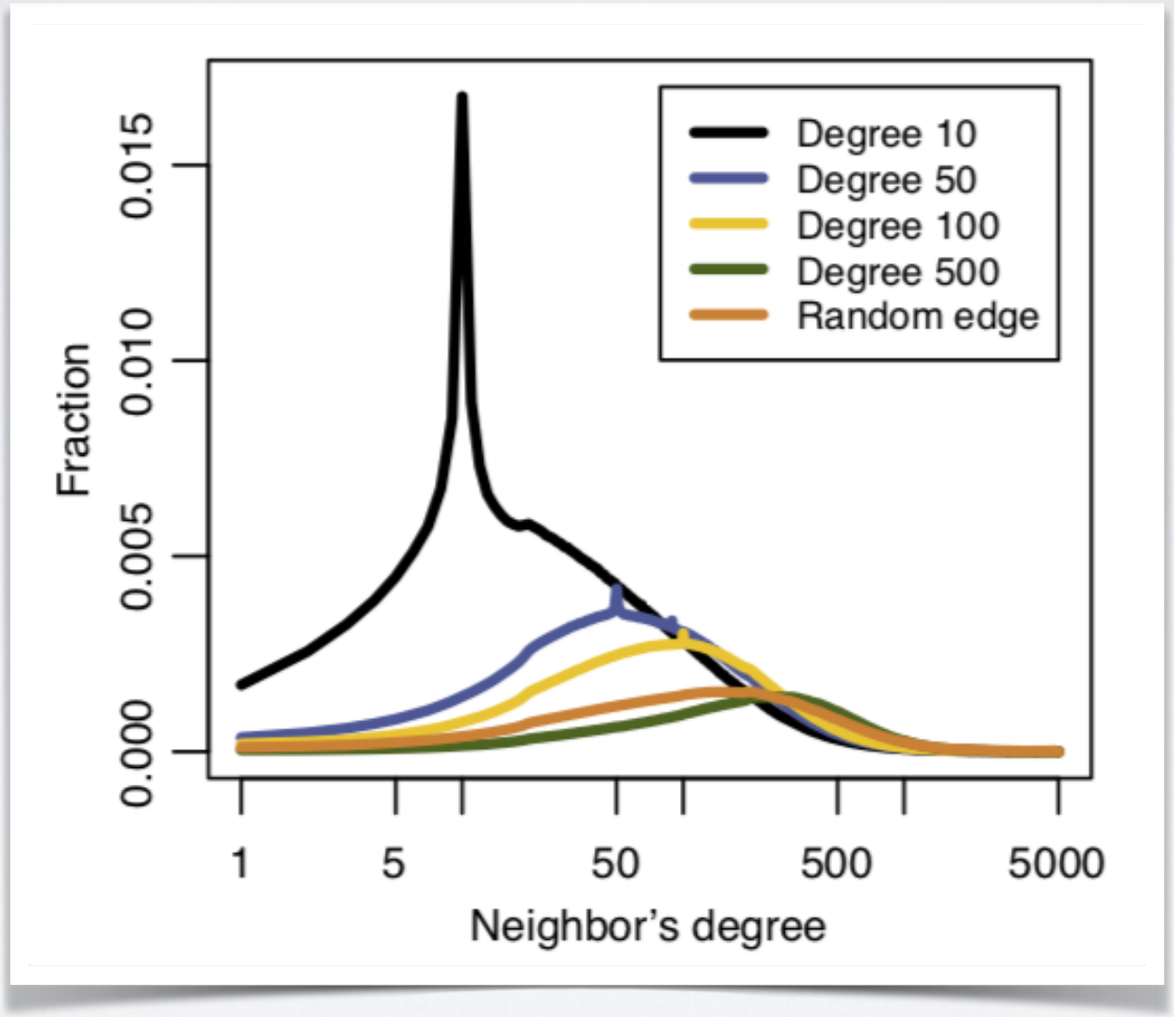
EXEMPLE OF GRAPH ANALYSIS



Age homophily

(More next class)

EXEMPLE OF GRAPH ANALYSIS



Many of my friends have the Same # of friends than me!

CENTRALITIES

Characterizing/Discovering important nodes

CENTRALITY

- We can measure nodes importance using so-called **centrality**.

- Poor terminology: nothing to do with being central in general

- Usage:
  - Some centralities have straightforward interpretation
  - Centralities can be used as *node features* for machine learning on graph
    - (Classification, link prediction, …)

NODE DEGREE

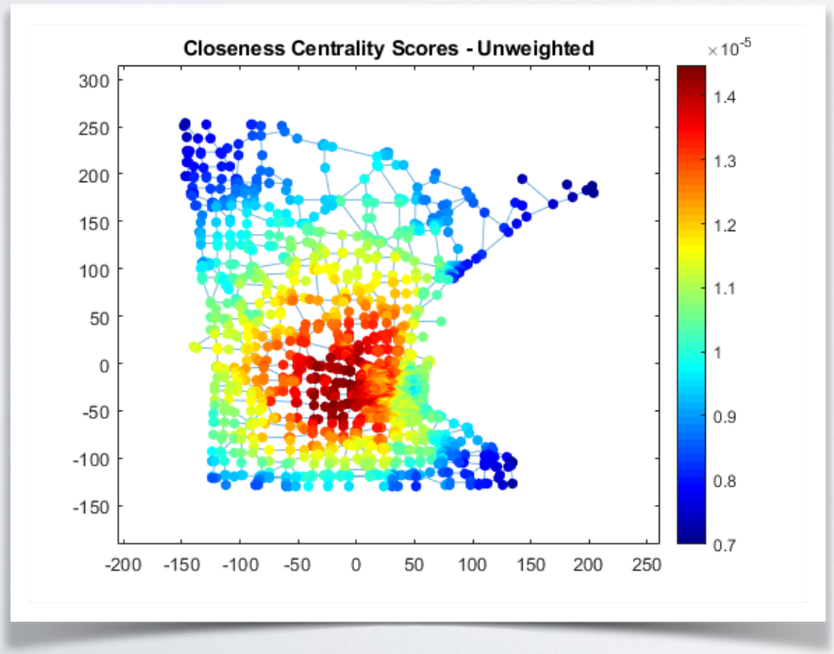
- **Degree**: how many neighbors
- Often enough to find important nodes
  - Main characters of a series talk with the more people
  - Largest airports have the most connections
  - …
- But not always
  - Facebook users with the most friends are spam
  - Webpages/wikipedia pages with most links are simple lists of references
  - …

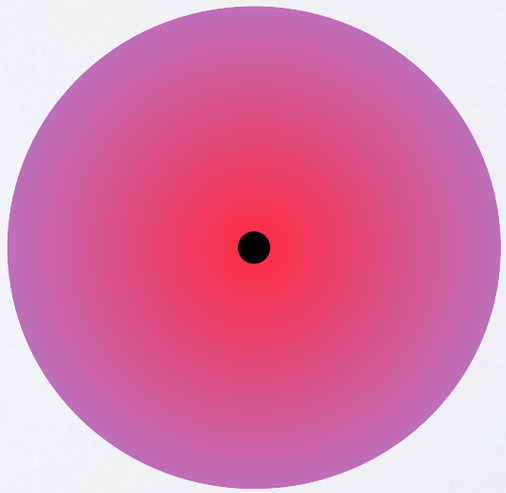
FARNESS, CLOSENESS

HARMONIC CENTRALITY

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





FARNESS, CLOSENESS

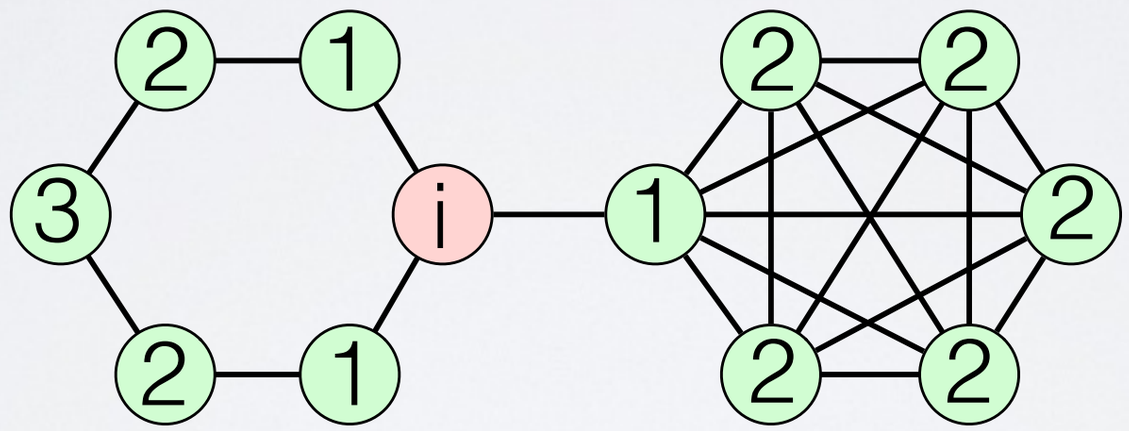
Farness: Average distance to all other nodes in the graph

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

CLOSENESS CENTRALITY

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

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



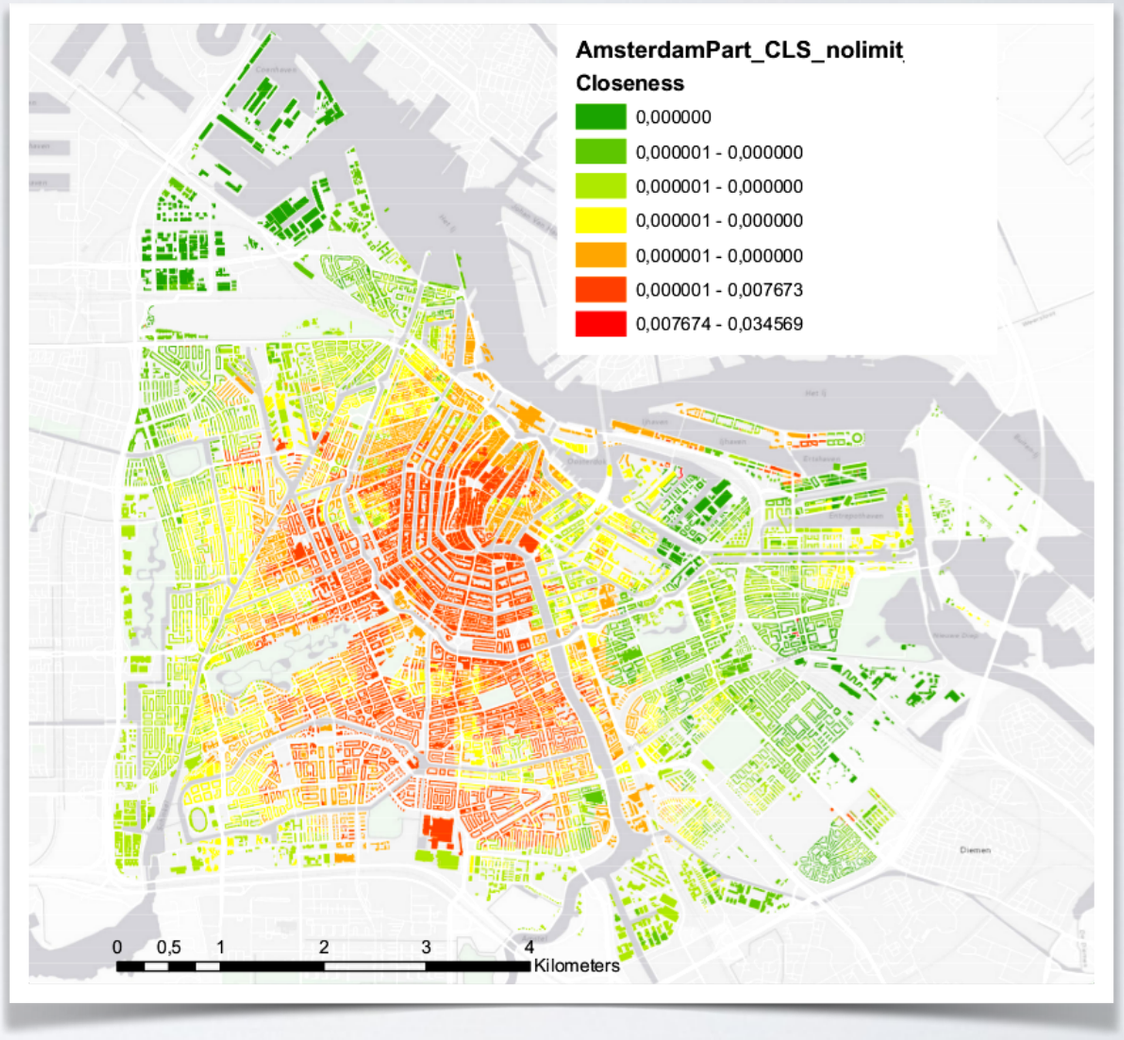
$$C_{cl}(i) = \frac{12-1}{(3\times 1 + 7\times 2 + 1\times 3)} = \frac{11}{20} = 0.55$$

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: f*raction of all the shortest paths between all the pairs of nodes going through u.

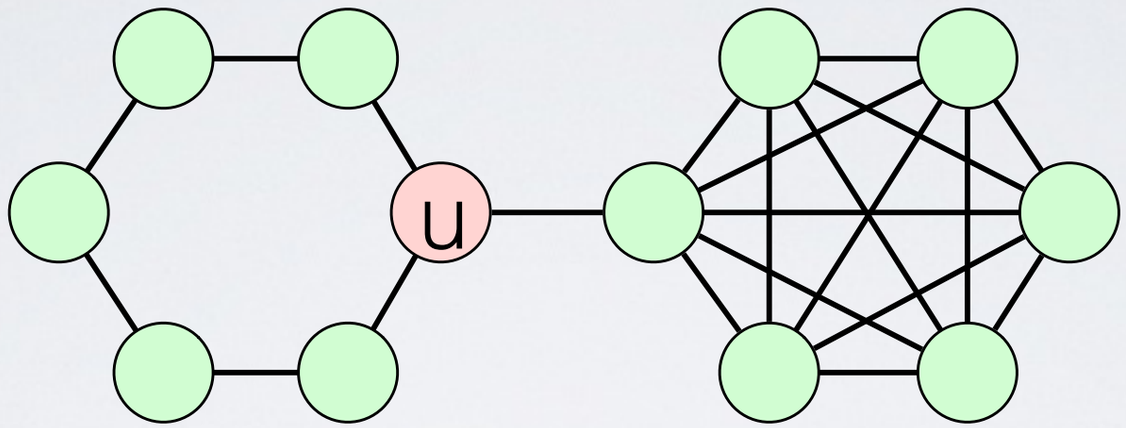
$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

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



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

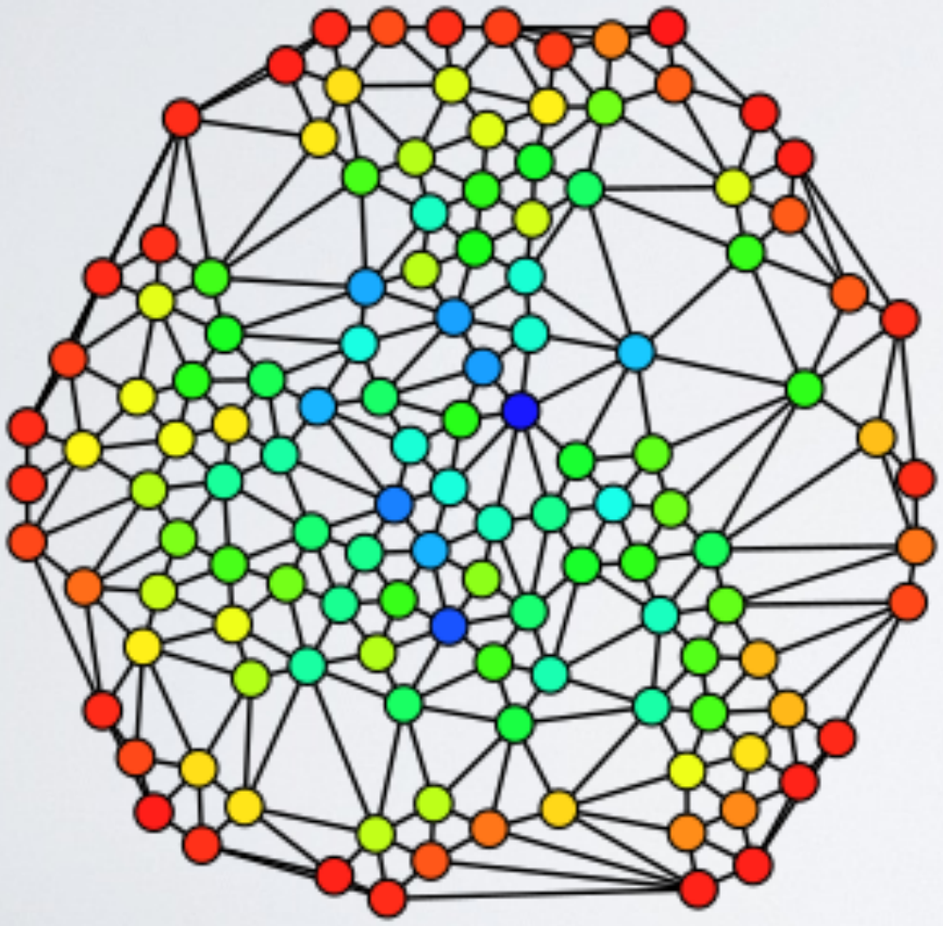
Exact computation:

Floyd-Warshall: *O(n³) time complexity*
O(n²) space complexity

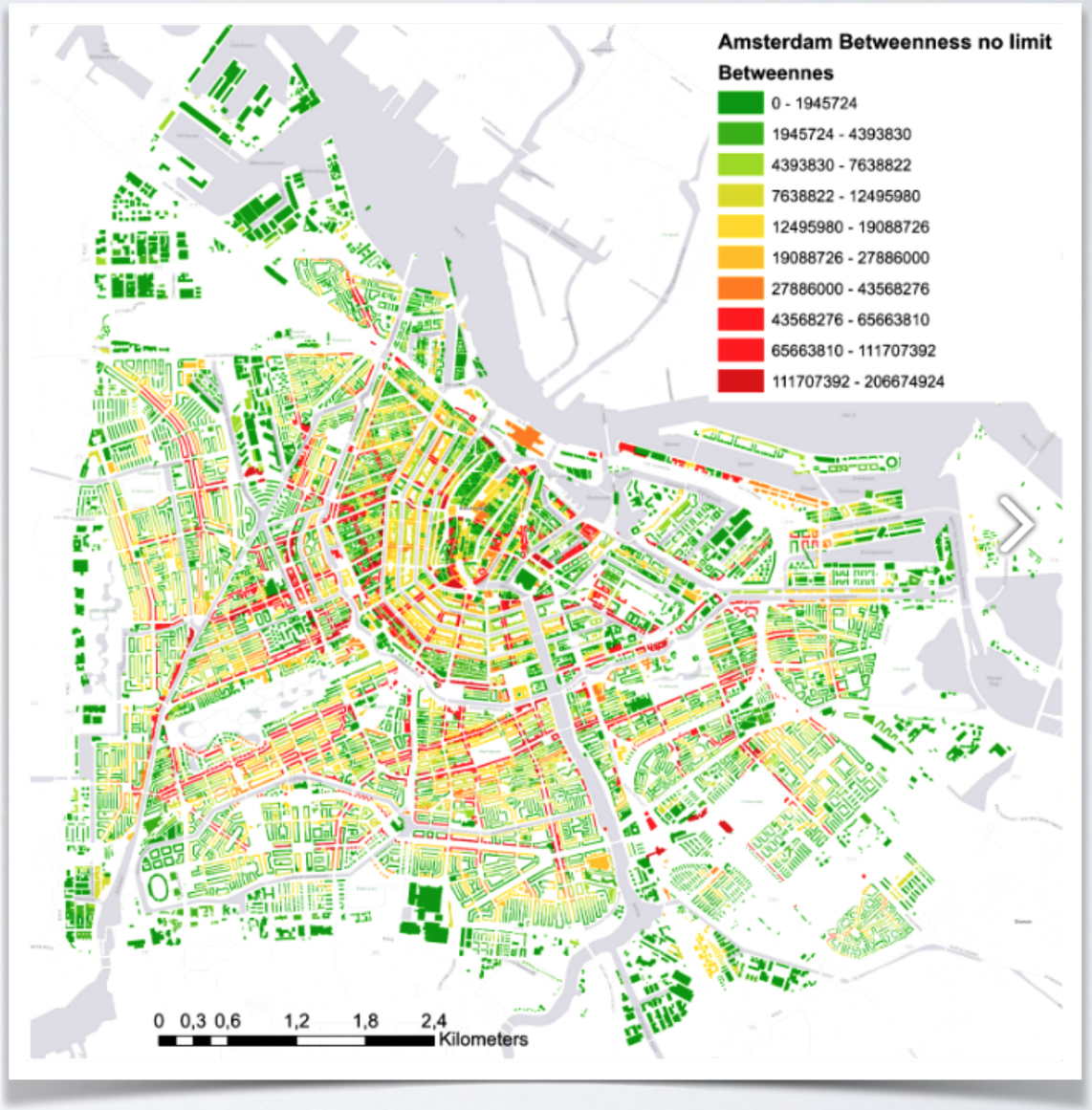
Approximate computation

Dijskstra: *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

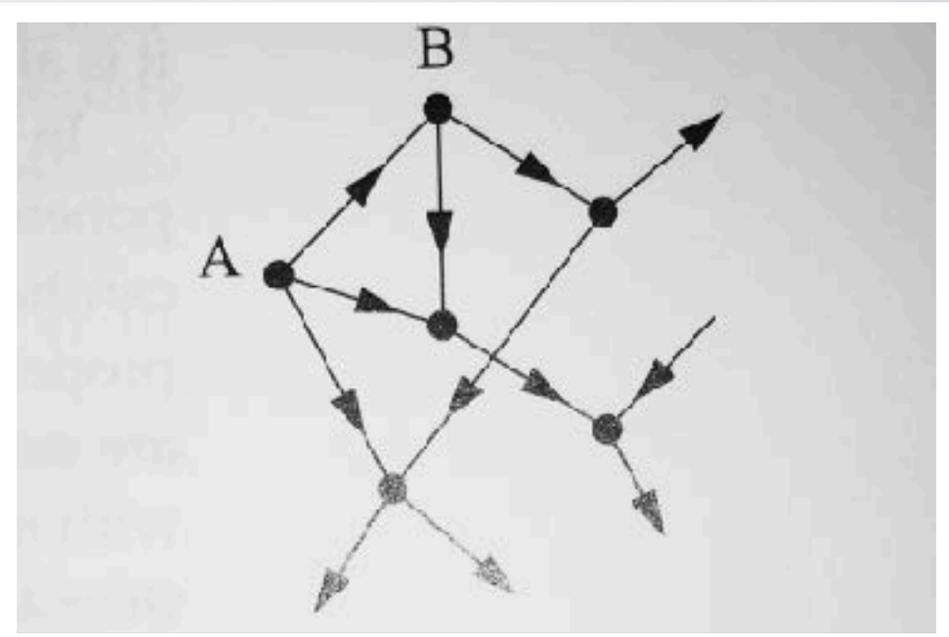
EIGENVECTOR CENTRALITY

- 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 Centrality

(Side notes)

-"We chose our system name, Google, because it
is a common spelling of googol, or 10^{100} and fits well with our goal of building very large-scale search "

-"[…] at the same time, search engines have migrated from the academic domain to the commercial. **Up until now 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 and to be advertising oriented (see Appendix A). With Google, we have a strong goal to push more development and understanding into the academic realm."**

-"[...], we expect that advertising funded search engines will be inherently biased towards the advertisers and away from the needs of the consumers."

PageRank Centrality

(Side notes)

Sergey Brin received his B.S. degree in mathematics and computer science from the University of Maryland at College Park in 1993. Currently, he is a Ph.D. candidate in computer science at Stanford University where he received his M.S. in 1995. He is a recipient of a National Science Foundation Graduate Fellowship. His research interests include search engines, information extraction from unstructured sources, and data mining of large text collections and scientific data.

Lawrence Page was born in East Lansing, Michigan, and received a B.S.E. in Computer Engineering at the University of Michigan Ann Arbor in 1995. He is currently a Ph.D. candidate in Computer Science at Stanford University. Some of his research interests include the link structure of the web, human computer interaction, search engines, scalability of information access interfaces, and personal data mining.

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)

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

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

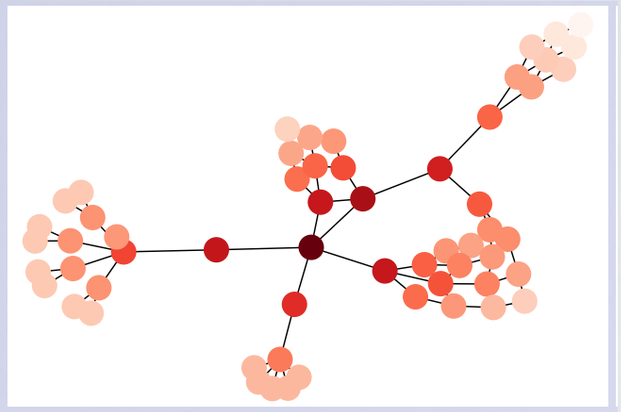
PAGERANK

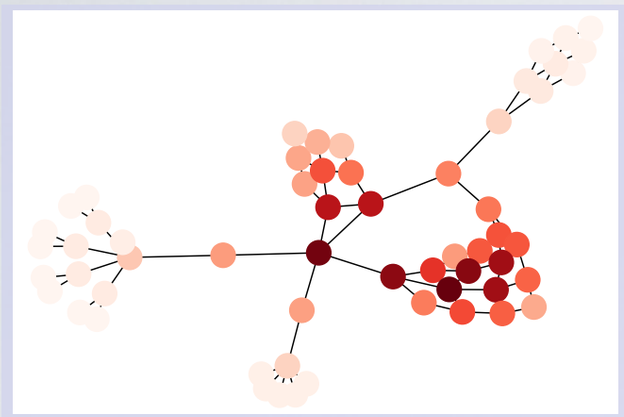
- Then how do Google rank when we do a research?

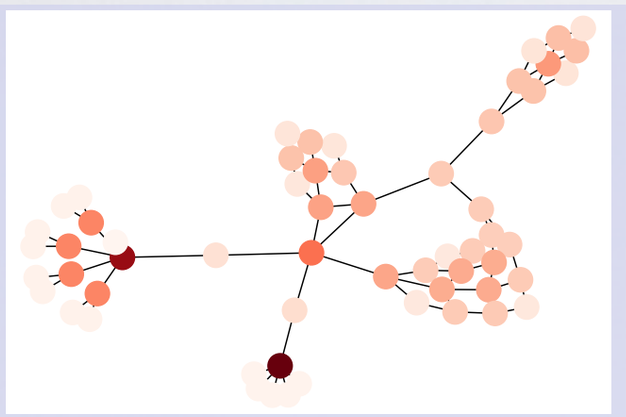
- Compute Pagerank (using the power method for scalability)

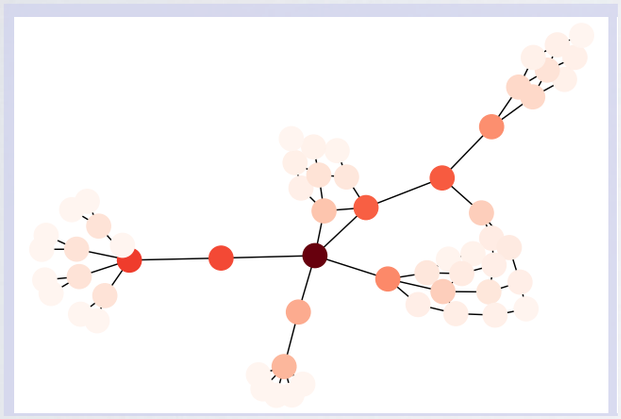
- Create a subgraph of documents related to our topic

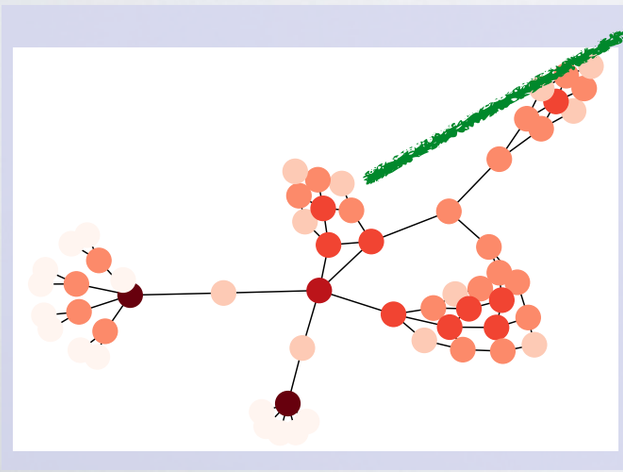
- Of course now it is certainly much more complex, but we don't really know:
  "Most search engine development has gone on at companies with little publication of technical details. This causes search engine technology to remain largely a black art" [Page, Brin, 1997]

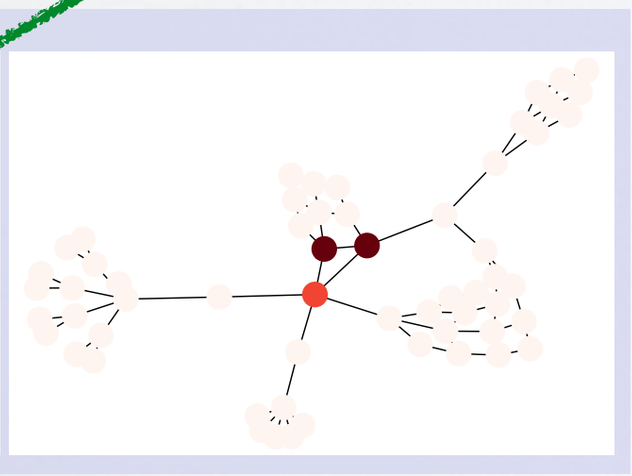






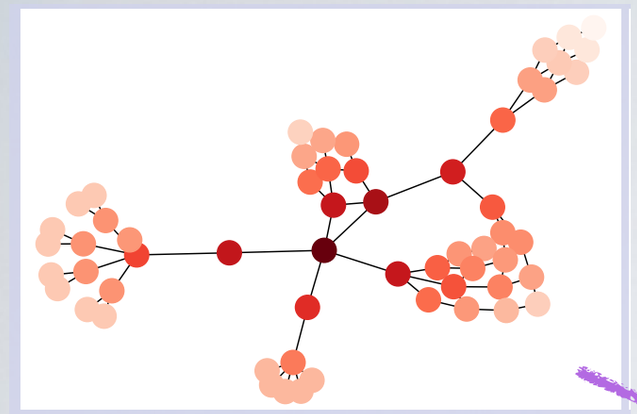


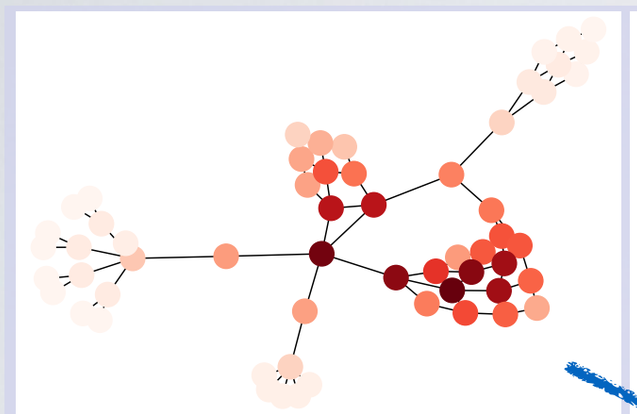


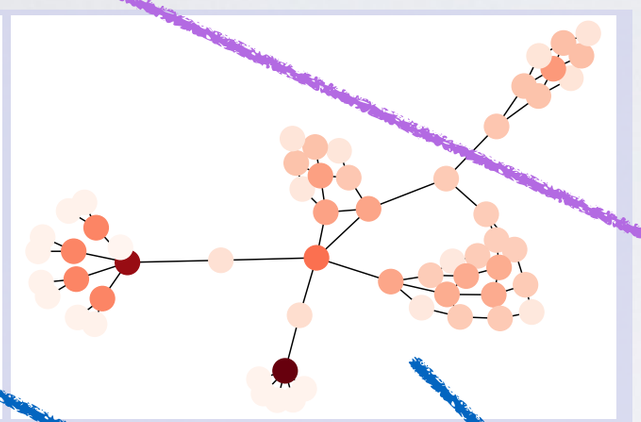


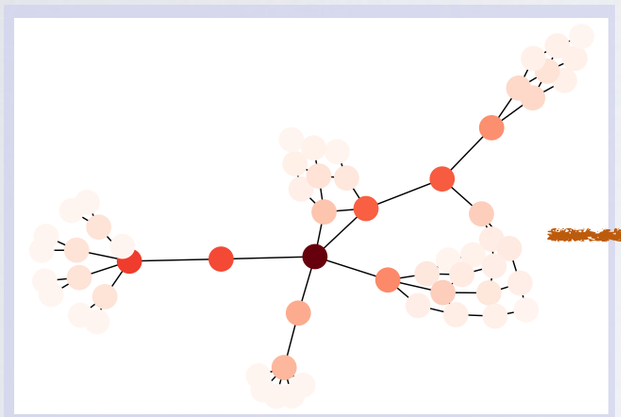
Which is which ?

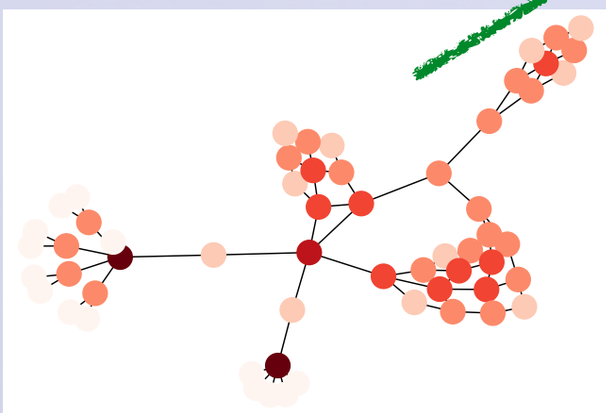
Degree
Clustering coefficient
Closeness
Betweenness
Eigenvector
PageRank

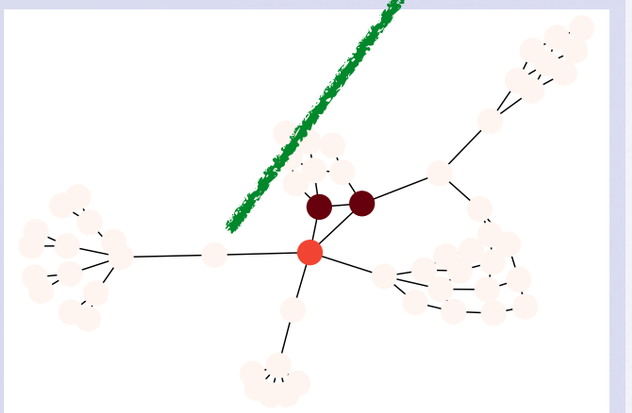












Which is which ?

Degree

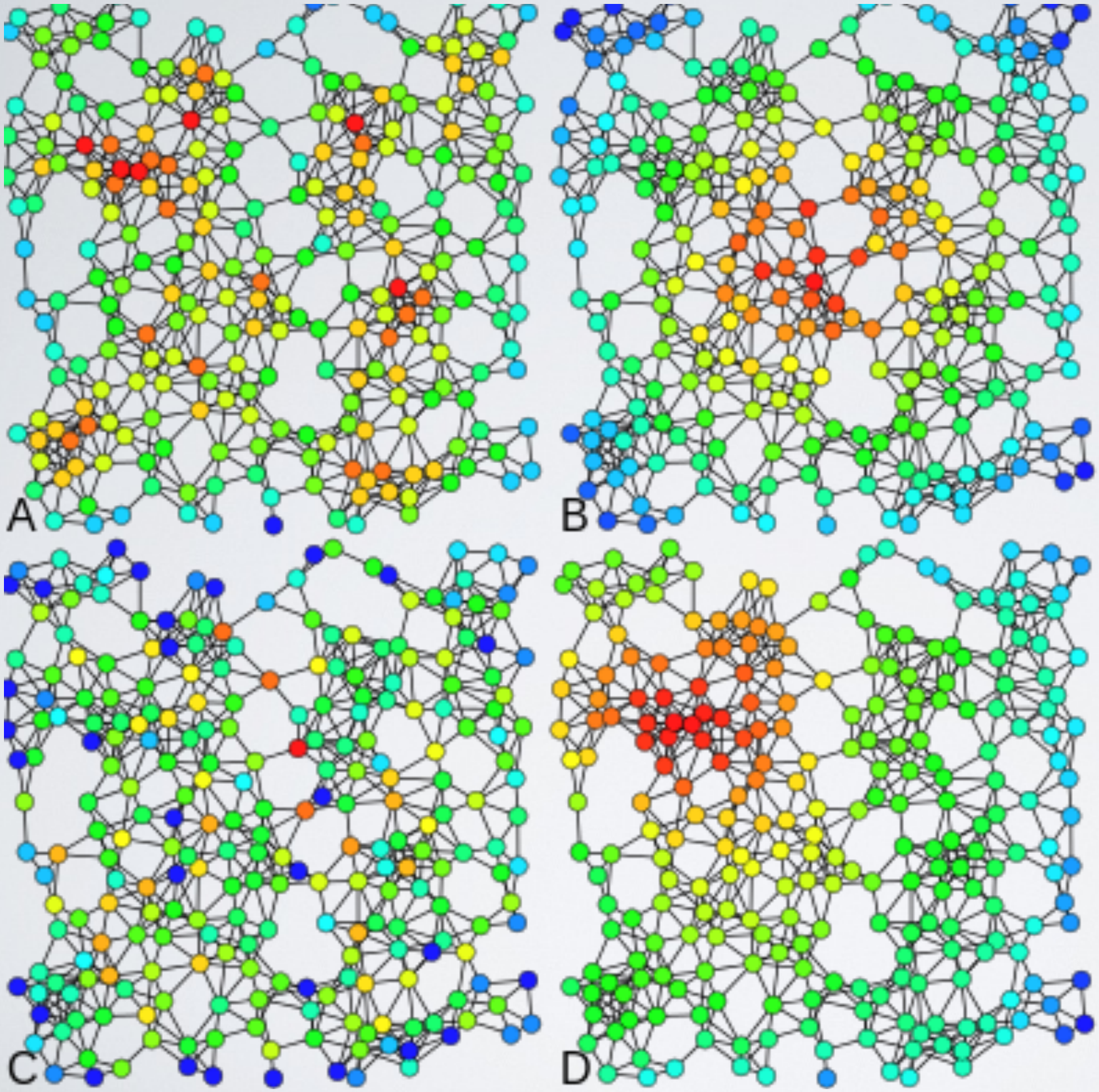
Clustering coefficient

Closeness

Betweenness

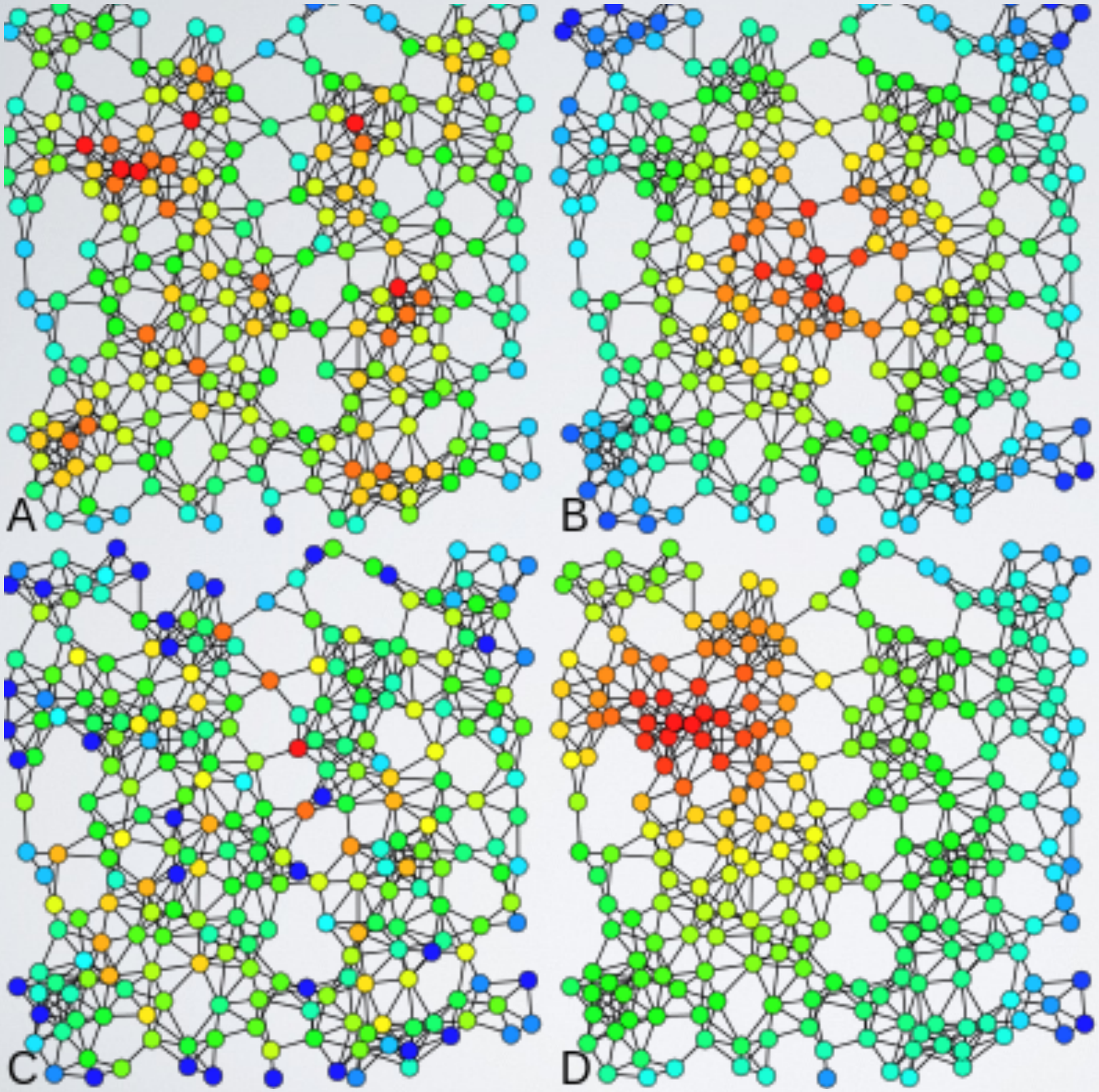
Eigenvector

PageRank



Try again :)

Degree
Betweenness
Closeness
Eigenvector



Try again :)

A: Degree
B:Closeness
C: Betweenness
D: Eigenvector

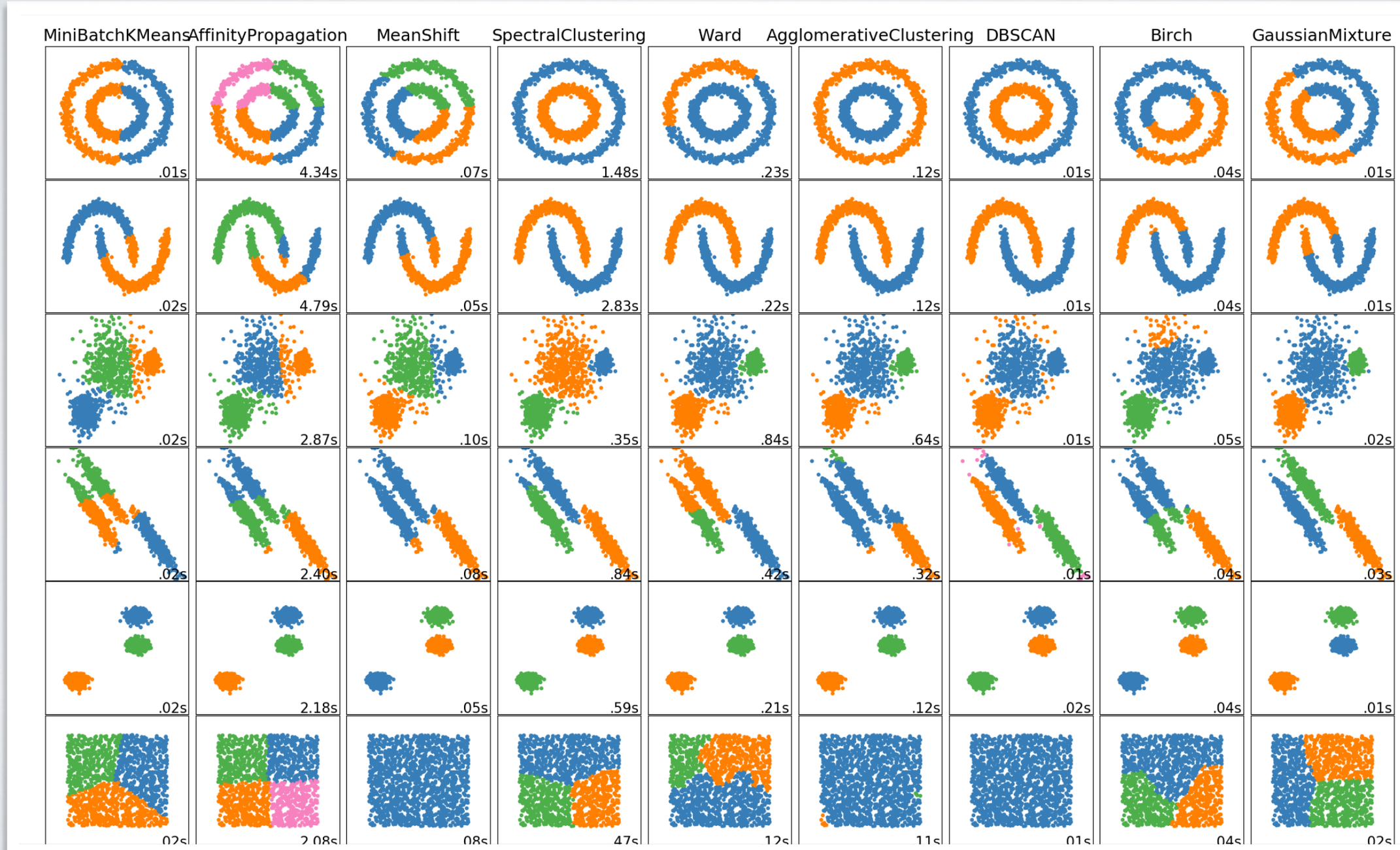
COMMUNITY DETECTION (GRAPH CLUSTERING)

COMMUNITY DETECTION

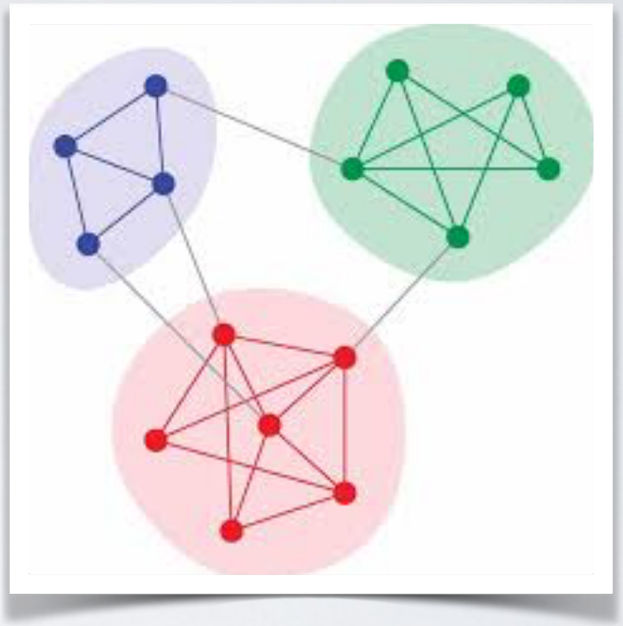
- Community detection is equivalent to "clustering" in unstructured data

- 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



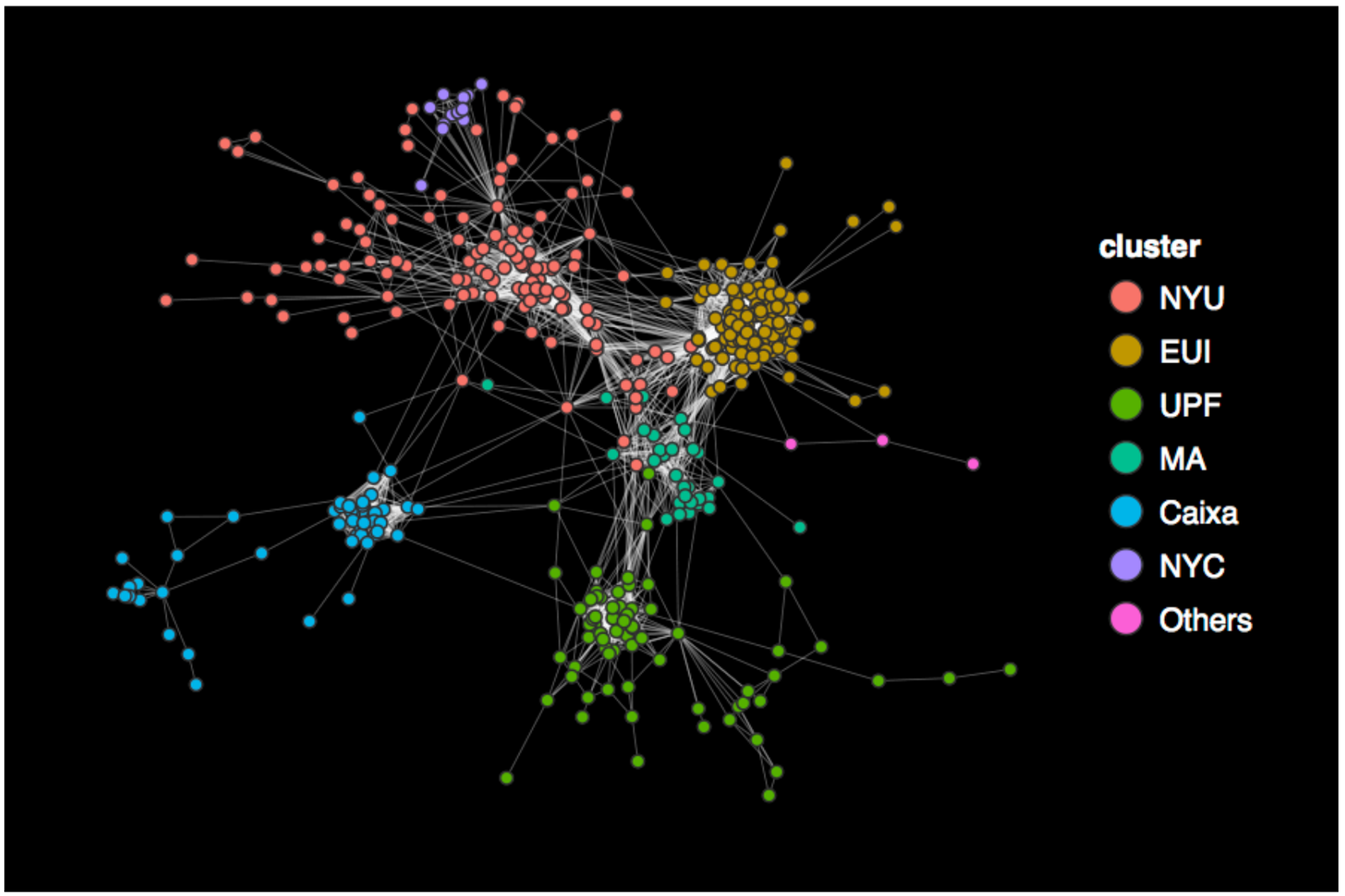
COMMUNITY DETECTION



- 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

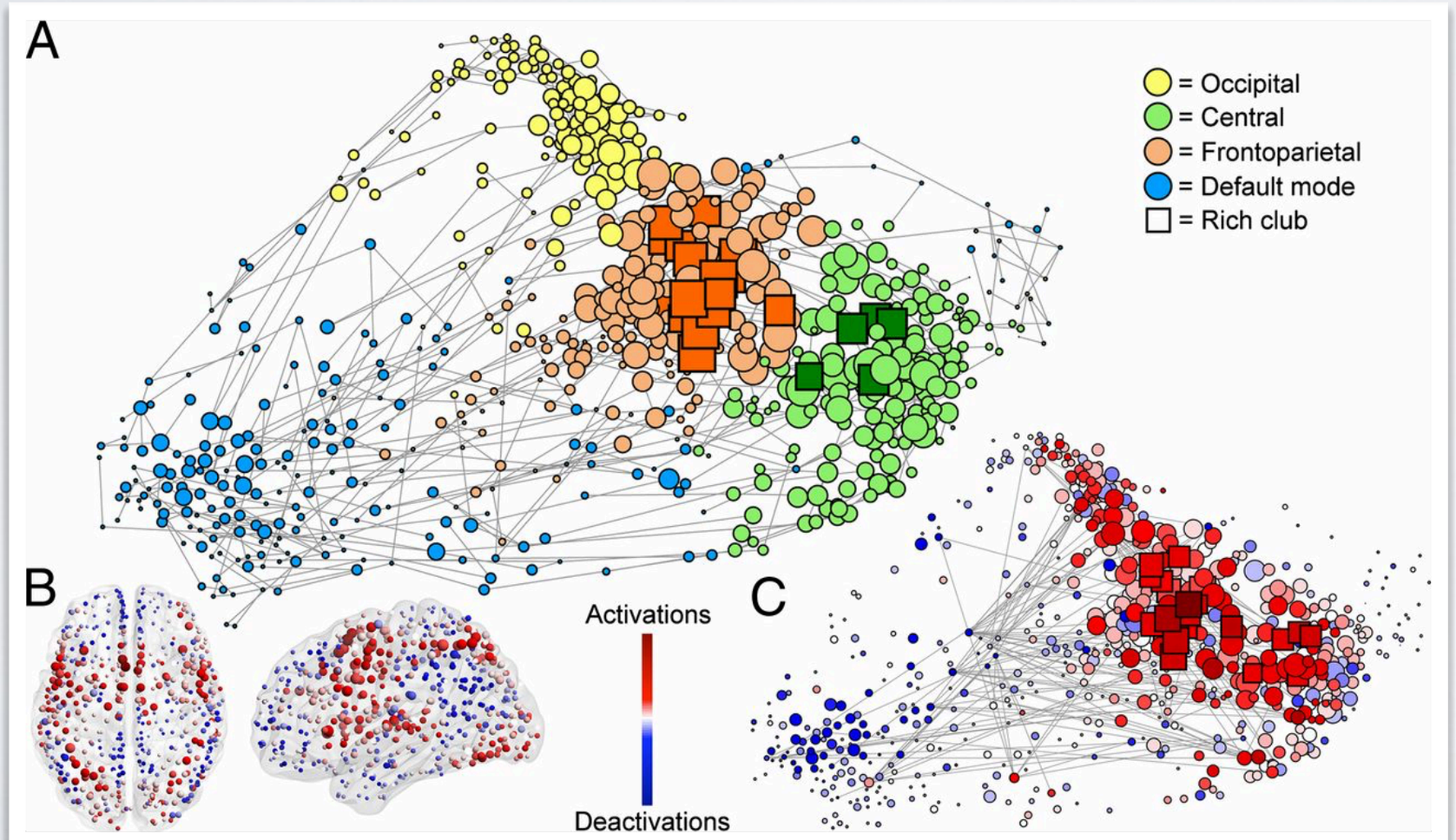
COMMUNITY STRUCTURE IN REAL GRAPHS

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



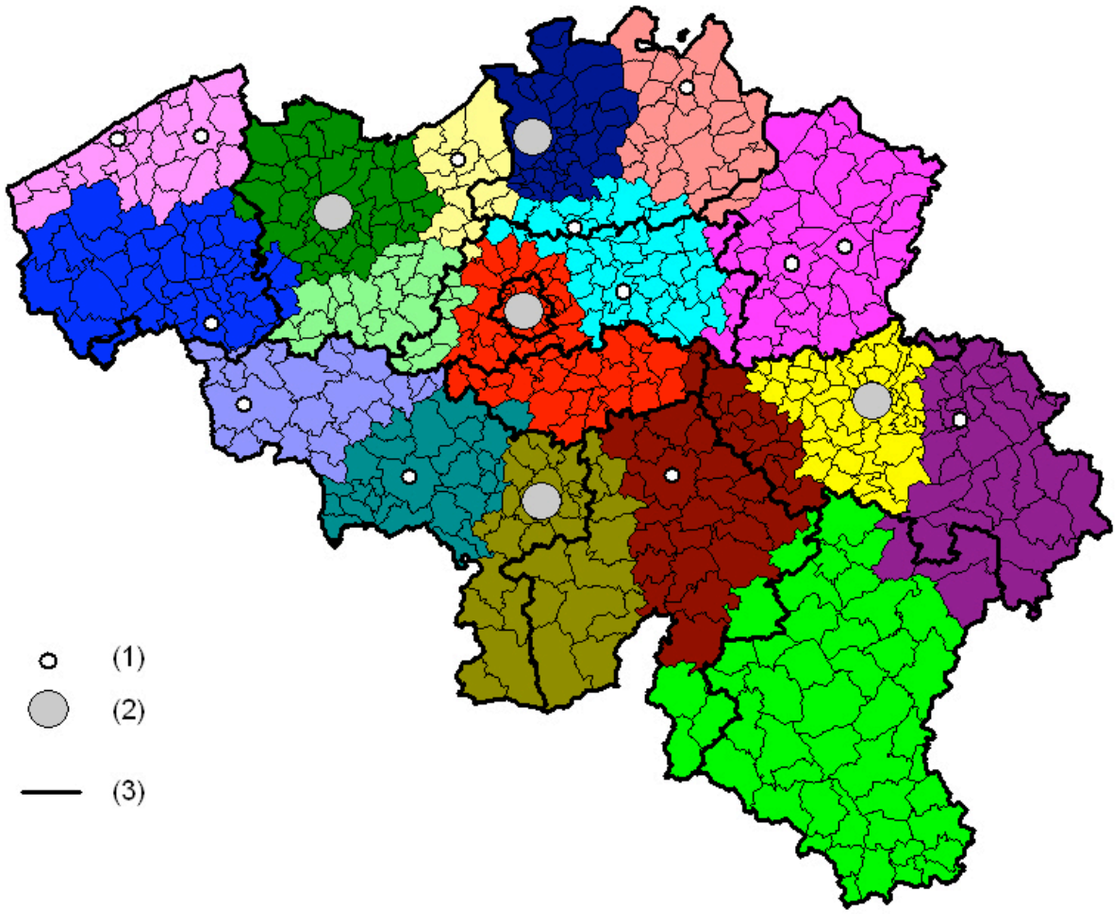
COMMUNITY STRUCTURE IN REAL GRAPHS

- Connections in the brain ?



COMMUNITY STRUCTURE IN REAL GRAPHS

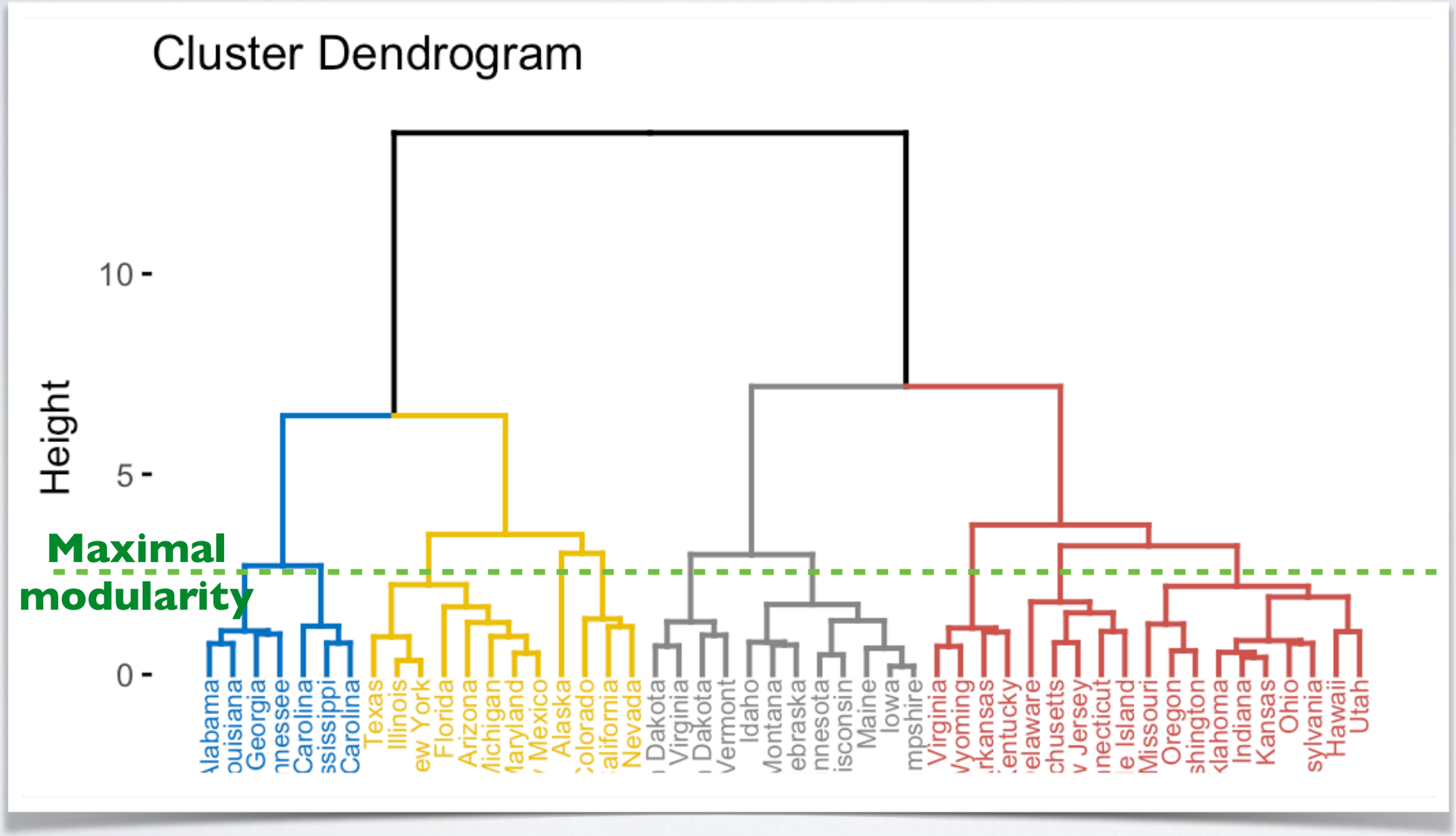
- Phone call communications in Belgium ?



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**

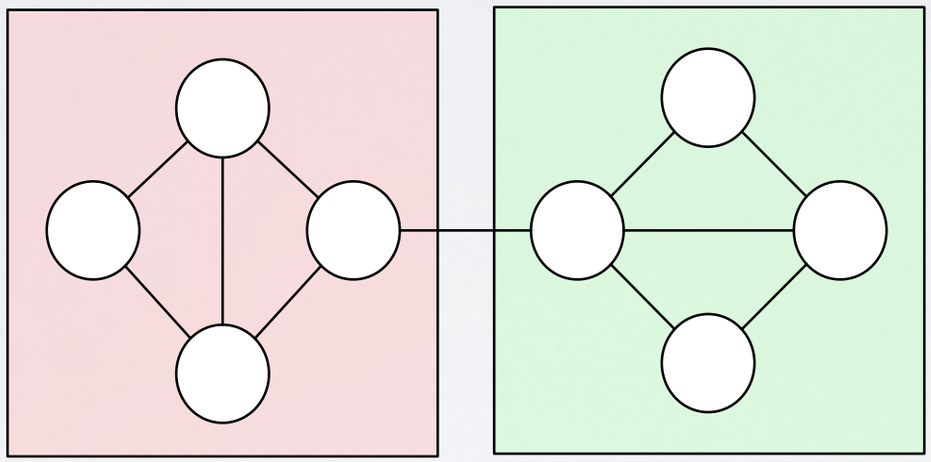
- The modularity is computed for a partition of a graph
  - (each node belongs to one and only one community)

- It compares :
  - The **observed** *fraction of edges inside communities*
  - To the **expected** *fraction of edges inside communities* in a random network

MODULARITY INTUITION

$n = 8$

$m = 11$



ER random graph

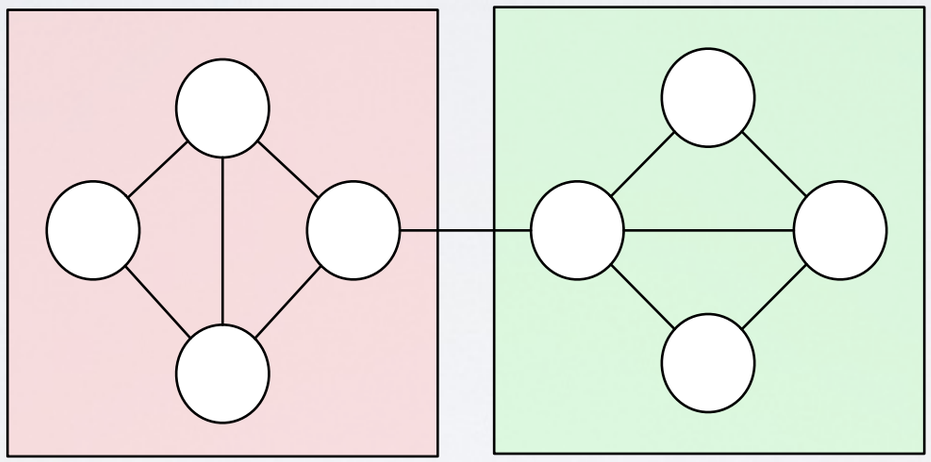
MODULARITY INTUITION

$n = 8$

$m = 11$

$p(u, v) \approx 0.39$

$$d(G) = p(u, v) = \frac{11}{\frac{1}{2}8(8-1)} = \frac{11}{28} \approx 0.39$$

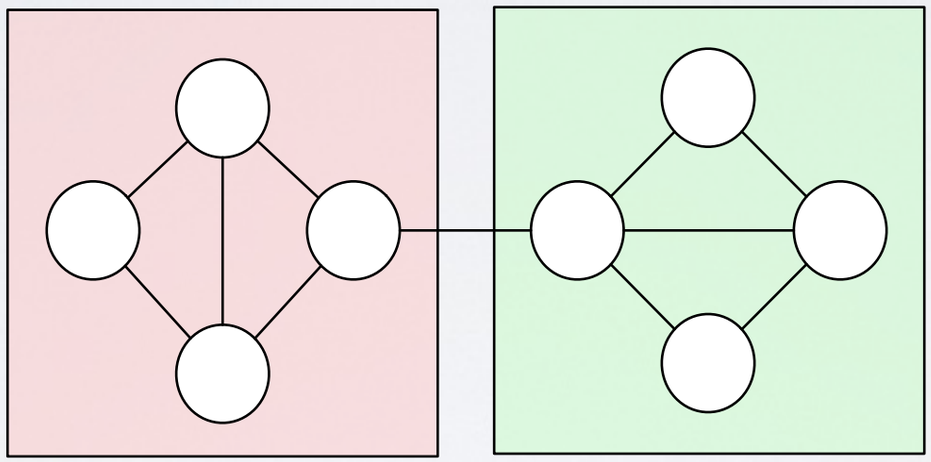


MODULARITY INTUITION

$n = 8$

$m = 11$

$p(u, v) \approx 0.39$



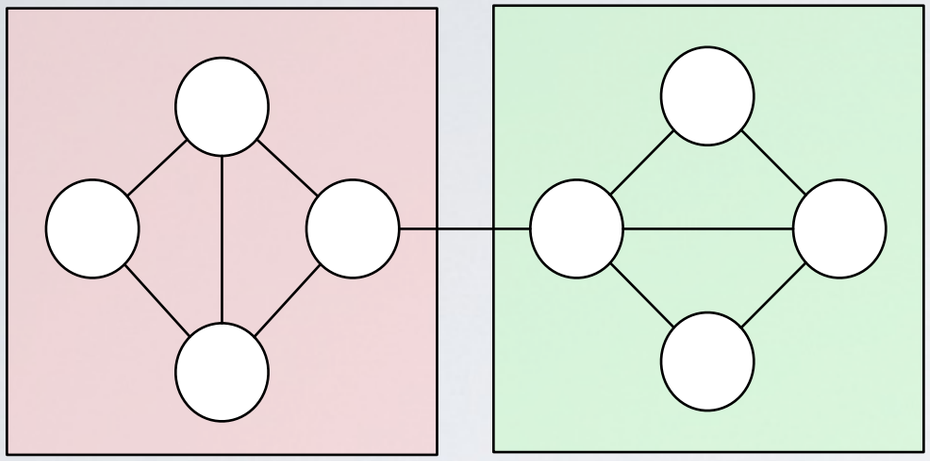
ER random graph

Expected edges inside red (or green)
(#node pairs * prob to observe an edge)

$$\frac{4(4-1)}{2} * p(u, v) = 2.34$$

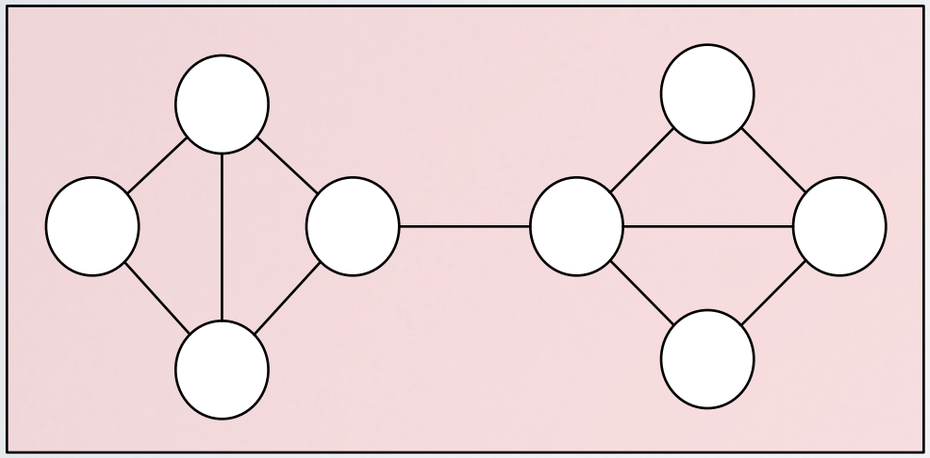
$$\text{Modularity} = \frac{2(5 - 2.34)}{m} = 0.48$$

MODULARITY INTUITION

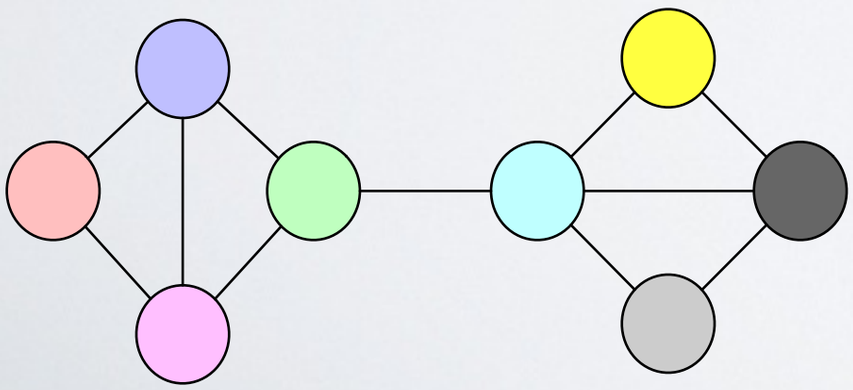


$n = 8$
$m = 11$

Q= 0.48

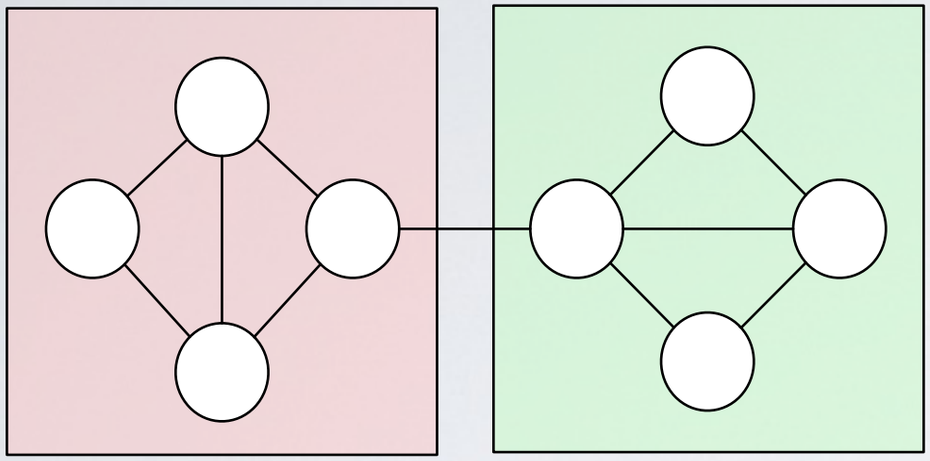


Q= ?



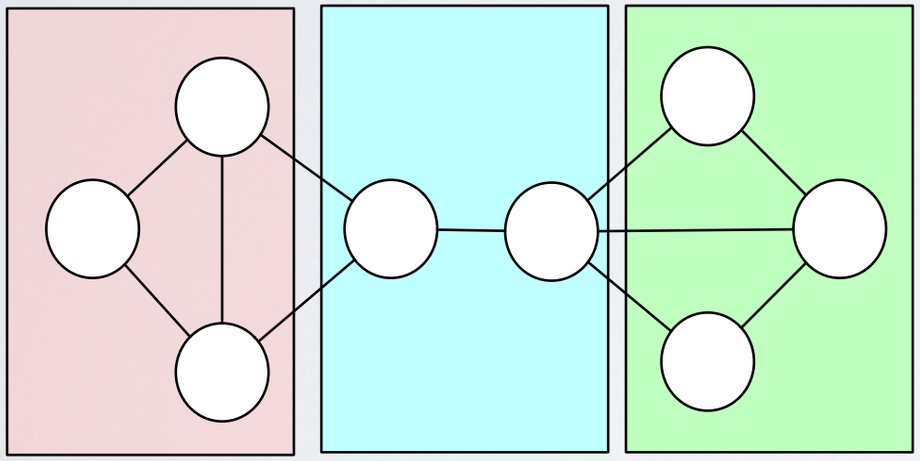
Q= ?

MODULARITY INTUITION

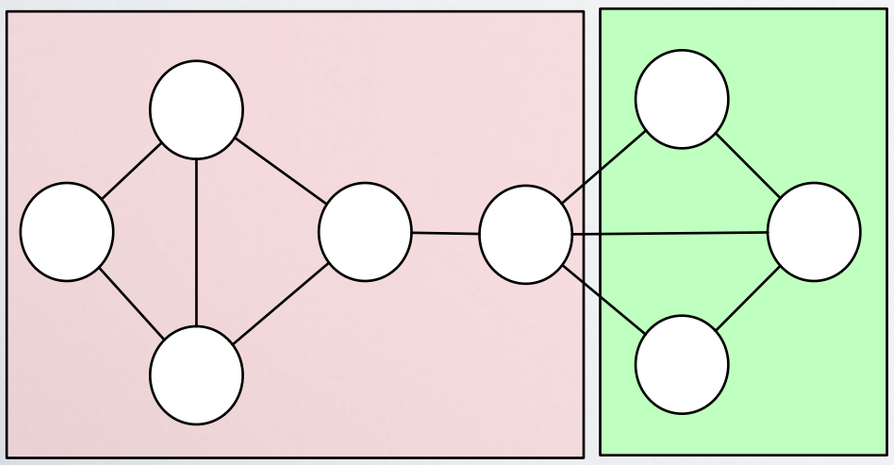


p=0.39

Q= (5-6p)+(5-6p)=10-12p=5.32

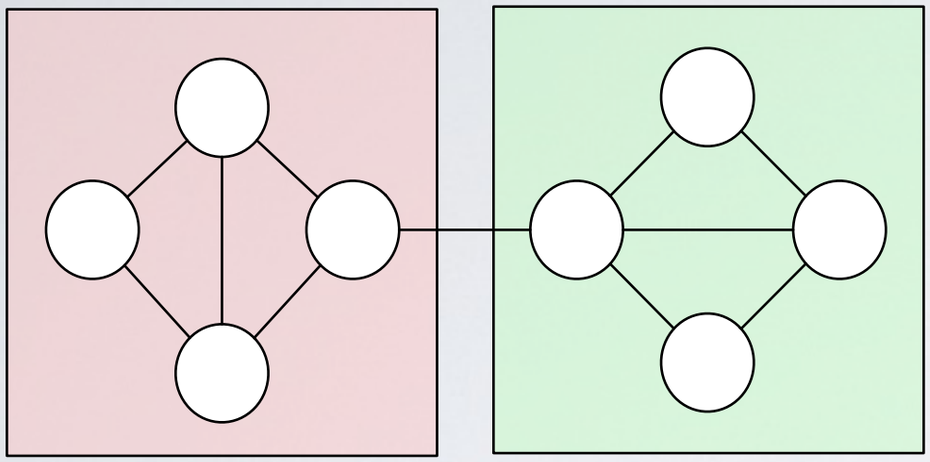


Q=(3-3p)+(1-p)+(2-3p)=7-7p=4.27



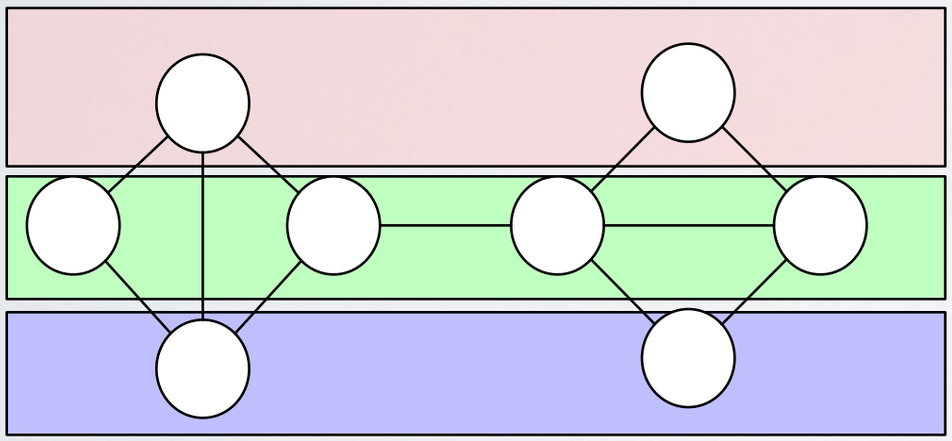
Q=(6-10p)+(2-3p)=8-13p=2.93

MODULARITY INTUITION



p=0.39

Q= (5-6p)+(5-6p)=10-12p=5.32



Q=(0-p)+(2-6p)+0-p=2-8p= - 0.34

MODULARITY NULL MODEL

- In previous examples, we used ER as a null model

- Usual approach: configuration model as null model
  - Preserves each node's degree
  - $p(u,v) = \frac{k_u k_v}{2m}$

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

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

1 if in same community

MODULARITY

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

1 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 ?

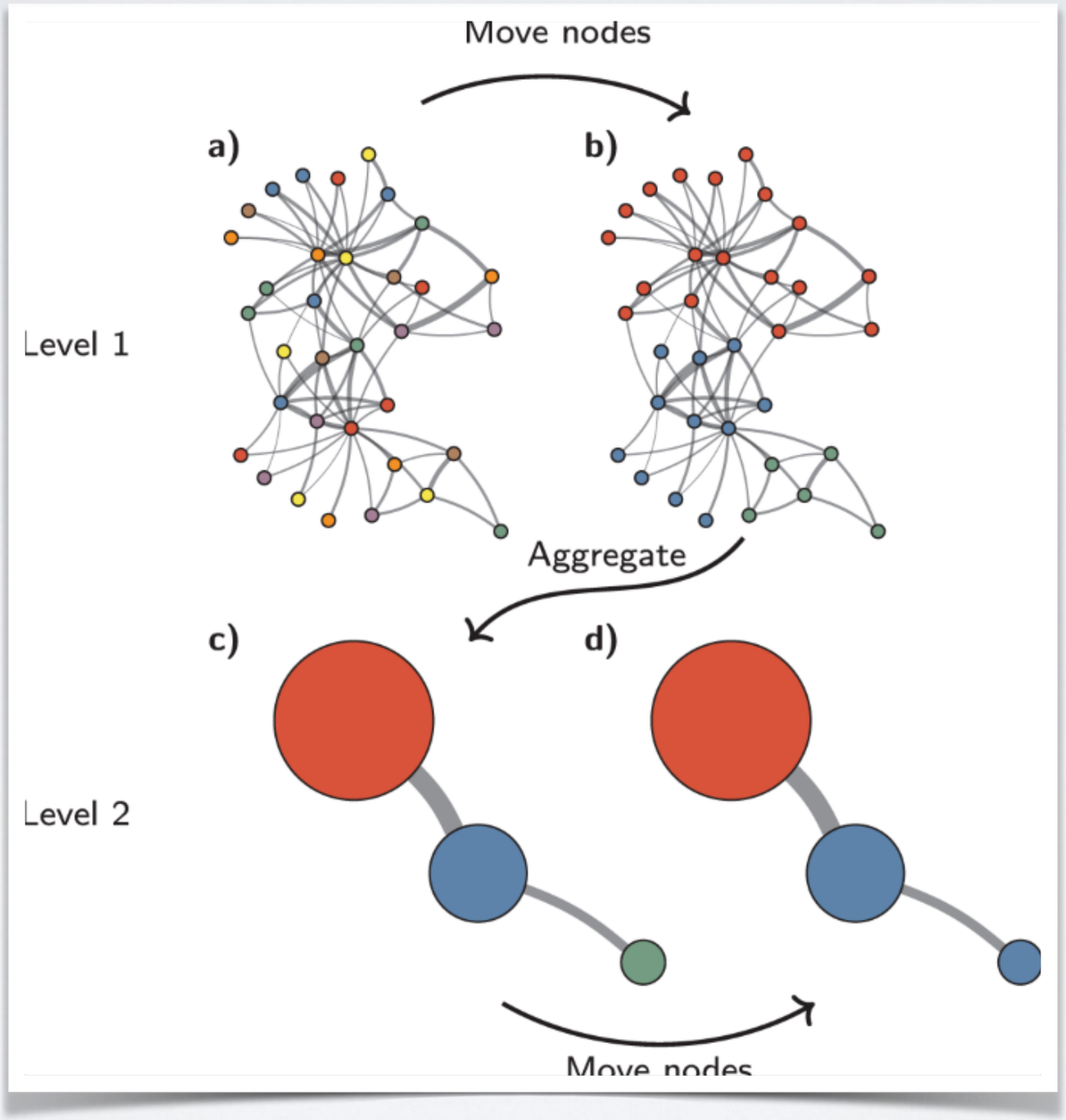
- Why not optimizing it directly !

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*

Blondel, Vincent D., et al. "Fast unfolding of communities in large networks." *Journal of statistical mechanics: theory and experiment* 2008.10 (2008): P10008.

LOUVAIN ALGORITHM



Blondel, Vincent D., et al. "Fast unfolding of communities in large networks." *Journal of statistical mechanics: theory and experiment* 2008.10 (2008): P10008.

ALTERNATIVES

- Most serious alternatives
  - Infomap (based on information theory —compression)
  - Stochastic block models (bayesian statistical inference)

- These methods have a clear definition of what are good communities. Theoretically grounded

EVALUATION OF COMMUNITY STRUCTURE

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