

# Introduction to Social Media Analytics (Lec 11)

Hao PENG

Department of Data Science

City University of Hong Kong

<https://haoopen.github.io/>

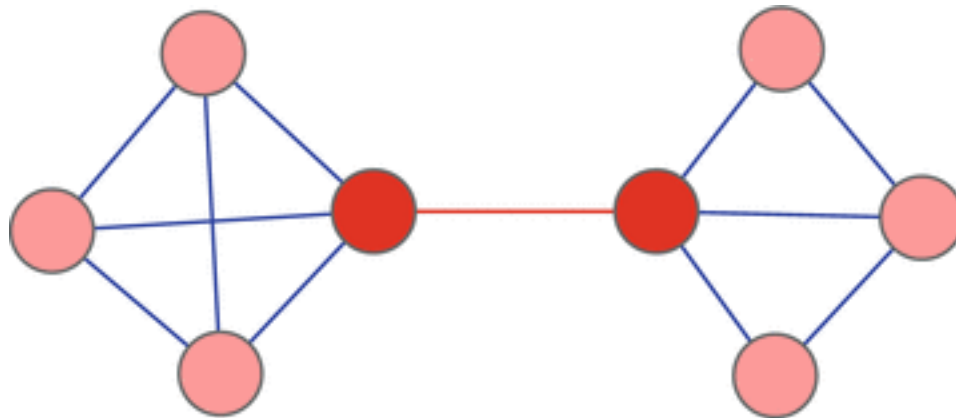
# Topics for this week

- Network Science
  - Background
- Characteristics of Social Networks
  - Short average distance
  - High clustering coefficient
  - Power-law degree distribution
  - Social influence/hubs
  - Strong and weak ties
  - Community structure
- Applications
  - Community detection
  - Cascade prediction
  - Information diffusion
  - Network visualization

# Edge b/w centrality (structure)

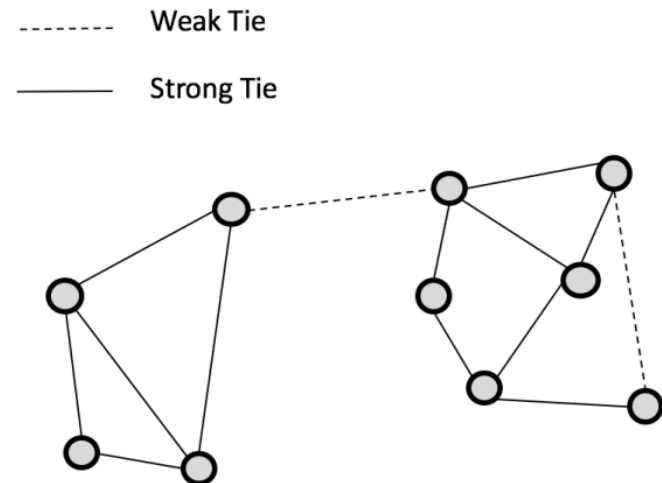
Betweenness centrality of an edge is the number of all-pairs shortest paths that pass through the edge.

(Structural definition)



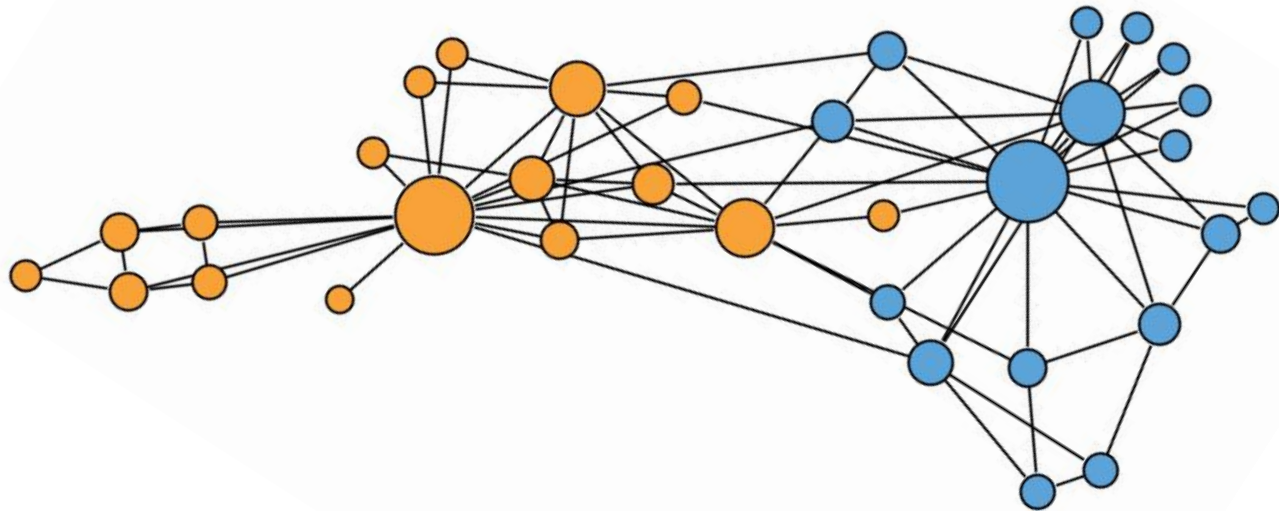
# Strong/weak ties (frequency)

- Measured by edge weights such as the contact frequency, the number of mutual friends, **instead of structural positions**.
- Strong ties are often seen within communities; whereas weak ties often connect/bridges different communities.
- Weak ties often have high betweenness centrality; they tend to be structurally more important! (weak in relationship but strong in global connectivity)



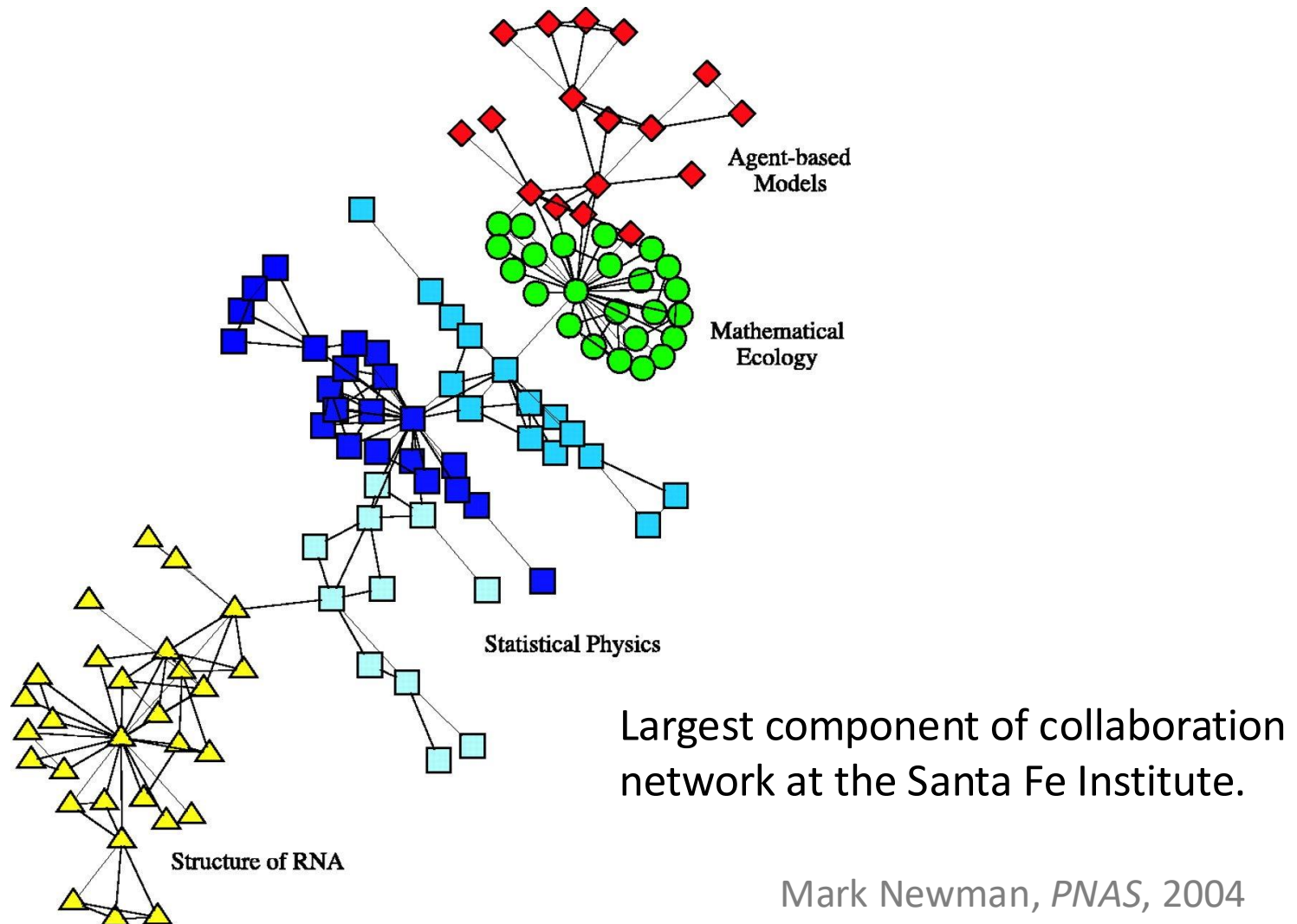
# Community structure

**Network communities** are groups of nodes in a network with lots of connections within the groups and not too many between groups.



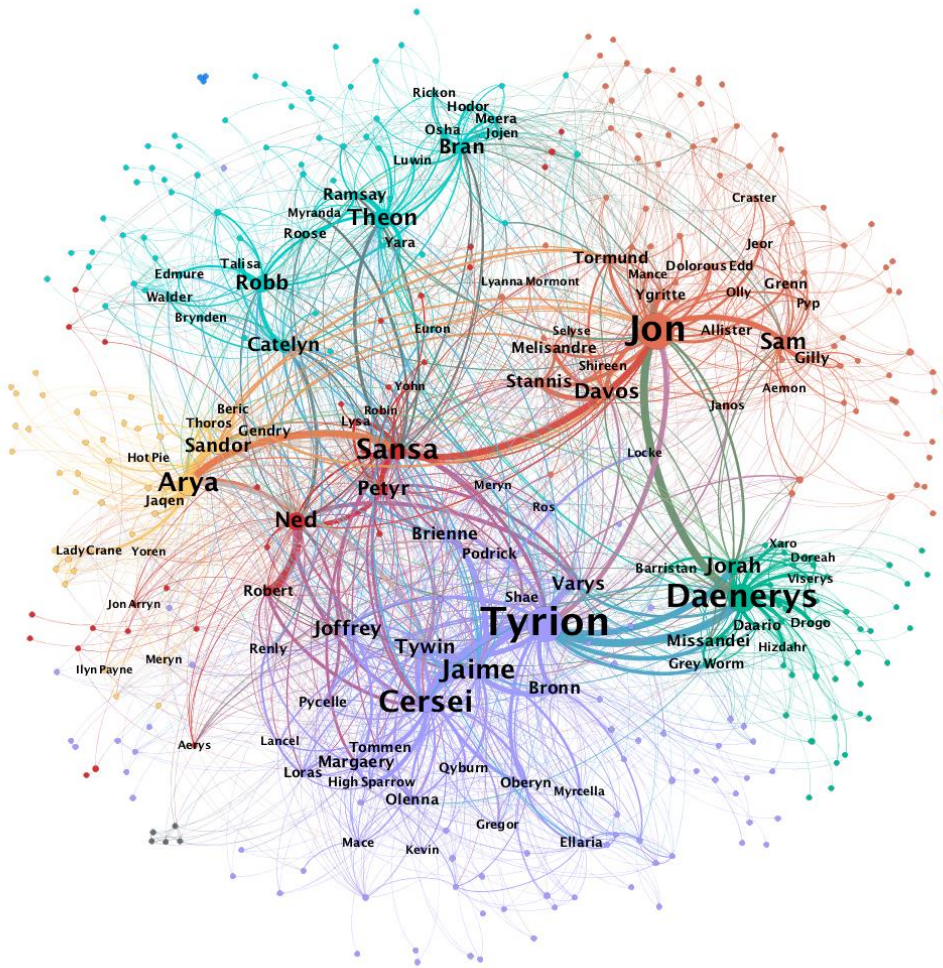
Friendship network in a karate club (Zachary 1977)

# Scientific collaboration networks



Mark Newman, *PNAS*, 2004

# Networks in Game of Thrones



How could we know about the strength of community structure?

Does the observed structure exist due to a random chance?

<https://medium.com/web-mining-is688-spring-2021/web-of-thrones-255139b930bb>

# Network modularity (Q)

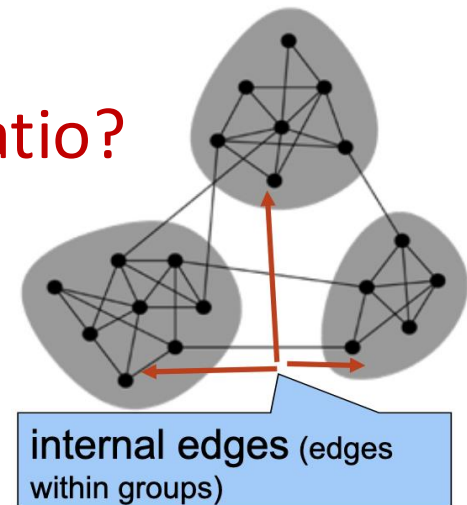
Modularity  $Q$  is the fraction of all edges that fall within the same groups minus the expected fraction if edges were distributed at random.

How to compute the edge density ratio?

$k_i$  = degree of node  $i$

$M = \sum k_i = 2|E|$

$A_{ij} = 1$  if  $(i,j) \in E$ , 0 otherwise



$$Q = \frac{\sum (A_{ij} - k_i k_j / M \mid i, j \text{ in the same group})}{M}$$

Mark Newman, "Modularity and community structure in networks". *PNAS*, 2006

# Community detection

**Network communities** are groups of nodes in a network with lots of connections within groups and not too many between groups.

Algorithms for identifying communities:

1. Deploying a **hierarchical clustering** idea:
  - a) **Agglomerative** (bottom up: successive pairing nodes by adding edges)
  - b) **Divisive** (top down: successive splitting clusters by removing edges)
2. Based on **information theory** (using random walks)

What are real-world applications of community detection?

# Hierarchical clustering approach

Start from each node as a community:



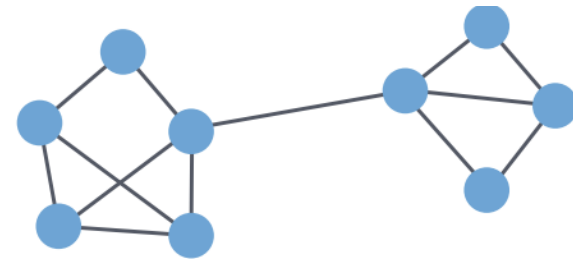
Agglomerative clustering



Suboptimal Partition  
 $M=0.22$



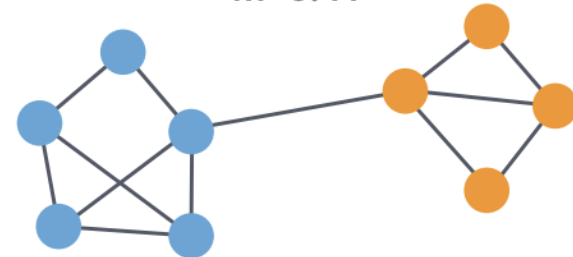
Start from all nodes in a single community:



Divisive clustering



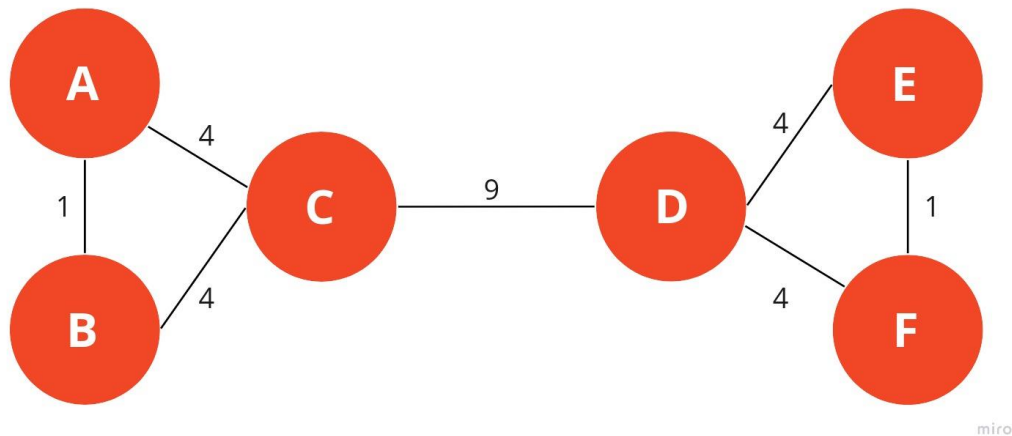
Optimal Partition  
 $M=0.41$



# The Newman-Girvan algorithm

Repeatedly **removing** edges (divisive):

1. Calculate betweenness scores for all edges in the network.
2. Remove the edge with the highest betweenness centrality.
3. Recalculate betweenness for all remaining edges.
4. Repeat steps 2-4 until there are no more edges left.

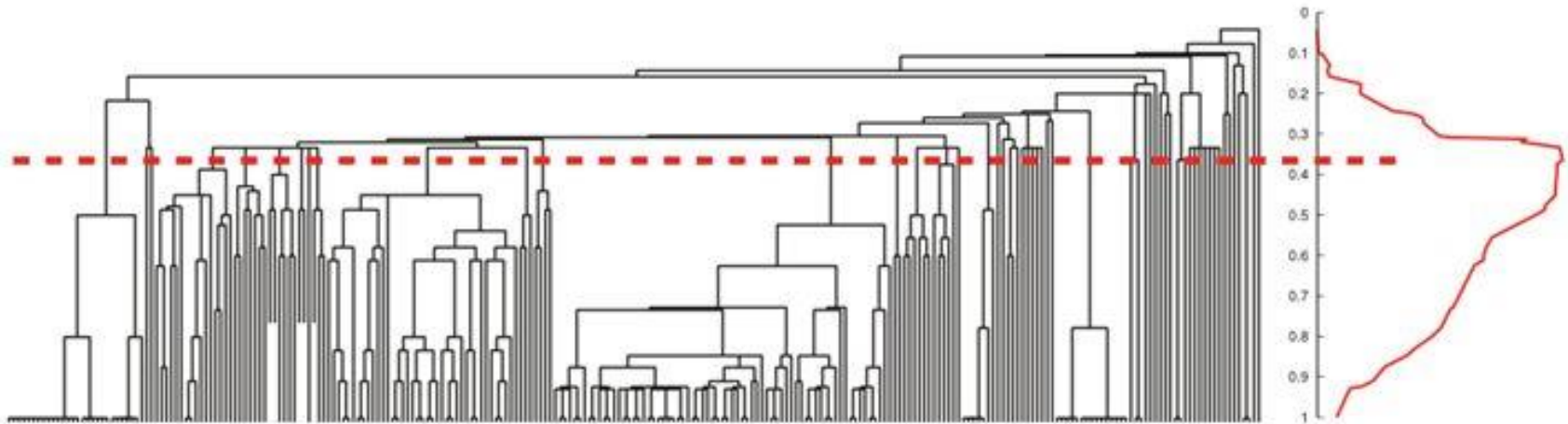


Mark Newman, *PNAS*, 2006; Girvan and Newman, *PNAS*, 2002

<https://memgraph.github.io/networkx-guide/algorithms/community-detection/girvan-newman/>

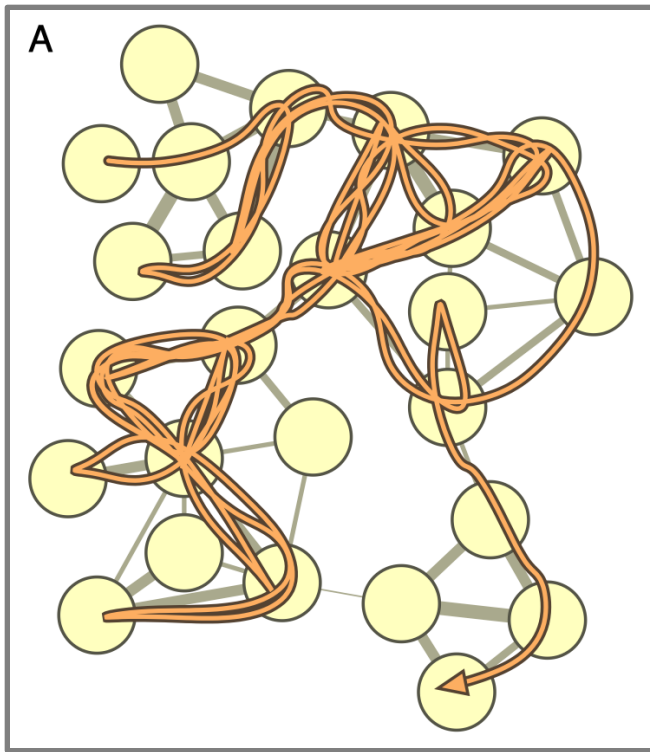
# The Newman-Girvan algorithm

- Repeatedly remove edges based on betweenness centrality.
- It produces a dendrogram for different depth of network partition.
- Can use **modularity** to measure the strength of community structure.



# The InfoMap algorithm

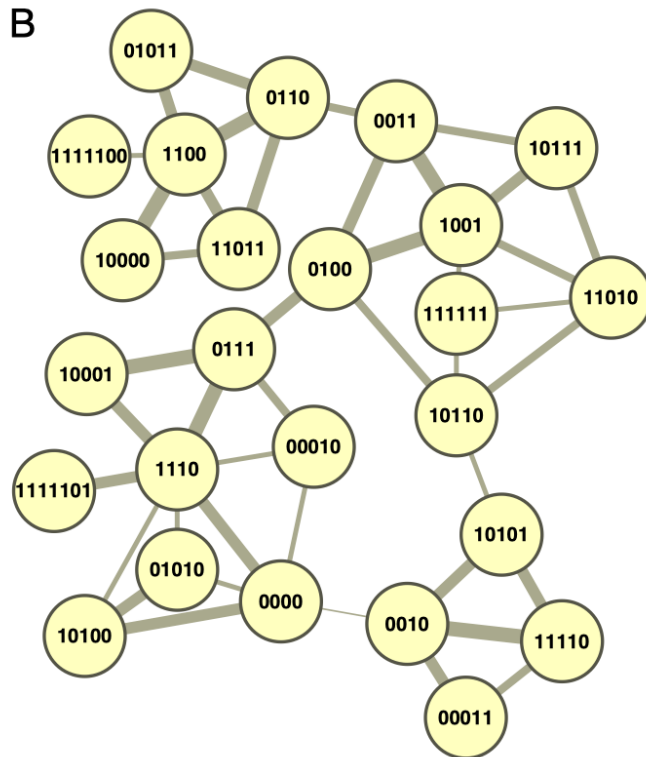
- Use information theory to describe trajectory of random walks
- Task: **finding an efficient coding scheme to compress walk paths**



Rosvall and Bergstrom. “Maps of random walks on complex networks reveal community structure.” *PNAS* 2008.

# The InfoMap algorithm

Baseline method: *give a unique name to each node in the network*



For example, use Huffman coding:

```
1111100 1100 0110 11011 10000 11011 0110 0011 10111 1001 0011  
1001 0100 0111 10001 1110 0111 10001 0111 1110 0000 1110 10001  
0111 1110 0111 1110 1111101 1110 0000 10100 0000 1110 10001 0111  
0100 10110 11010 10111 1001 0100 1001 10111 1001 0100 1001 0100  
0011 0100 0011 0110 11011 0110 0011 0100 1001 10111 0011 0100  
0111 10001 1110 10001 0111 0100 10110 111111 10110 10101 11110
```

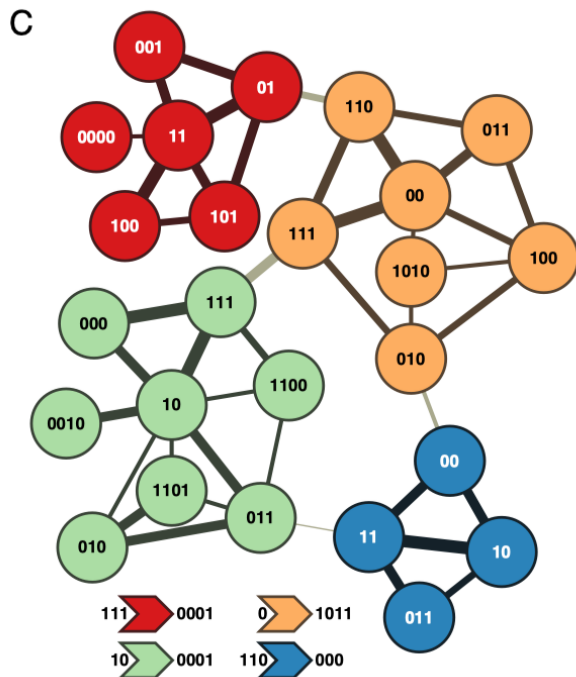
How can we do better?

Rosvall and Bergstrom. "Maps of random walks on complex networks reveal community structure." *PNAS* 2008.

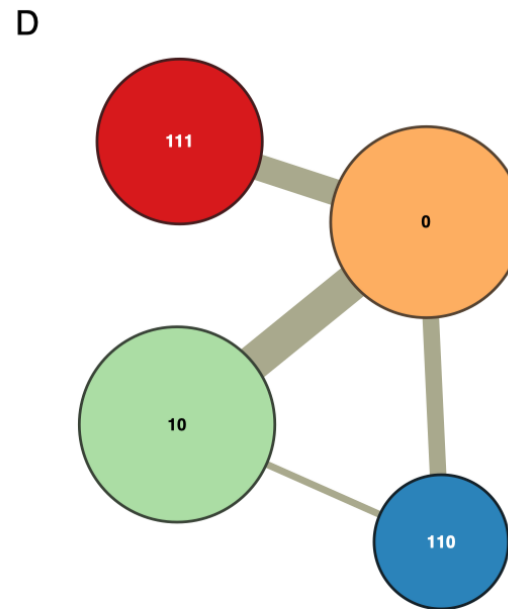
# The InfoMap algorithm

A two-level coding scheme:

- Major clusters receive unique names (enter/exit codes)
- But the names of nodes within clusters are reused



111 0000 11 01 101 100 101 01 0001 0 110 011 00 110 00 111 1011 10  
111 000 10 111 000 111 10 011 10 000 111 10 111 10 0010 10 011 010  
011 10 000 111 0001 0 111 010 100 011 00 111 00 011 00 111 00 111  
110 111 110 1011 111 01 101 01 0001 0 110 111 00 011 110 111 1011  
10 111 000 10 000 111 0001 0 111 010 1010 010 1011 110 00 10 011



243 vs. 314 bits.

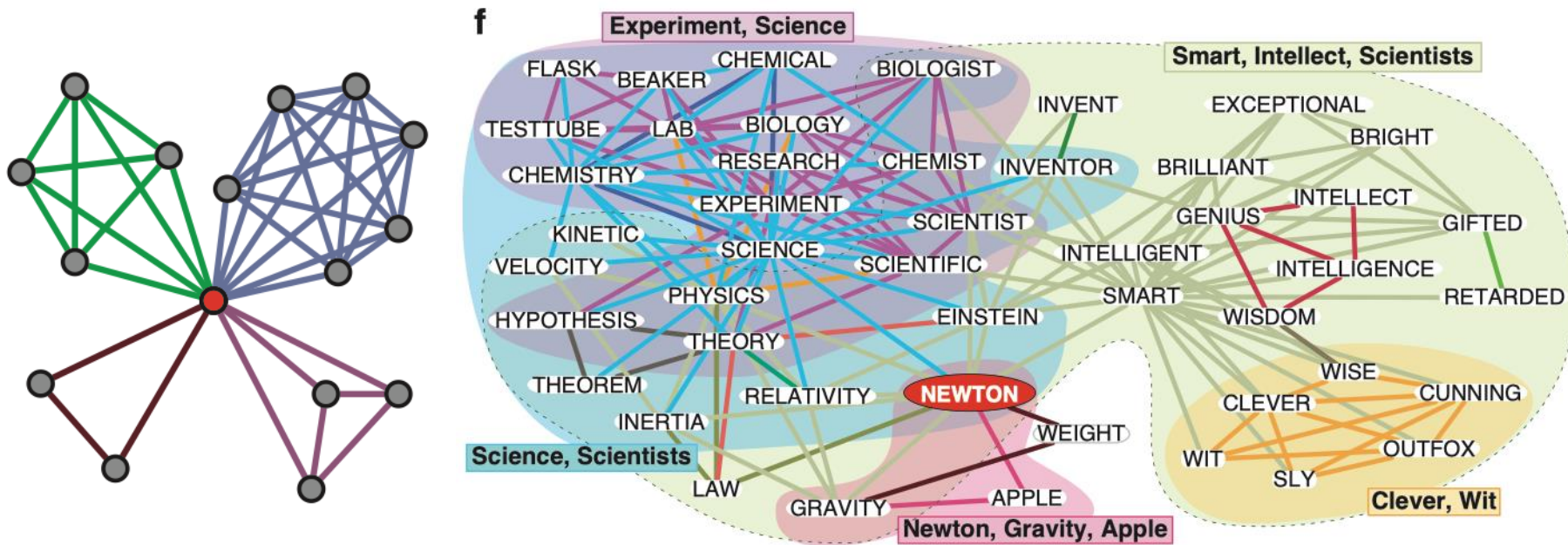
Big improvement!

Treat major clusters  
as the communities.

111 0000 11 01 101 100 101 01 0001 0 110 011 00 110 00 111 1011 10  
111 000 10 111 000 111 10 011 10 000 111 10 111 10 0010 10 011 010  
011 10 000 111 0001 0 111 010 100 011 00 111 00 011 00 111 00 111  
110 111 110 1011 111 01 101 01 0001 0 110 111 00 011 110 111 1011  
10 111 000 10 000 111 0001 0 111 010 1010 010 1011 110 00 10 011

# Community detection based on links

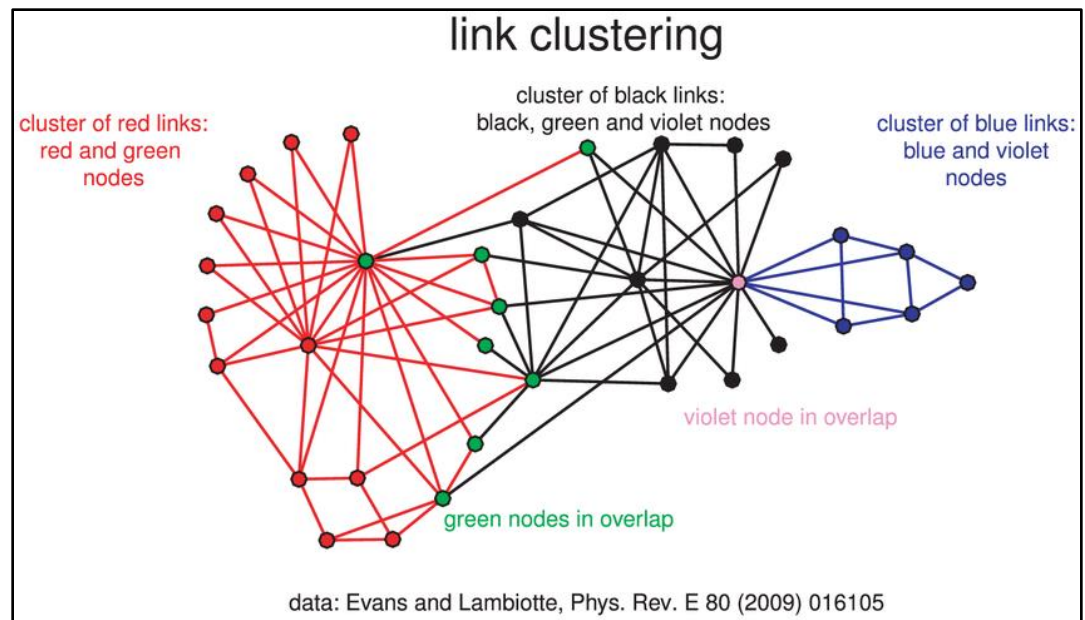
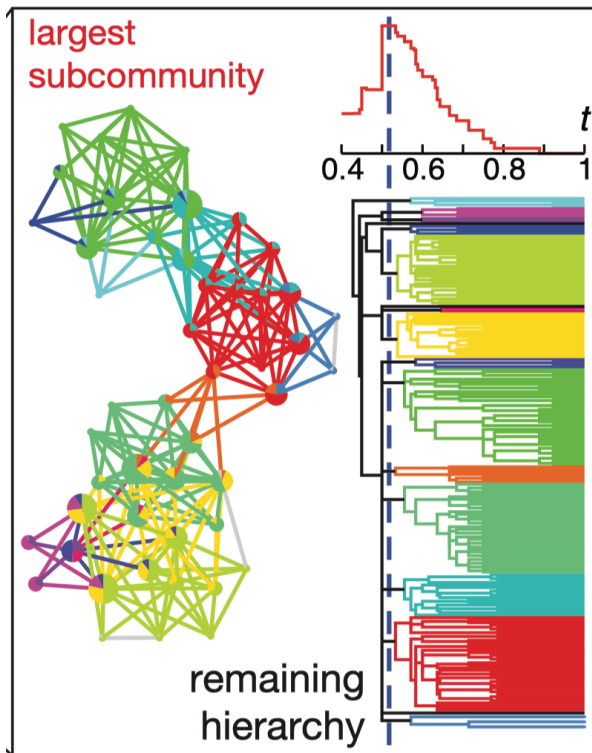
- The node-level methods partition nodes into **disjoint** groups.
- But a node can (often) belong to multiple communities.



Ahn et al. "Link communities reveal multi-scale complexity in networks." *Nature* 2010.

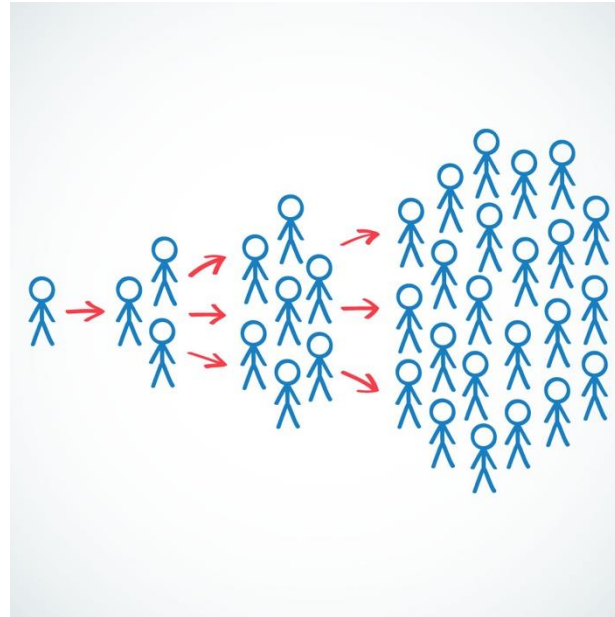
# Community detection based on links

- Clustering links for finding **overlapping node communities**
- A node can belong to multiple identified link communities



Ahn et al. "Link communities reveal multi-scale complexity in networks." *Nature* 2010.

# Viral marketing / information diffusion



Knowledge about **the network structure** can be useful for designing effective marketing campaigns. **Why?**

# Simple vs. complex contagion

Two common information diffusion models:

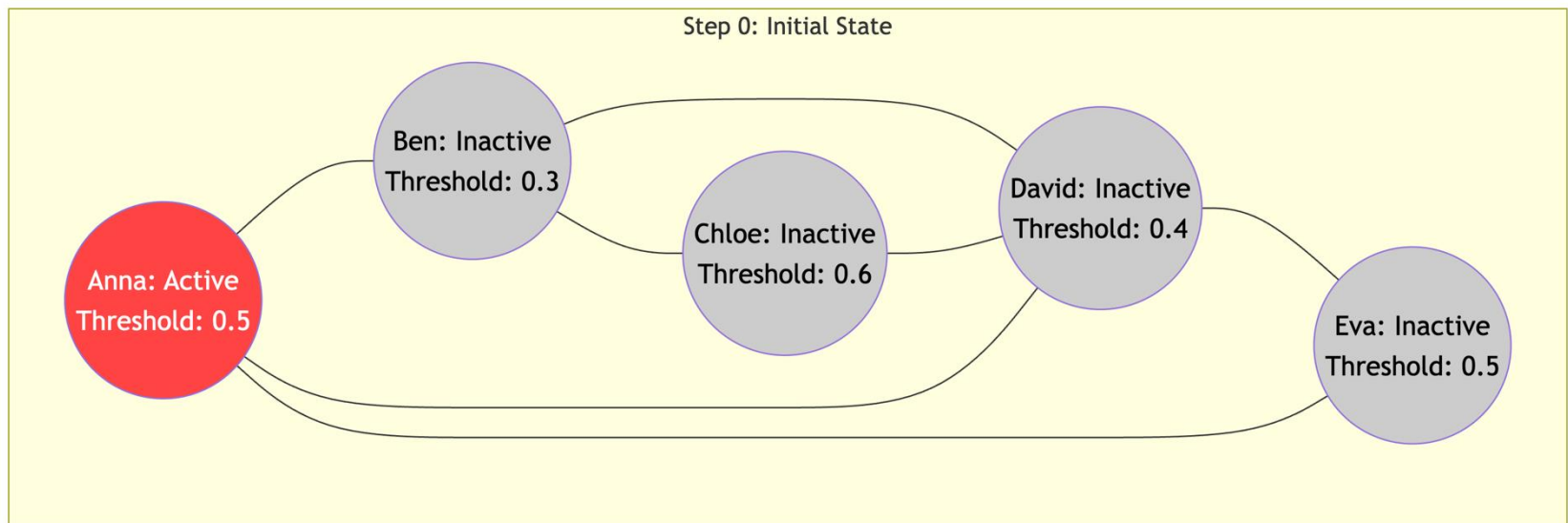
- Independent cascade model (simple contagion)
  - Information spreads like **disease transmission**
  - A node has **only one chance** to “infect” its neighbors
  - **Each exposure is independent from each other**
- Linear threshold model (complex contagion **Our focus!**)
  - Exposure to multiple neighbors is needed for adoption
  - Each node has a **threshold** for activation (**count** or **fraction**)
  - More applicable to the spread of **contentious** behaviors  
(such as *wearing mask, voting, jaywalking, new fashion*)

**More examples?**

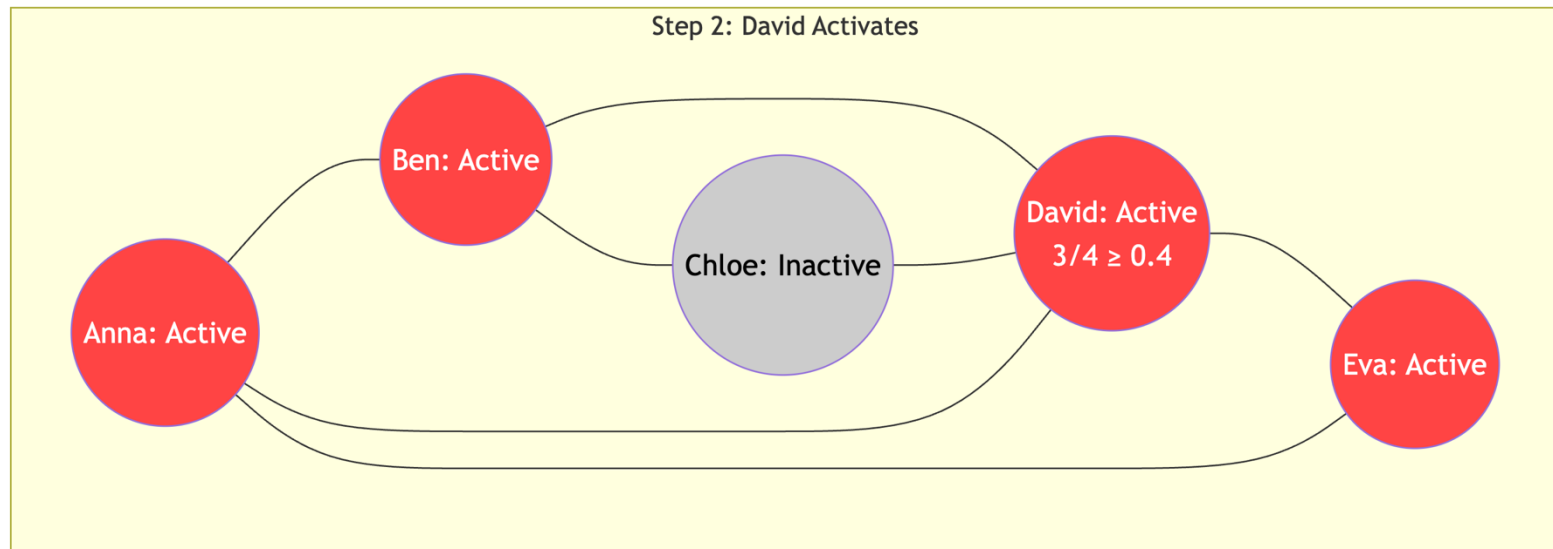
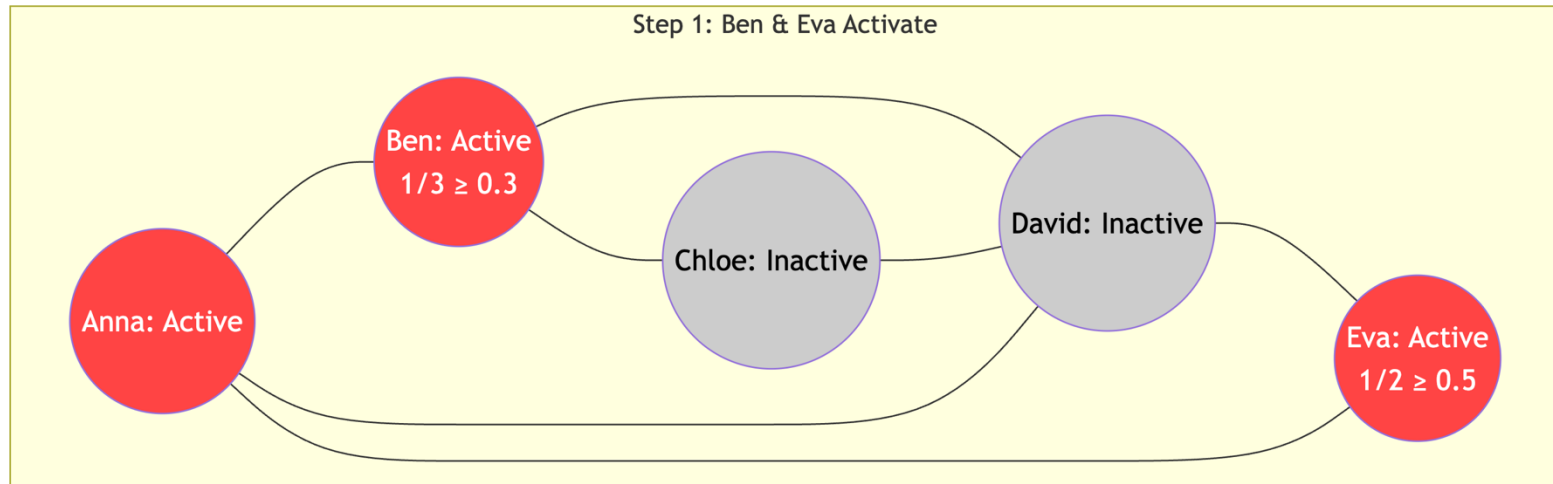
# Agent-based modeling

Simulating complex contagion in discrete steps:

- Start with a few seed nodes at step 0.
- In each step, for each inactive node, check if the fraction of its active friends meets its threshold.
- If so, activate the node (it remains active permanently later).
- The process unfolds deterministically until no change happens.



# Complex contagion illustration



# Strategies for global / large cascade

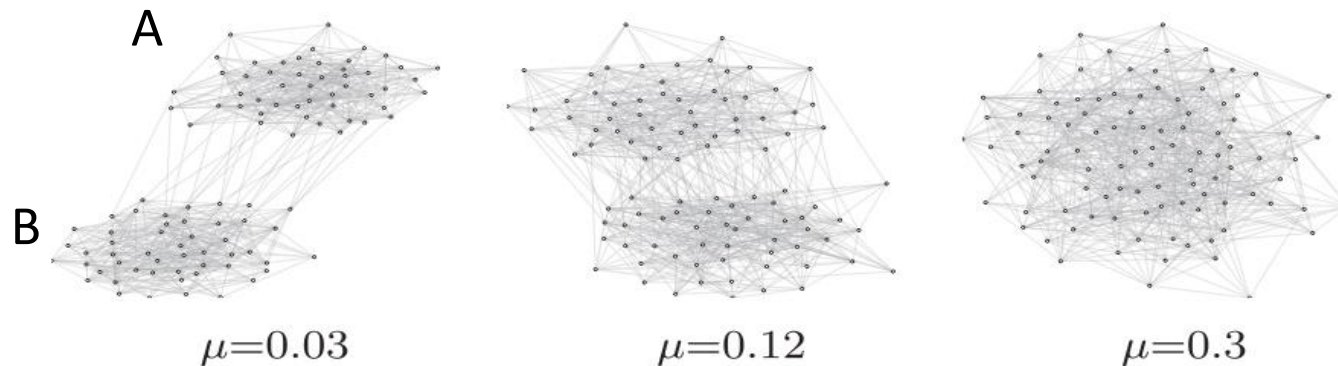
A **global cascade** means that every node gets infected in the end.

How can we “infect” the whole network with a limited budget?

- How to select seeds? Pick only influential nodes?
- Are super stars as important as we think?
- Shall we pay attention to the whole network structure?

# Community structure impacts global cascades more than seed nodes!

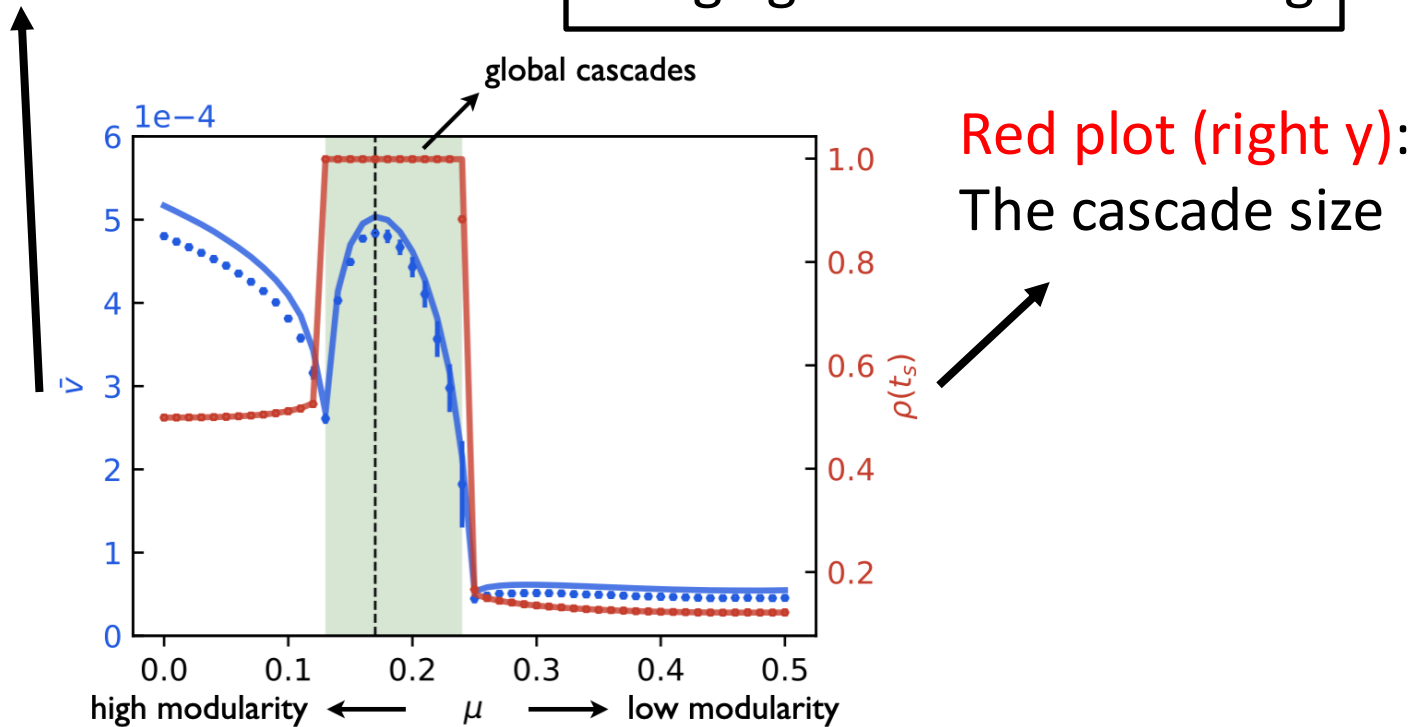
- ❑ A network with 2 equal-sized communities A and B;
- ❑ Number of nodes and edges are fixed;
- ❑ Each node has the same threshold 0.35;
- ❑ Define  $u$  as the fraction of edges running between A and B;
- ❑ Let  $u$  control strength of modularity (large  $u$  --> weak community);
- ❑ Randomly activate 20% nodes (as seeds) in A; Calculate cascade size.



# Optimal modularity for global cascade

Blue plot (left y axis):  
The diffusion speed

Run 1000 simulations per  $u$   
using agent-based modeling



Peng et al. "Network modularity controls the speed of information diffusion". *PRE*, 2020.

# Optimal modularity for global cascade

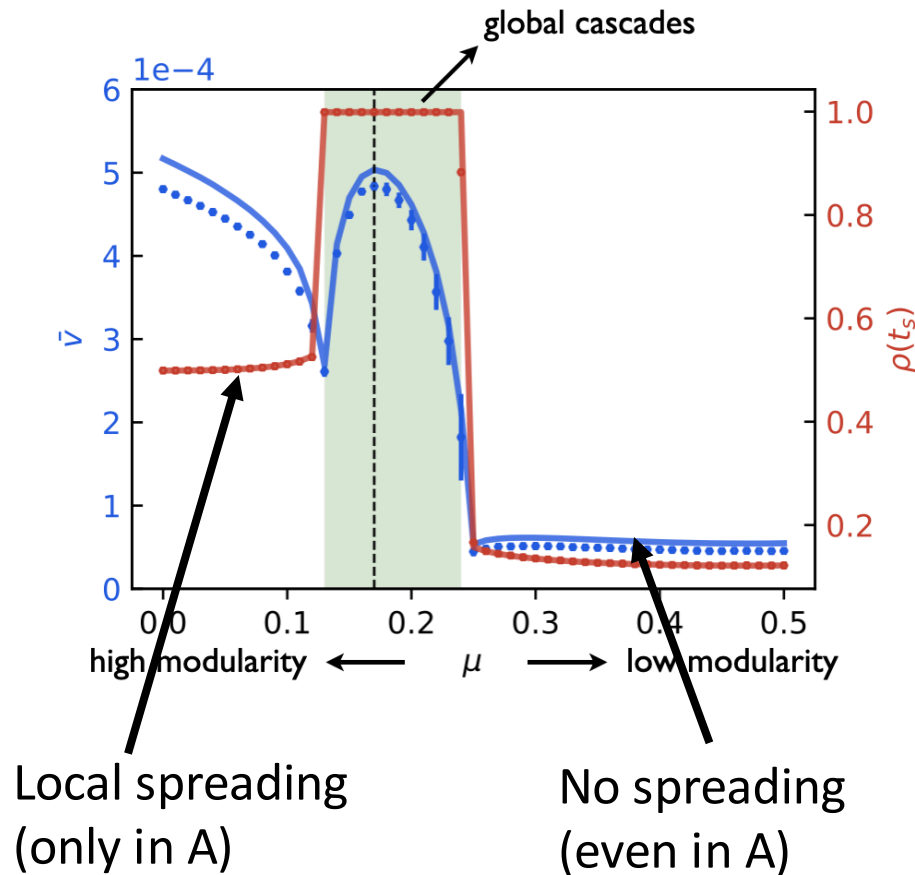


Fig: With a medium level of community structure, global cascades are guaranteed to happen, no matter how you pick the seeds. **Why?**

What if you fix the seed nodes, but vary the  $u$ ?

Peng et al. "Network modularity controls the speed of information diffusion". *PRE*, 2020.

# Complex contagion & Cascade

- ❑ Large cascades don't depend on highly influential individuals.
- ❑ Instead, they require a **critical mass** (**early adopter**) of easily influenced people connected to other easy-to-influence people.
- ❑ The key to the spread of complex contagion is to find **tightly connected groups of easily influenced adopters where social influence/reinforcement can flow effectively and efficiently.**

*Community structure can provide enough social influence needed to trigger a global cascade from the early adopters.*

- Can social influence exist without a network?
- Are there examples of cascades without a network?

# Virality prediction

**Memes** are units of transmissible information (product, norm, behavior).

**Virality prediction:** *Is this meme going to “infect” x% of the whole network?*

Potential useful factors based on the initial stage of sharing:

- Meme content
- User characteristics
- Temporal variation
- Network topology
- Community structure



<https://www.youtube.com/watch?v=XqZsoesa55w>

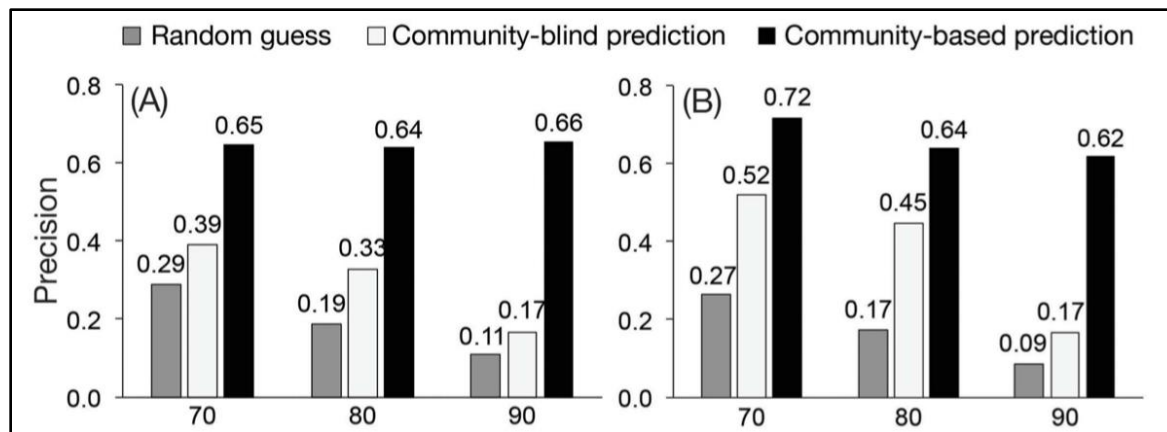
# Community structure matters

## Basic network features:

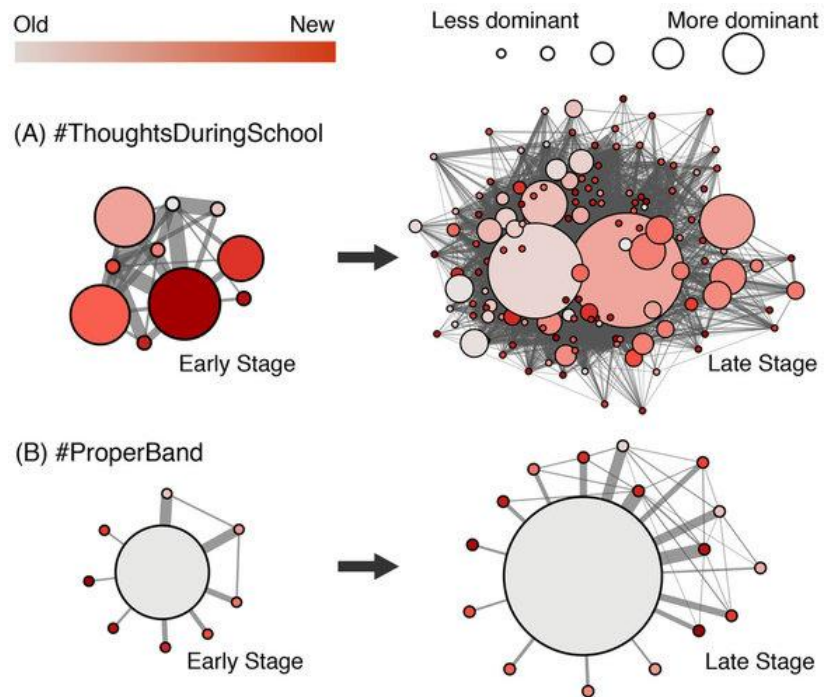
- Number of early adopters
- Number of uninfected neighbors of early adopters

## Community-based features (structural):

- Number of infected communities
- Usage and adoption entropy (measure the strength of concentration)
- Fraction of intra-community user interactions



# Influential early adopters are not required to produce large cascades



The evolution of a viral meme ([#ThoughtsDuringSchool](#)) from the early stage (30 tweets) to the late stage (200 tweets) of diffusion.

The evolution of a non-viral meme ([#ProperBand](#)) from the early stage to the final stage (65 tweets in total).

The effect of hubs is often exaggerated on online platforms;

# Visualizing networks



TITUS ANDRONICUS  
Number of characters 36 | 50% Network density



ROMEO AND JULIET  
Number of characters 41 | 37% Network density



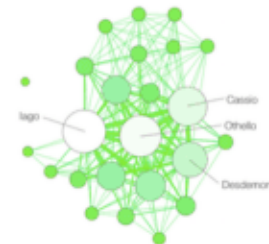
JULIUS CAESAR  
Number of characters 46 | 34% Network density



HAMLET  
Number of characters 37 | 39% Network density

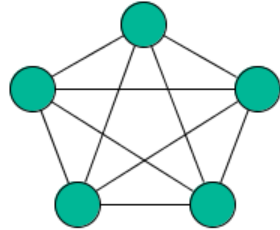
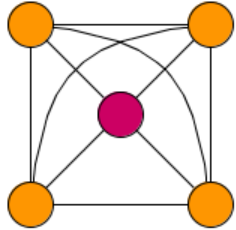


TROILUS AND CRESSIDA  
Number of characters 35 | 40% Network density



OTHELLO  
Number of characters 24 | 55% Network density

# Visualizing networks



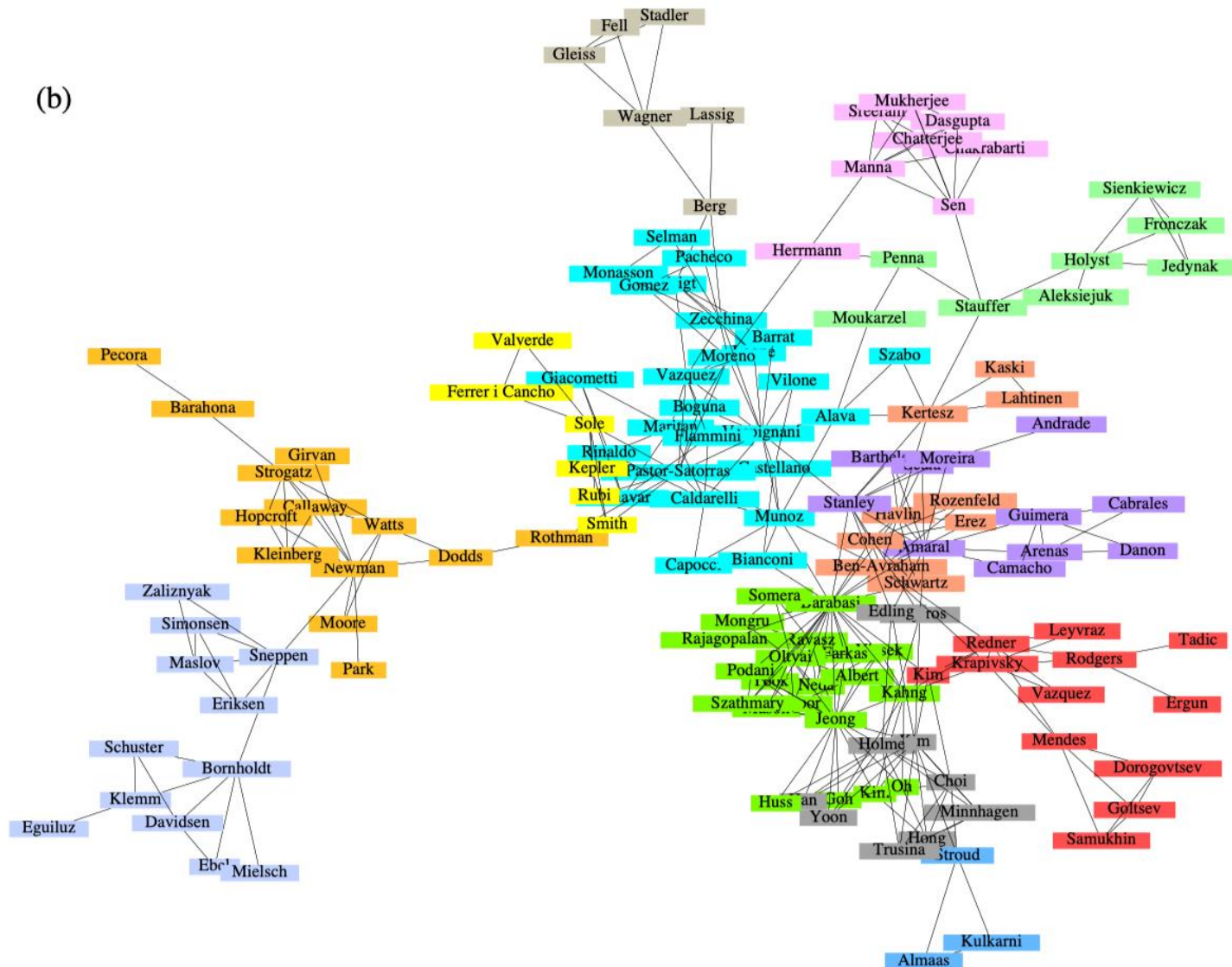
*Which node is the most central in the left graph?*

*Which node is the most central in the right graph?*

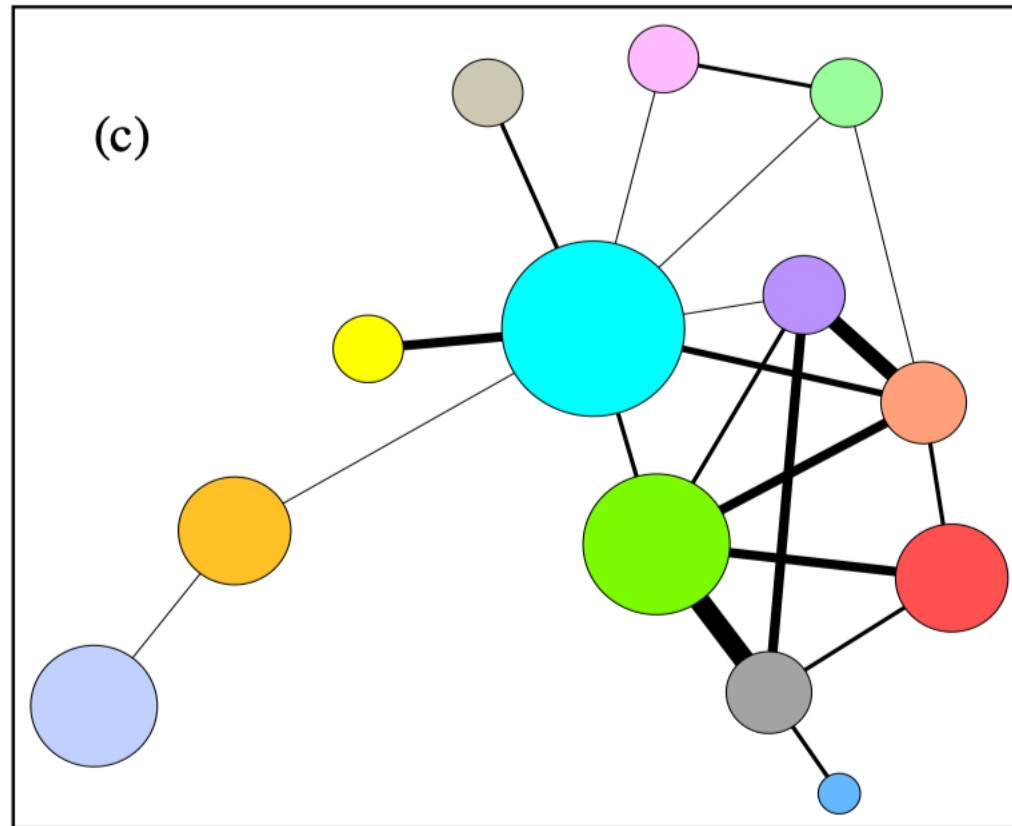
- A visualization of a network can be both revealing and deceiving
- Visualization is useful for exploration and hypothesis building
- Statistical analyses are needed for hypothesis testing

# Focusing on communities

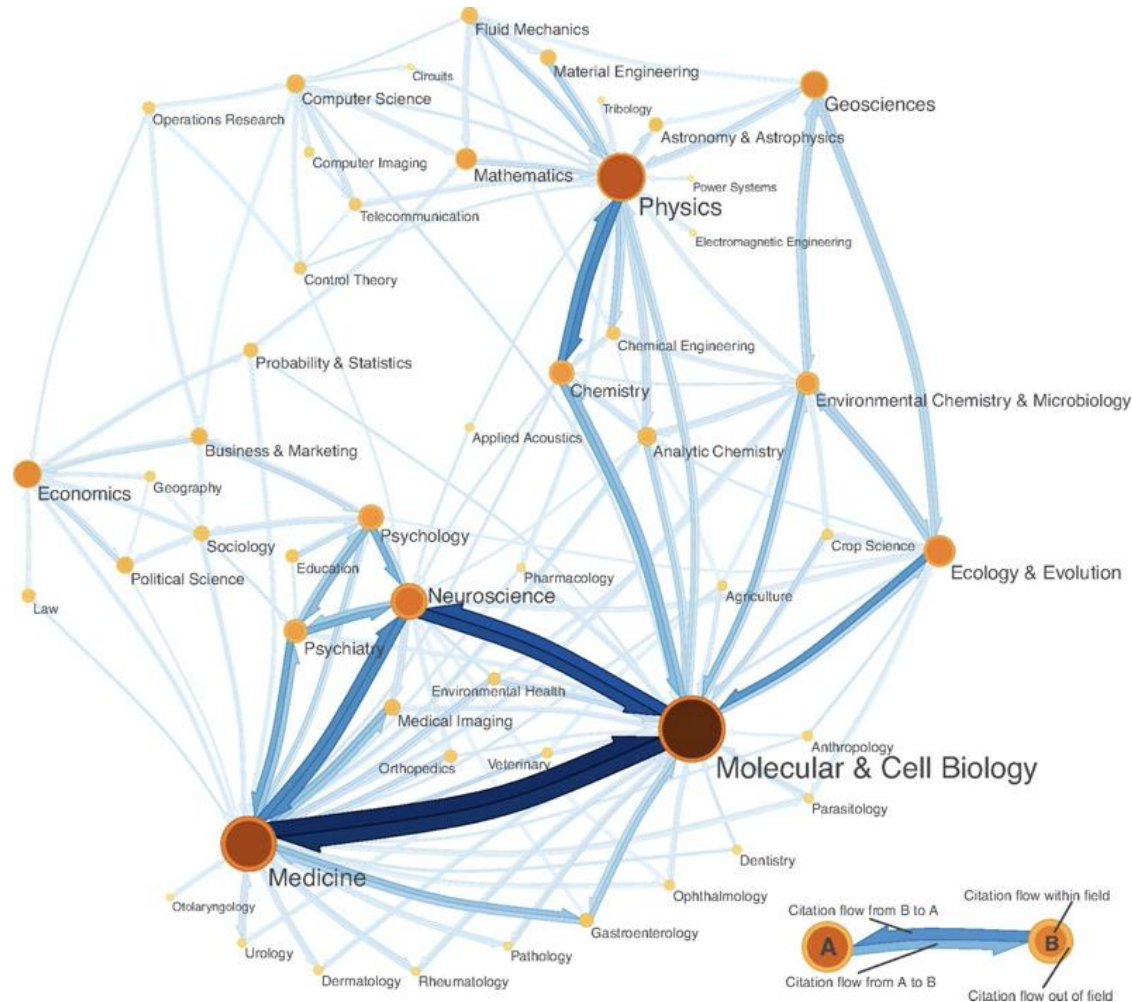
(b)



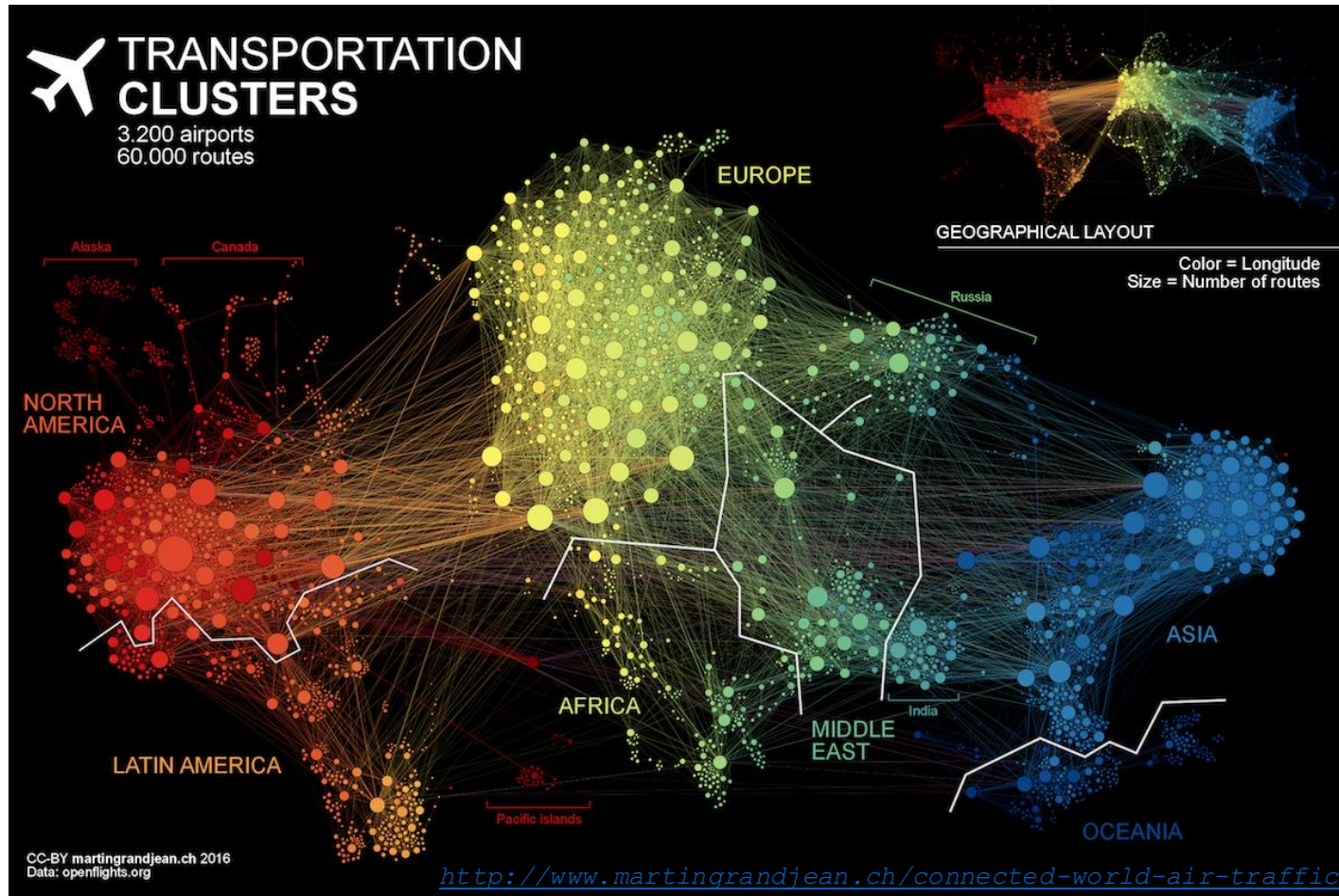
# Focusing on communities



# Show node size and link weights

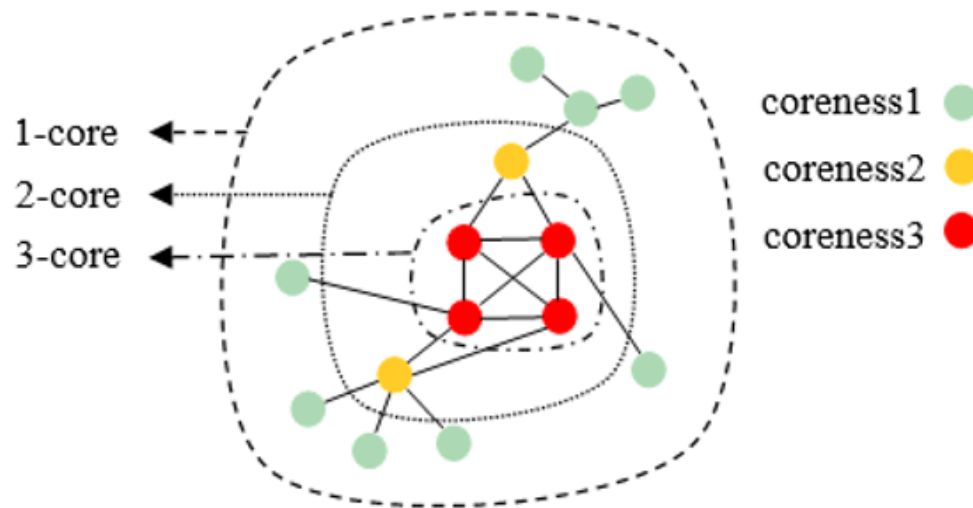


# Show node size and link weights



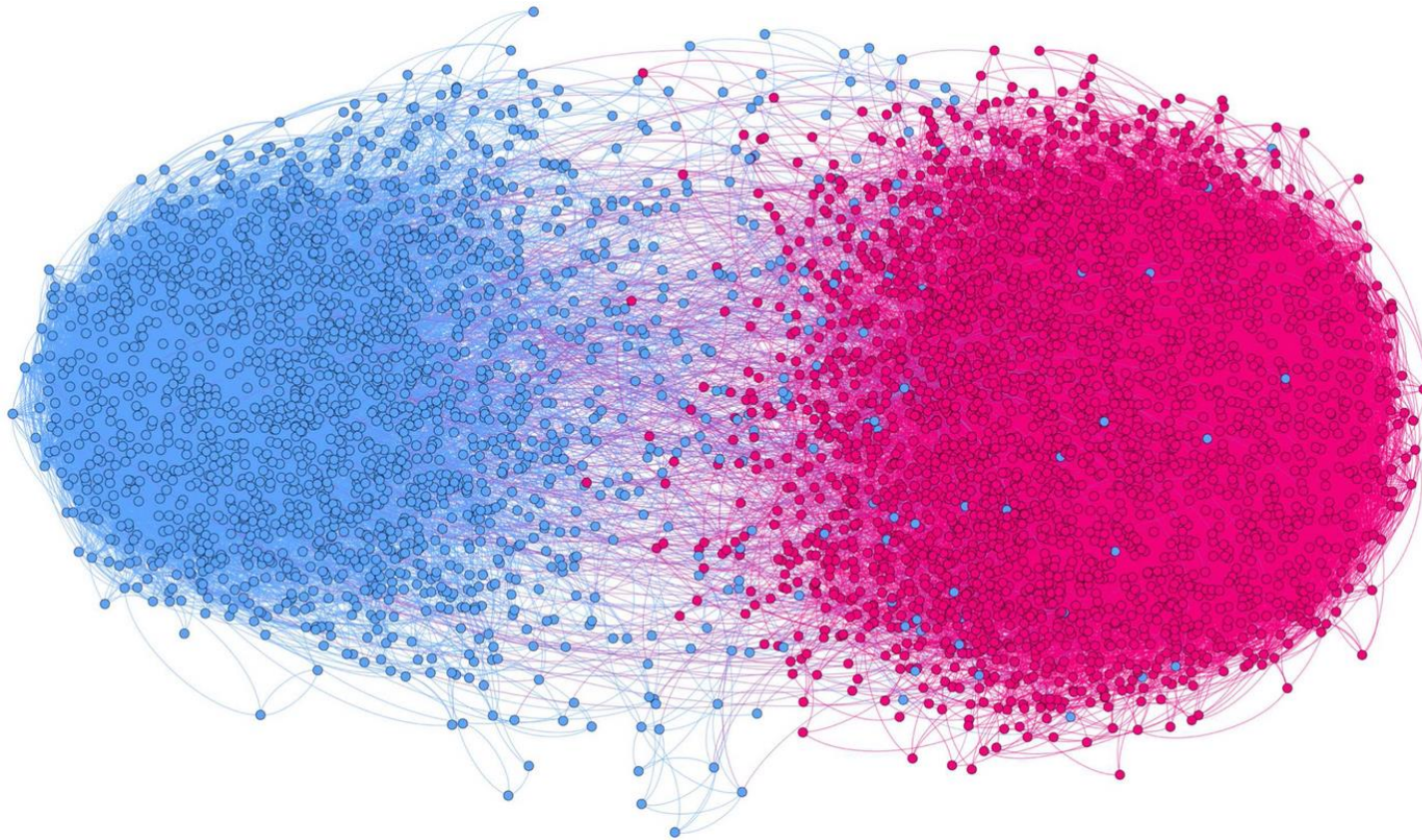
# K-score decomposition

- K-core: recursively removing all nodes of degree smaller than  $k$ , until the degree of all remaining nodes is at least  $k$ .
- Nodes are said to have coreness  $k$  (or, to belong to the  $k$ -shell) if they belong to the  $k$ -core but not to the  $(k + 1)$ -core.



# K-score visualization

Example of a politically polarized retweet network (only 3-core nodes are visualized).

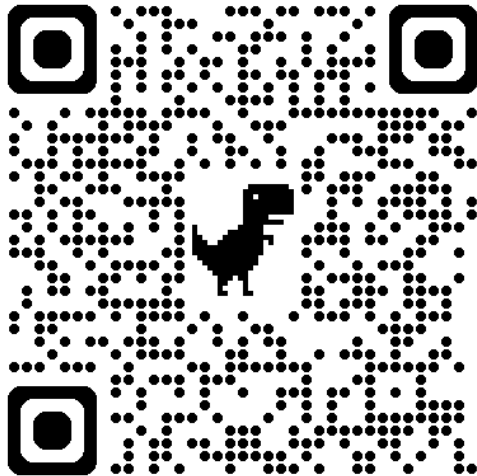


<https://link.springer.com/article/10.1007/s42001-020-00084-7>

# Interactive network visualization

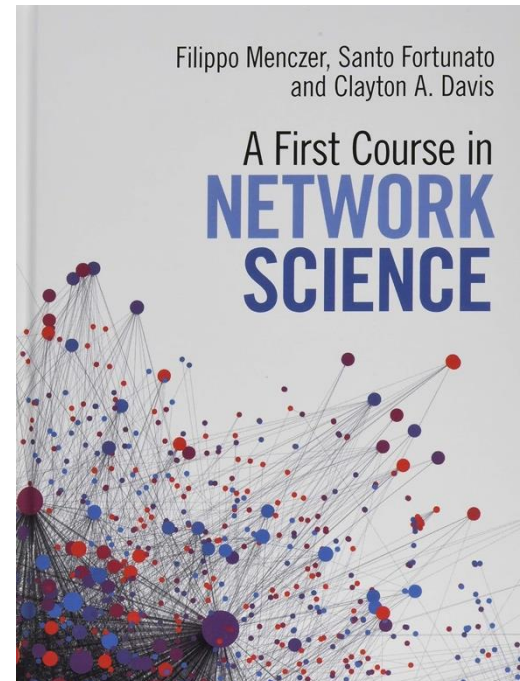
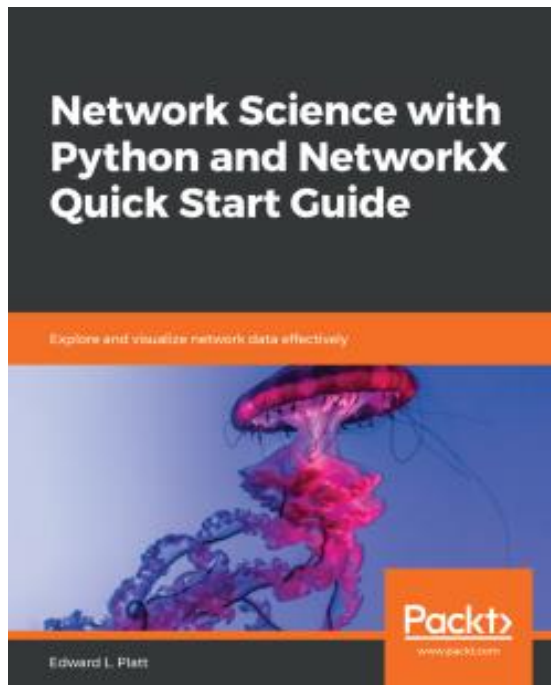
- **D3.js**: A JavaScript library for producing dynamic and interactive data visualizations in web browsers (cross-platform).
  - Pre-requisite skills: html, css, javascript
  - Books: “*Javascript by Example*”
  - Online sources: “*D3 in Depth*” (<https://www.d3indepth.com/>)

An interactive visualization of the formation of Echo Chambers:



# NetworkX

- A Python package to create, manipulate, and study the structure, dynamics, and functions of complex networks.
- NetworkX has the capacity to operate on very large graphs with more than 10 million nodes and 100 million edges.



# LOQ course evaluation survey

- Appreciate your valuable feedback!
- LOQ system: <https://onlinesurvey.cityu.edu.hk/>
- Also available on Canvas - our course site.



# Course notes

- HW3 due next Friday
  - Start early!
  - Pay attention to our late policy
- Next week agenda:
  - SMA Ethics
  - final exam review