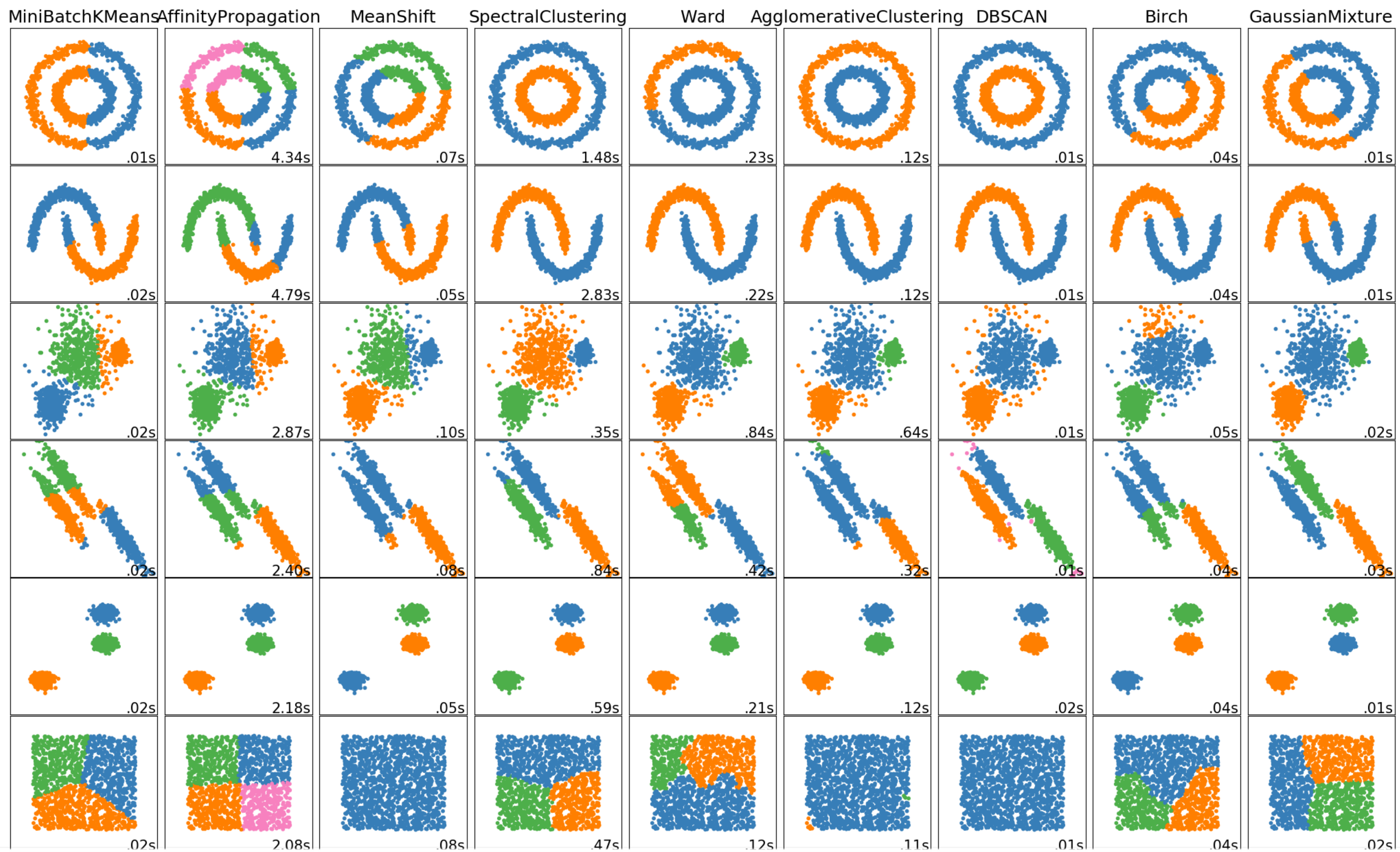
COMMUNITY DETECTION (GRAPH CLUSTERING)

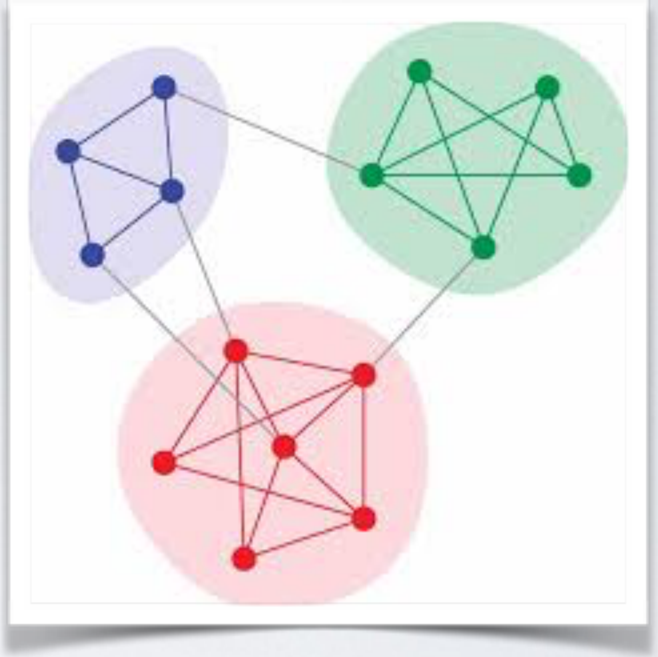
COMMUNITY DETECTION

- 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

WHY COMMUNITY DETECTION ?

- One of the key properties of complex networks was
  - High clustering coefficient
  - (friends of my friends are my friends)
- Different from random networks. How to explain it ? Evenly distributed ?
  - Watts strogatz (spatial structure?)
  - Forest fire, copy mechanism ?
- => In real networks, presence of dense groups: communities
  - Small, dense (random) networks have high density.
  - Large networks could be interpreted as aggregation of smaller, denser networks, with much fewer edges between them

SOME HISTORY

- The *graph partitioning problem* was a classic problem in graph theory

- It goes like this:
  - How to split a network in **k** <u>equal</u> parts such that there is a <u>minimal</u> number of edges between parts.
  - It was one problem among many others
  - Variants were proposed:
    - What if partitions are not exactly same size ?
    - What if the number of parts is not exactly k ?
    - …

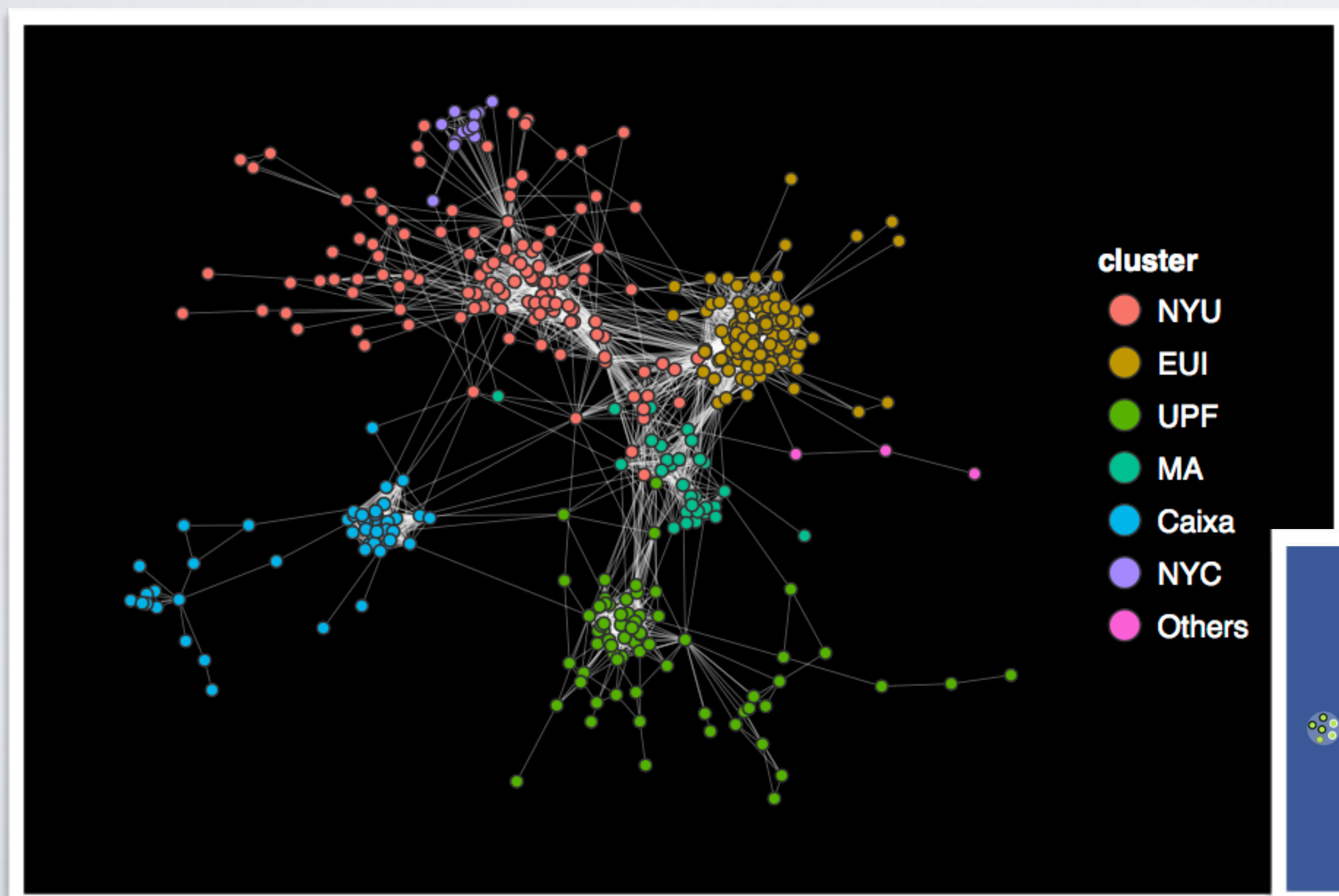
SOME HISTORY

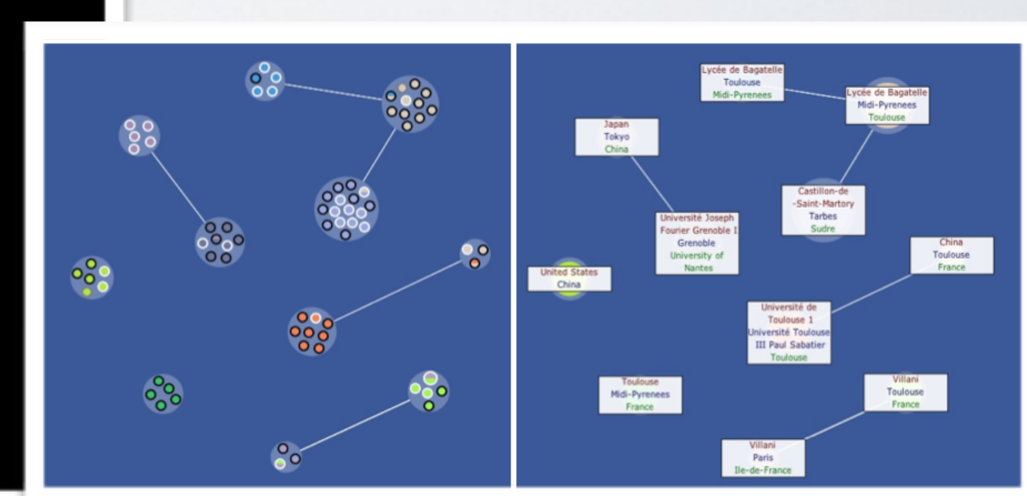
- Then in 2002, [Girvan & Newman 2002], introduction of the problem of "community discovery":
  - Observation that social networks are very often composed of groups
  - The number and the size of these groups is not known in advance
  - Can we design an algorithm to discover automatically those groups ?

Girvan, Michelle, and Mark EJ Newman. "Community structure in social and biological networks." *Proceedings of the national academy of sciences* 99.12 (2002): 7821-7826.

COMMUNITY STRUCTURE IN REAL GRAPHS

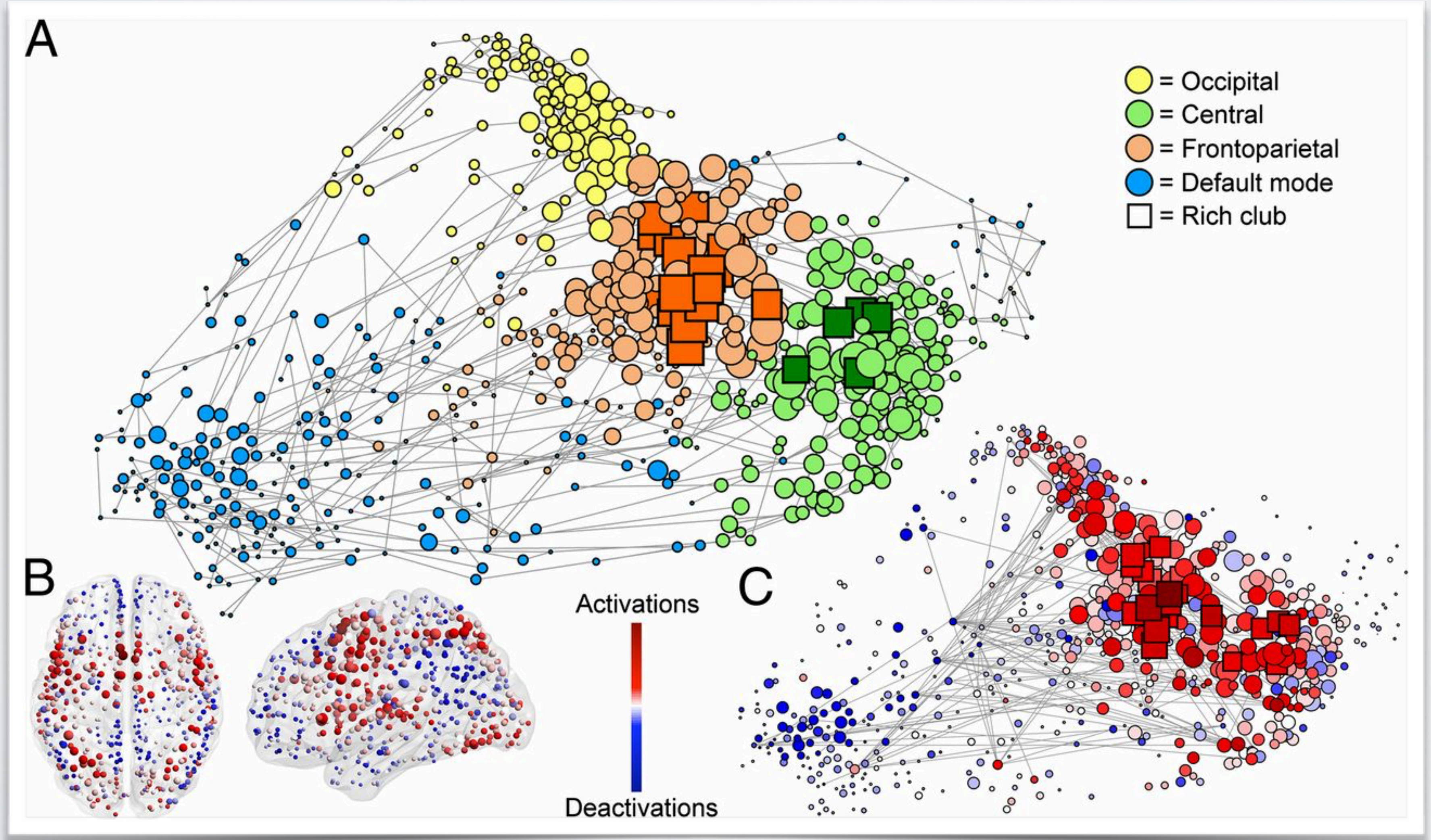
- If you plot the graph of your facebook friends, it looks like this





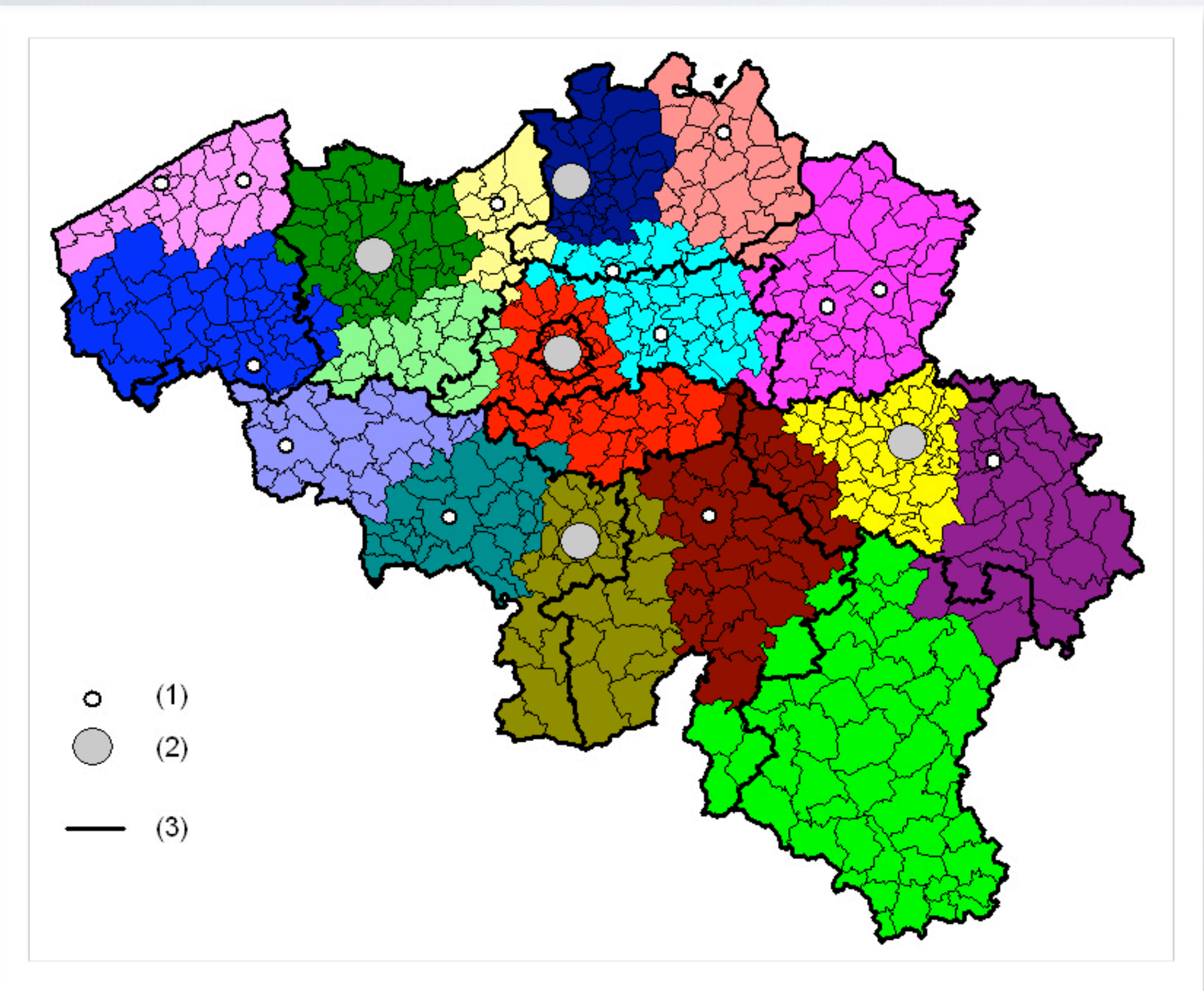
COMMUNITY STRUCTURE IN REAL GRAPHS

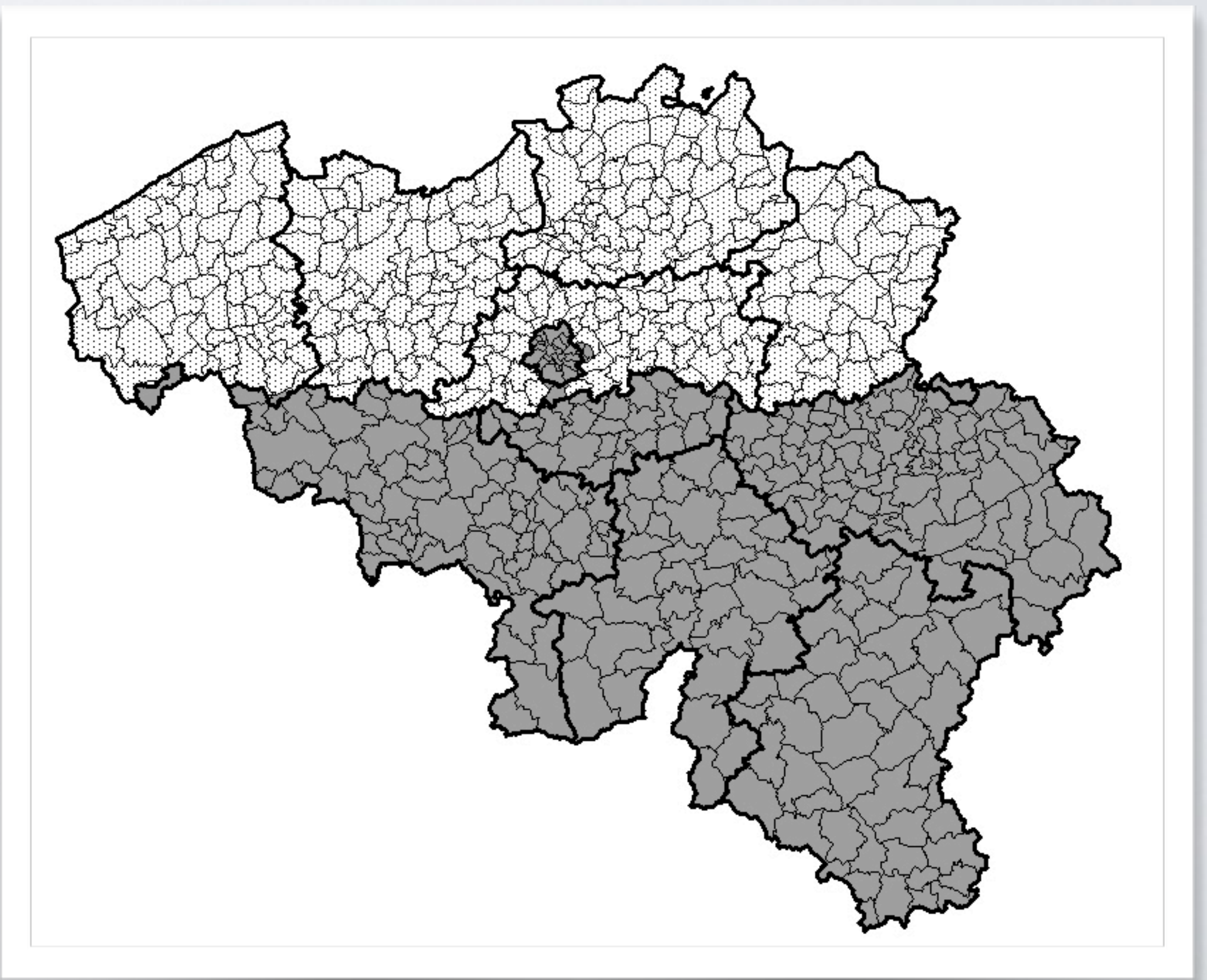
- Connections in the brain ?



COMMUNITY STRUCTURE IN REAL GRAPHS

- Phone call communications in Belgium ?

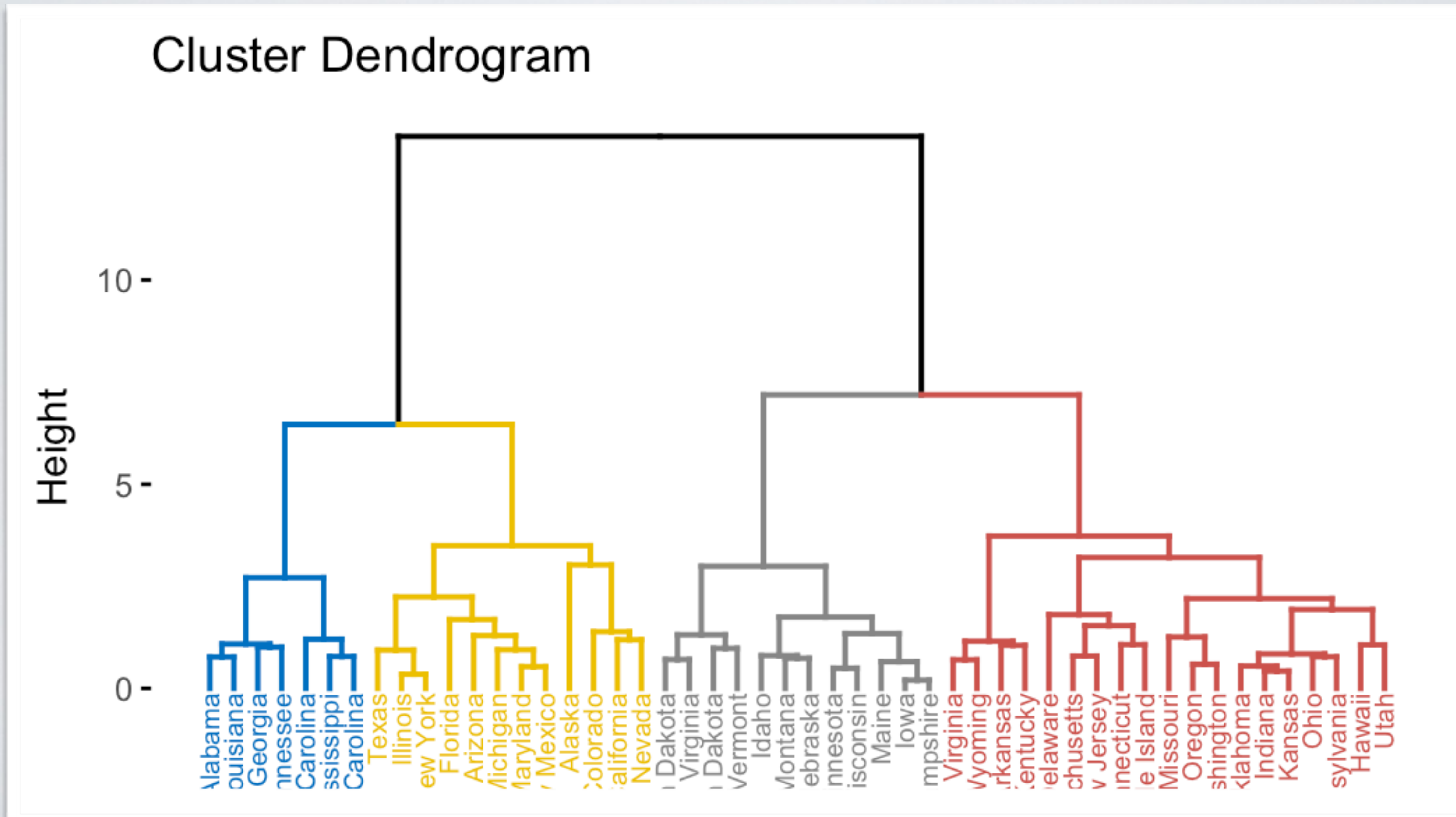




FIRST METHOD BY GIRVAN & NEWMAN

- 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

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

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 random network

MODULARITY

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

$$e_{ij} = \sum_{vw} \frac{A_{vw}}{2m} 1_{v \in c_i} 1_{w \in c_j}$$

$$a_i = \frac{k_i}{2m} = \sum_j e_{ij}$$

MODULARITY

- One point to note:
  - Number of edges in a **random** network: what type of random network ?
- Original (and still mostly used) null model for modularity:
  - The **Configuration model**, or *degree preserving* random model
  - The degrees of nodes is conserved.
  - Multi-edges and loops are allowed (for practical reasons)
- No trivial solution:
  - Too many/too few communities: comparable to a random model
- 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 !

MODULARITY OPTIMIZATION

- From 2004 to 2008: The golden age of Modularity
- Scores of methods proposed to optimize it
  - Graph spectral approaches
  - Meta-heuristics approches (simulated annealing, multi-agent…)
  - Local/Gloabal approaches…
- => 2008: the Louvain algorithm

LOUVAIN ALGORITHM

- Simple, greedy approach
  - Easy to implement
  - Extremely fast
- Yields a hierarchical community structure
- Beats state of the art on all aspects (when proposed)
  - Speed
  - Max modularity obtained
  - Do not fall in some traps (see later)

LOUVAIN ALGORITHM

- 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*
  - Global Modularity = sum of modularities of each community

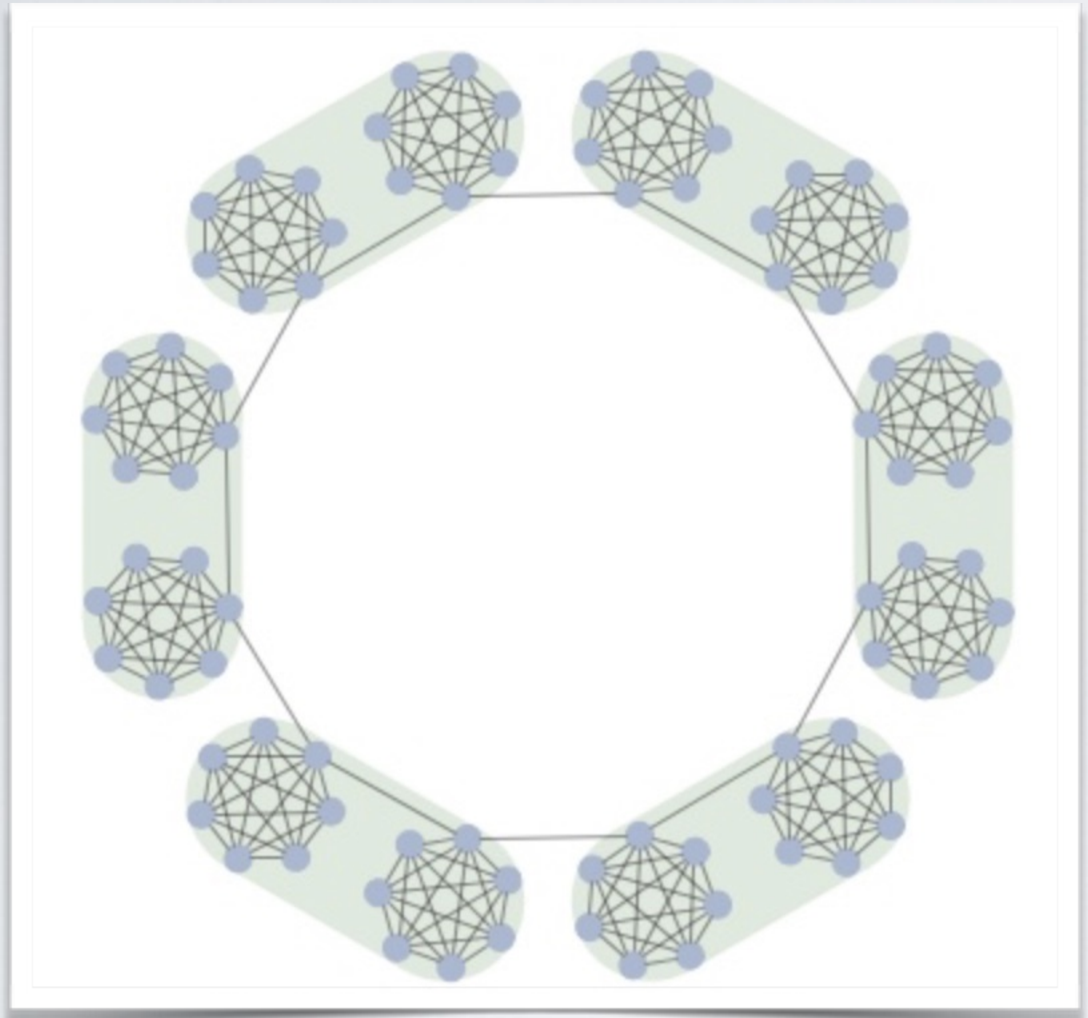
Blondel, Vincent D., et al. "Fast unfolding of communities in large networks." *Journal of statistical mechanics: theory and experiment* 2008.10 (2008): P10008.

RESOLUTION LIMIT

- Modularity == Definition of good communities ?
- 2006-2008: series of articles [Fortunato,Lancicchinetti,Barthelemy]
    - Resolution limit of Modularity
- => Modularity has intrinsic flaws, it is only *one* measure of the quality of communities
- Let's see an example

Fortunato, Santo, and Marc Barthelemy. "Resolution limit in community detection." *Proceedings of the national academy of sciences* 104.1 (2007): 36-41.

RESOLUTION LIMIT



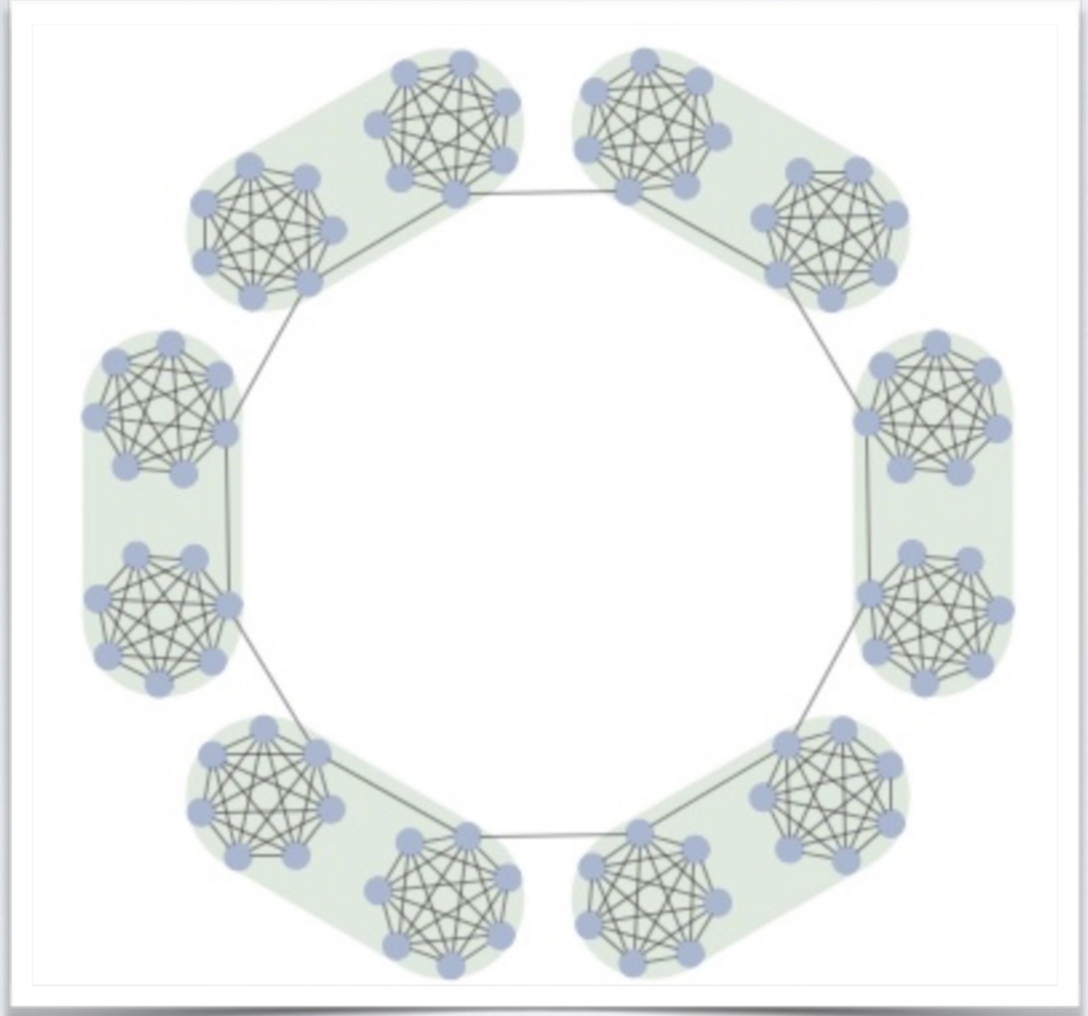
Let's consider a ring of cliques

Cliques are as dense as possible

Single edge between them:
=>As separated as possible

Any acceptable algorithm=>Each clique is a community

RESOLUTION LIMIT



But with modularity:

Small graphs=> OK

Large graphs=>
The max of modularity obtained by merging cliques

RESOLUTION LIMIT

- Discovery that Modularity has a “favorite scale”:
- For a graph of given **density** and **size:**
  - Communities cannot be smaller than a fraction of nodes
  - Communities cannot be larger than a fraction of nodes
- Modularity optimisation will never discover
  - Small communities in large networks
  - Large communities in small networks

RESOLUTION LIMIT

- Multi-resolution modularity

$$\sum_{i}^{c} e_{ii} - a_i^2 \quad \Rightarrow \quad \sum_{i}^{c} e_{ii} - \lambda a_i^2$$

λ = Resolution parameter

More a patch than a solution…

OTHER WEAKNESSES

- Modularity has other controversial/not-intuitive properties:
  - Global measure => a difference in one side of the network can change communities at the other end (imagine a growing clique ring…)
  - Unable to find no community:
    - Network without community structure: Max modularity for partitions driven by random noise
- To this day, Louvain and modularity still most used methods
  - Results are usually "good"/useful

ALTERNATIVES

- 1000+ Algorithms published, 2+ more every month (not an exaggeration)
- What unfortunately many methods still do:
  - They define their own criteria of good communities without being grounded on existing literature
  - They show empirically on a few networks using a single validation method that their method is better than Louvain (10y.o. algorithm)
- Common saying: "no algorithm is better than other, it depends on the network" (I don't really agree)

ALTERNATIVES

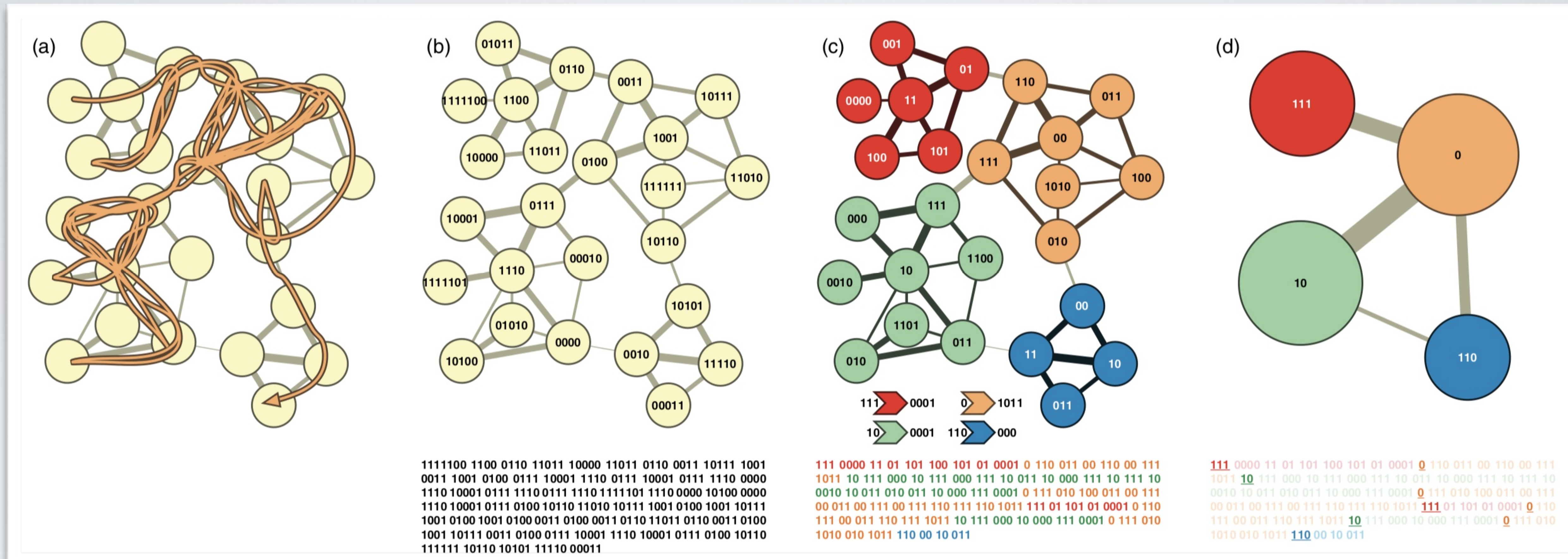
- Most serious alternatives (in my opinion)
  - Infomap (based on information theory —compression)
  - Stochastic block models (bayesian inference)
- These methods have a clear definition of what are good communities. Theoretically grounded
- Most other methods are ad hoc=>They define a process, without a clear definition

INFOMAP

- [Rosvall & Bergstrom 2009]
- Find the partition minimizing the *description* of any *random walk* on the network
- We want to *compress* the description of random walks

Rosvall, Martin, and Carl T. Bergstrom. "Maps of random walks on complex networks reveal community structure." *Proceedings of the National Academy of Sciences* 105.4 (2008): 1118-1123.

INFOMAP



Random walk

Description Without Communities

With communities

Huffman coding: short codes for frequent items

Prefix free: no code is a prefi x of another one (avoid fi x length/separators)

The Infomap method

Finding the optimal partition M:

- Shannon's source coding theorem (Shannon's entropy)

for a probability distribution P = $\{p_i\}$ such that $\Sigma_i p_i = 1$, the lower limit of the per-step code-length is

$$L(\mathcal{P}) = H(\mathcal{P}) \equiv -\sum_i p_i \log p_i$$

- Minimise the expected description length of the random walk

Sum of Shannon entropies of multiple codebooks weighted by the rate of usage

probability of between modules movements of a RW, i.e. the rate of usage of the index codebook

probability of within modules movements of a RW, i.e. the rate of usage of the module codebook

$$L(\mathbf{M}) = q_{\curvearrowright} H(\mathcal{Q}) + \sum_{i=1}^{m} p^{i}_{\circlearrowright} H(\mathcal{P}^{i})$$

Expected decryption length of partition M

Entropy of movement between modules, i.e. the frequency weighted average length of codewords

Entropy of movement inside modules, i.e. the frequency weighted average length of codewords in the module codebook

Algorithm

1. Compute the fraction of time each node is visited by the random walker (Power-method on adjacency matrix)
2. Explore the space of possible partitions (deterministic greedy search algorithm - similar to Louvain but here we join nodes if they decrease the description length)
3. Refine the results with simulated annealing (heat-bath algorithm)

INFOMAP

- To sum up:
  - Infomap defines a *quality function* for a partition different than modularity
  - Any algorithm can be used to optimize it (like Modularity)
- Advantage:
  - Infomap can recognize random networks (no communities)
  - It is nearly as fast as Louvain
- Drawback:
  - It seems to suffer from a sort of resolution limit
    - Variants: hierarchical, overlapping, etc.

STOCHASTIC BLOCK MODELS

- Stochastic Block Models (SBM) are based on statistical models of networks
- They are in fact more general than usual communities.
- The model is:
  - Each node belongs to 1 and only 1 community
  - To each pair of communities, there is an associated density (probability of each edge to exist)

Stochastic block models

Parameters:

k : scalar denoting the number of blocks/groups/communities in the network

z : a $n \times 1$ vector where $z(l)$ describes the block index for node l

M : a $k \times k$ stochastic block matrix, where M_{ij} gives the probability that nodes of type i are connected to nodes of type j (where i and j are indexes of modules)

Generating networks

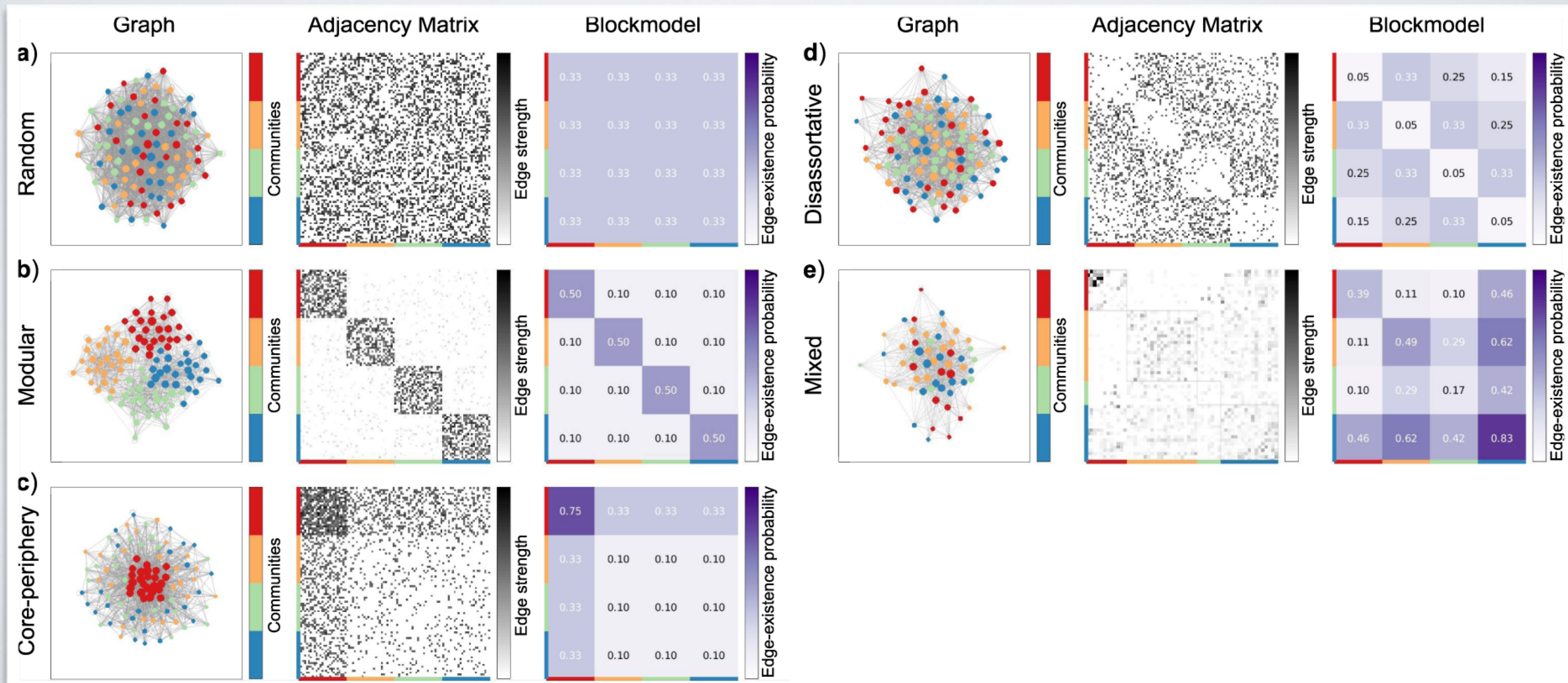
1. Take N disconnected nodes
2. Connect each $u,v \in V$ nodes with probability $M_{z(u),z(v)}$

Properties:

- Every vertices in a same module are statistically equivalent
- Vertices in a module are connected by a random graph
- Emergent degree distribution is a combination of Poisson distributions

STOCHASTIC BLOCK MODELS

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



STOCHASTIC BLOCK MODELS

- SBM can represent different things:
  - Associative SBM: density inside nodes of a same communities >> density of pairs belonging to different communities.
- This is very powerful and potentially relevant
- Problem: Often hard to interpret in real situations.
  - SBM can be “constrained”: we impose that intra d.>inter d.

STOCHASTIC BLOCK MODELS

- General idea of SBM community detection:
  - Specify the desired number of cluster
  - Find parameters to optimize the maximum likelihood
    - Principle: parameters such as the probability to generate the observed network is maximal.

- Underlying idea: Communities are "random sub-networks"

- Again, question is: what type of random networks ?
  - Erdos Renyi vs Degree corrected ?
    - DG gives better results on real networks
  - Micro-canonical/canonical ensemble
    - Micro-canonical: all networks than can be generated are generated with the same probability
    - Canonical: Probability to generate different networks can be different

STOCHASTIC BLOCK MODELS

- Main weakness of SBM:
  - Number of clusters must be specified (avoid trivial solution)
- Usual approach to solve it
  - Similar to k-means in clustering: try different k and measure improvement (elbow-method)
  - Not satisfying
- [2016 Peixoto]
  - Non-parametric SBM
  - Bayesian inference
  - Minimum Description Length (MDL) (Occam's razor)

STOCHASTIC BLOCK MODELS

Bayesian Formulation

Priors: $P(k \mid e, b)P(e \mid b)P(b)$

$$P(A, k, e, b) = P(A \mid k, e, b)\underbrace{P(k \mid e, b)P(e \mid b)P(b)}_{\text{Priors}}$$

$$P(b \mid A) = \frac{P(A \mid b)}{P(A)}$$ Posterior distribution

A: adjacency matrix
k: degree sequence
e: Matrix of edges between blocks
b: partitions

Peixoto, Tiago P. "Bayesian stochastic blockmodeling." *arXiv preprint arXiv:1705.10225* (2017).

STOCHASTIC BLOCK MODELS

Information Theoretic Formulation

$$P(A,k,e,b) = 2^{-\Sigma}$$

$$\Sigma = S + L$$

$$S = -\,log_2 P(A\,|\,k,e,b)$$

bits necessary to encode the graph knowing the model

$$L = -\,log_2 P(k,e,b)$$

bits necessary to encode the model

Objective = maximize the graph compression.
-Too many communities: over-complexifying the model
-Too few communities: Harder to encode the graph, since the model provides few useful information
Occam's razor

Peixoto, Tiago P. "Bayesian stochastic blockmodeling." *arXiv preprint arXiv:1705.10225* (2017).

STOCHASTIC BLOCK MODELS

- To sum up:
  - SBM have a convincing definition of communities
  - In practice, slower than louvain/infomap
  - But more powerful
  - Can also say if there is no community
  - And also suffer from a form of resolution limit
- Less often used, but regain popularity since works by Peixoto.

EVALUATION OF COMMUNITY STRUCTURE

EVALUATION

- Two main approaches:
  - Intrinsic 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

INTRINSIC EVALUATION

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

- Community quality function
  - **Contraction**: Average in-degree $|E_{in}|/|c|$
  - **Expansion**: Average out-degree $|E_{out}|/|c|$
  - Conductance: $\frac{|E_{out}|}{|E_{out}|+|E_{in}|}$
    - Fraction of external edges

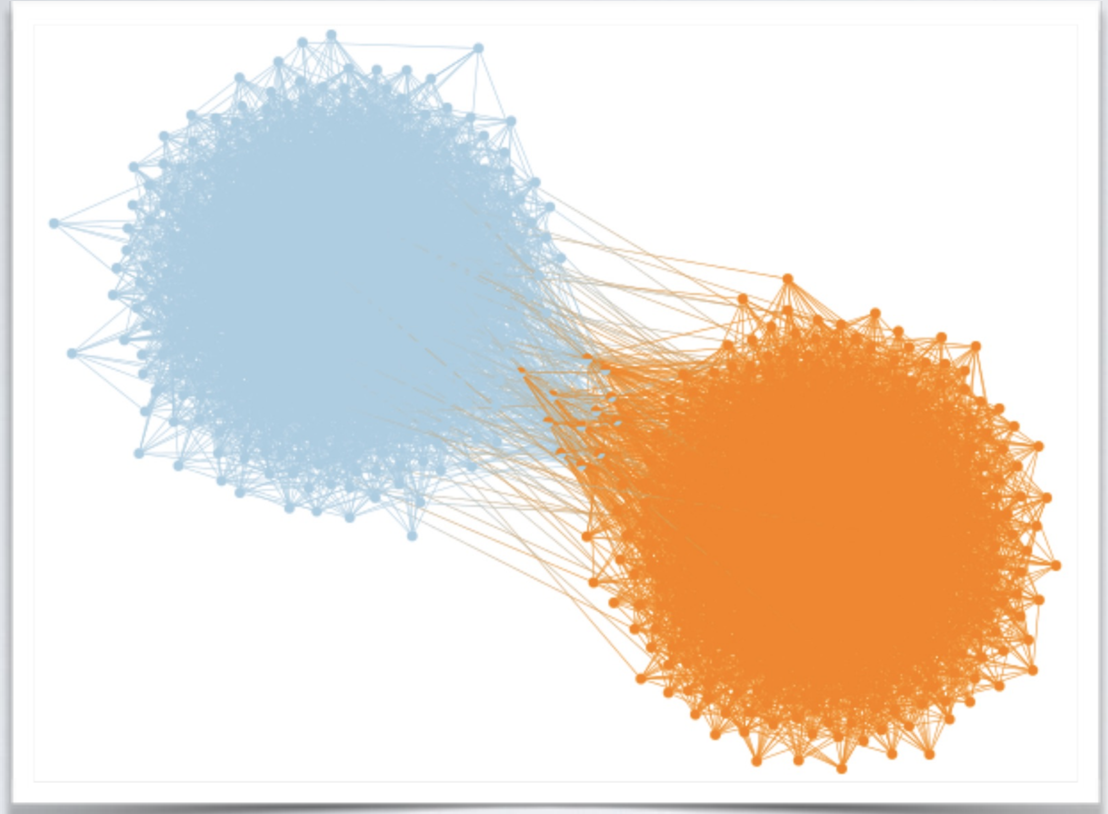
$|E_{in}|, |E_{out}|$:
# of links to nodes inside (respectively, outside) the community

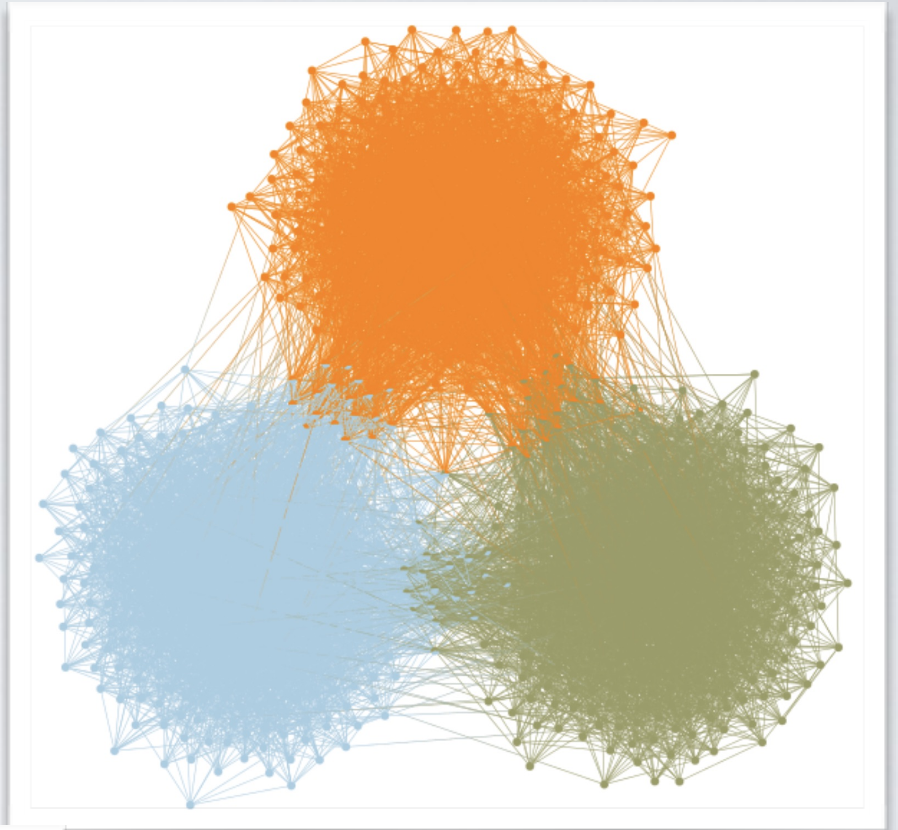
COMPARISON WITH GROUND TRUTH

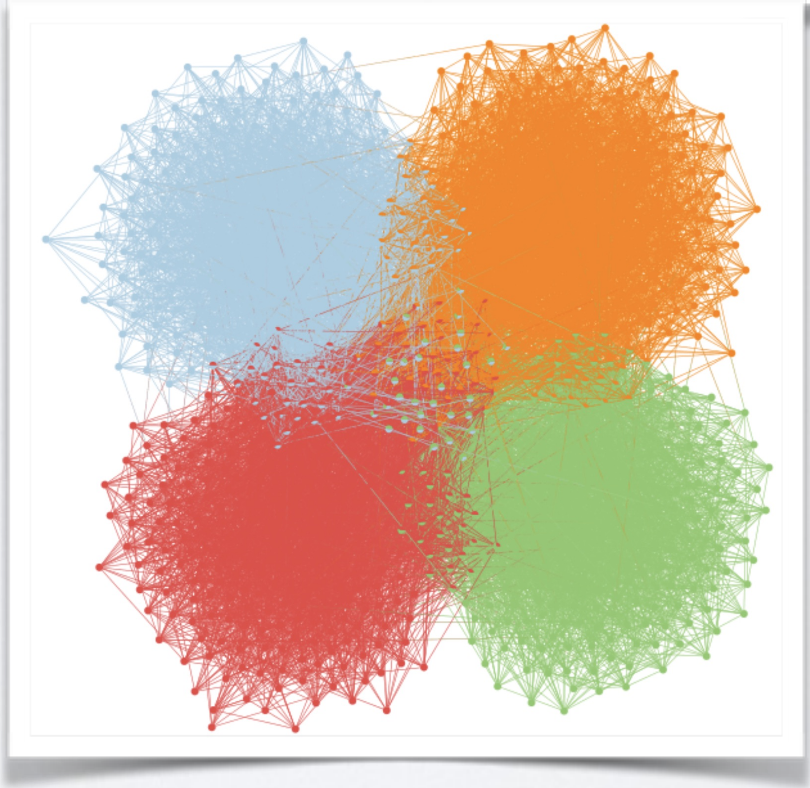
SYNTHETIC NETWORKS

- Planted Partition models:
  - Another name for SBM with manually chosen parameters
    - Assign degrees to nodes
    - Assign nodes to communities
    - Assign density to pairs of communities
    - Attribute randomly edges
  - Problem: how to choose parameters?
    - Either oversimplifying (all nodes same degrees, all communities same #nodes, all intern densities equals…)
    - Or ad-hoc process (sample values from distributions)

SYNTHETIC NETWORKS





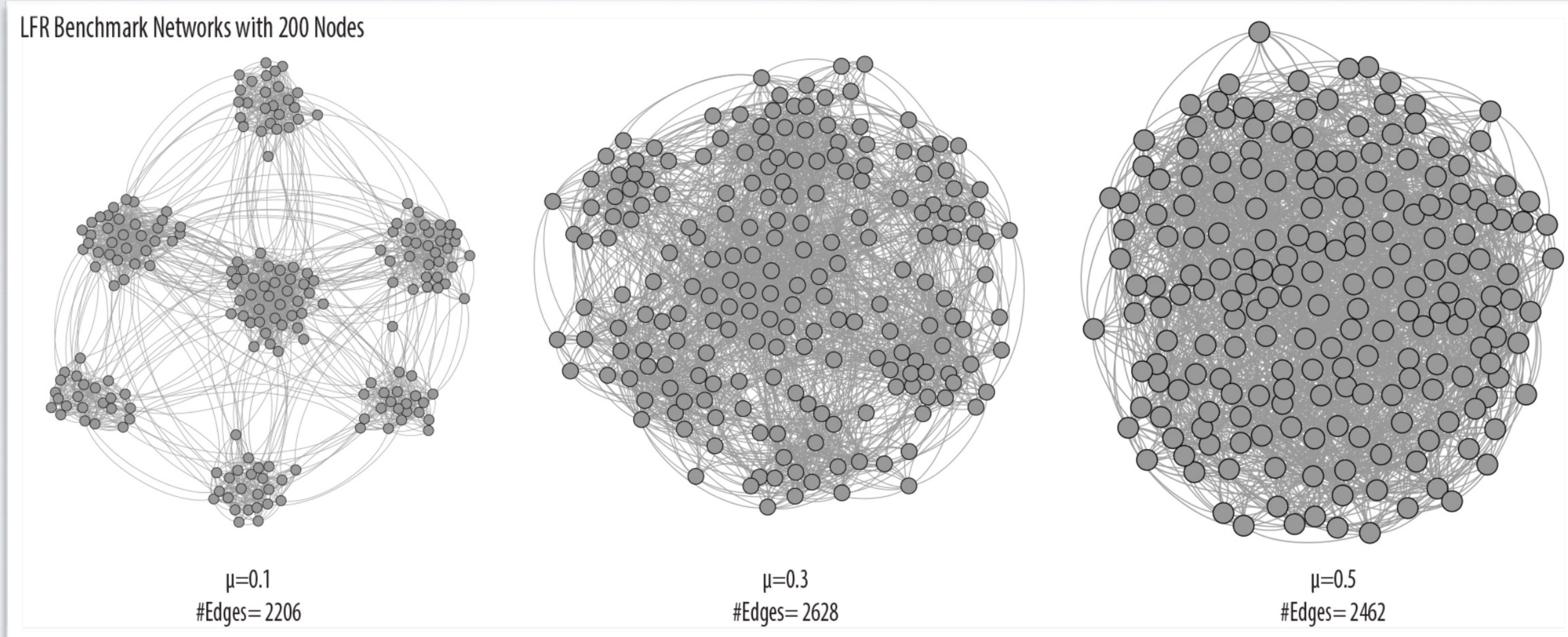


SYNTHETIC NETWORKS

- LFR Benchmark [Lancichinetti 2008]
  - High level parameters:
    - Slope of the power law distribution of degrees/community sizes
    - Avg Degree, Avg community size
    - Mixing parameter: fraction of external edges of each node
  - Varying the mixing parameter makes community more or less well defined

- Currently the most popular

SYNTHETIC NETWORKS



SYNTHETIC NETWORKS

- Pros of synthetic generators:
  - We know for sure the communities we should find
  - We can control finely the parameters to check robustness of methods
    - For instance, resolution limit…
- Cons:
  - Generated networks are not realistic: simpler than real networks
    - LFR: High CC, scale free, but all nodes have the same mixing coefficient, no overlap, …
    - SBM: depend a lot on parameters, random generation might lead to unexpected ground truth (it is *possible* to have a node with no connections to other nodes of its own community…)

REAL NETWORKS WITH GT

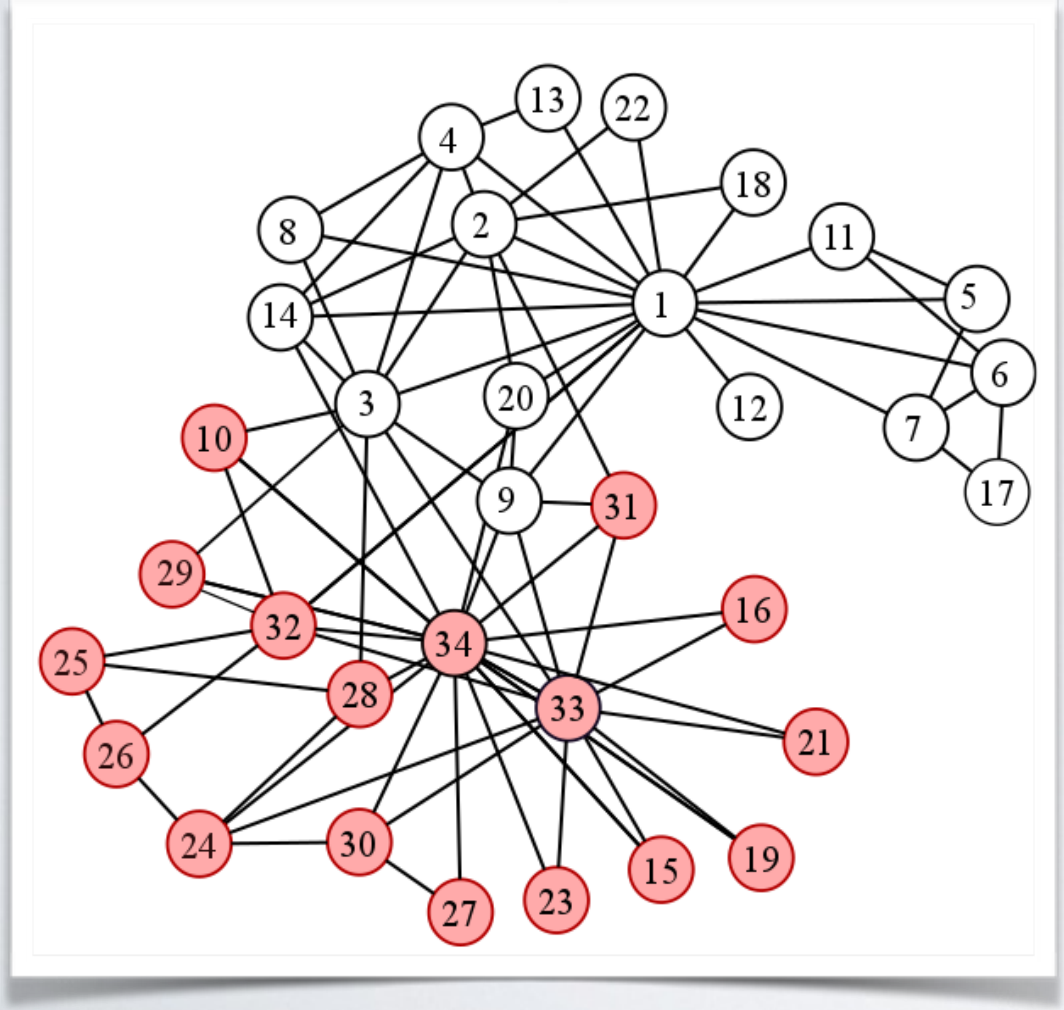
- In some networks, **ground truth** communities are known:
  - Social networks, people belong to groups (Facebook, Friendsters, Orkut, students in classes…)
  - Products, belonging to categories (Amazon, music…)
  - Other resources with defined groups (Wikipedia articles, Political groups for vote data…)
- Some websites have collected such datasets, e.g.
  - http://snap.stanford.edu/data/index.html

REAL NETWORKS WITH GT

- Pros of GT communities:
  - Retain the full complexity of networks and communities
- Cons:
  - No guarantee that communities are **topological** communities.
  - In fact, they are not: some GT communities are not even a single connected component…
- Currently, controversial topic
  - Some authors say it is non-sense to use them for validation
  - Some others consider it necessary

REAL NETWORKS WITH GT

- Example: the most famous of all networks: Zackary Karate Club

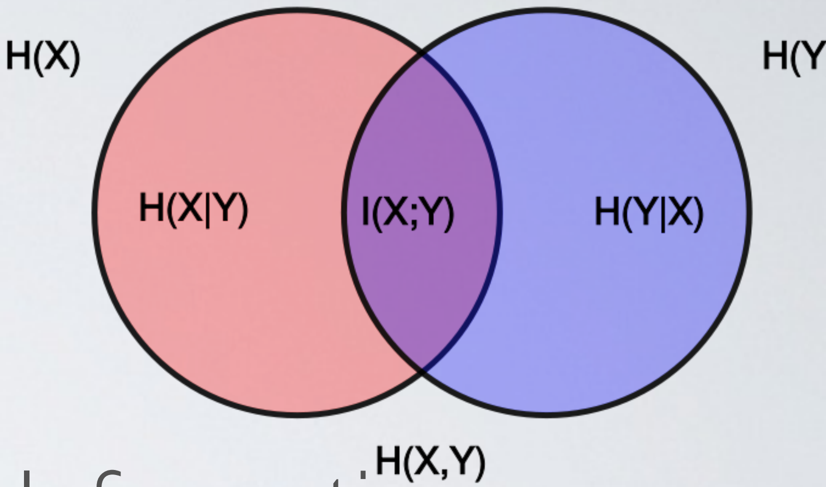


If your algorithm find the right communities,
Then it is wrong…

MEASURING PARTITION SIMILARITIES

- Synthetic or GT, we get:
  - Reference communities
  - Communities found by algorithms
- How to measure their similarity ?
  - NMI
  - aNMI
  - F1-score

MEASURING PARTITION SIMILARITIES



- NMI: Normalized Mutual Information

- Classic notion of Information Theory: Mutual Information
  - How much knowing one variable reduces uncertainty about the other
  - Or how much in common between the two variables

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log\left(\frac{p(x,y)}{p(x)\,p(y)}\right)$$

- Normalized version: NMI
  - 0: independent, 1: identical

- Adjusted for chance: aNMI

$$AMI(U,V) = \frac{MI(U,V) - E\{MI(U,V)\}}{\max\{H(U),H(V)\} - E\{MI(U,V)\}}$$

MEASURING PARTITION SIMILARITIES

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log\left(\frac{p(x,y)}{p(x)\,p(y)}\right)$$

For all pairs of clusters (y,x)

Probability for a node picked at random to belong to both x and y

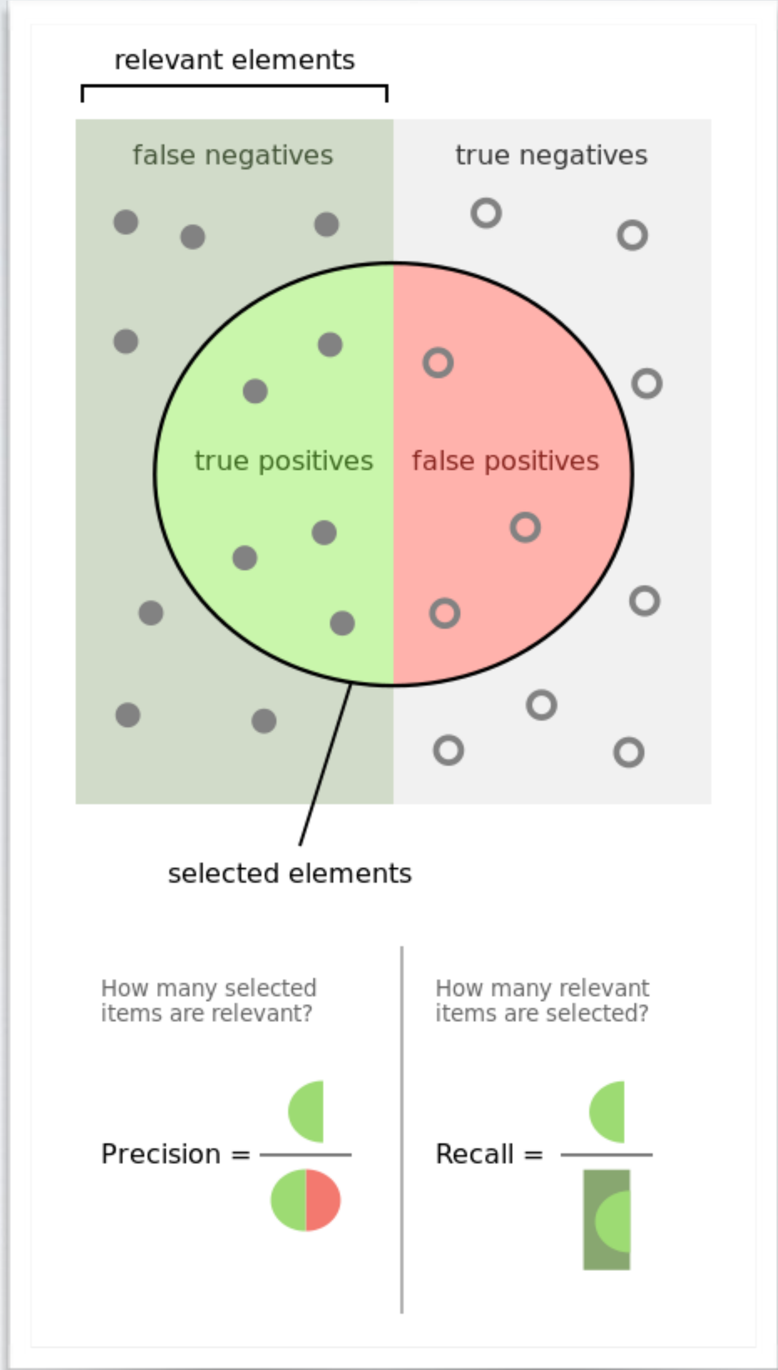
Probably for a node picked at random to belong to x

MEASURING PARTITION SIMILARITIES

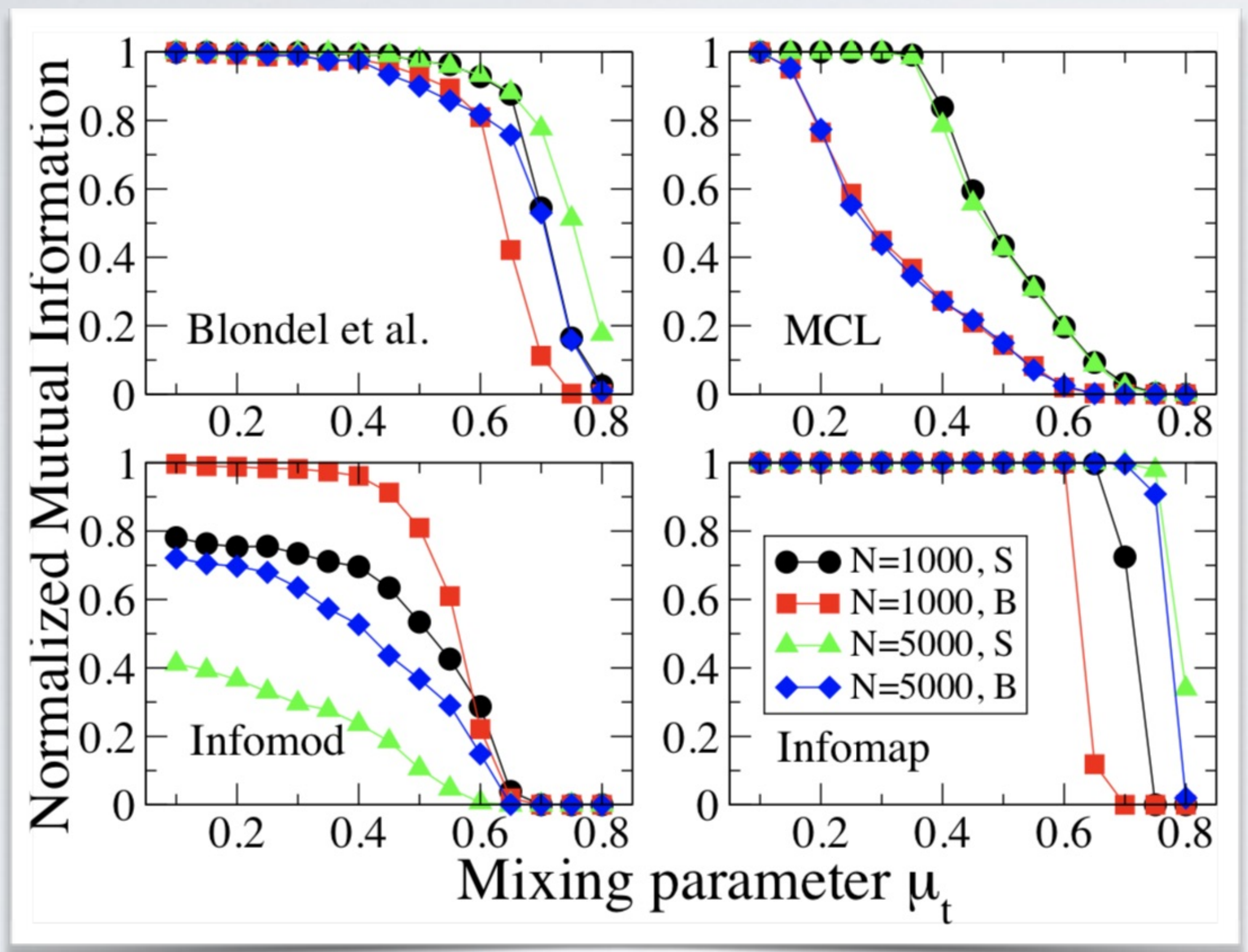
- F1-score: Borrowed from machine learning
  - Harmonic mean of Precision & Recall

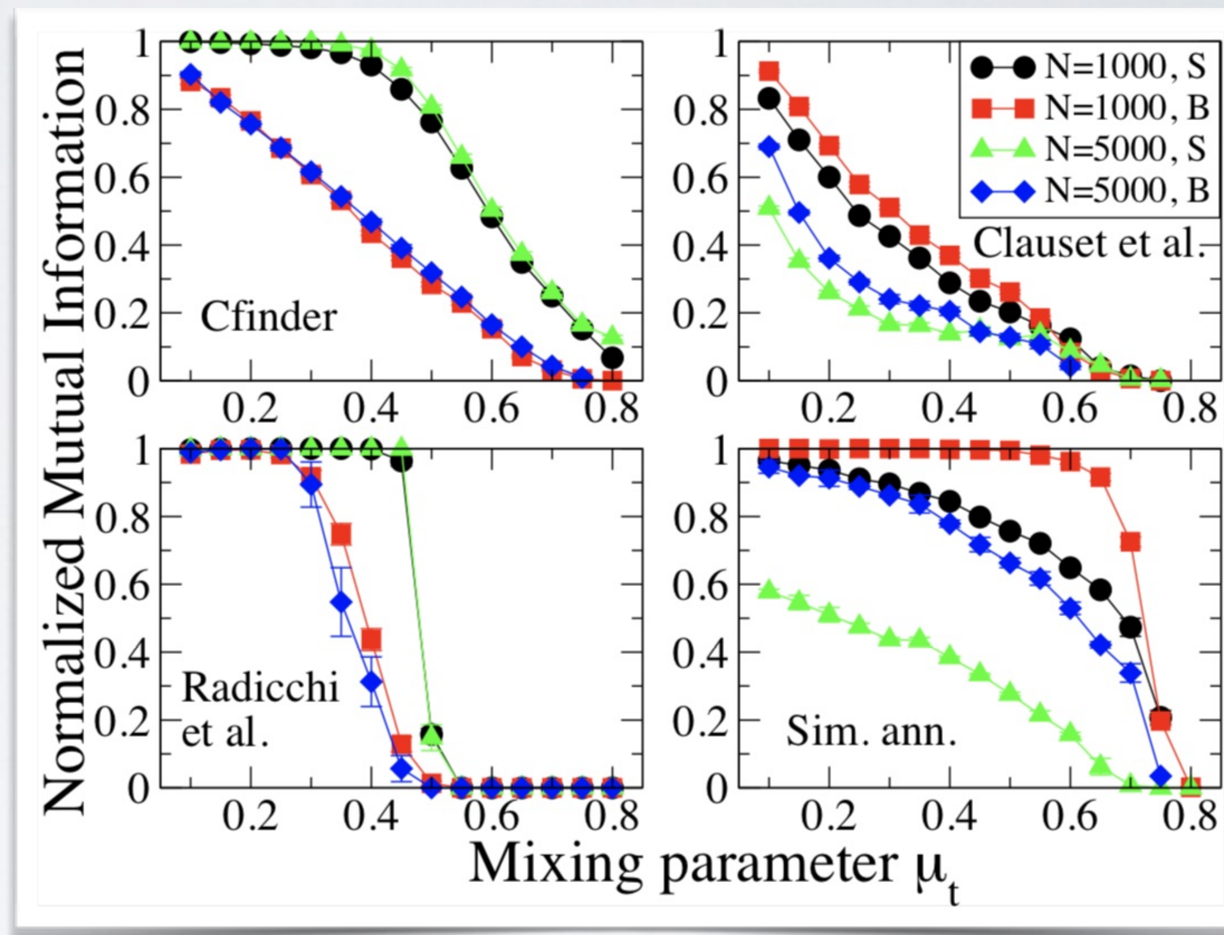
$$F_1 = \frac{2}{\frac{1}{\mathrm{recall}} + \frac{1}{\mathrm{precision}}} = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$$

Precision/Recall for Communities:
Pairs of nodes in the same clusters



ALGORITHMS COMPARATIVE ANALYSIS

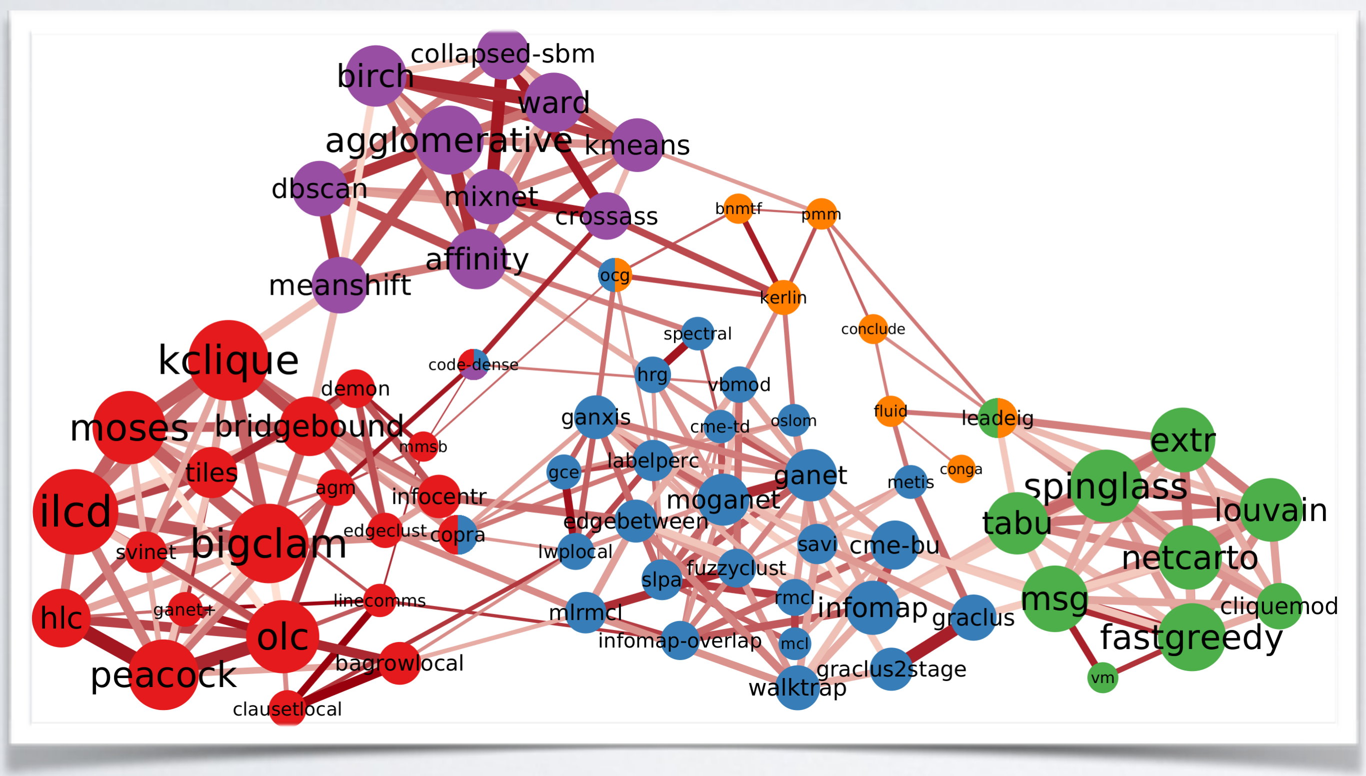




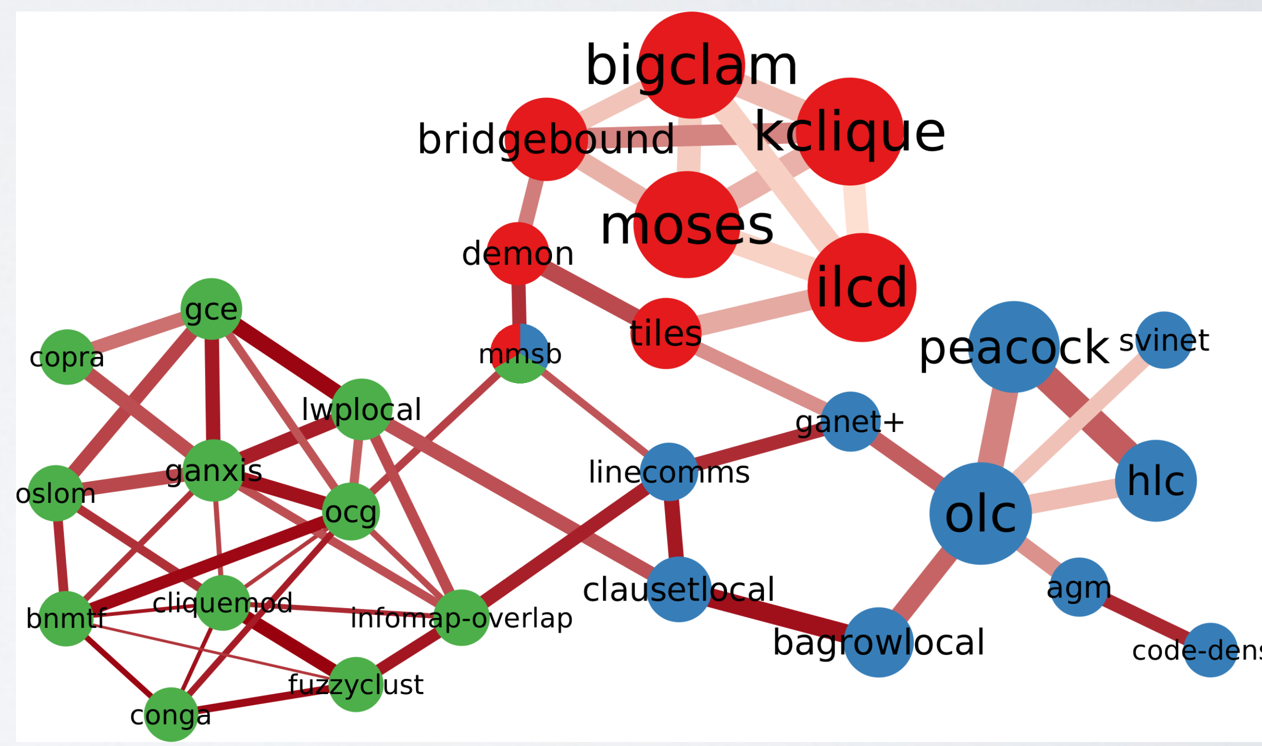
Lancichinetti, Andrea, and Santo Fortunato. "Community detection algorithms: a comparative analysis." *Physical review E* 80.5 (2009): 056117.

ALGORITHMS COMPARATIVE ANALYSIS

Rank	Algorithm	oNMI MAX
1	linecomms	165
2	oslom	73
3	infomap-overlap	64
4	savi	62
5	labelperc	57
6	rmcl	54
7	edgebetween	41
7	leadeig	41
7	vbmod	41
10	gce	32



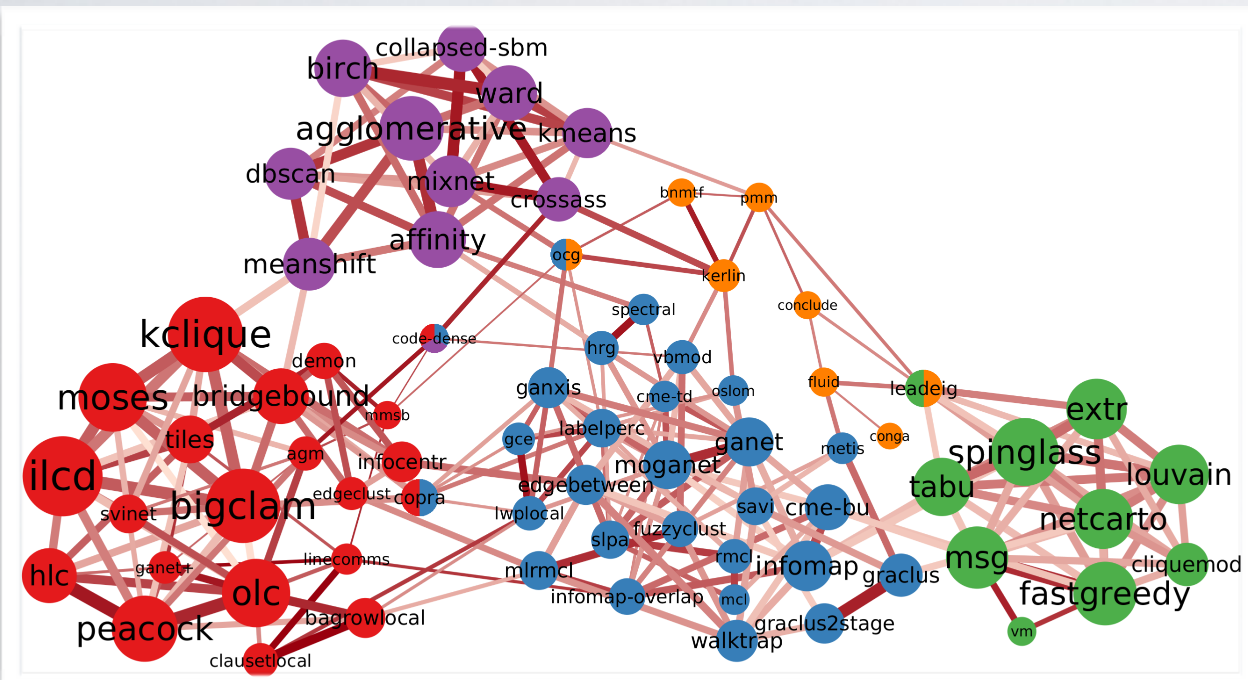
All methods



Overlapping only

Coscia, Michele. "Discovering Communities of Community Discovery." *arXiv preprint arXiv:1907.02277* (2019).

ALGORITHMS COMPARATIVE ANALYSIS



ID	Col	n	Over	Spr	Q	NSim
1	Red	21	0.9048	0.1429	0.0952	0.0952
2	Blue	28	0.3214	0.5357	0.1429	0.0357
3	Green	10	0.1000	0.0000	1.0000	0.0000
4	Purple	11	0.0909	0.0000	0.0000	0.7273
5	Orange	8	0.3750	0.2500	0.3750	0.0000

Table 1: Features of the communities of *ASN*. *n*: # of nodes. Over: % overlapping algorithms. Spr: % algorithms based either on centrality measures (including edge betweenness and random walks) or some sort of spreading process (e.g. label percolation). Q: % algorithms based on modularity maximization. NSim: % algorithms based on neighborhood similarity. Algorithms can be part of multiple/no classes, so the rows do not sum to one.

ID	Col	$\lvert\bar{C}\rvert$	Avg Size	$\bar{d}$	$\bar{Q}$	$\bar{c}$	Avg Ncut
1	Red	19.7979	9.0942	0.3220	0.2200	0.7423	0.7674
2	Blue	5.6520	16.4769	0.2627	0.1102	0.5542	0.7100
3	Green	4.8948	11.9844	0.2580	0.1118	0.6288	0.7407
4	Purple	10.3702	11.0140	0.2917	0.0333	0.7555	0.8033
5	Orange	4.2852	17.0505	0.2329	0.0863	0.5963	0.7483

Table 2: The averages of various community descriptive statistics per algorithm group. $\lvert\bar{C}\rvert$: Average number of communities. Avg Size: Average number of nodes in the communities. $\bar{d}$: Average community density. $\bar{Q}$: Average modularity – when the algorithm is overlapping we use the overlapping modularity instead of the regular definition. $\bar{c}$: Average conductance – from [24]. Avg Ncut: Average normalized cut – from [24].

NODE/COMMUNITY RELATION

- Embeddedness : $e = \frac{k_{int}}{k}$
  - (fraction of internal edges)
- Hub dominance: $h(C) = \frac{max(k_{int})}{n_c - 1}$
  - Is the community star-like?

OTHER MESO-SCALE ORGANIZATIONS

MESO-SCALE

- MACRO properties of networks:
  - degree distribution, density, average shortest path…
- MICRO properties of networks:
  - Centralities
- MESO-scale: what is in-between
  - Community structure
  - Overlapping Community Structure
  - Core-Periphery
  - Spatial Organization (another class)

OVERLAPPING COMMUNITIES

- In real networks, communities are often overlapping
  - Some of your High-School friends might be also University Friends
  - A colleague might be a member of your family
  - …
- Overlapping community detection is considered much harder
  - And is not well defined
- Difference between "attributes" and overlapping communities ?
  - Community of Women, Community of 17-19yo, Community of fans of X…

OVERLAPPING COMMUNITIES

- Many algorithms
  - Adaptations of modularity, random walks, label propagations…
  - Original methods
  - Many local methods (local criterium) compare with global optimisation for partitions

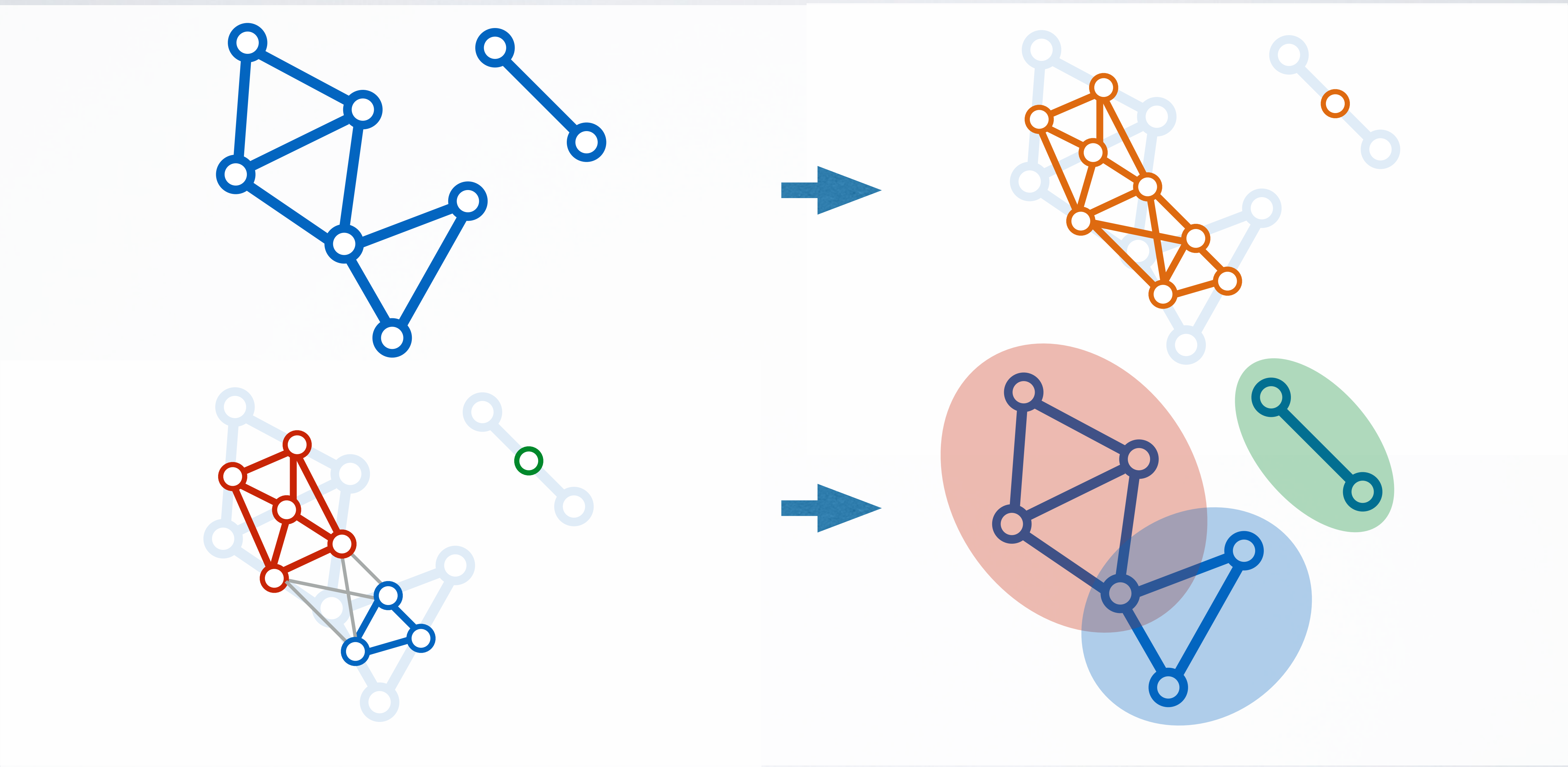
OVERLAPPING COMMUNITIES

- Motif-based definitions:
  - Cliques
    - Of a given size
    - Maximal cliques
  - N-cliques
    - Set of nodes such as there is at least a path of length <=N between them
    - Generalization of cliques for N>1
    - Computationally expensive

Link clustering - overlapping communities

Link graphs

- Links are replaced by nodes which are connected if the original links share a node

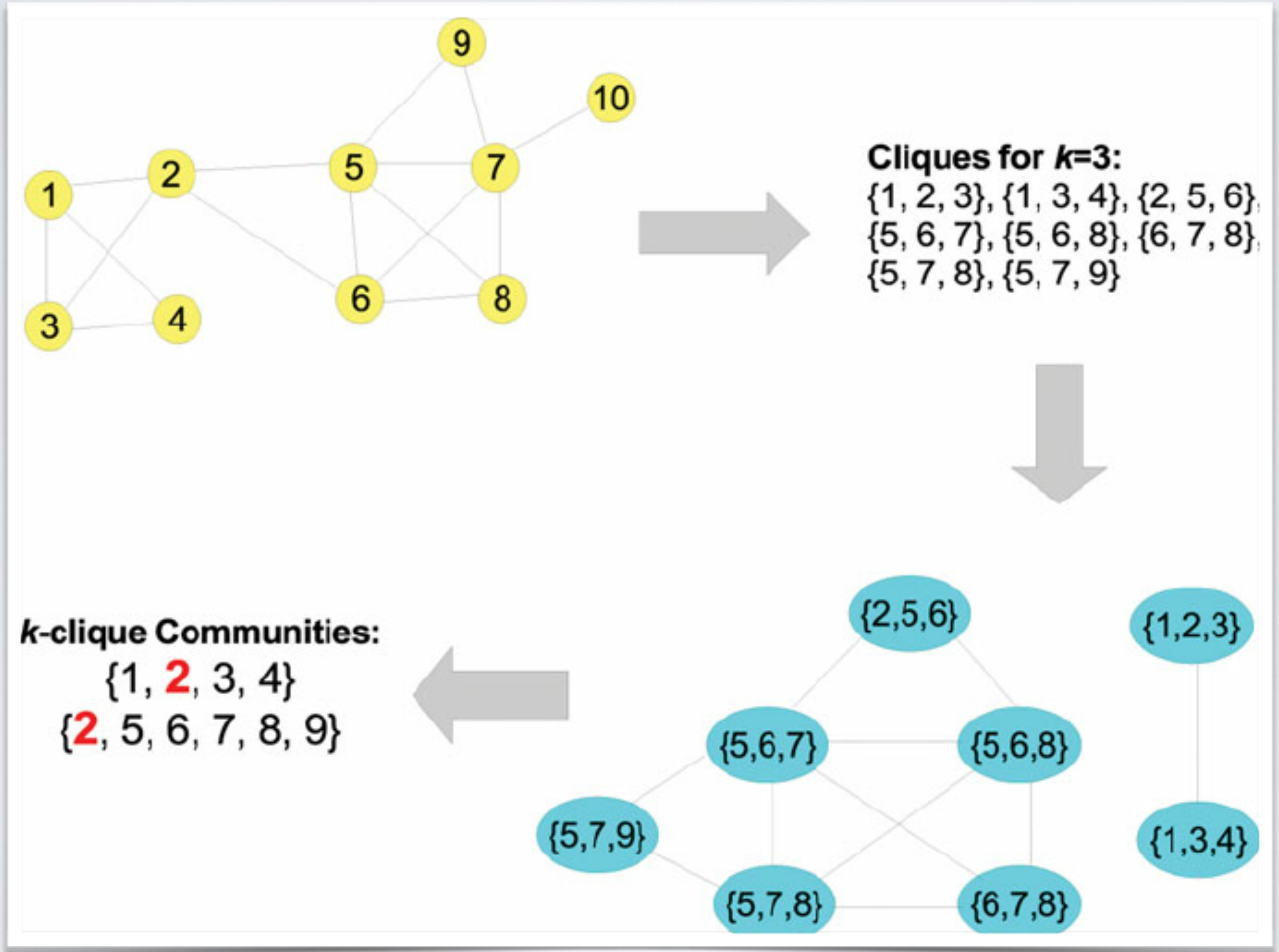


- Community detection on link graphs allows for overlapping communities

K-CLIQUE PERCOLATION

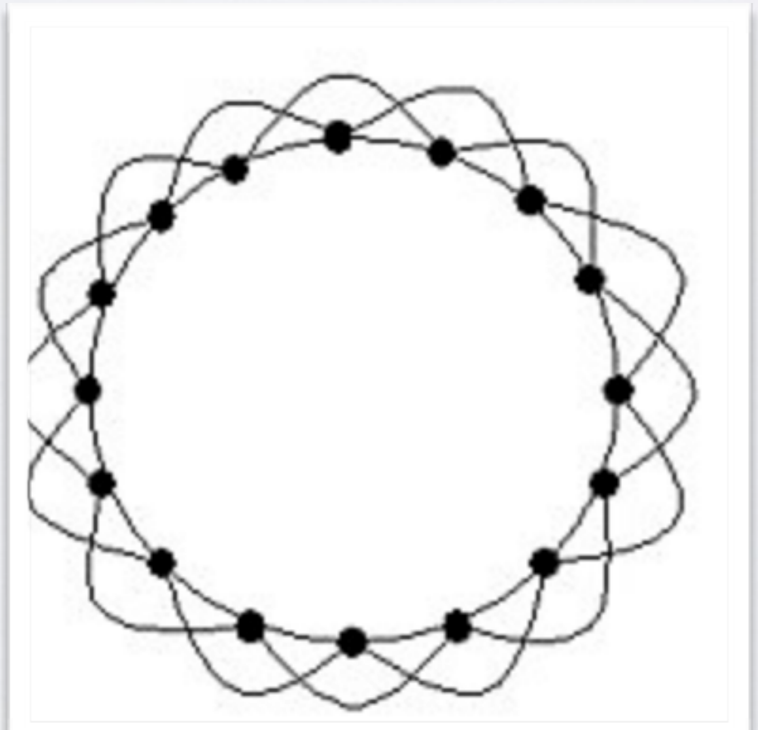
- (Other name: CPM, C-finder)
- Parameter: size k of atomic cliques
- 1)Find all cliques of size k
- 2)merge iteratively all cliques having k-1 nodes in common

K-CLIQUE PERCOLATION



K-CLIQUE PERCOLATION

- Obvious weakness: communities can be very far from random networks



OVERLAPPING COMMUNITIES

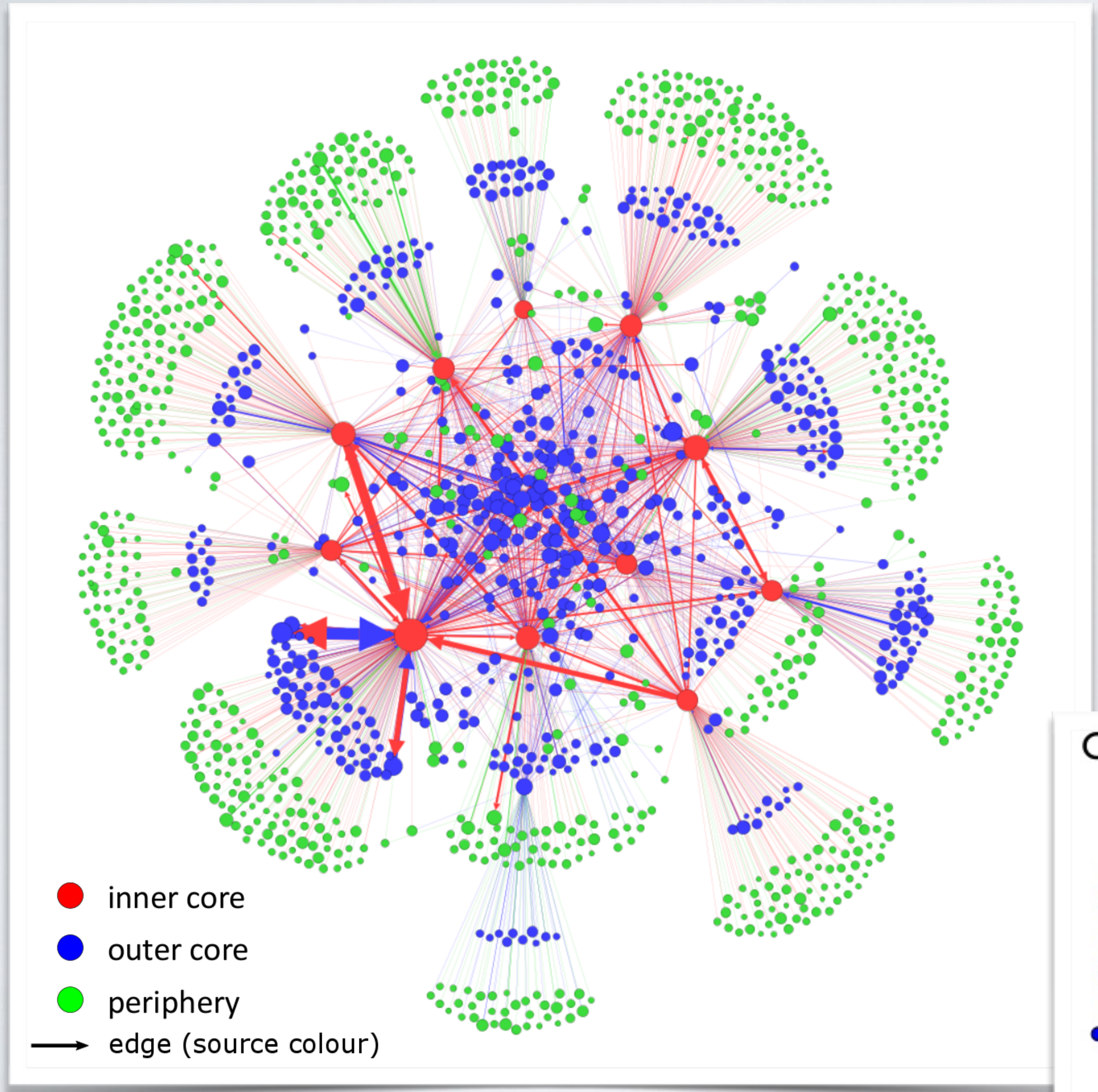
- Another general approach
- Each community is defined intrinsically.
  - Must verify a property
  - Try to add and remove randomly nodes
  - Until the property is maximized.
  - Natural overlap, low complexity
  - Problem: which property ?

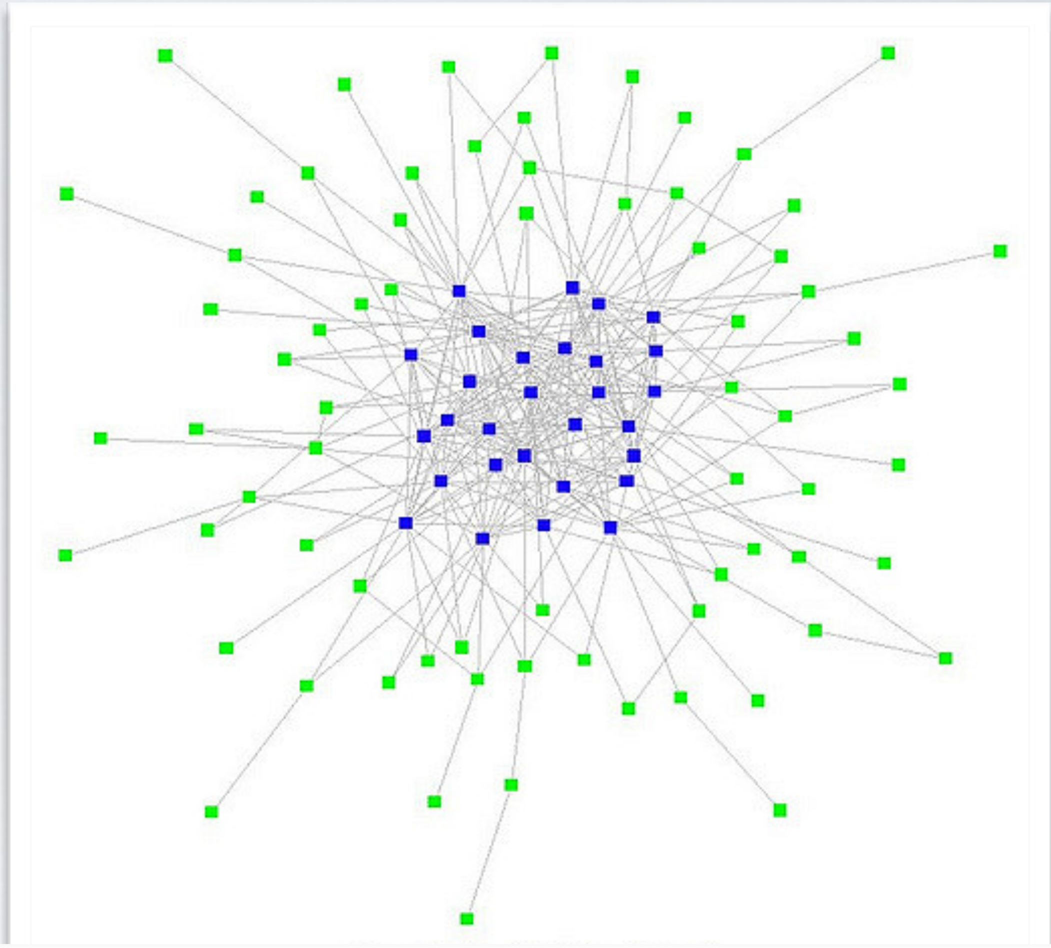
Lancichinetti, Andrea, et al. "Finding statistically significant communities in networks." *PloS one* 6.4 (2011): e18961.

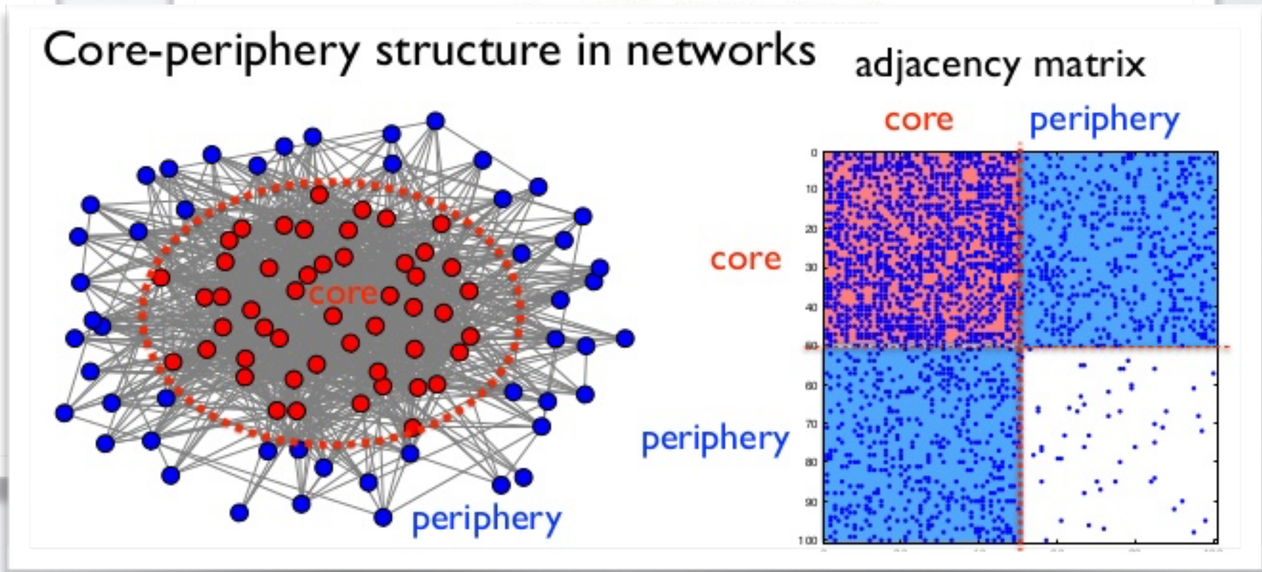
HIERARCHICAL COMMUNITIES

Lancichinetti, Andrea, et al. "Finding statistically significant communities in networks." *PloS one* 6.4 (2011): e18961.

CORE-PERIPHERY







NESTEDNESS