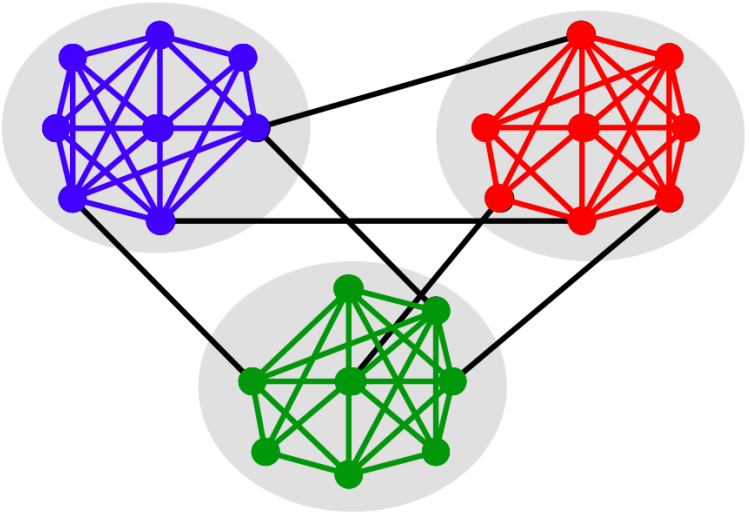
DEI
DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
TÉCNICO LISBOA

Community structure

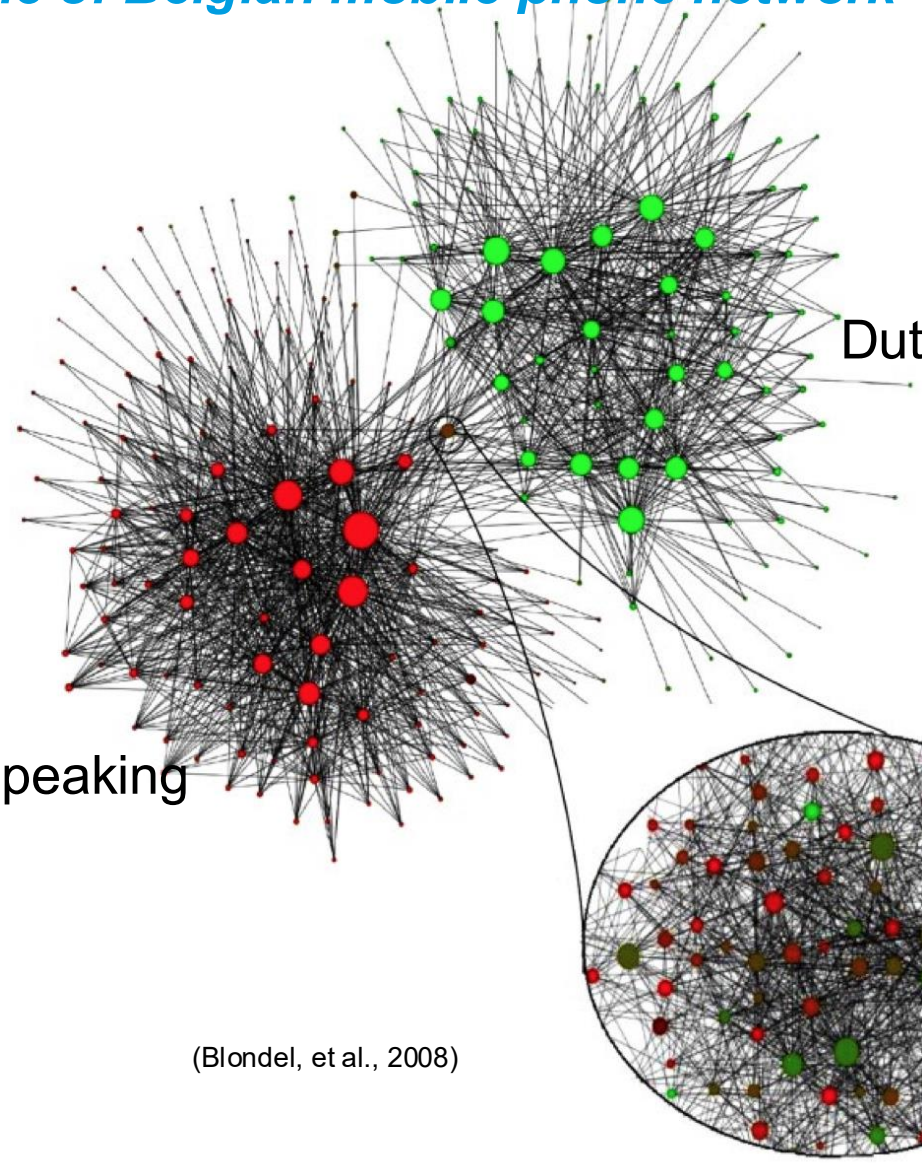
The building blocks of complex networks



Network Science, 2025/2026

Modules and communities

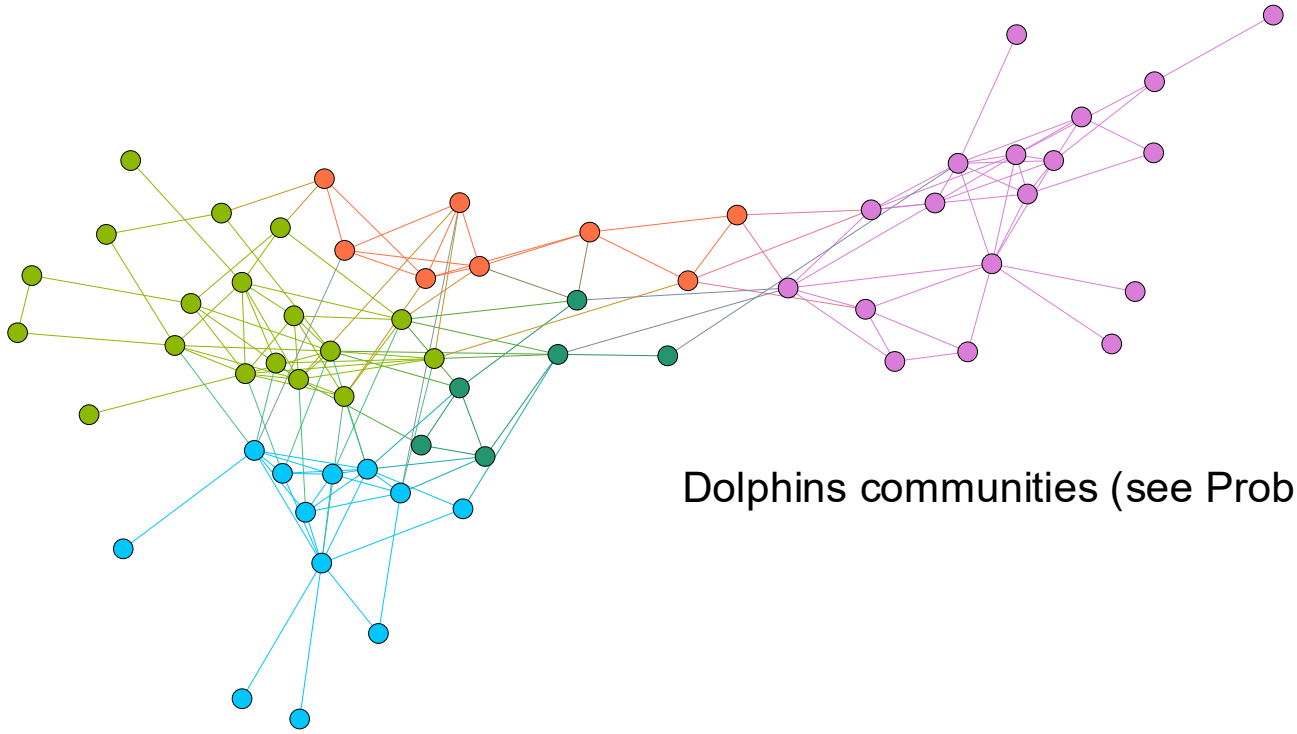
Sample of Belgian mobile phone network



(Blondel, et al., 2008)

Modules and communities

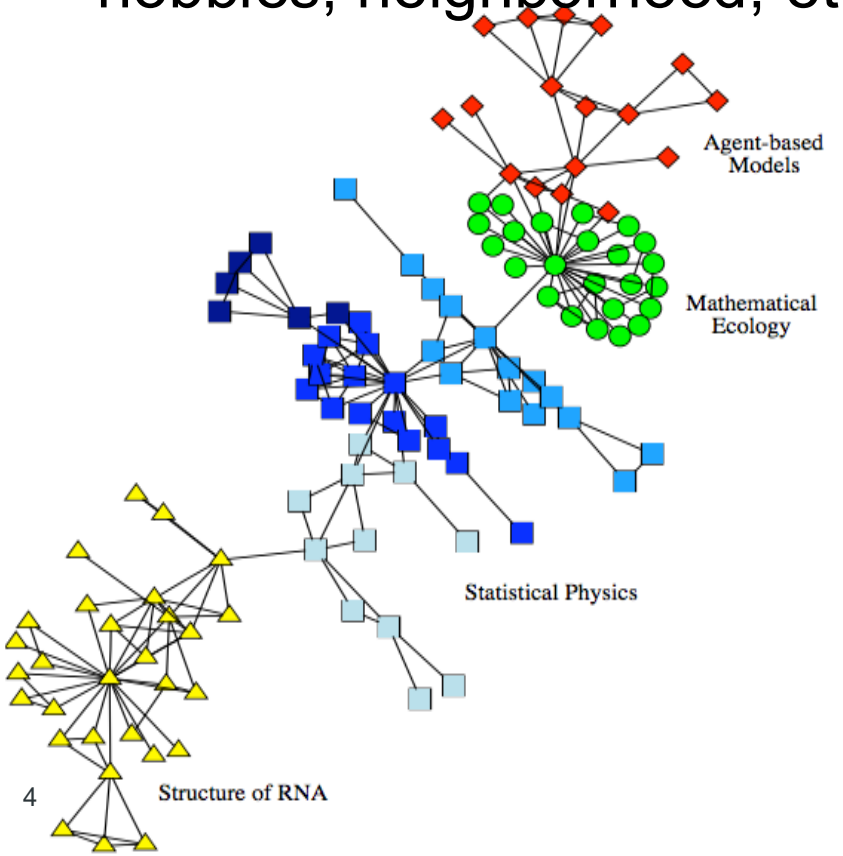
In network science we call a ***community*** a group of nodes that have a higher likelihood of connecting to each other than to nodes from other communities.



Dolphins communities (see Problem set 1)

Modules and communities

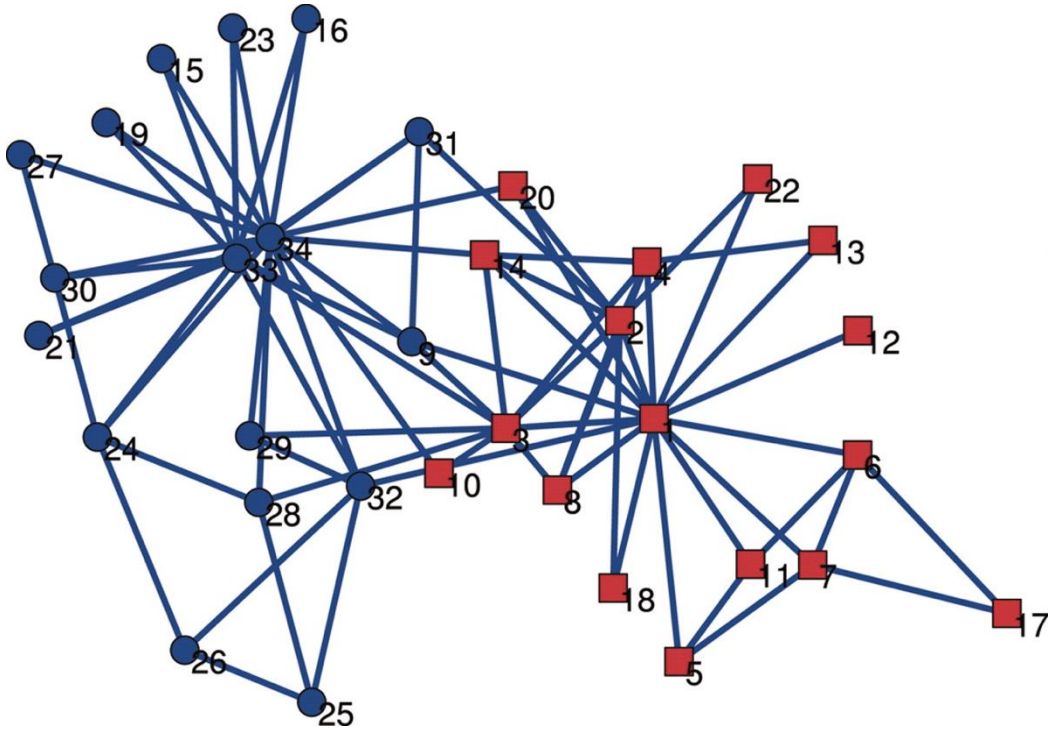
- ***Social networks*** are full of easy to spot communities related, for instance, to our friends' circles (work, family, hobbies, neighborhood, etc).



Collaboration network between scientists working at the Santa Fe Institute.

Modules and communities

- ***Social networks*** are full of easy to spot communities related, for instance, to our friends' circles (work, family, hobbies, neighborhood, etc).

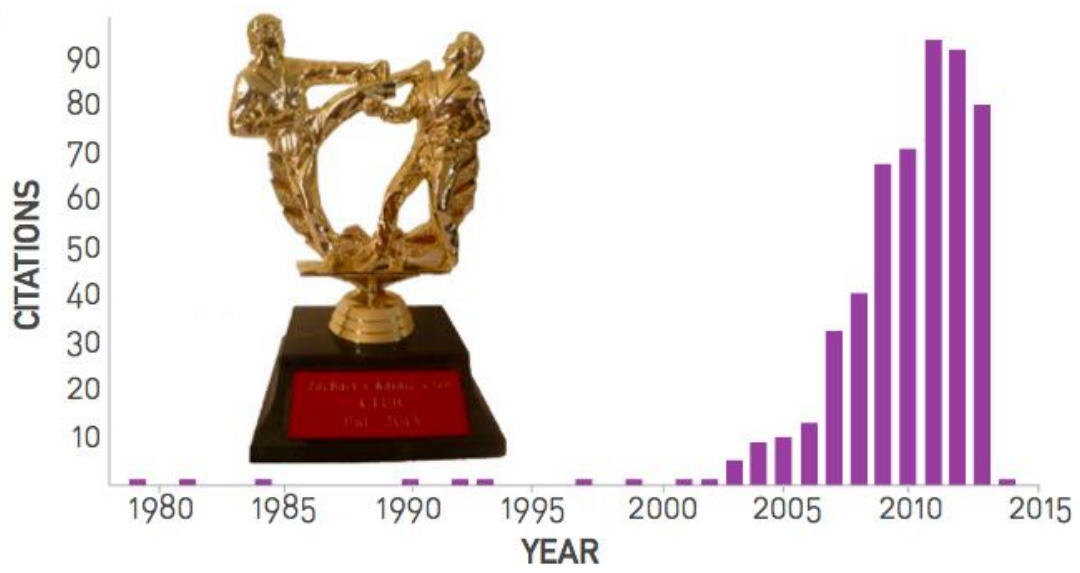


A famous dataset (70's):
The Zachary's Karate Club

Zachary recorded the split of the club in two during his observation period.

Modules and communities

- ***Social networks*** are full of easy to spot communities related, for instance, to our friends' circles (work, family, hobbies, neighborhood, etc).

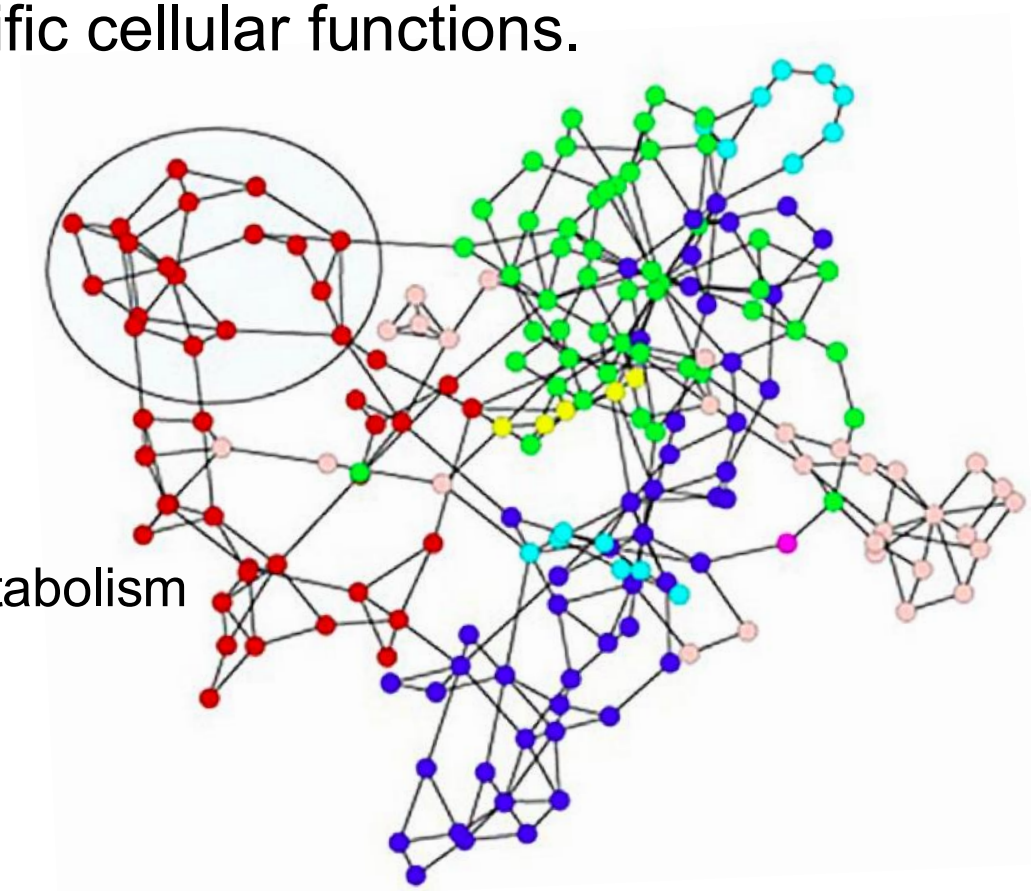


A famous dataset (70's):
The Zachary's Karate Club

The 1st scientist at any conference on networks who uses Zachary's karate club as an example is awarded a price!

Modules and communities

- ***Biological networks***: Biology is moving beyond genes focusing on how groups of molecules form functional modules to carry specific cellular functions.



E. coli metabolism

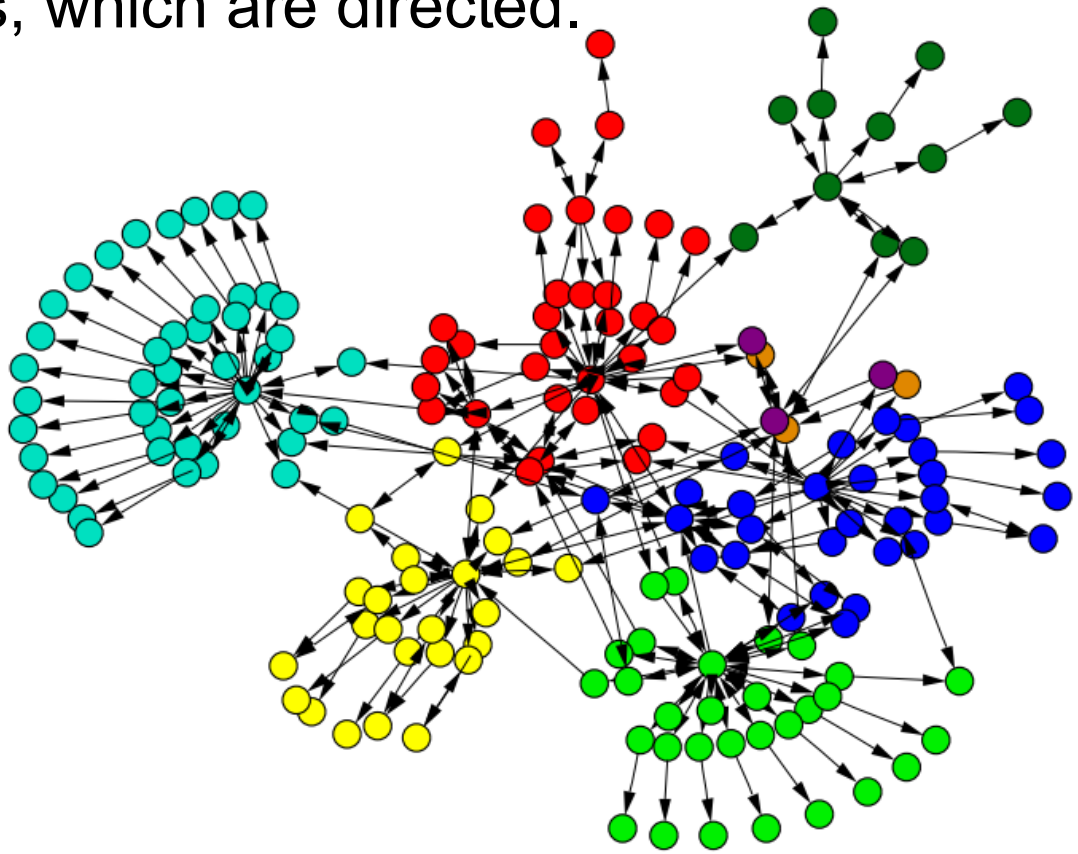
Modules and communities

- ***Biological networks***: Biology is moving beyond genes focusing on how groups of molecules form functional modules to carry specific cellular functions.

- E.g., proteins that are involved in the same diseases tend to interact with each other, inspiring researchers to seek for ***disease modules***. The idea is that each disease may be linked to a well-defined neighborhood of the cellular network.

Modules and communities

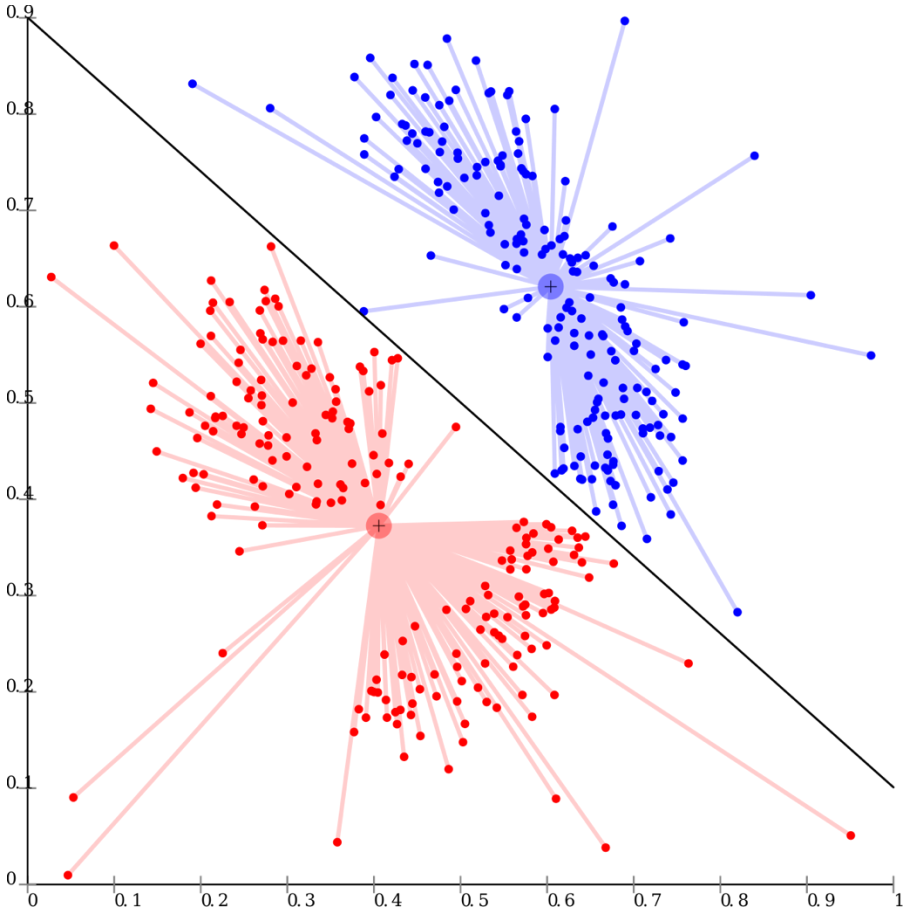
- ***Technological networks***: e.g. sample of the web graph consisting of the pages of a web site and their mutual hyperlinks, which are directed.



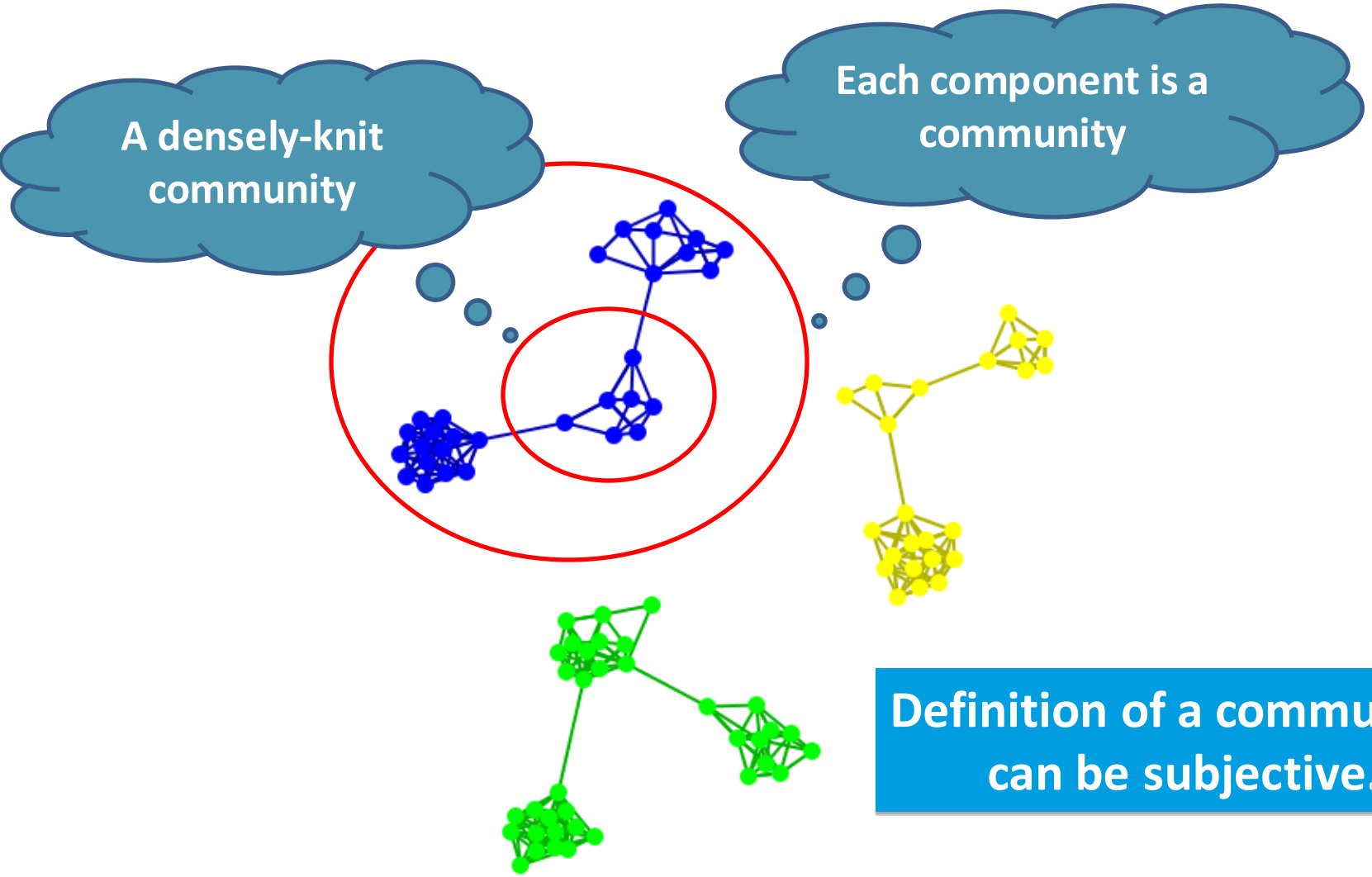
Modules and communities

You can also see it as general unsupervised clustering technique

Grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense) to each other than to those in other groups (clusters).



Subjectivity of Community Definition



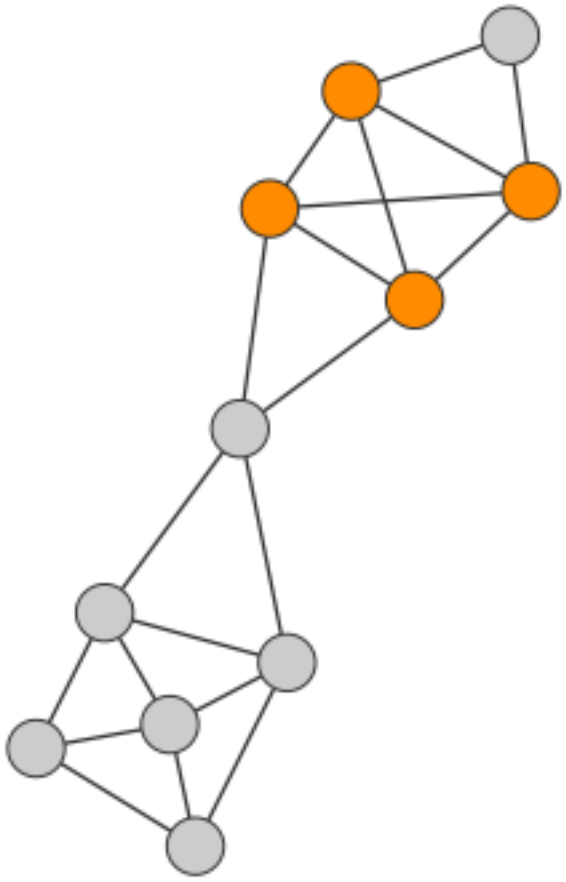
Definition of a community can be subjective.

Defining communities (examples)

Connectedness + density hypothesis

Maximum Cliques

Group of nodes with direct links to all members of the community.



Note: Here we're highlighting the highest order clique of this network (a square), yet there are several three-node cliques on this network.

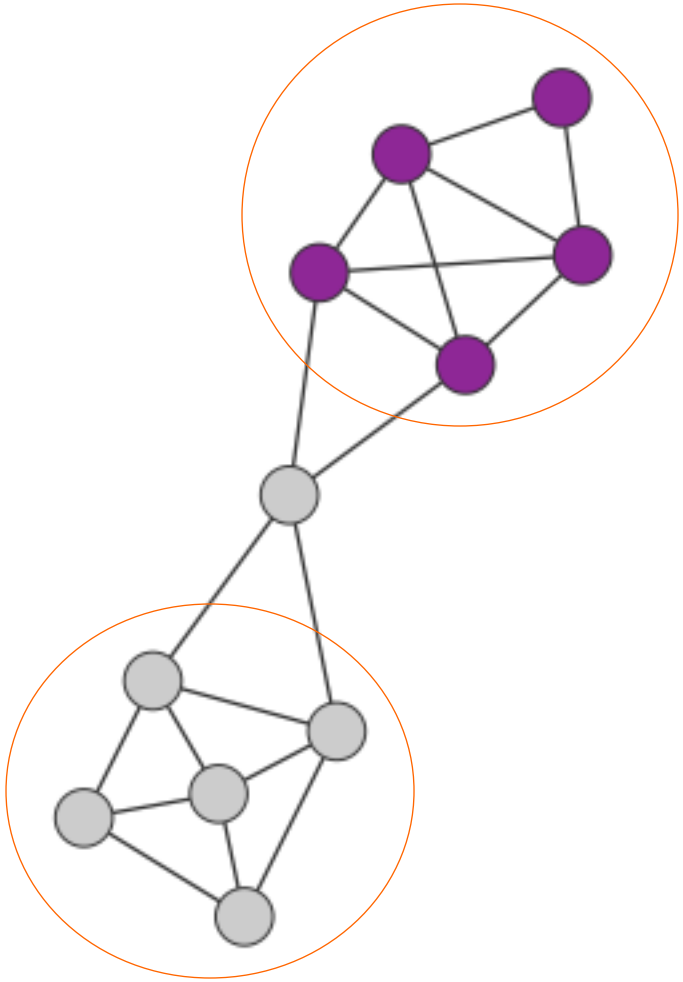
Defining communities (examples)

Connectedness + density hypothesis

Strong communities

Each node *i* in a community C has more links within the community that with the rest of the graph.

$$k_i^{\text{internal}}(C) > k_i^{\text{external}}(C)$$



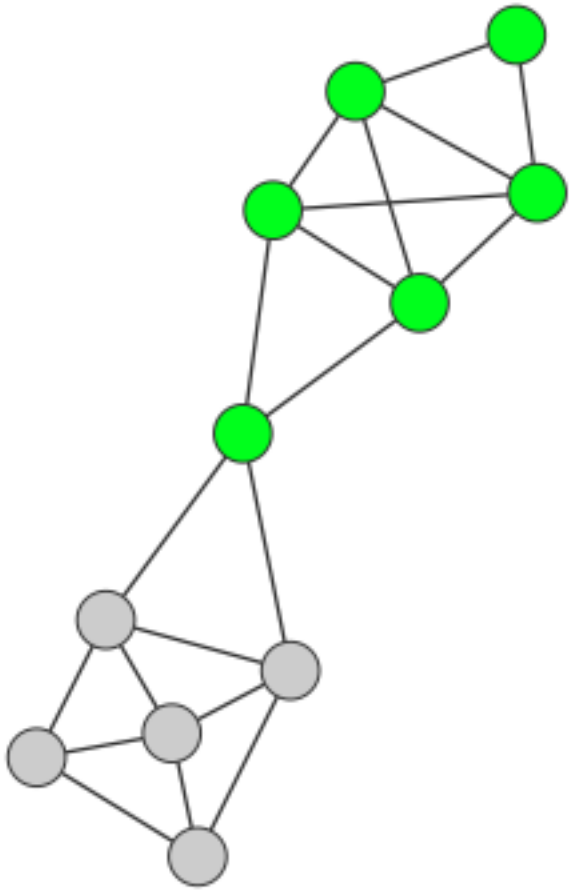
Defining communities (examples)

Connectedness + density hypothesis

Weak communities

The total internal degree of a community exceeds its total external degree

$$\sum_{i \in C} k_i^{\text{internal}}(C) > \sum_{i \in C} k_i^{\text{external}}(C)$$



Note: The green nodes represent one of many possible weak communities.

How many communities do we have?

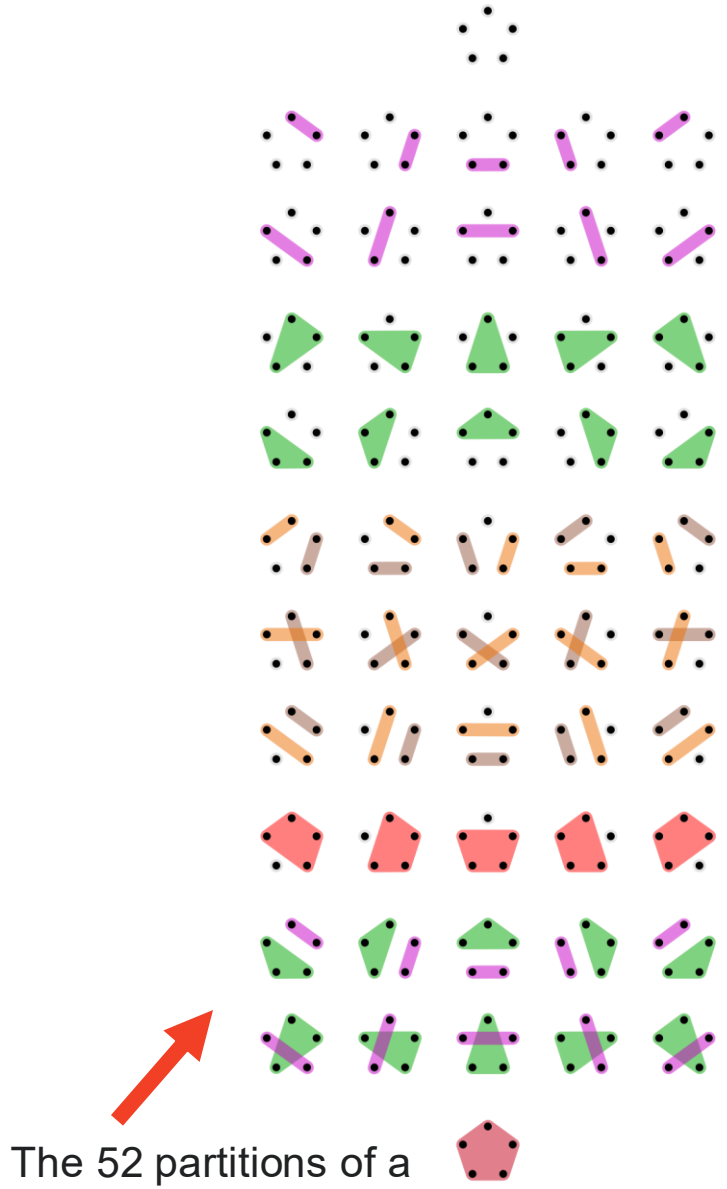
A partition is a division of a network into an arbitrary number of groups, such that each node belongs to a single group.

How many partitions can we have in a network of size N?

(this is provided by the Bell Number)

We are surfing in very large landscape of possible solutions…

Any brute-force technique will fail!



The 52 partitions of a set with 5 elements

How many communities do we have?

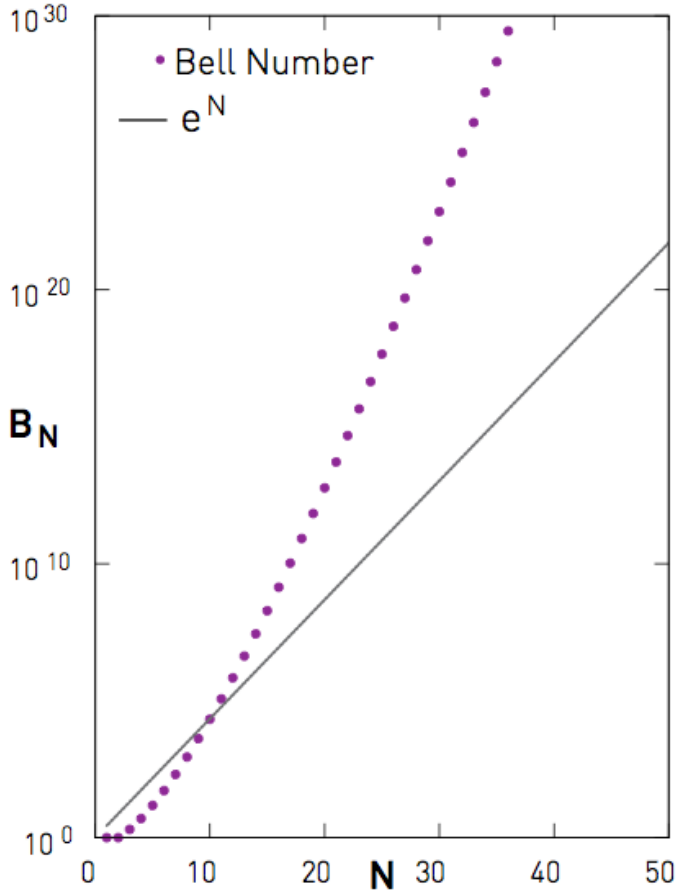
A partition is a division of a network into an arbitrary number of groups, such that each node belongs to a single group.

How many partitions can we have in a network of size N?

(this is provided by the Bell Number)

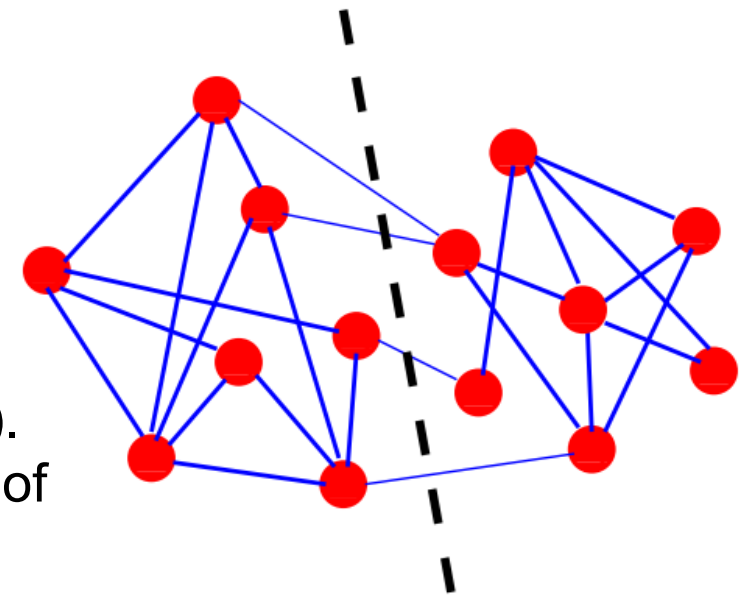
We are surfing in very large landscape of possible solutions…

Any brute-force technique will fail!



Graph partitioning

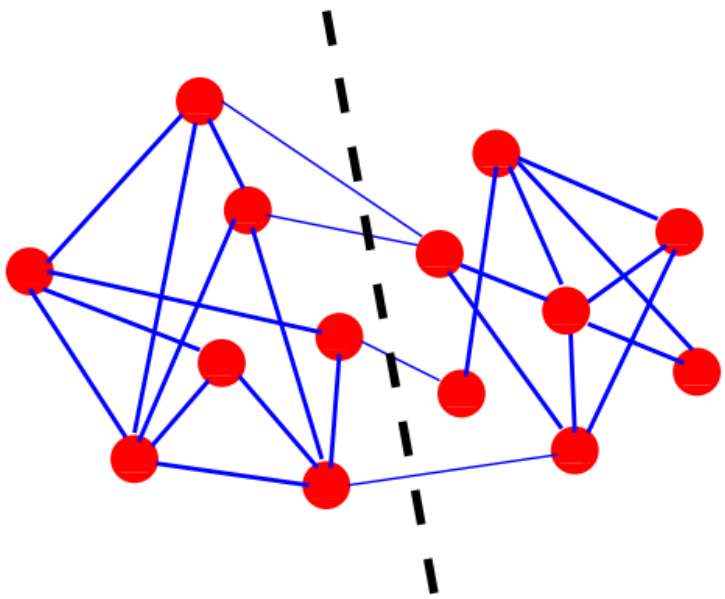
In standard graph partitioning (e.g. Kernighan-Lin Algorithm) the number and size of communities are pre-defined; in community detection both parameters are unknown.



Kernighan-Lin Algorithm (ex: 2 partitions). Randomly assign partitions. Test all swaps of of nodes among partitions recording the resulting change in the number of links connecting different communities (the cut size). Greedily accept the swap which offers the best reduction.

Graph partitioning

In standard graph partitioning (e.g. Kernighan-Lin Algorithm) the number and size of communities are pre-defined; in community detection both parameters are unknown.



The main content of hundreds of published papers on network communities consists on the search for the algorithm which provides the best partitioning given a certain measure of "quality".

Modularity

*Also referred to as **Q-modularity***

hard clustering, i.e., no overlap!

Compare two networks: our given network and its zero modularity counterpart (the uniformly random network with the same sequence of degrees).

Let us assume that a network of E links is divided into n modules, r = 1, 2,... ,n, with E_r links in module r.

$$\frac{(k_r = \text{total degree of nodes in r})^2}{2E}$$

$$M = \frac{1}{E}\sum_{r=1}^{n} E_{r\,|\,\text{net}} - \frac{1}{E}\sum_{r=1}^{n} E_{r\,|\,\text{random counterpart}}$$

Internal links within module r

Internal links in module r for a randomized network

Modularity

Compare two networks: our given network and its zero modularity counterpart (the uniformly random network with the same sequence of degrees).

The previous expression can be shown to be equivalent to

$$M = \sum_{r=1}^{n} \left[\frac{E_{r\,|\,\text{net}}}{E} - \left(\frac{k_r}{2E} \right)^2 \right]$$

k_r = total degree of nodes in r

Internal links within module r

Internal links in module r for a randomized network

Modularity

Compare two networks: our given network and its zero modularity counterpart (the uniformly random network with the same sequence of degrees).

The previous expression can be shown to be equivalent to

$$M = \sum_{r=1}^{n} \left[\frac{E_{r\,|\,\text{net}}}{E} - \left(\frac{k_r}{2E} \right)^2 \right]$$

k_r = total degree of nodes in r

One can also imagine this quantity as a "local" modularity M_r associated with community r

It provides a measure of the difference between the real wiring of the links and the expected number of links between each pair of vertices (belonging to this community) if the network is randomly rewired

Modularity (extensions)

In the most recent literature on graph clustering several modifications and extensions of modularity can be found.

Example: Modularity can be easily extended to graphs with ***weighted edges***. The degrees are replaced by the sum of the weights of edges adjacent to the vertex (a.k.a., *strength*), and the number of edges by the sum of weights (W).

$$M = \sum_{r=1}^{n} \left[\frac{W_{r|\text{net}}}{W} - \left(\frac{S_r}{2W} \right)^2 \right]$$

sum of the weights of the internal edges of module r

sum of the strengths of the vertices of r.

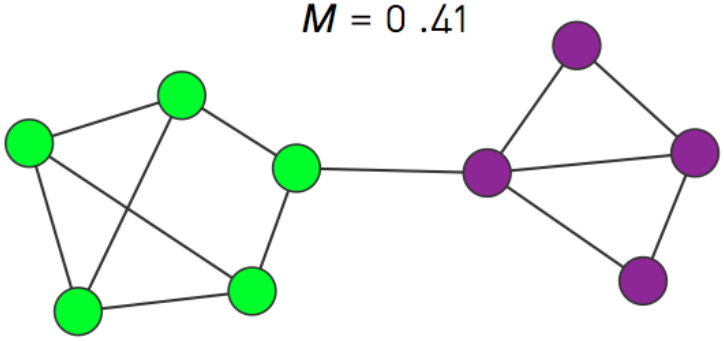
Modularity

$$M = \sum_{r=1}^{n} \left[\frac{E_r}{E} - \left(\frac{k_r}{2E} \right)^2 \right]$$

1

OPTIMAL PARTITION

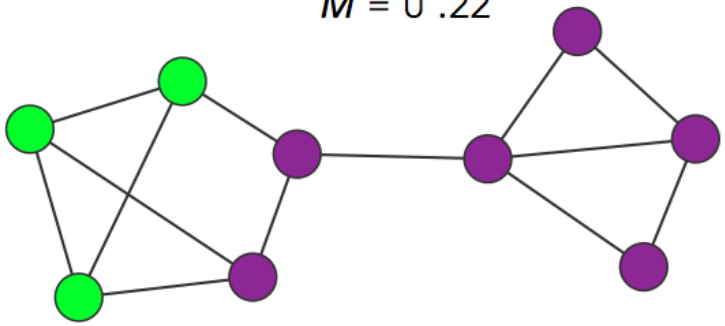
M = 0 .41



2

SUBOPTIMAL PARTITION

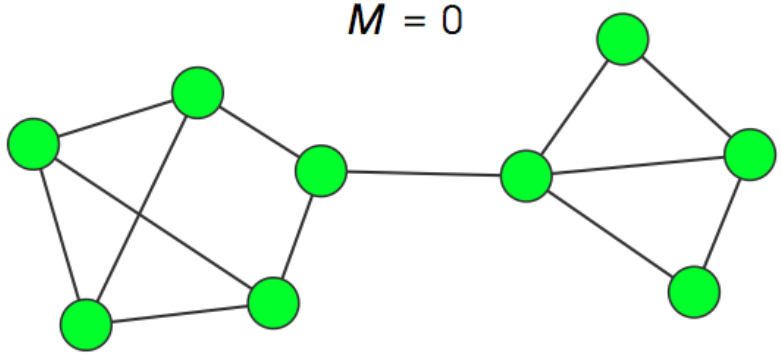
M = 0 .22



3

SINGLE COMMUNITY

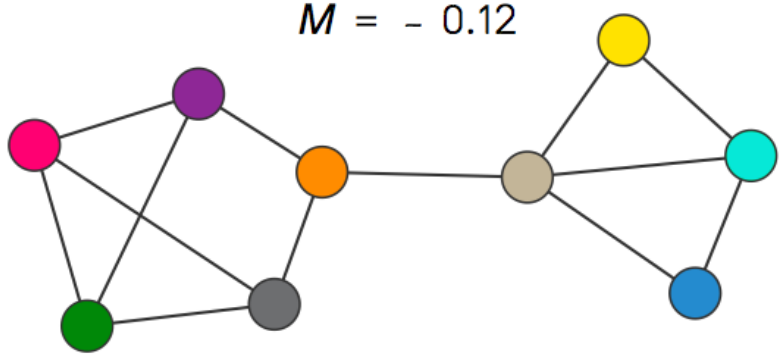
M = 0



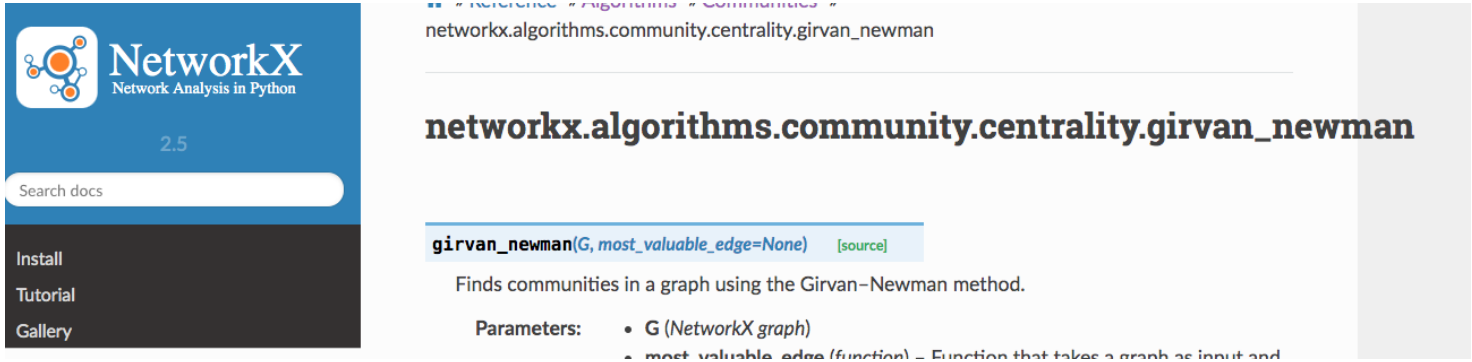
4

NEGATIVE MODULARITY

M = - 0.12

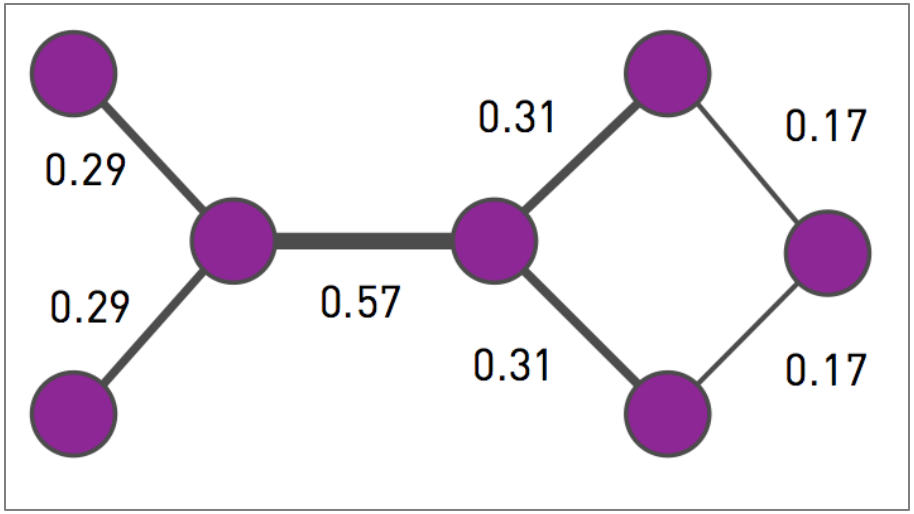


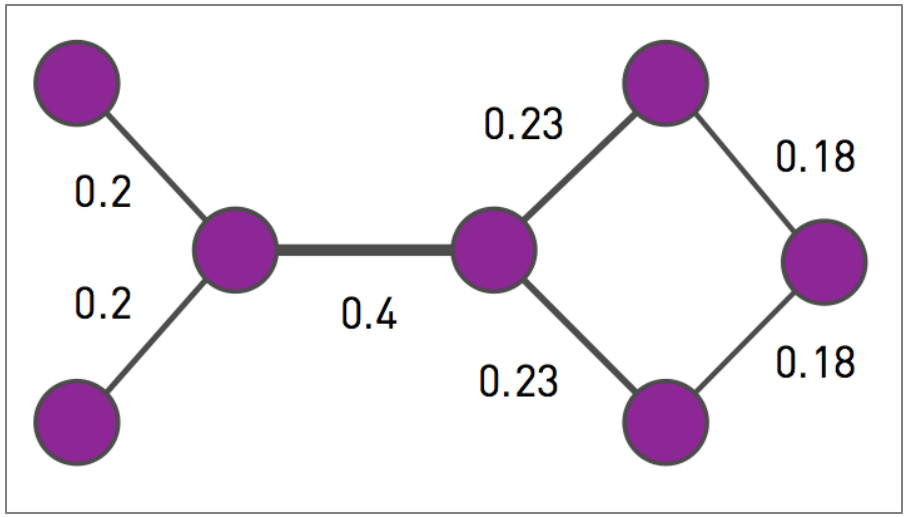
Let us analyze some popular algorithms…



The Girvan-Newman algorithm

Based on ***link centrality*** measures. Examples:





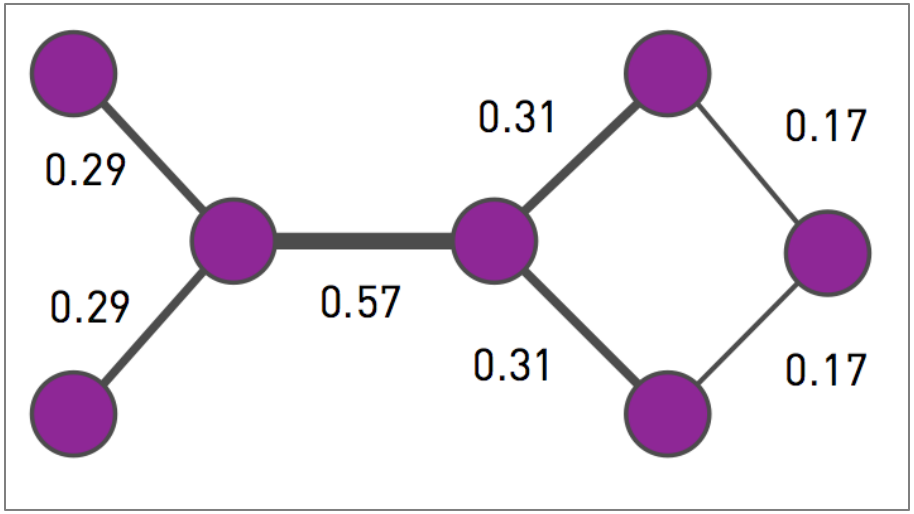
Link Betweenness

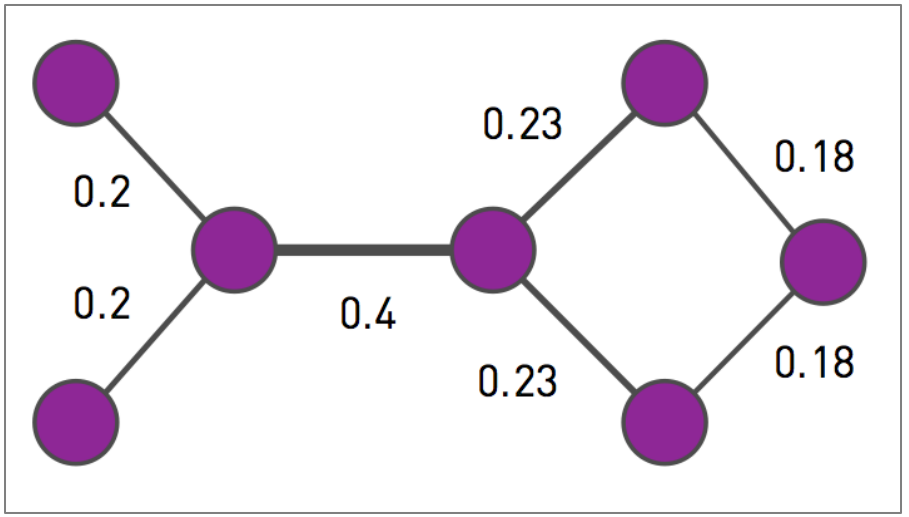
Random-Walk Betweenness

As node betweeness, link betweeness of a link (i,j) is proportional to the number of shortest paths between all node pairs that run along the link (i,j).

The Girvan-Newman algorithm

Based on ***link centrality*** measures. Examples:





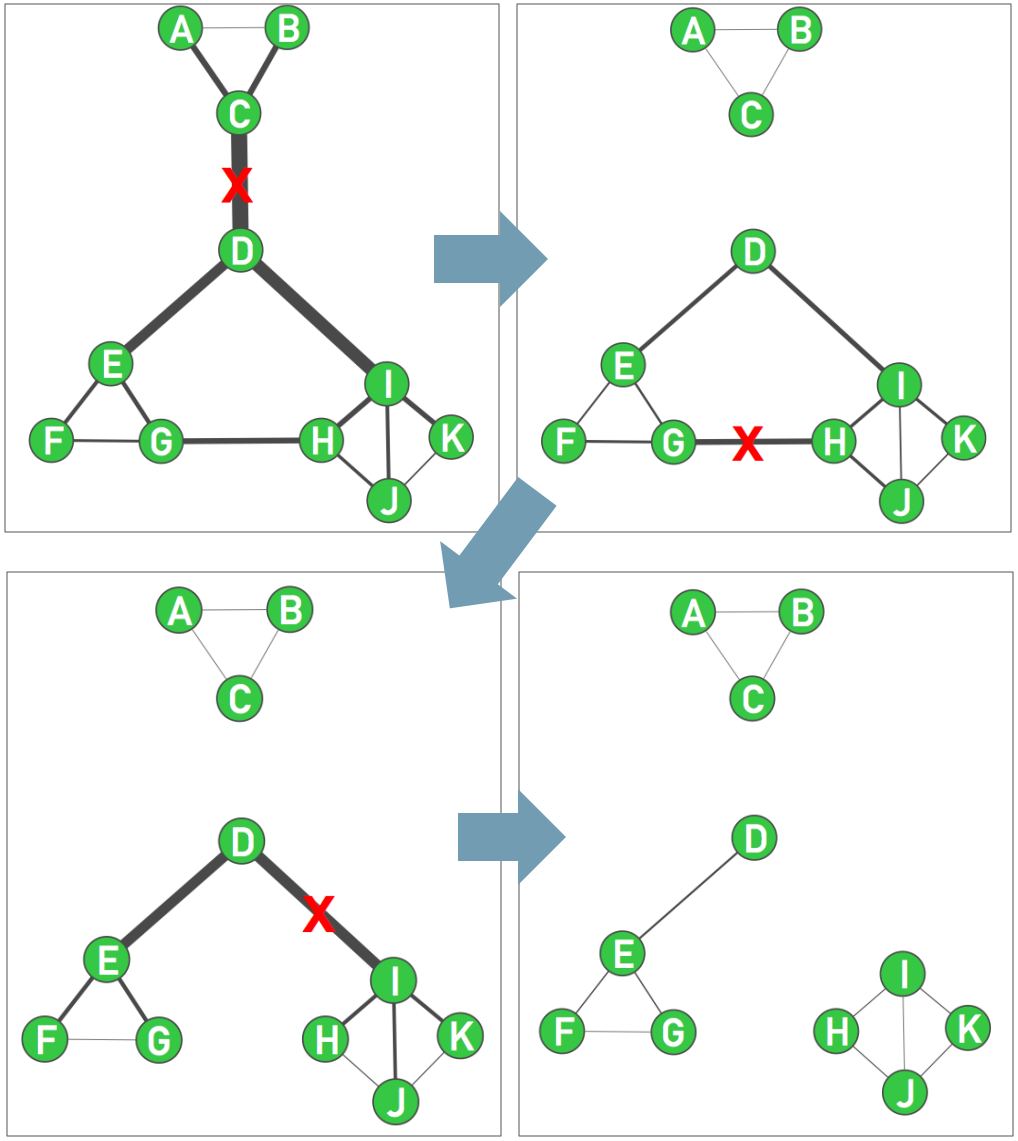
Link Betweenness

Random-Walk Betweenness

A pair of nodes (m, n) are chosen at random. A walker starts at m, randomly following links until it reaches n. Betweenness of a link i,j is the probability that the link is crossed by the walker after averaging over all possible choices for the starting nodes m and n.

The Girvan-Newman algorithm

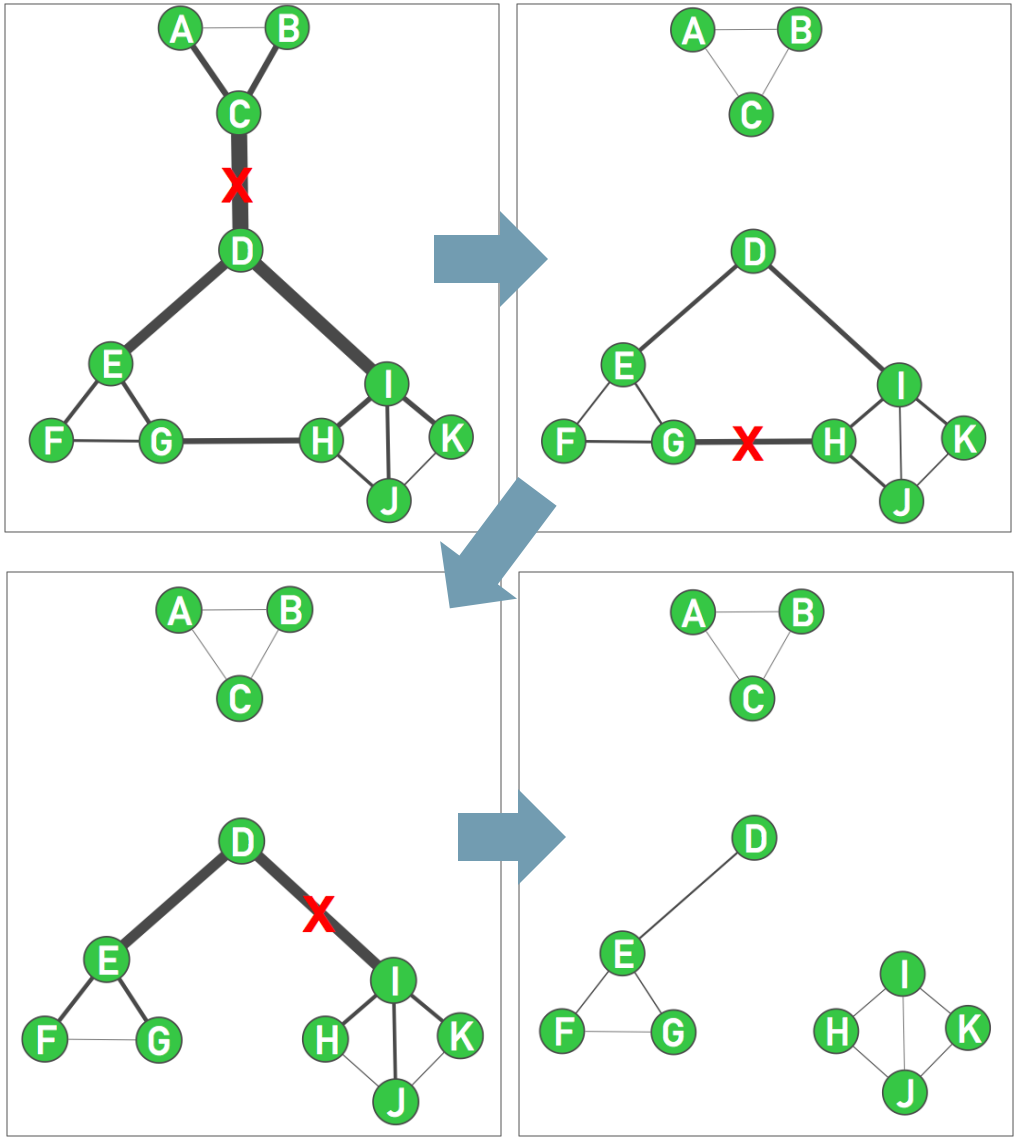
(w/ betweeness centrality)



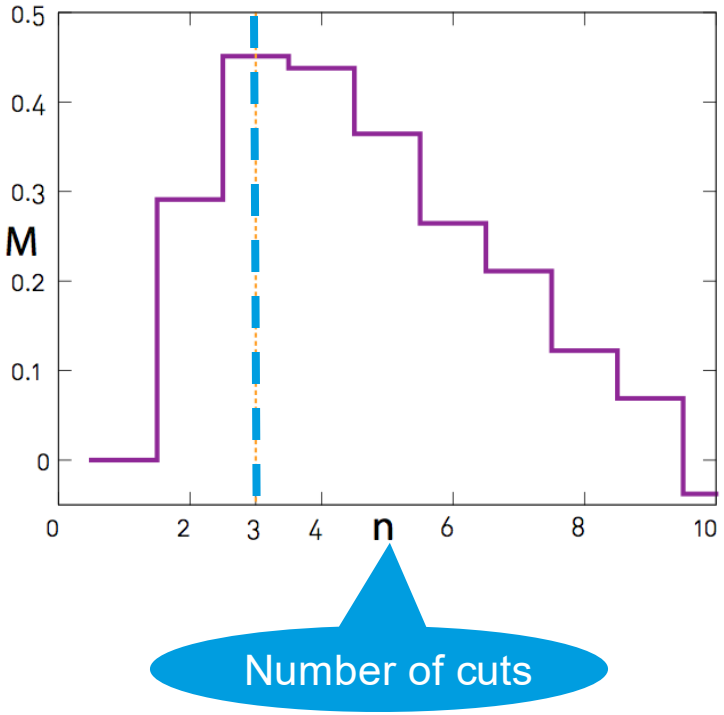
1. Compute the (betweeness) centrality of each link.
2. Remove the link with highest centrality.
3. Repeat steps 1 and 2 until all links are removed.

The Girvan-Newman algorithm

(w/ betweeness centrality)

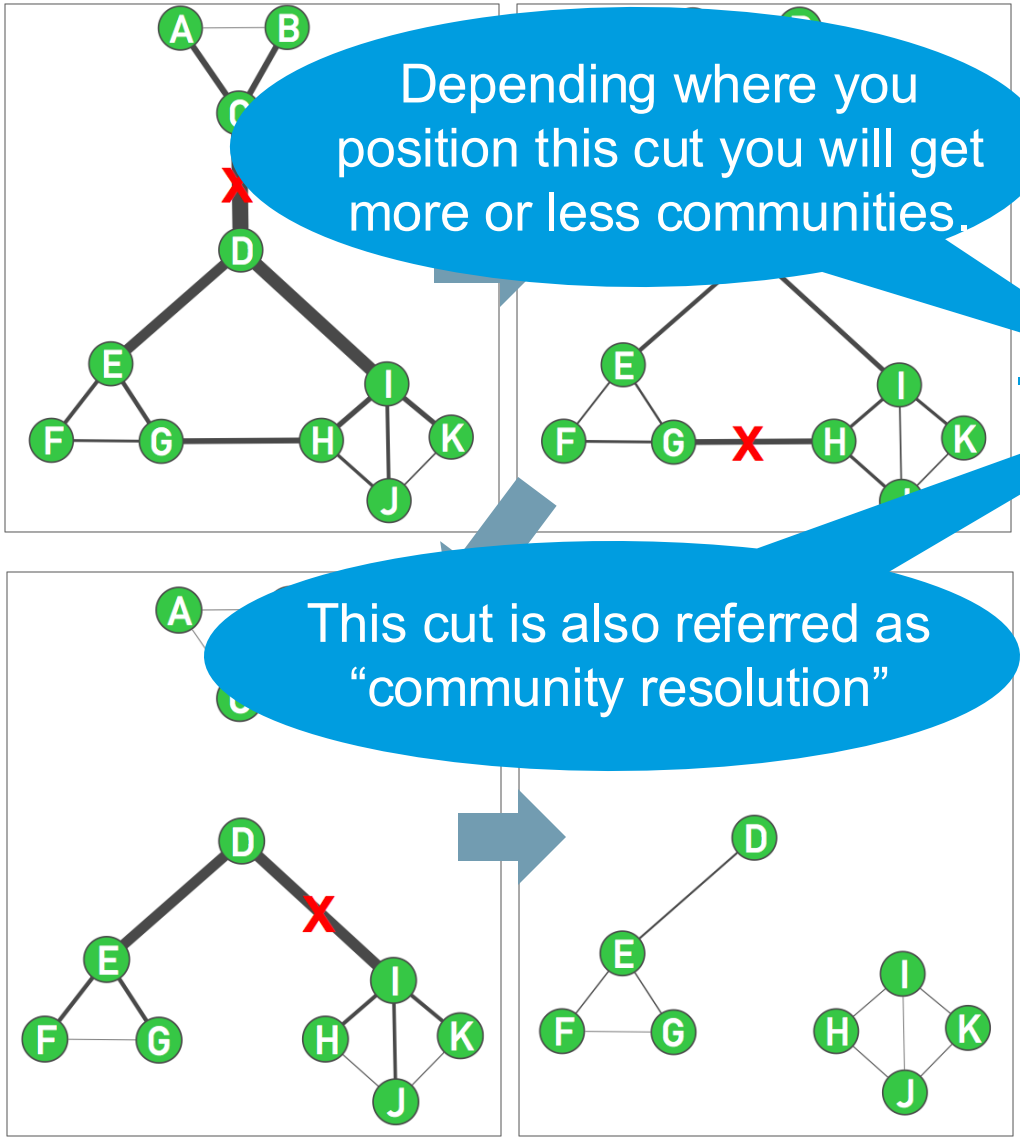


The modularity function, M, helps us select the optimal cut.

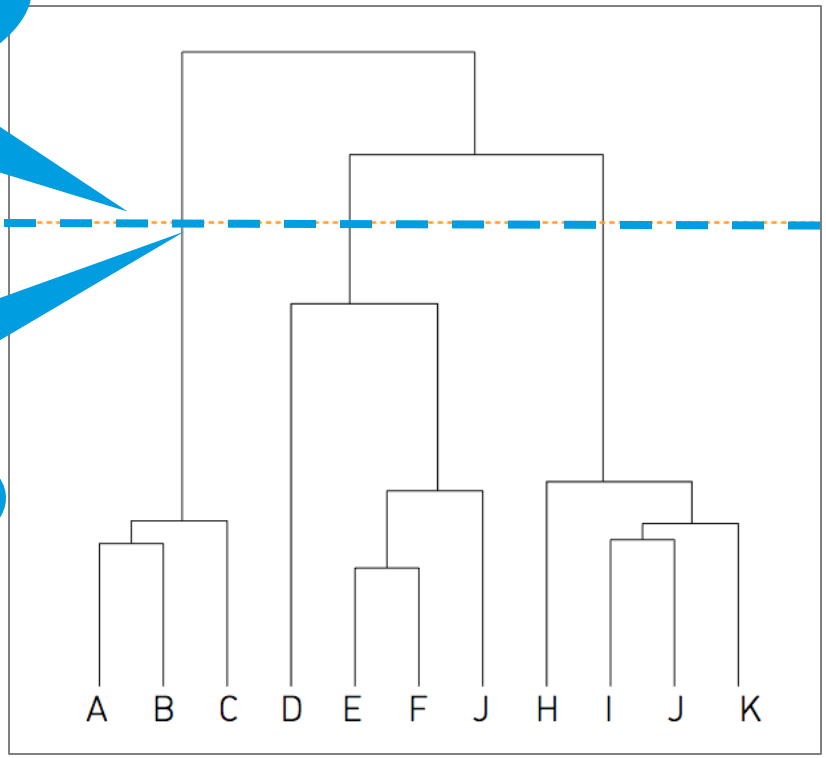


The Girvan-Newman algorithm

(w/ betweeness centrality)



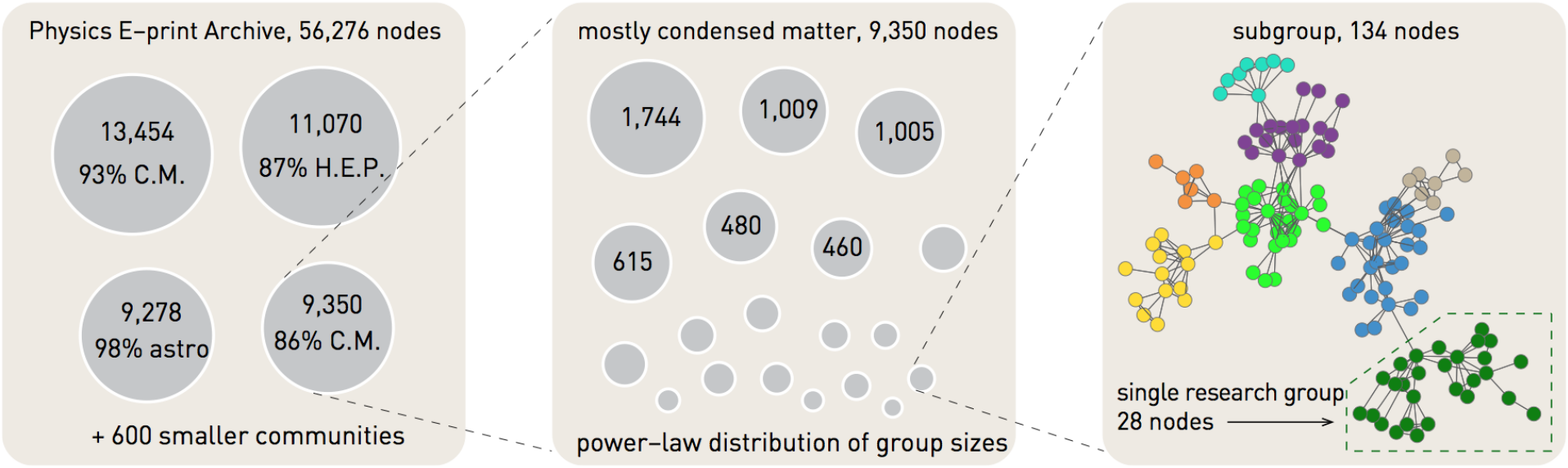
The dendrogram of communities



Horizontal cuts correspond to partitions of the graph in communities.

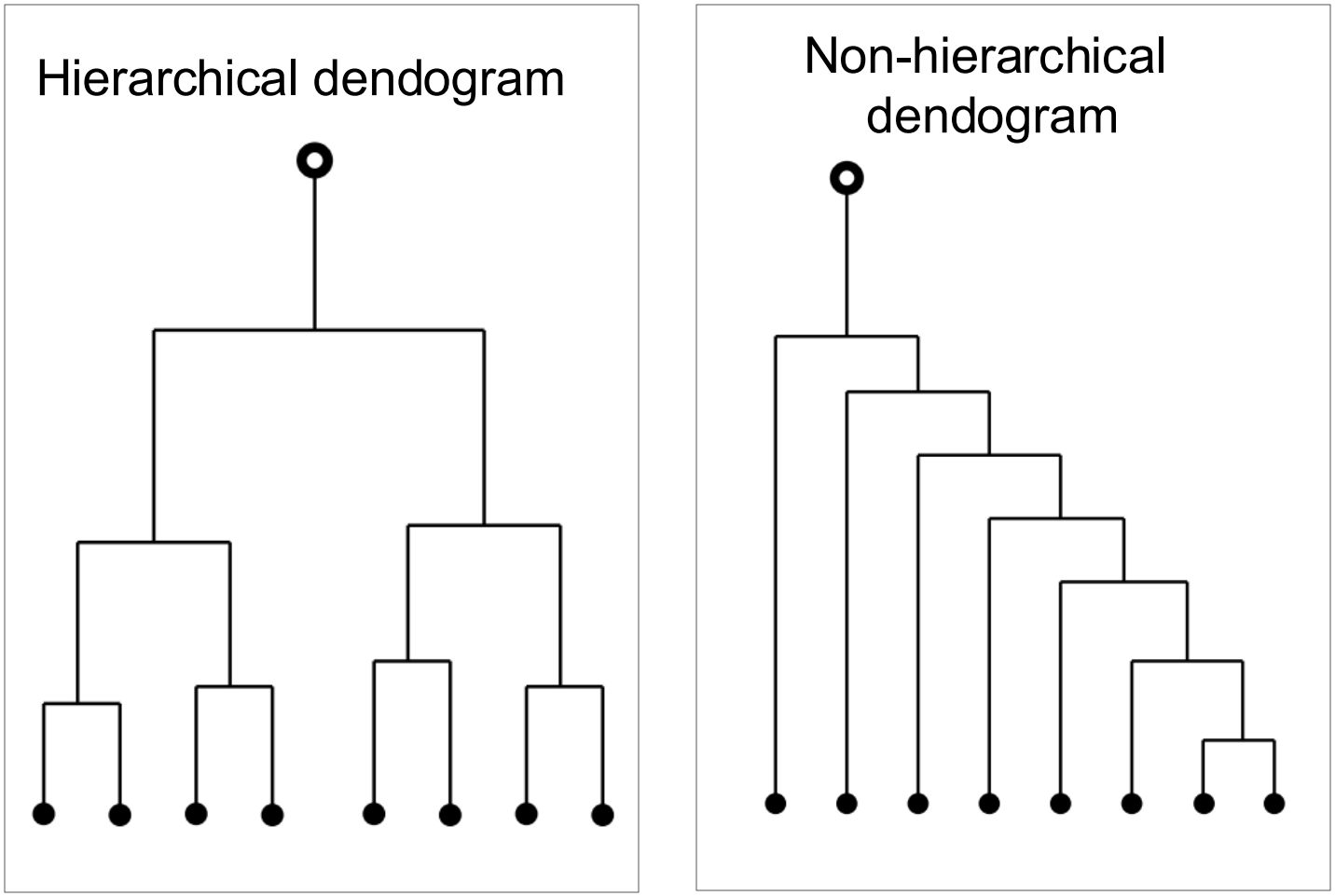
A small parenthesis on hierarchical structures

All these layers, irrespectively of its modularity value, may provide interesting features. Example: Collaboration networks in science.

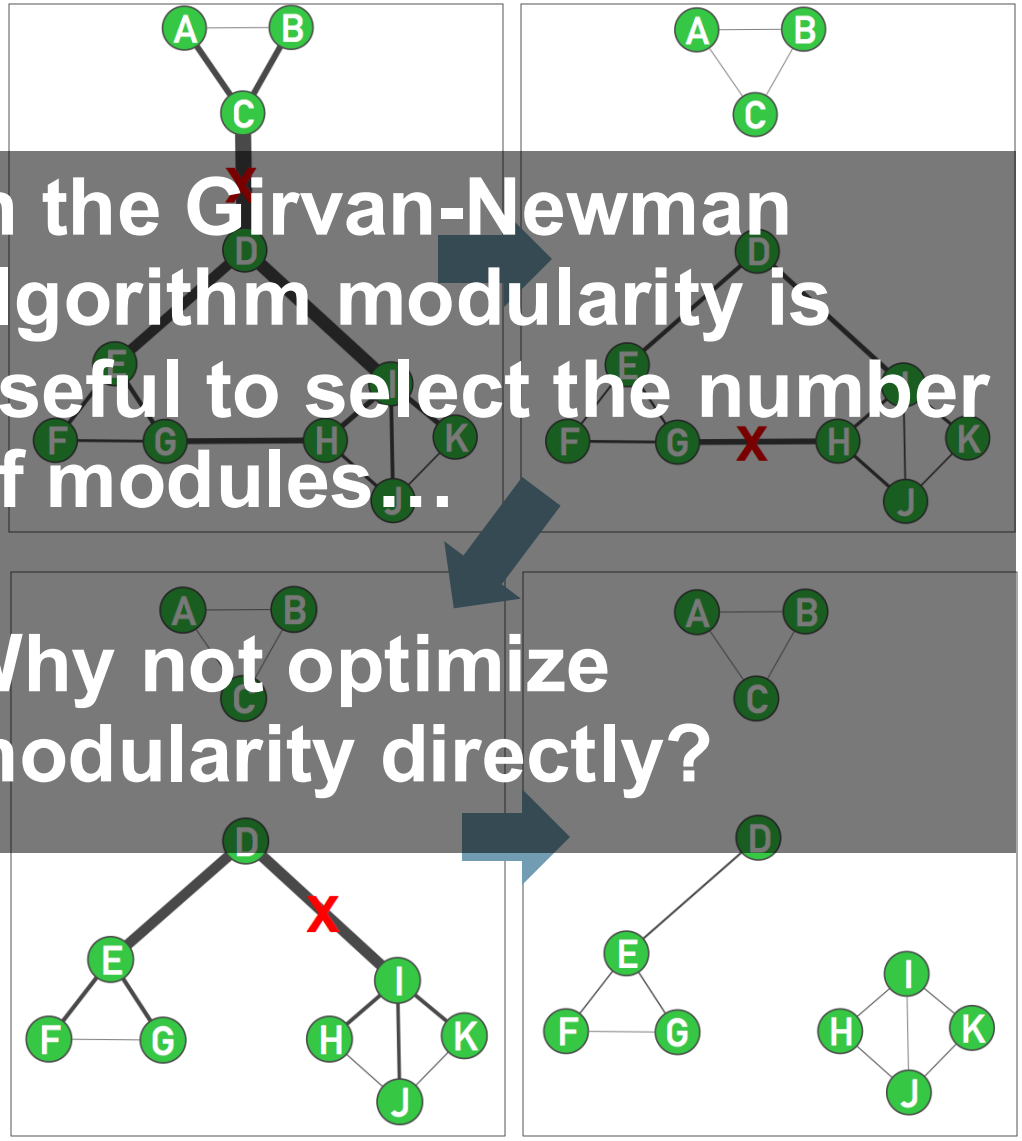


M.E.J. Newman. Fast algorithm for detecting community structure in networks. Physical Review E, 69:066133, 2004.

A small parenthesis on hierarchical structures



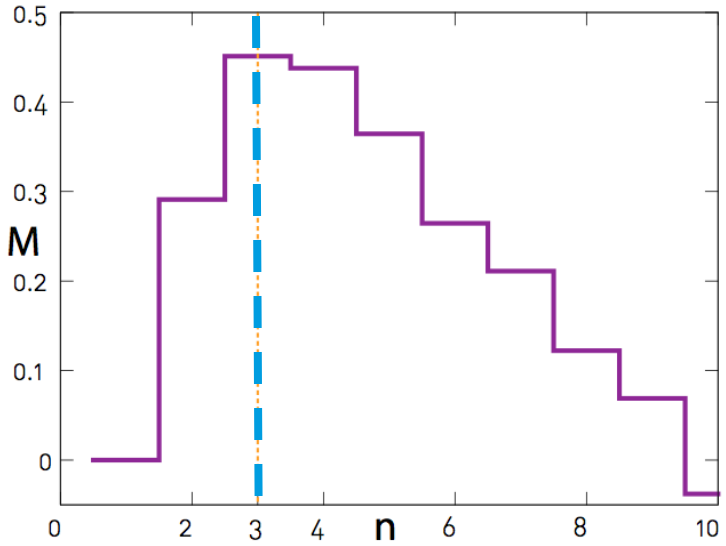
Two contrasting examples of dendrograms for two networks. The left dendrogram indicates what is often referred as a hierarchical organization of a network.



In the Girvan-Newman algorithm modularity is useful to select the number of modules…

Why not optimize modularity directly?

The modularity function, M, helps us select the optimal cut.

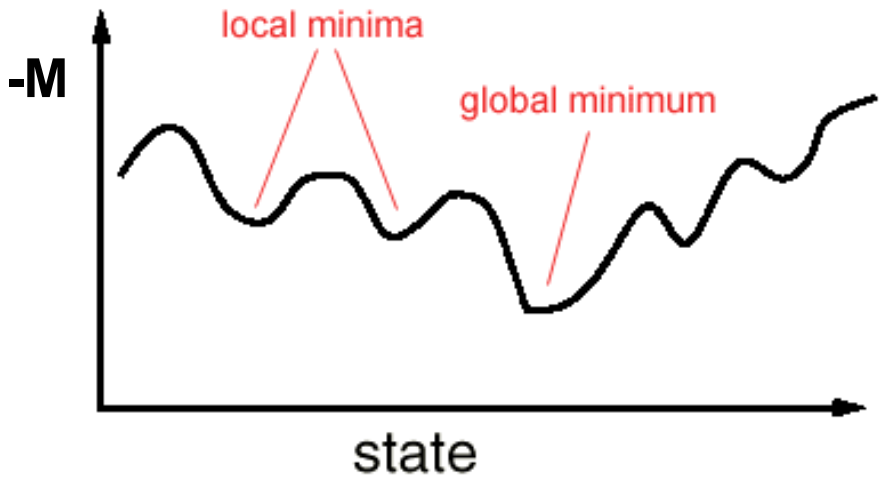


Optimizing modularity w/ a greedy algorithm

Newman, Phys Rev E 69:066133 (2004).

This time I won't break links. I will just assign a community to each node.

1. Start with all vertices as isolates (number of nodes = number of communities)
2. Identify the community pair for which ΔM is maximal and merge them.
3. Repeat 2 until all nodes merge into a single community.
4. Select the partition for which M is maximal (or –M is minimal)



Optimizing modularity w/ a greedy algorithm

Newman, Phys Rev E 69:066133 (2004).

This time I won't break links. I will just assign a community to each node.

1. Start with all vertices as isolates (number of nodes = number of communities)
2. Identify the community pair for which ΔM is maximal and merge them.
3. Repeat 2 until all nodes merge into a single community.
4. Select the partition for which M is maximal (or –M is minimal)

This processes can be improved by using simulated annealing rather than greedy search.

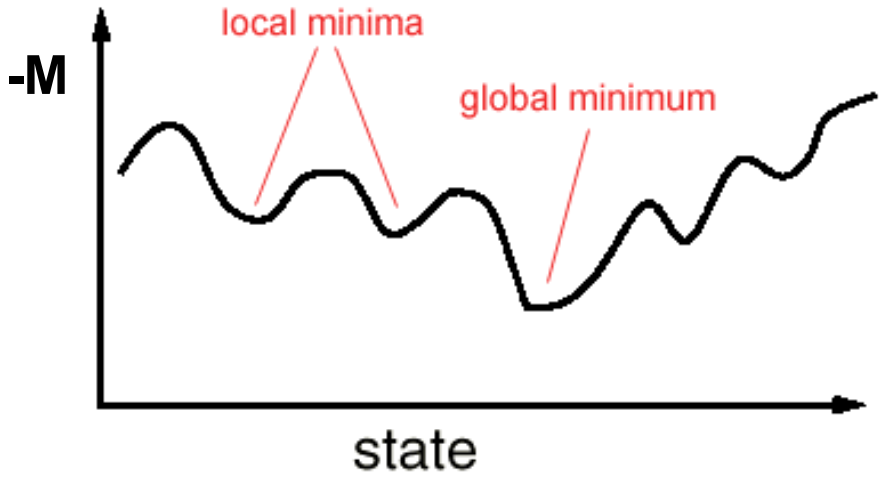
Guimerà & Amaral, Nature 2005

Optimizing modularity w/ simulated annealing

main idea

Guimerà & Amaral, Nature 2005

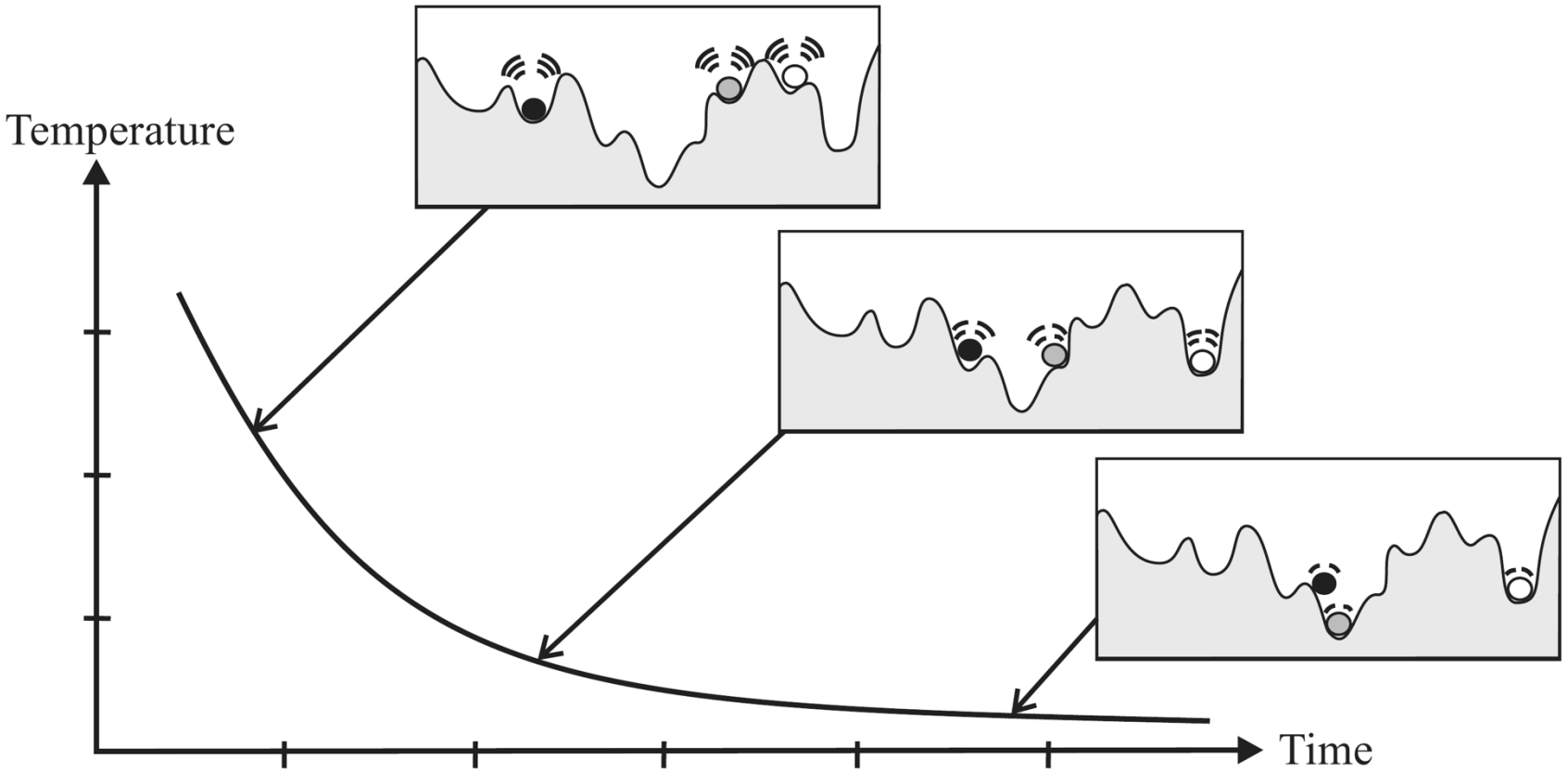
1. Start with all vertices as isolates (number of nodes = number of communities)
2. Select randomly a community pair and merge them if $\Delta M>0$. If $\Delta M<0$ accept the change anyway with a probability $p=Exp[-(M_{new}-M_{old})/T]$.
3. Repeat 2 until all nodes merge into a single community, progressively reducing *T* (temperature) until *T*=0.



Source code available here: https://amaral.northwestern.edu/resources/software/netcarto

Optimizing modularity w/ simulated annealing

Simulated Annealing



Source code available here: https://amaral.northwestern.edu/resources/software/netcarto

Optimizing modularity w/ simulated annealing

Guimerà & Amaral, Nature 2005

In its standard implementation Guimerá and Amaral combine two types of “moves” (with a SA at each level):

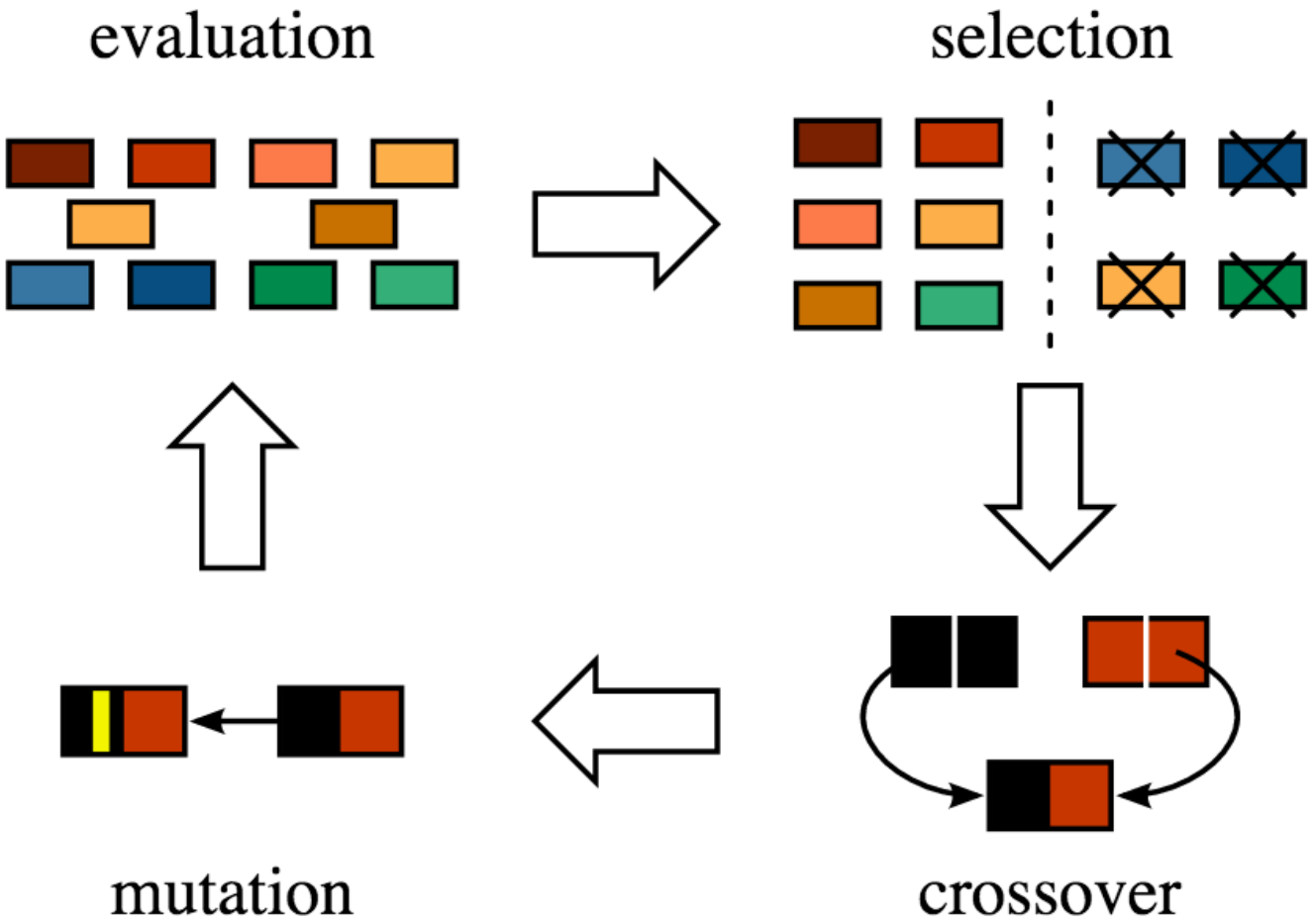
local moves: where a single vertex is shifted from one cluster to another, taken at random;

global moves: consisting of merges and splits of communities.

- In practical applications, one typically combines n^2 local moves with n global ones in one iteration.
- The method can potentially come very close to the true modularity maximum, but it is slow (max number of nodes ~10^4).

Source code available here: https://amaral.northwestern.edu/resources/software/netcarto

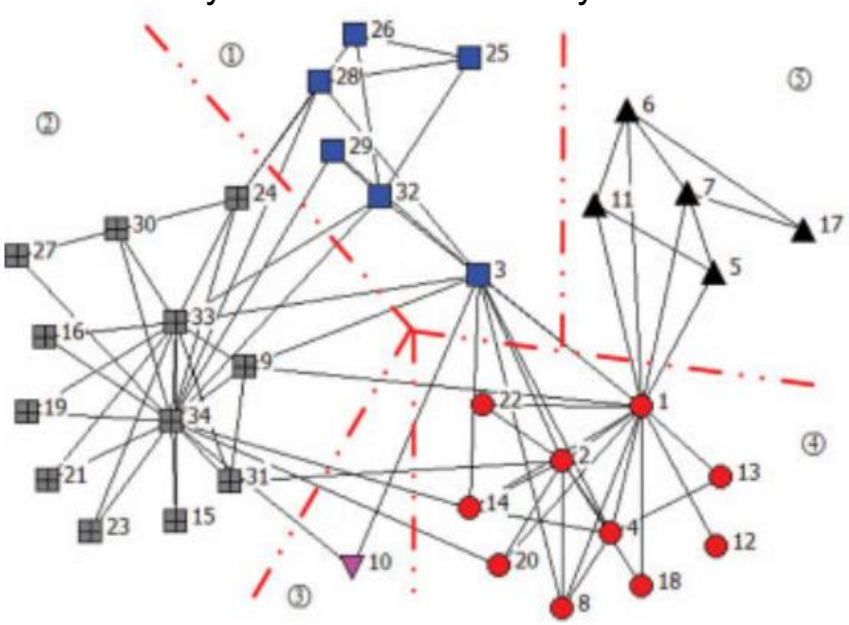
Optimizing modularity w/ Genetic algorithms



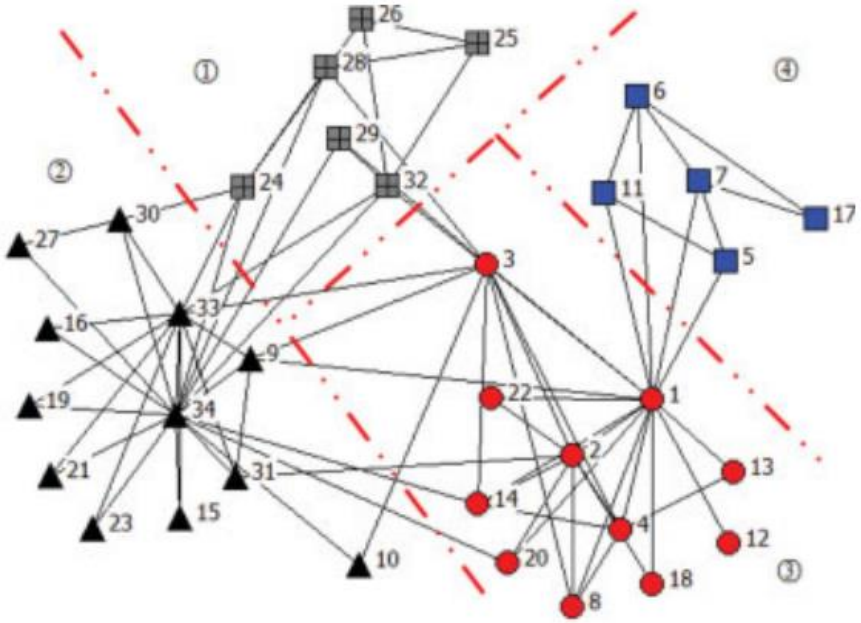
Shuzhuo et al., A genetic algorithm with local search strategy for improved detection of community structure, Complexity 2010

Optimizing modularity w/ Genetic algorithms

Community structures for Zachary karate club network



Girvan-Newman algorithm



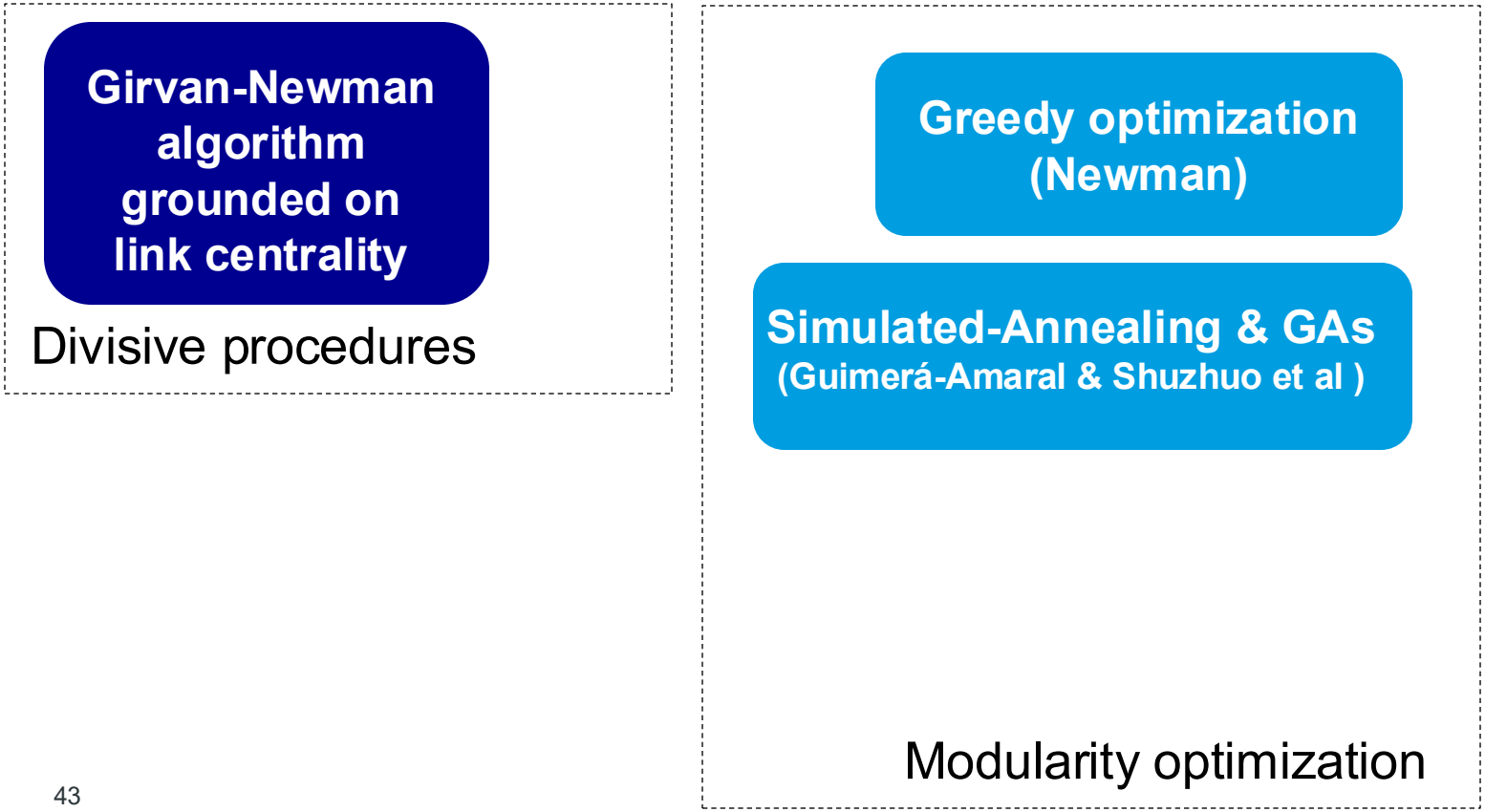
Genetic algorithm

As the SA it can be quite accurate, yet very slow for large N!

Shuzhuo et al., A genetic algorithm with local search strategy for improved detection of community structure, Complexity 2010

Conclusions so far

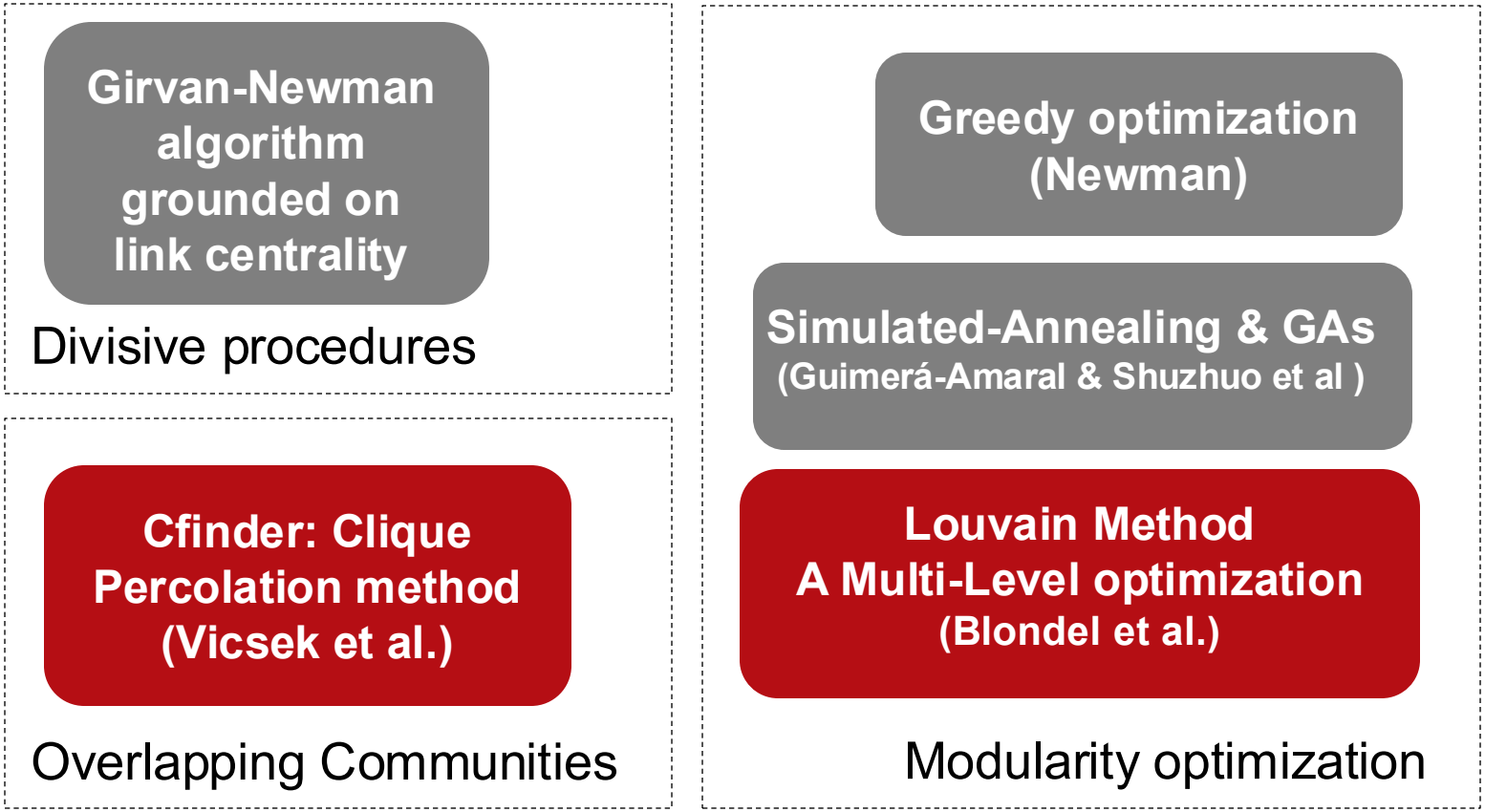
Several algorithms compete to be the most used method in community detection. **We have discussed a few:**



Conclusions so far

Several algorithms compete to be the most used method in community detection. **Before getting into a few more…**

Let's return back to the nature of the problem we're facing.



The limits of modularity

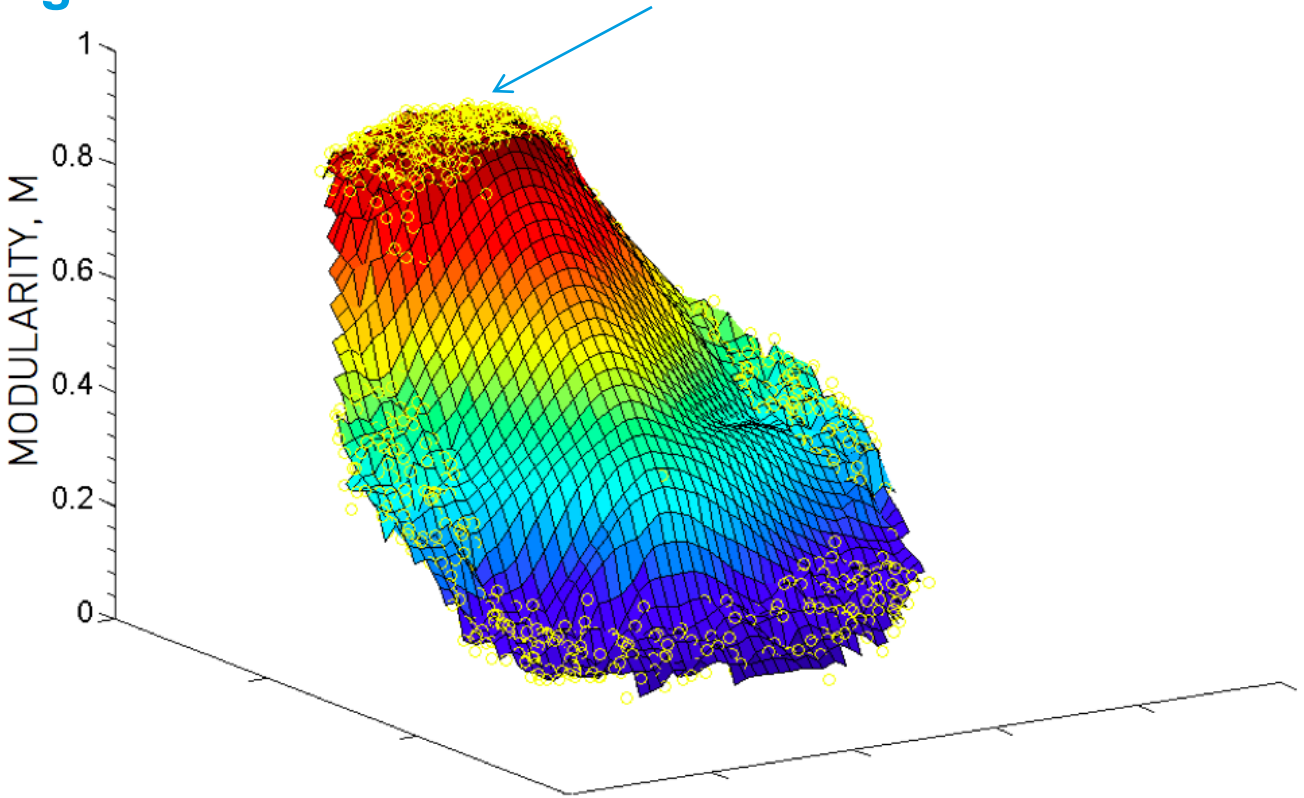
Why is this optimization problem so hard to solve?

- Extremely large landscape.
- **Problematic landscapes for optimization**
- **The maxima of modularity is often misleading**
- **Resolution limit**

Problematic landscapes

(Good, Montjoye, and Clauset, 2009)

We are trying to find a maximum in here:

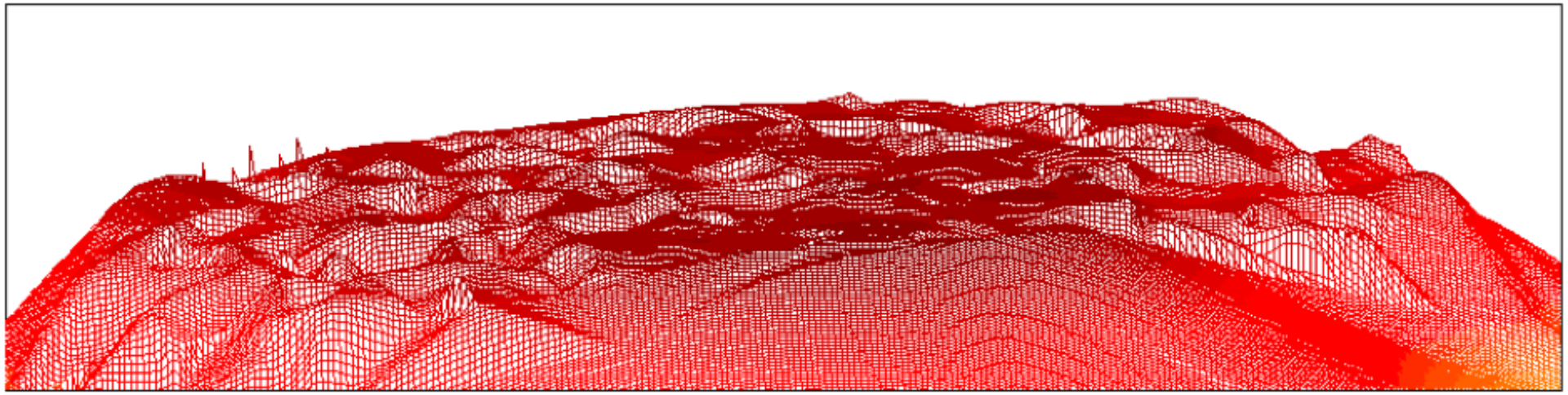


the modularity function is not peaked around a single optimal partition, but has a high modularity plateau

Problematic landscapes

(Good, Montjoye, and Clauset, 2009)

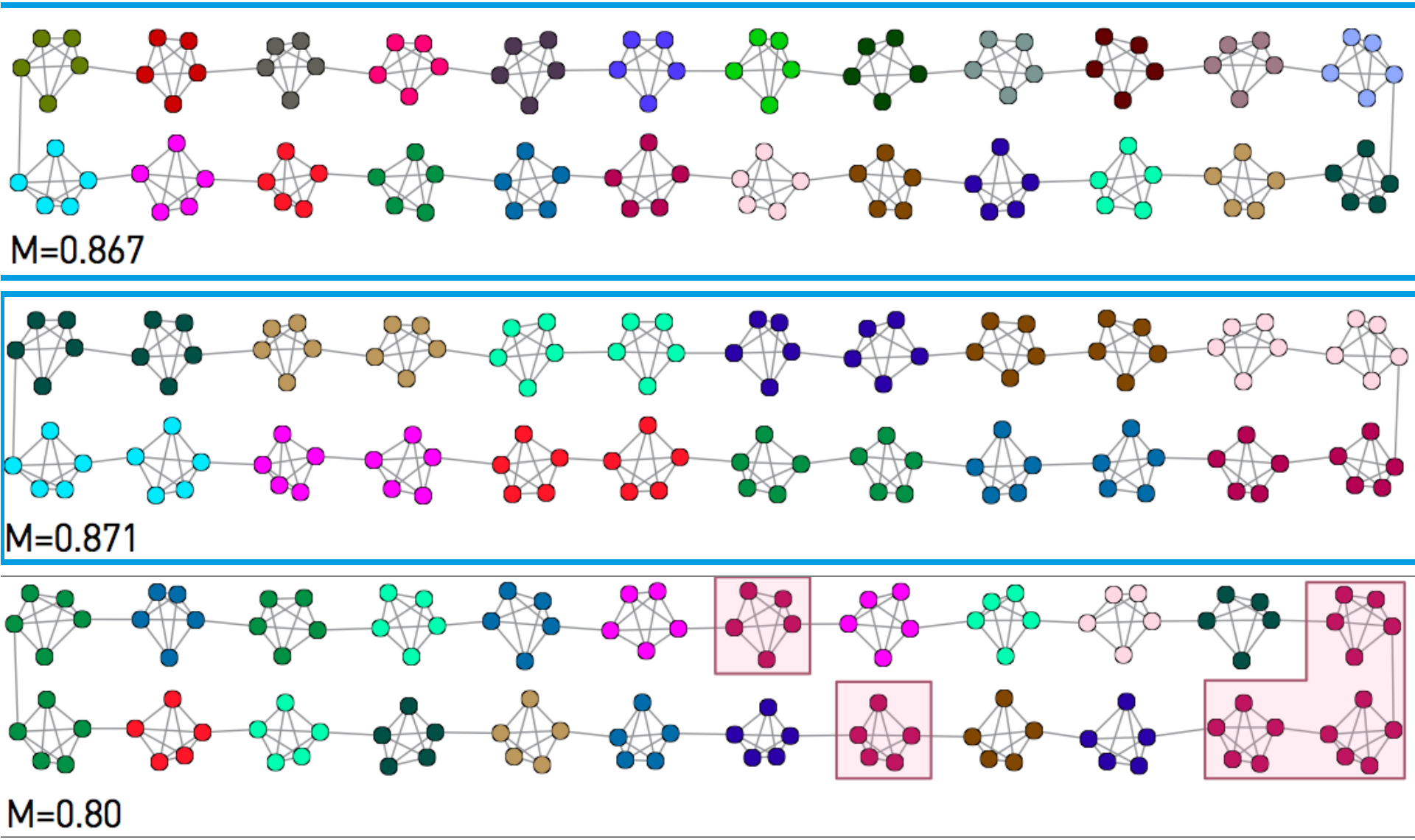
We are trying to find a maximum in here:



the modularity function is not peaked around a single optimal partition, but has a high modularity plateau

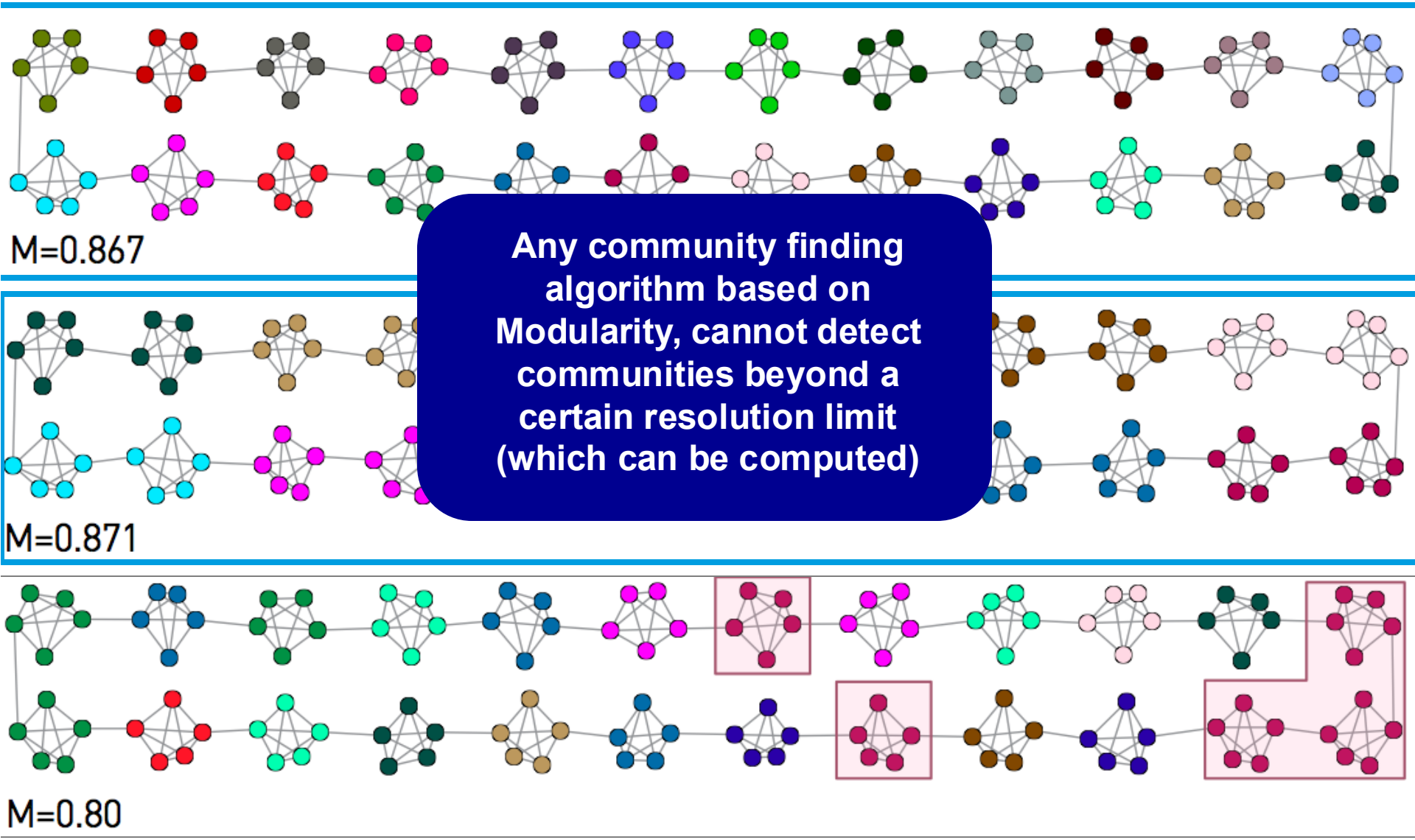
Resolution limit

Fortunato & Barthélemy, Resolution limit in community detection. PNAS,104:36–41, 2007.



Resolution limit

Fortunato & Barthélemy, Resolution limit in community detection. PNAS,104:36–41, 2007.



Any community finding algorithm based on Modularity, cannot detect communities beyond a certain resolution limit (which can be computed)

Resolution limit

Fortunato & Barthélemy, Resolution limit in community detection. PNAS,104:36–41, 2007.

Modularity maximization forces small communities into larger ones. If we merge communities A and B into a single community, the network's modularity changes with

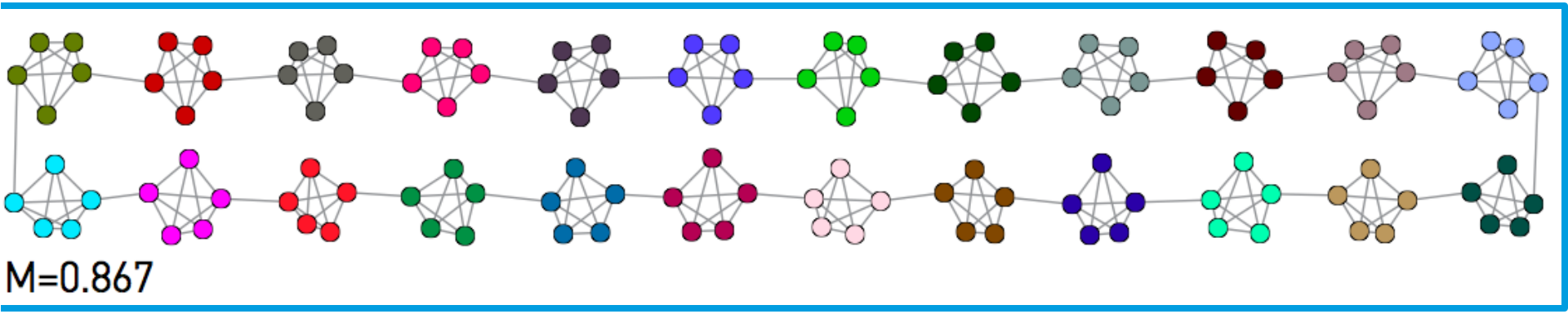
$$\Delta M_{AB} = \frac{E_{AB}}{E} - \frac{k_A k_B}{2E^2}$$

where E_{AB} is number of links that connect the nodes in community A with total degree k_A to the nodes in community B with total degree k_B.

If A and B are distinct communities, they should remain distinct when *M* is maximized. Yet, this is not always the case.

Resolution limit

Fortunato & Barthélemy, Resolution limit in community detection. PNAS,104:36–41, 2007.



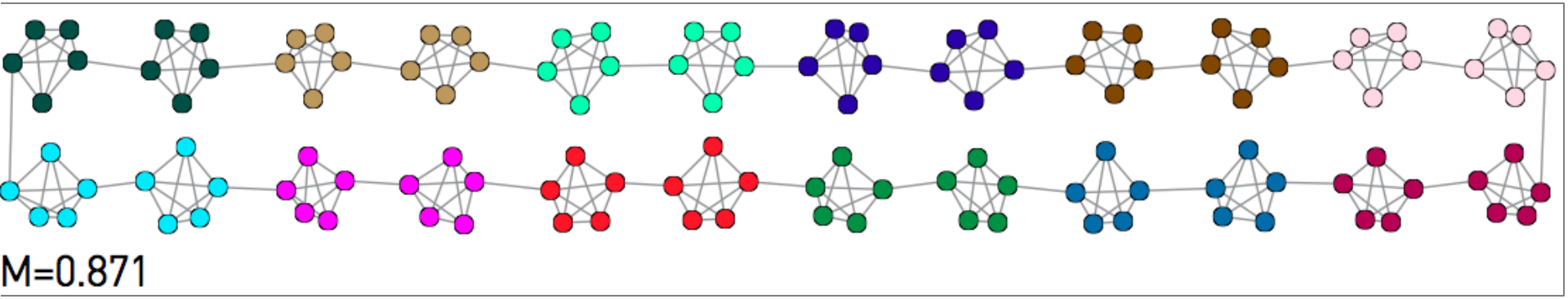
$$\Delta M_{AB} = \frac{E_{AB}}{E} - \frac{k_A k_B}{2E^2}$$

Merging two of the cliques k-cliques illustrated above gives

$$\Delta M_{AB} = \frac{1}{E} - \frac{k^2(k-1)^2}{2E^2} > 0 \xrightarrow{k \approx k-1} k > \sqrt{2E}$$

Resolution limit

Fortunato & Barthélemy, Resolution limit in community detection. PNAS,104:36–41, 2007.



- Thus, any modularity algorithm cannot detect communities below the resolution limit

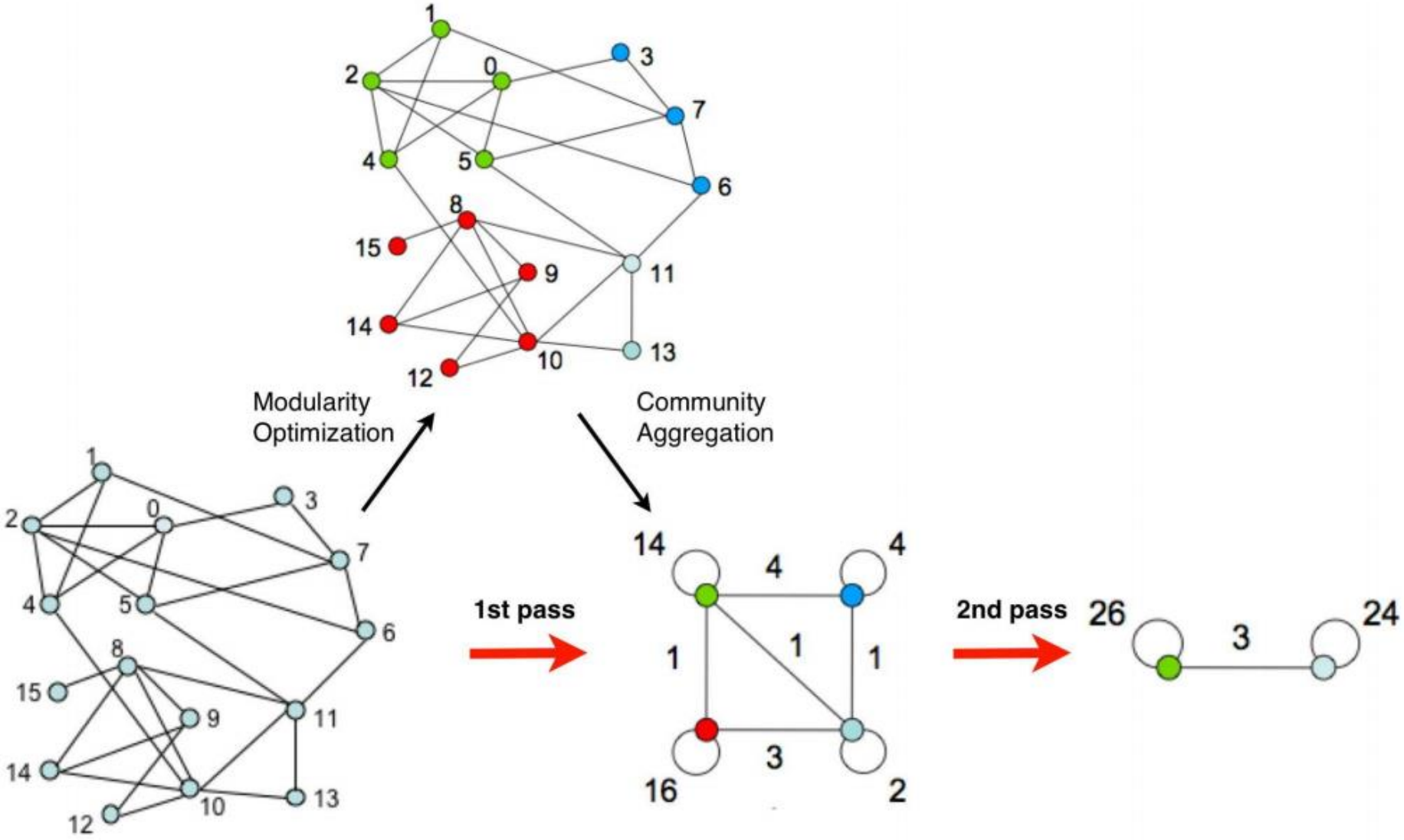
$$k \leq \sqrt{2E}$$

- Real networks contain numerous small communities and large absolute values of *E*. Given the resolution limit, these small communities are systematically forced into larger communities, offering a misleading characterization of the underlying community structure.

Blondel et al., J Stat Mech 2008

Louvain method

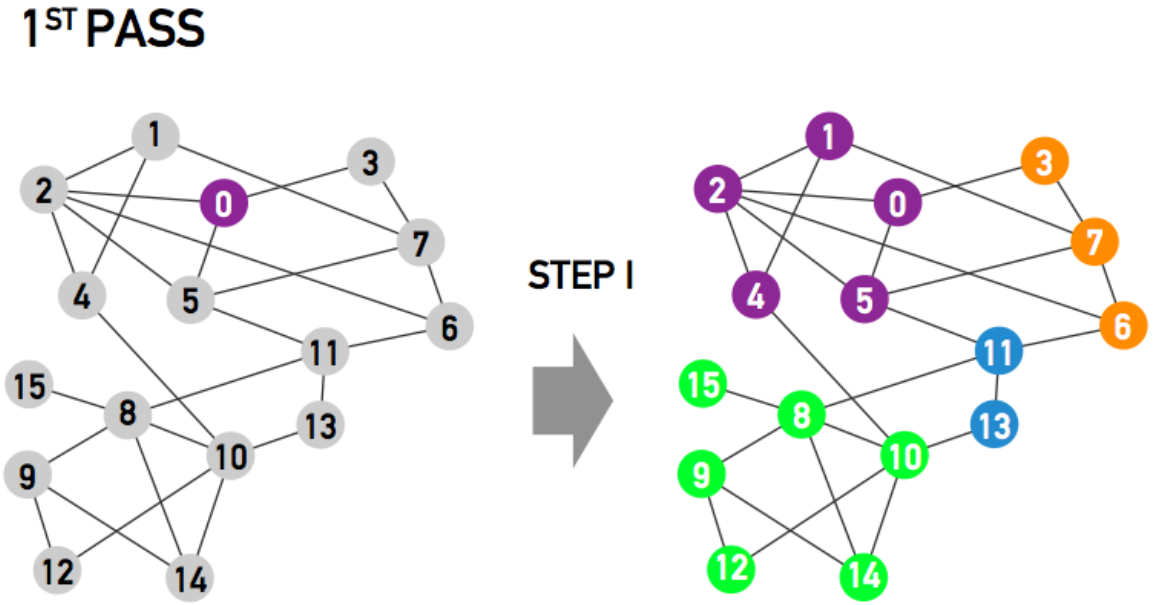
A Multi-Level Aggregation Method for optimizing modularity



Louvain method

Blondel et al., J Stat Mech 2008

The main steps of the Louvain algorithm. Each pass consists of two steps:

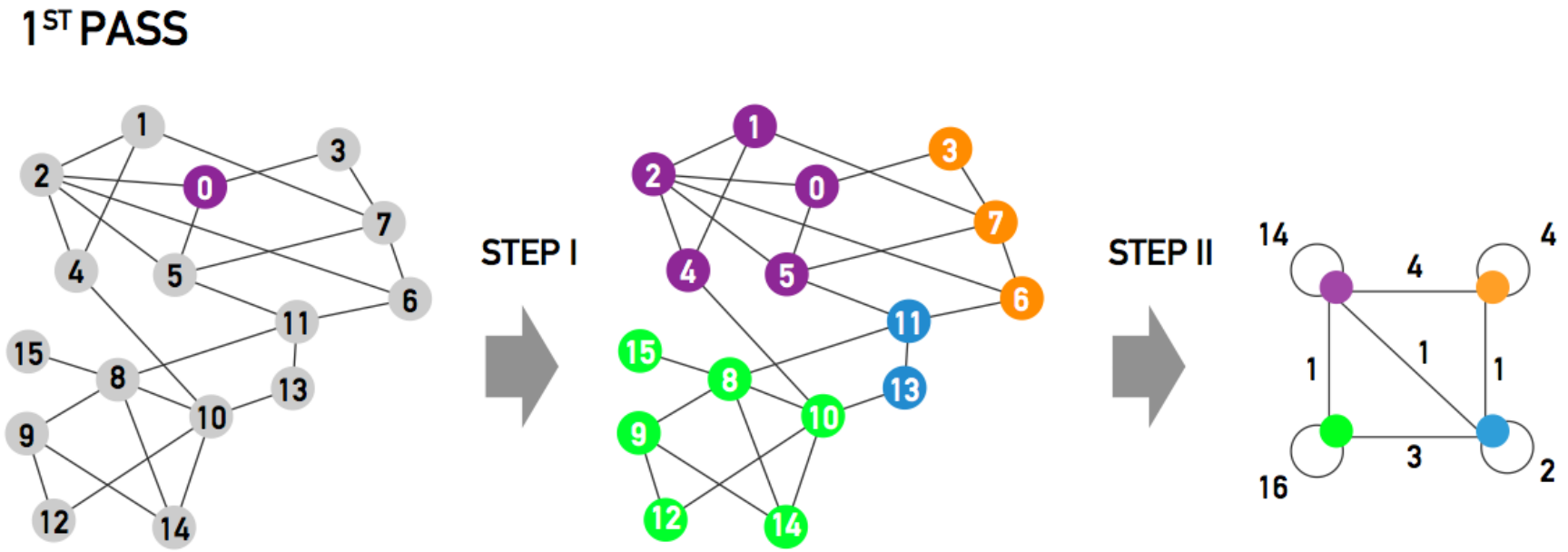


STEP I – Modularity Optimization
Modularity is optimized by local changes. For each node calculate the change in modularity in case the node joins the community of each immediate neighbor.

Louvain method

Blondel et al., J Stat Mech 2008

The main steps of the Louvain algorithm. Each pass consists of two steps:



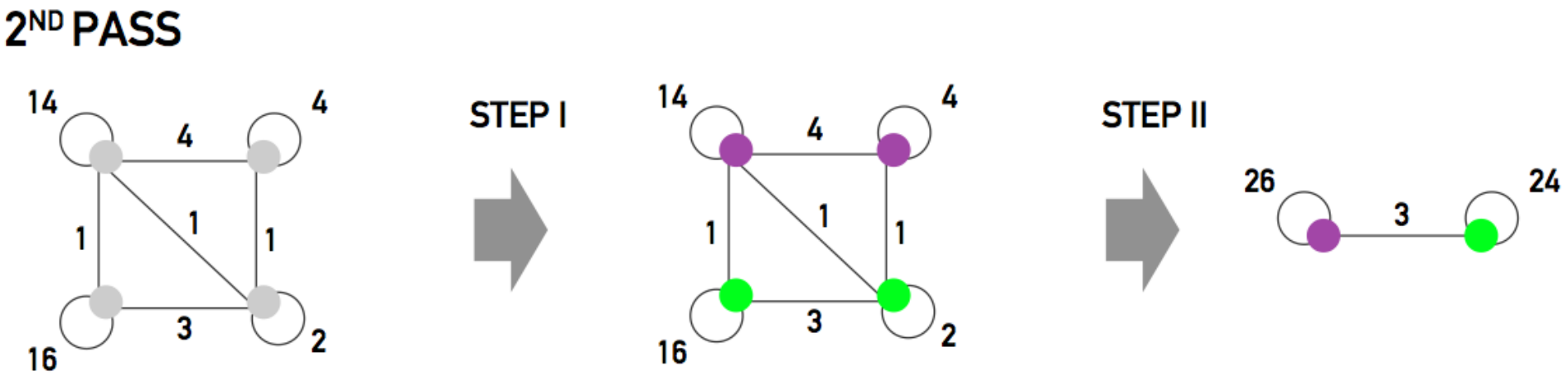
STEP II – Community aggregation

Construct a new network whose nodes are the communities identified in Step I. The weight of the link between two nodes is the sum of the weight of the links between the nodes in the corresponding communities.

Louvain method

Blondel et al., J Stat Mech 2008

The main steps of the Louvain algorithm. Each pass consists of two steps:



The sum of steps I and II is called a pass. The network obtained after each pass is processed again until no further increase in modularity is possible.

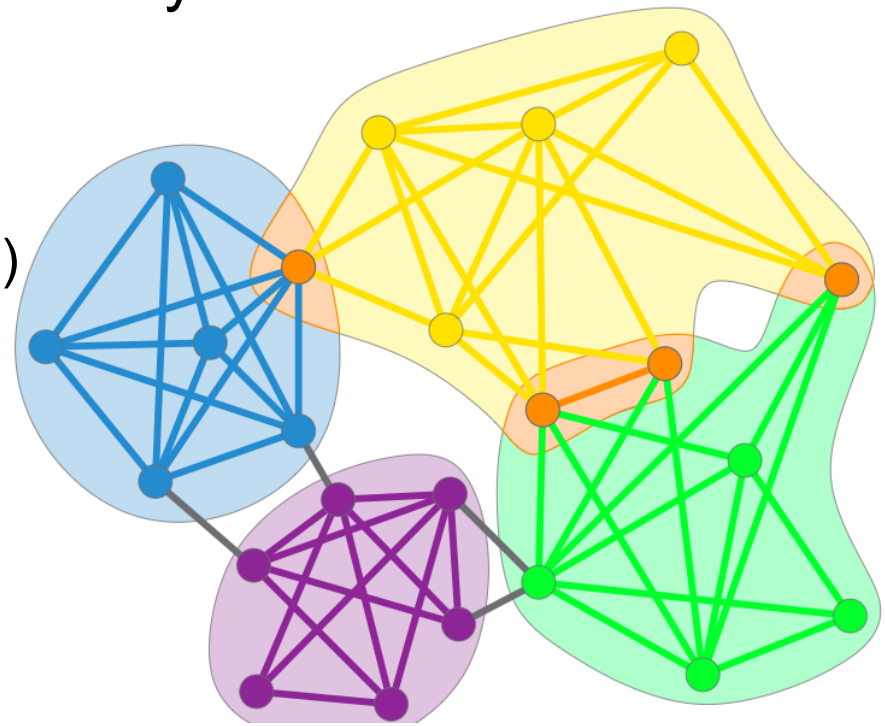
Overlapping communities

While the existence of a nested community structure has long been appreciated by sociologists and within graph partitioning methods, the algorithms discussed so far force each node into a single community.

Examples of solutions:

1. **CFinder** (clique percolation)
   [Palla et. al., Nature 446, 664 (2007)]

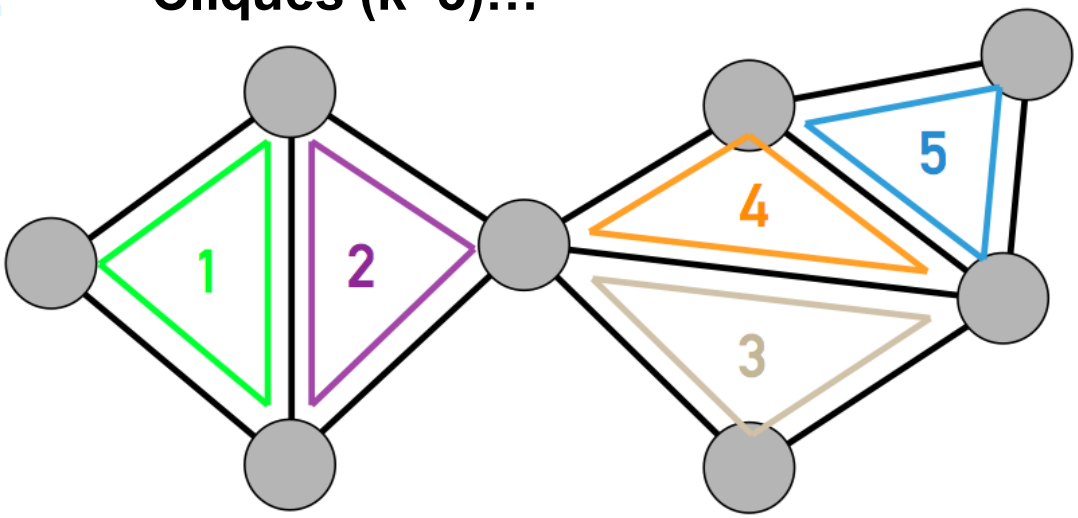
1. Link clustering
   [Ahn et al, Nature, 466, 761 (2010)]



Clique percolation algorithm (http://www.cfinder.org/)

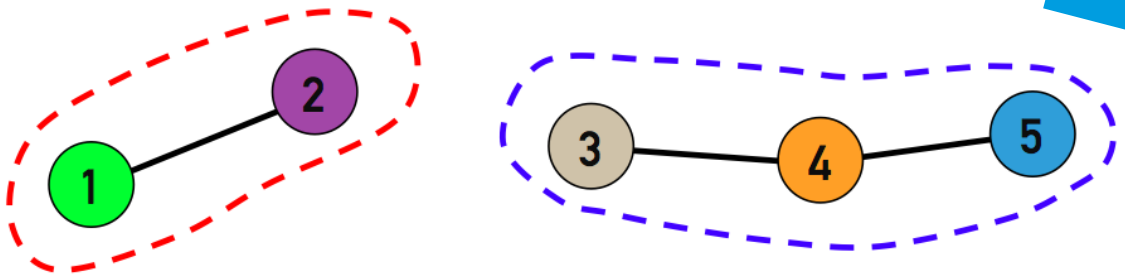
Palla et. al., Nature 446, 664 (2007)

Cliques (k=3)…



O=

	1	2	3	4	5
1	0	1	0	0	0
2	1	0	0	0	0
3	0	0	0	1	0
4	0	0	1	0	1
5	0	0	0	1	0

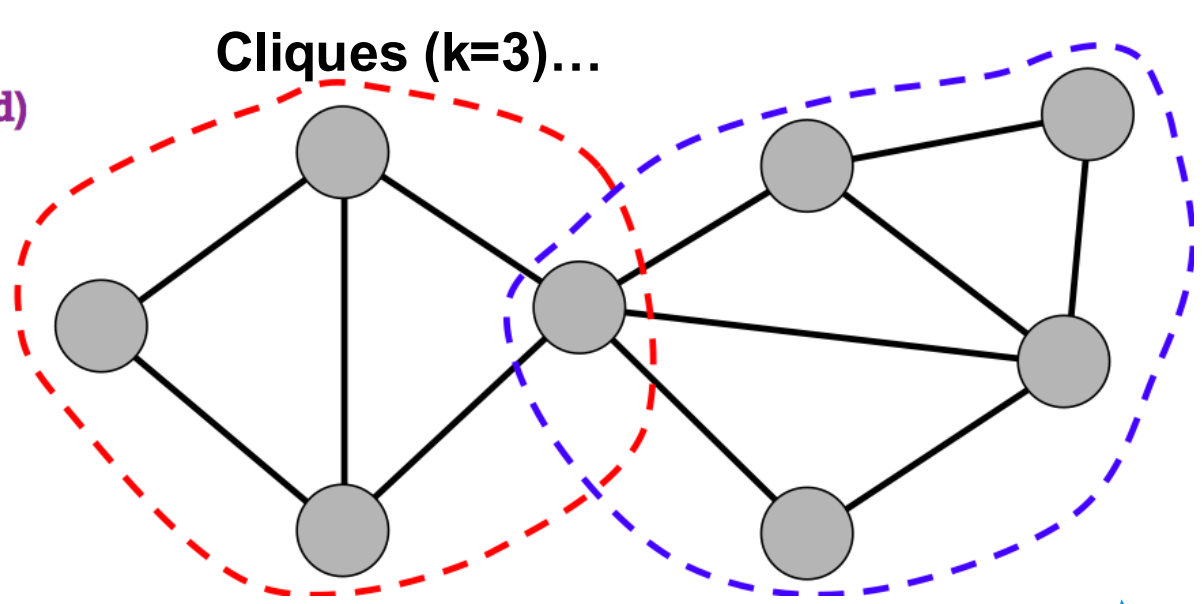


Two k cliques are considered adjacent if they share k-1 nodes

https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.community.kclique.k_clique_communities.html?highlight=community%20finding

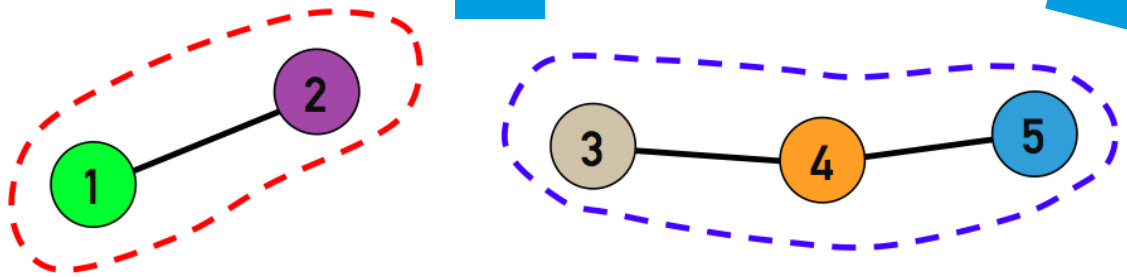
Clique percolation algorithm (http://www.cfinder.org/)

Palla et. al., Nature 446, 664 (2007)



O=

	1	2	3	4	5
1	0	1	0	0	0
2	1	0	0	0	0
3	0	0	0	1	0
4	0	0	1	0	1
5	0	0	0	1	0



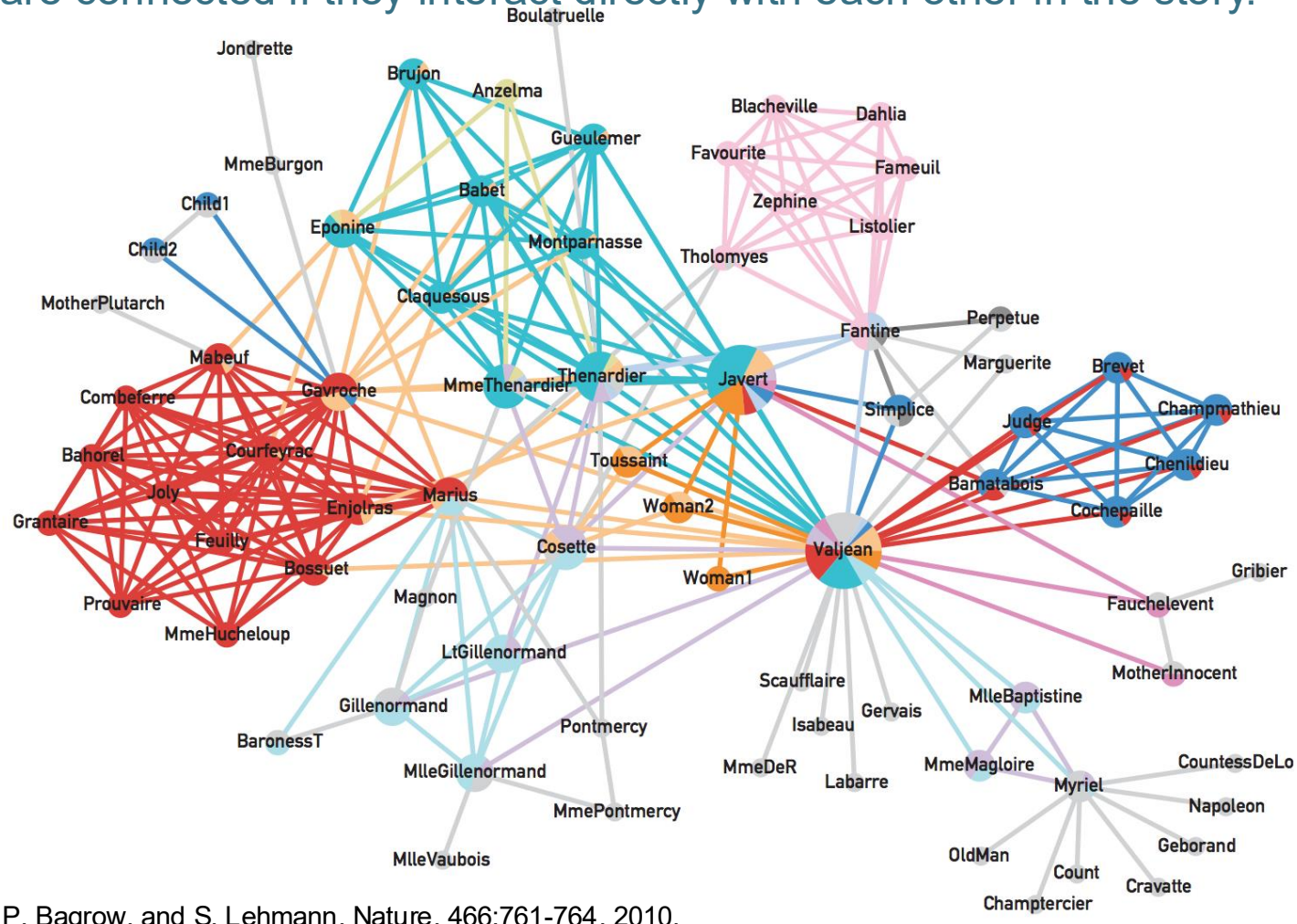
Two k cliques are considered adjacent if they share k-1 nodes

NetworkX Network Analysis in Python

https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.community.kclique.k_clique_communities.html?highlight=community%20finding

Representing overlapping communities

The network of characters in Victor Hugo's 1862 novel *Les Miserables*. Two characters are connected if they interact directly with each other in the story.



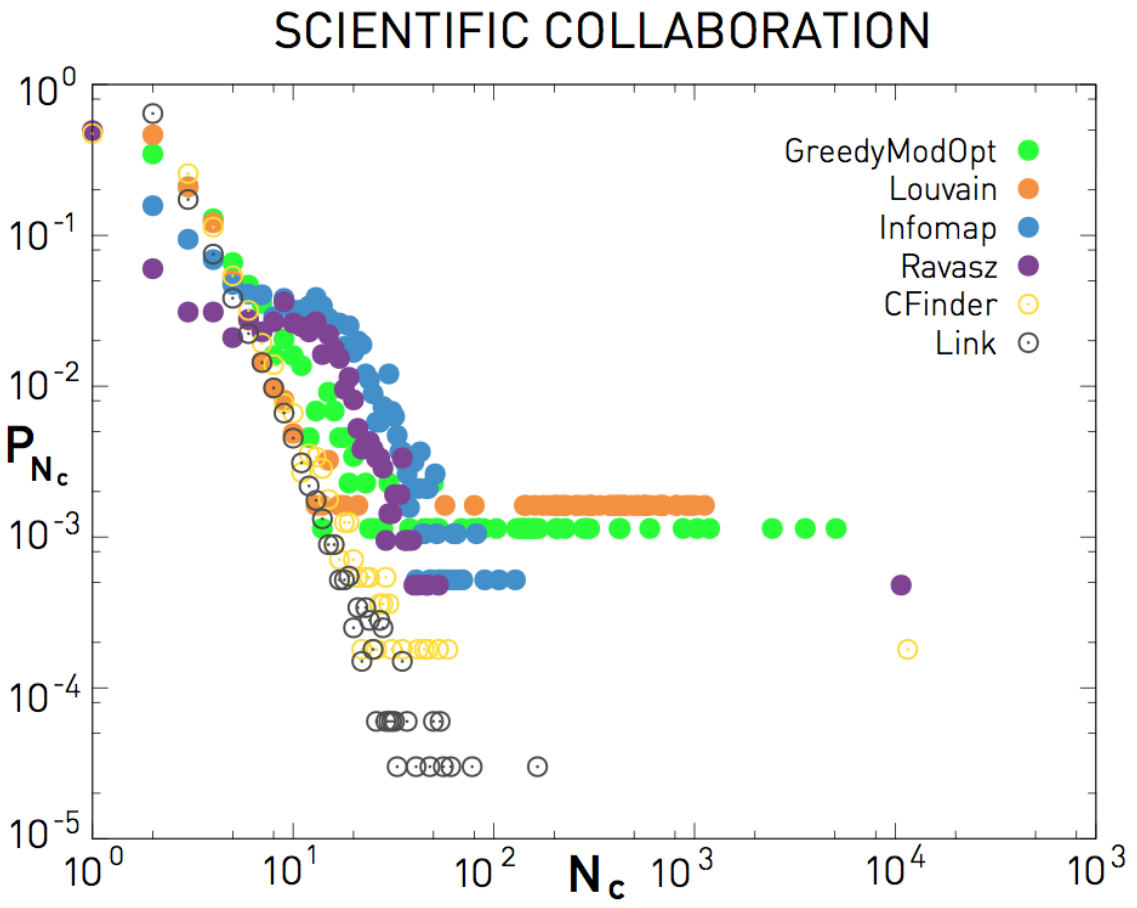
Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann, Nature, 466:761-764, 2010.

[Figure obtained with the *link clustering algorithm,* not discussed]

Community size distribution

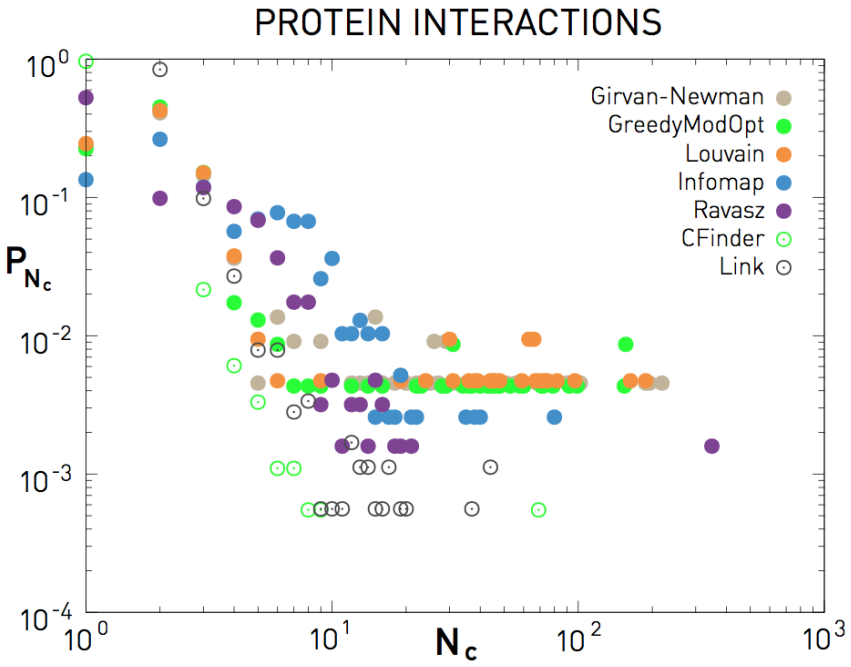
The community size distribution P_{Nc} predicted by different community finding algorithms.

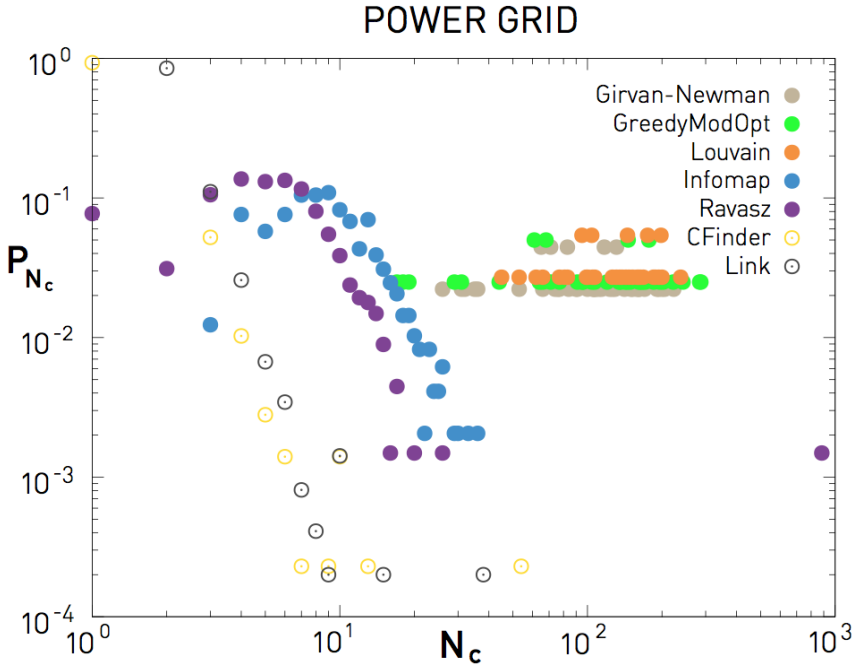
Many studies report fat tailed community size distributions, implying that numerous small communities coexist with a few very large ones.



Community size distribution

The community size distribution P_{Nc} predicted by different community finding algorithms.

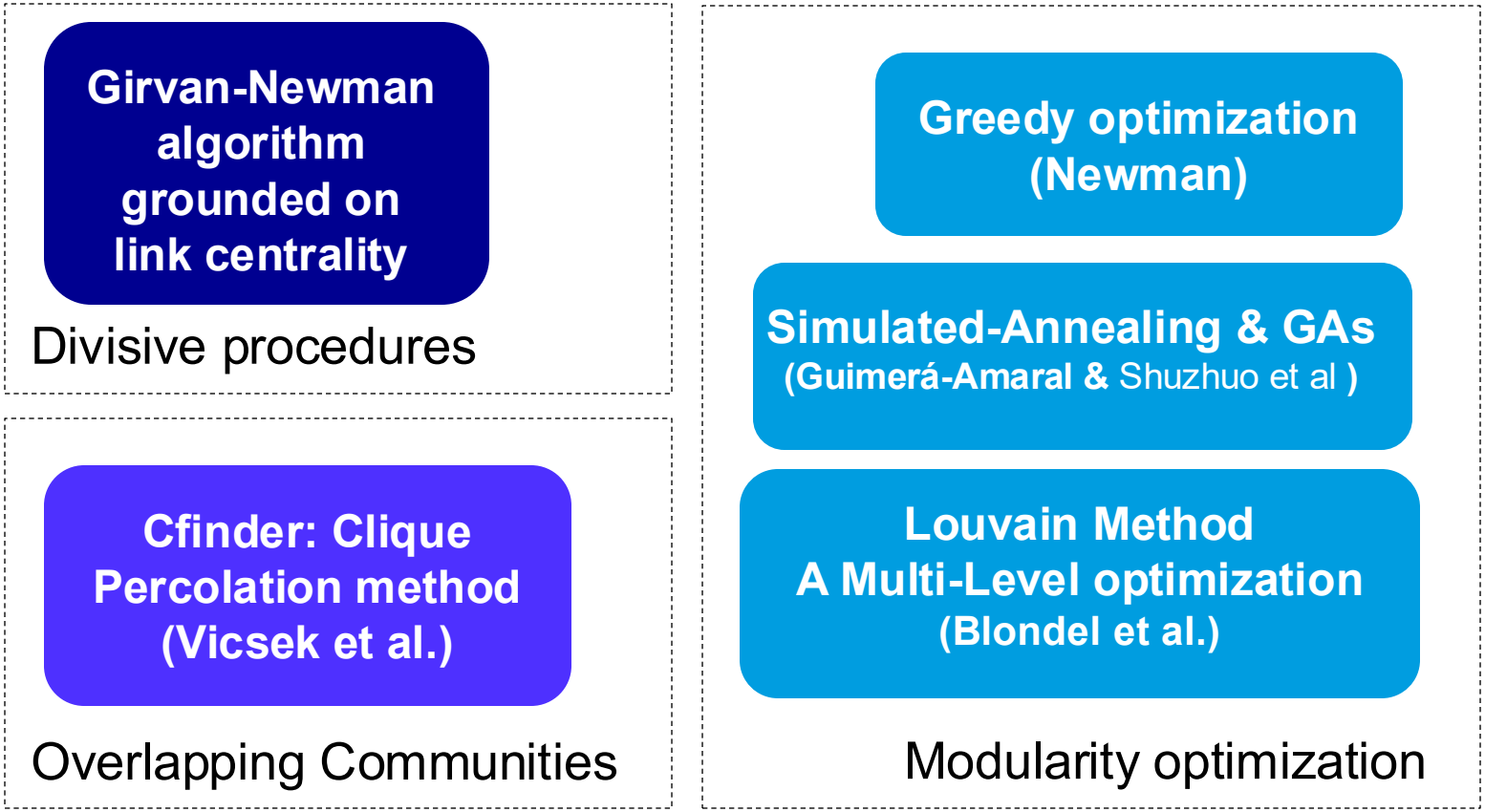




Different algorithms may offer conflicting results.

Conclusions so far

Several algorithms compete to be the most used method in community detection. Here we provided some well-known examples:



How should we assess the *running time* of each algorithm?

Algorithmic complexity

Several algorithms compete to be the most used method in community detection (n=# nodes, m=# edges)

Author	Ref.	Label	Order
Girvan & Newman	(Girvan and Newman, 2002; Newman and Girvan, 2004)	GN	$O(nm^2)$
Clauset et al.	(Clauset *et al.*, 2004)	Clauset et al.	$O(n \log^2 n)$
Blondel et al.	(Blondel *et al.*, 2008)	Blondel et al.	$O(m)$
Guimerà et al.	(Guimerà and Amaral, 2005; Guimerà *et al.*, 2004)	Sim. Ann.	parameter dependent
Radicchi et al.	(Radicchi *et al.*, 2004)	Radicchi et al.	$O(m^4/n^2)$
Palla et al.	(Palla *et al.*, 2005)	Cfinder	$O(\exp(n))$
Van Dongen	(Dongen, 2000a)	MCL	$O(nk^2)$, $k < n$ parameter
Rosvall & Bergstrom	(Rosvall and Bergstrom, 2007)	Infomod	parameter dependent
Rosvall & Bergstrom	(Rosvall and Bergstrom, 2008)	Infomap	$O(m)$
Donetti & Muñoz	(Donetti and Muñoz, 2004, 2005)	DM	$O(n^3)$
Newman & Leicht	(Newman and Leicht, 2007)	EM	parameter dependent
Ronhovde & Nussinov	(Ronhovde and Nussinov, 2009)	RN	$O(m^\beta \log n)$, $\beta \sim 1.3$

Algorithmic complexity

Several algorithms compete to be the most used method in community detection (n=# nodes, m=# edges)

Author	Ref.	Label	Order
Eckmann & Moses	(Eckmann and Moses, 2002)	EM	$O(m\langle k^2\rangle)$
Zhou & Lipowsky	(Zhou and Lipowsky, 2004)	ZL	$O(n^3)$
Latapy & Pons	(Latapy and Pons, 2005)	LP	$O(n^3)$
Clauset et al.	(Clauset *et al.*, 2004)	NF	$O(n \log^2 n)$
Newman & Girvan	(Newman and Girvan, 2004)	NG	$O(nm^2)$
Girvan & Newman	(Girvan and Newman, 2002)	GN	$O(n^2 m)$
Guimerà et al.	(Guimerà and Amaral, 2005; Guimerà *et al.*, 2004)	SA	parameter dependent
Duch & Arenas	(Duch and Arenas, 2005)	DA	$O(n^2 \log n)$
Fortunato et al.	(Fortunato *et al.*, 2004)	FLM	$O(m^3 n)$
Radicchi et al.	(Radicchi *et al.*, 2004)	RCCLP	$O(m^4/n^2)$
Donetti & Muñoz	(Donetti and Muñoz, 2004, 2005)	DM/DMN	$O(n^3)$
Bagrow & Bollt	(Bagrow and Bollt, 2005)	BB	$O(n^3)$
Capocci et al.	(Capocci *et al.*, 2005)	CSCC	$O(n^2)$
Wu & Huberman	(Wu and Huberman, 2004)	WH	$O(n+m)$
Palla et al.	(Palla *et al.*, 2005)	PK	$O(\exp(n))$
Reichardt & Bornholdt	(Reichardt and Bornholdt, 2004)	RB	parameter dependent

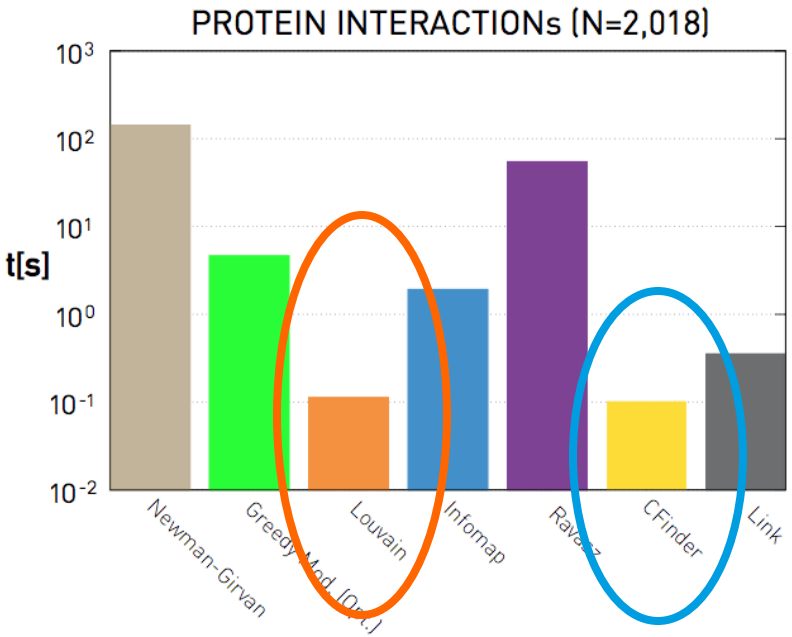
Running time

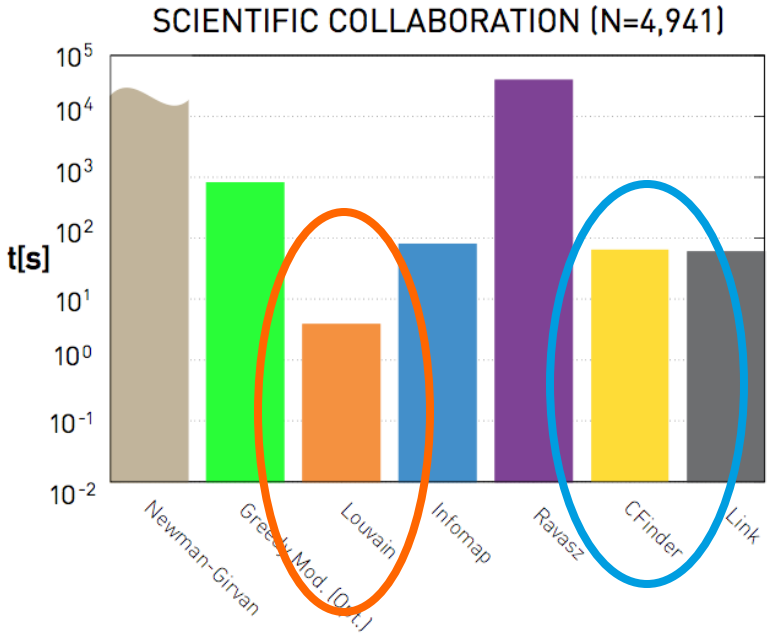
- While community identification algorithms do not check all partitions, their computational cost still varies widely, determining their speed and consequently the size of the network they can handle.

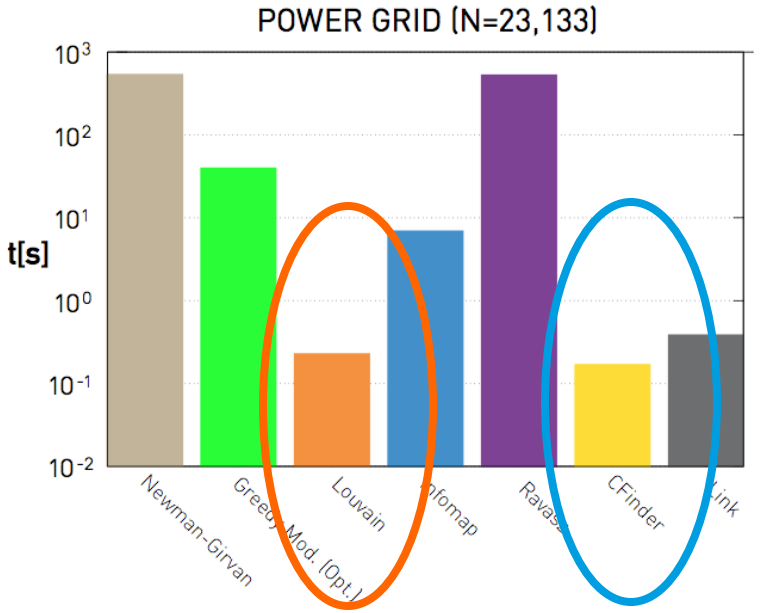
- The computational complexity of the algorithms shows that one of the most efficient is the Louvain algorithm (~O(Nlog$N$)). The least efficient is CFinder with O(e^N).

- This scaling matters if we need to find communities in very large networks. To get a sense of the true speed of these algorithms we may also measure their actual running time in real datasets.

Running time







How should we assess the *accuracy* of each algorithm?

Let us check two popular benchmarks, artificial nets with a pre-defined community structure.

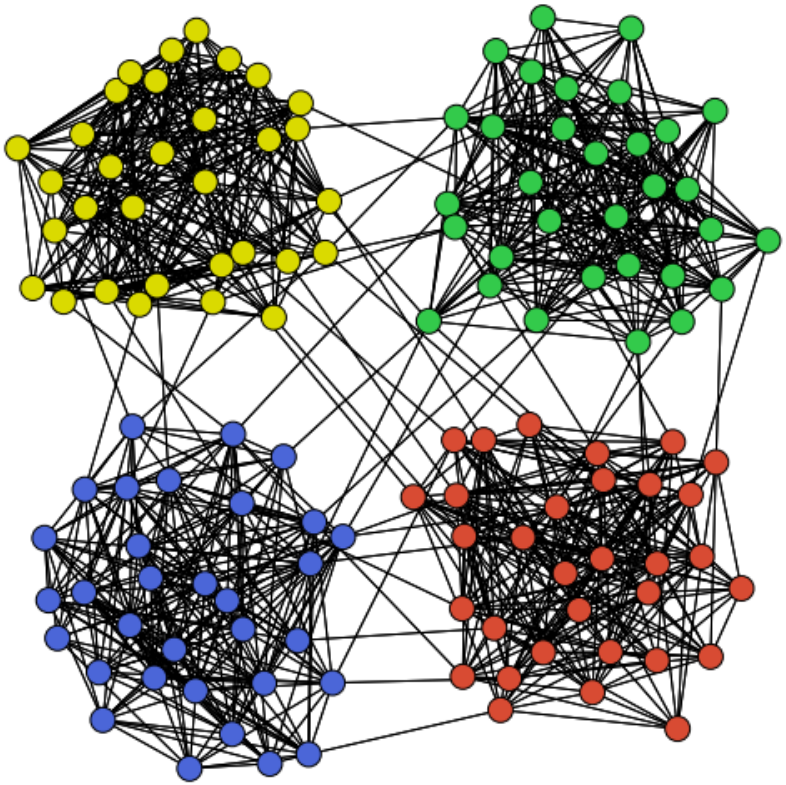
Girvan-Newman (GN) Benchmark

An example of a neat benchmark consists of N=128 nodes partitioned into n_c=4 communities of size N_c=32.

Each node is connected with probability p_{int} to the N_c–1 nodes in its community and with probability p_{ext} to the $3N_c$ nodes in the other three communities.

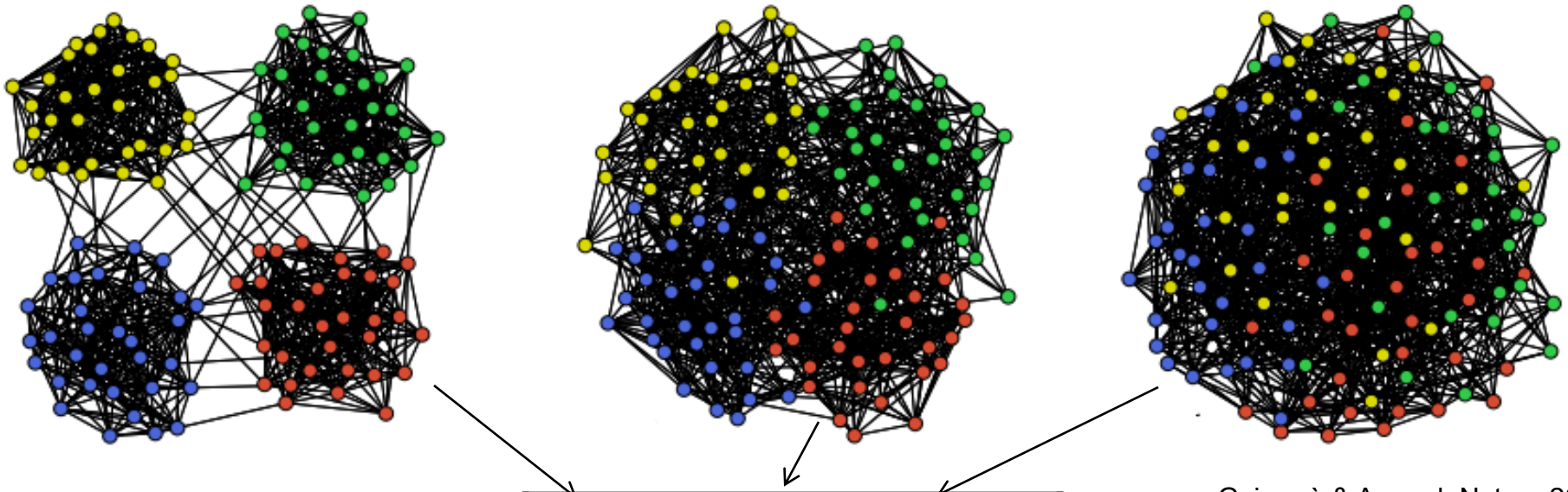
This creates a control parameter

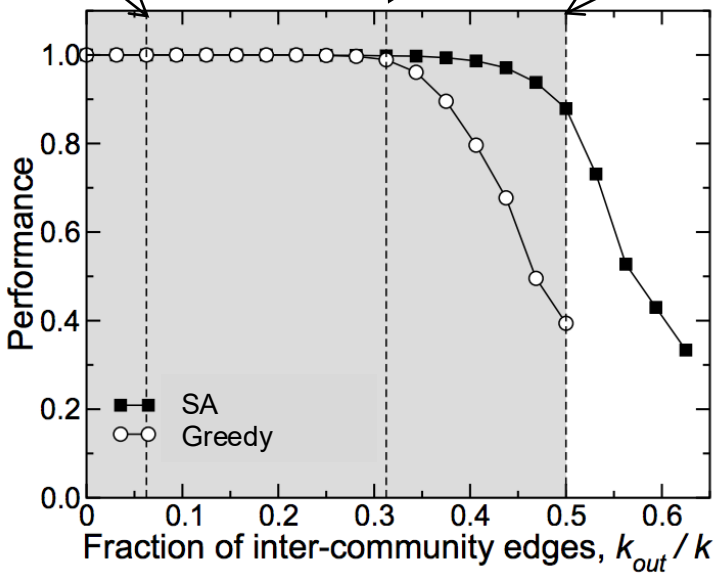
$$\mu = \frac{k^{ext}}{k^{ext} + k^{int}}$$



Girvan-Newman (GN) Benchmark

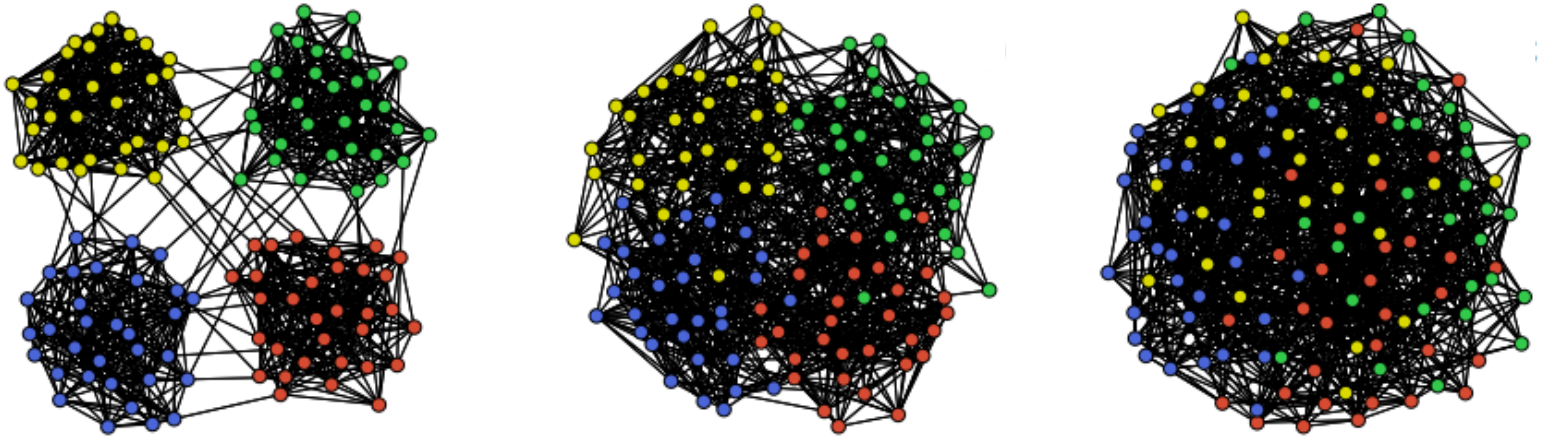
Example: Greedy optimization vs simulated annealing (SA)

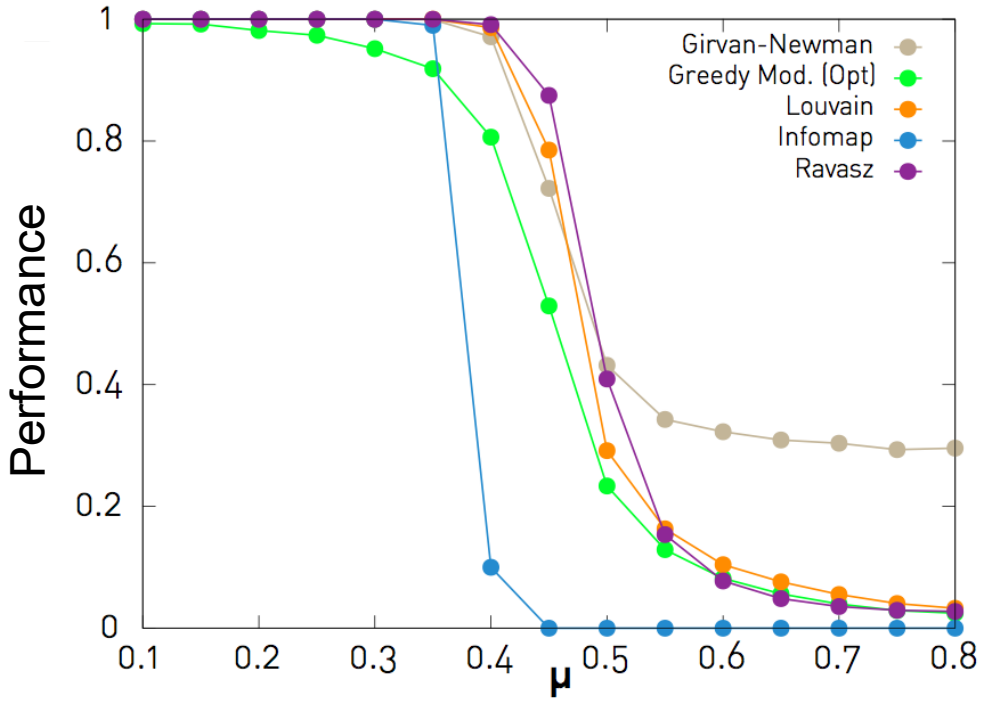




Guimerà & Amaral, Nature 2005

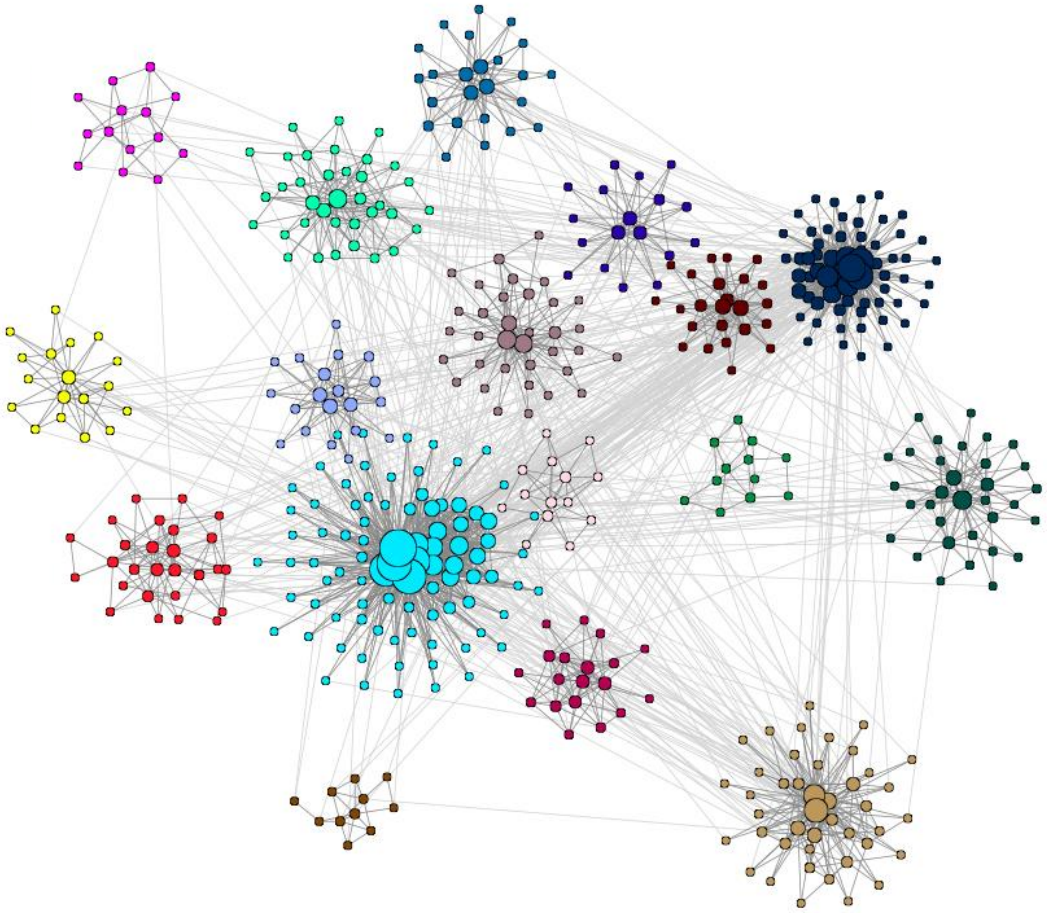
Girvan-Newman (GN) Benchmark





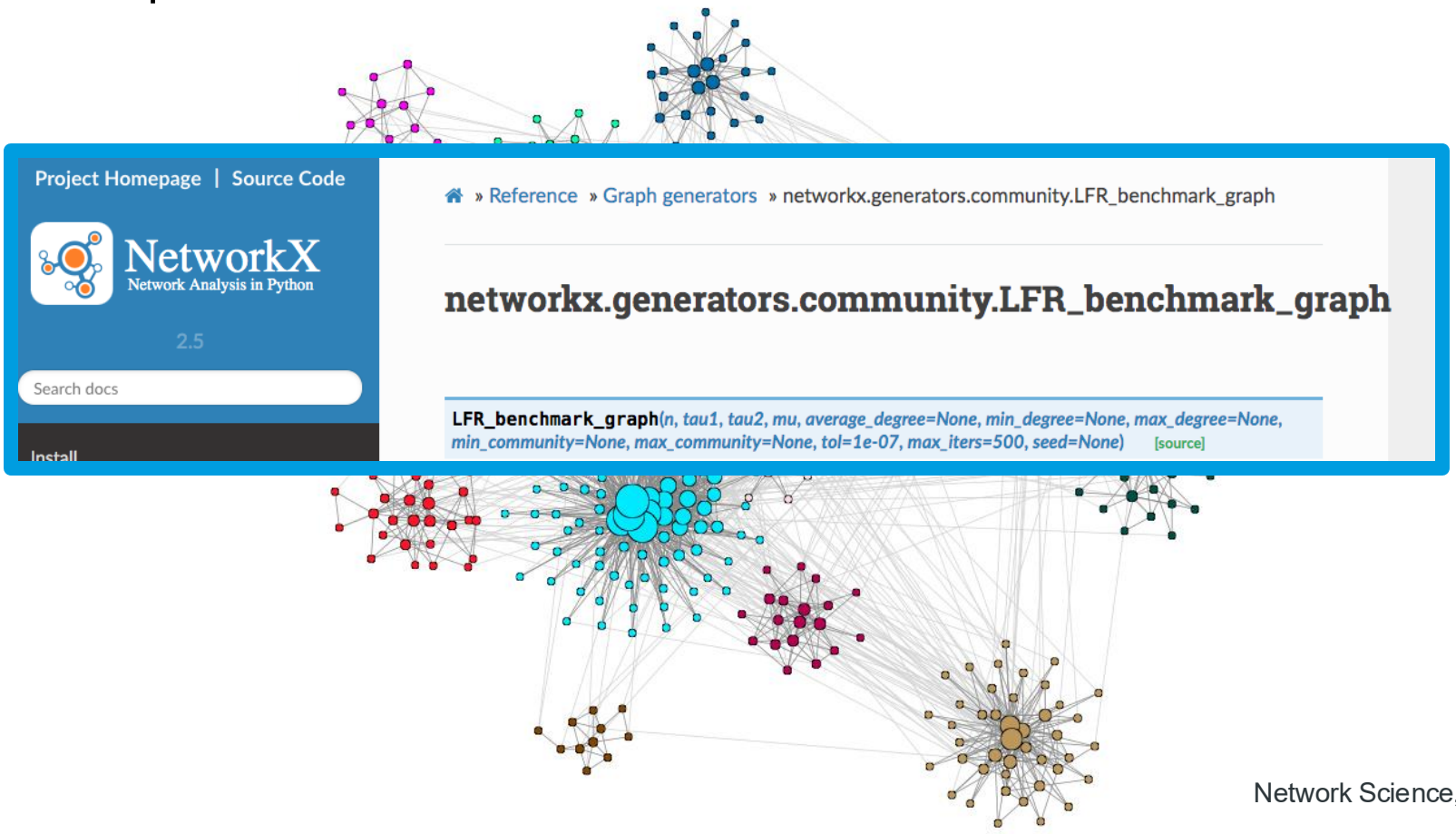
Lancichinetti-Fortunato-Radicchi (LFR) Benchmark

Generates networks in which both the node degrees and community sizes follow a power law.



Lancichinetti-Fortunato-Radicchi (LFR) Benchmark

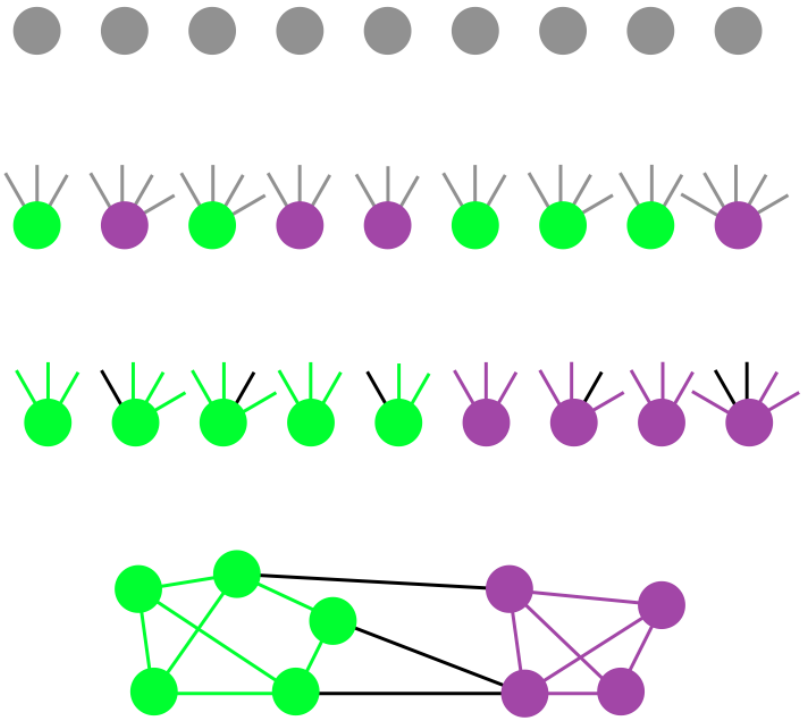
Generates networks in which both the node degrees and community sizes follow a power law.



Lancichinetti-Fortunato-Radicchi (LFR) Benchmark

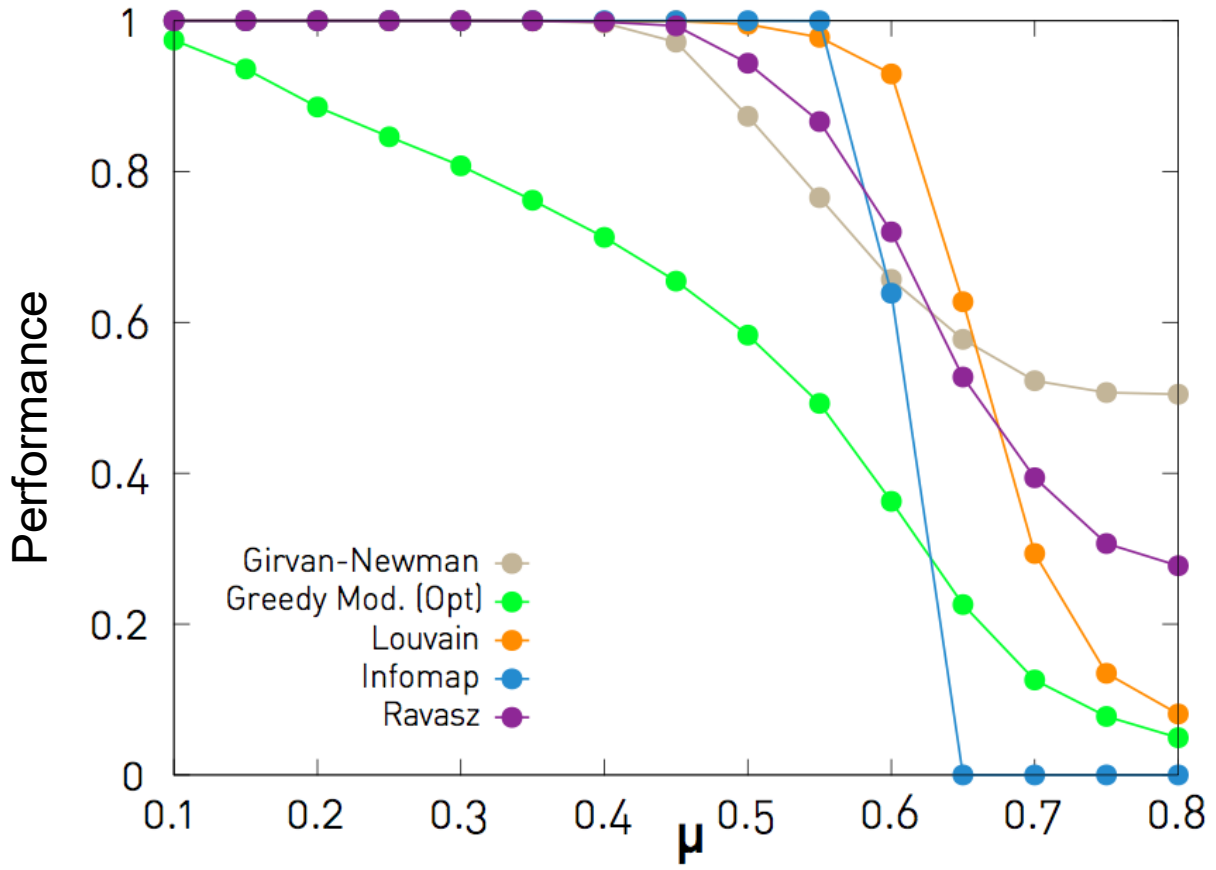
Generates networks in which both the node degrees and community sizes follow a power law.

① Start with N isolated nodes.
② Assign each node to a community (of size N_c) where N_c follows the power law dist. Also assign a degree to each node taken from a power-law dist.
③ Each receives an internal degree $(1-\mu)k_i$ and an external degree (μk_i).
④ Randomly attach open stubs taking into account internal and external links.



Lanchinetti-Fortunato-Radicchi (LFR) Benchmark

Once again de Louvain method provides good results.



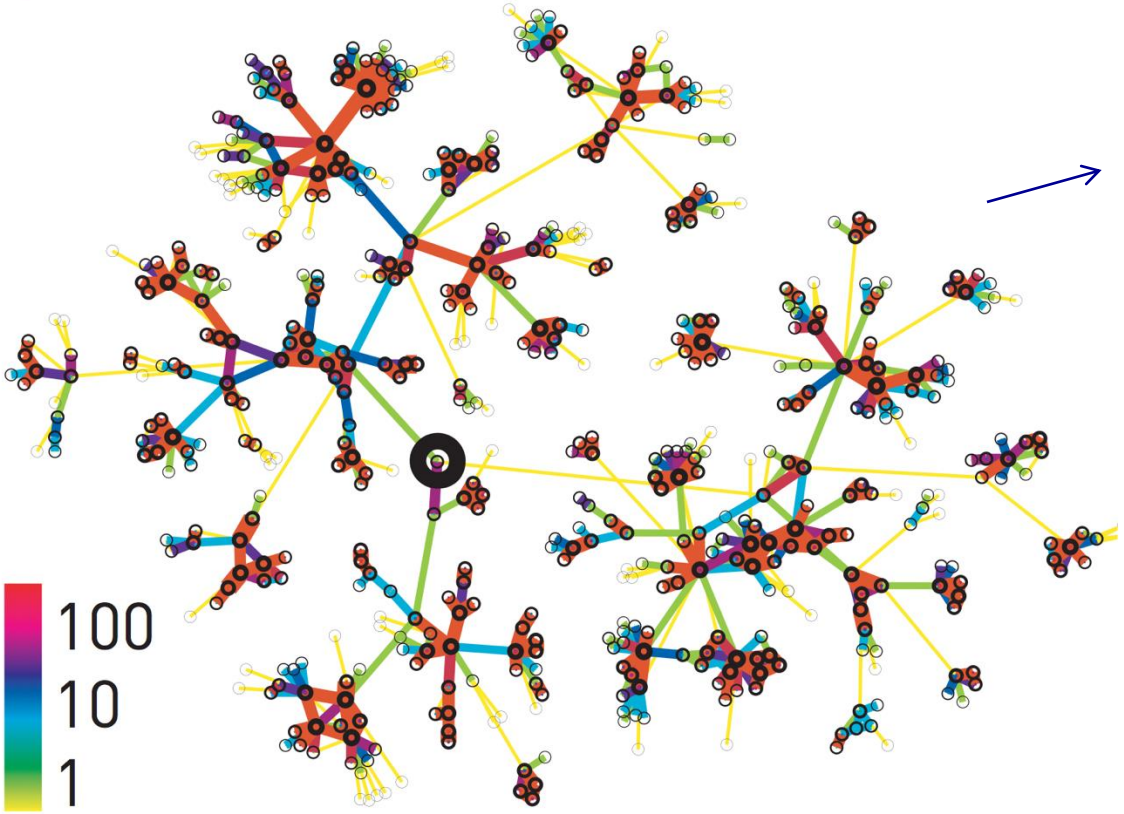
Other ways

Online tool for quantitative assessment of classification agreement

`comparingpartitions.info`

Final remarks: Communities and link weights

Social Networks. The more time two individuals spend together, the more likely that they share friends. Consequently communities in social networks tend to be nucleated around strong ties. Links connecting different communities are weaker in comparison.

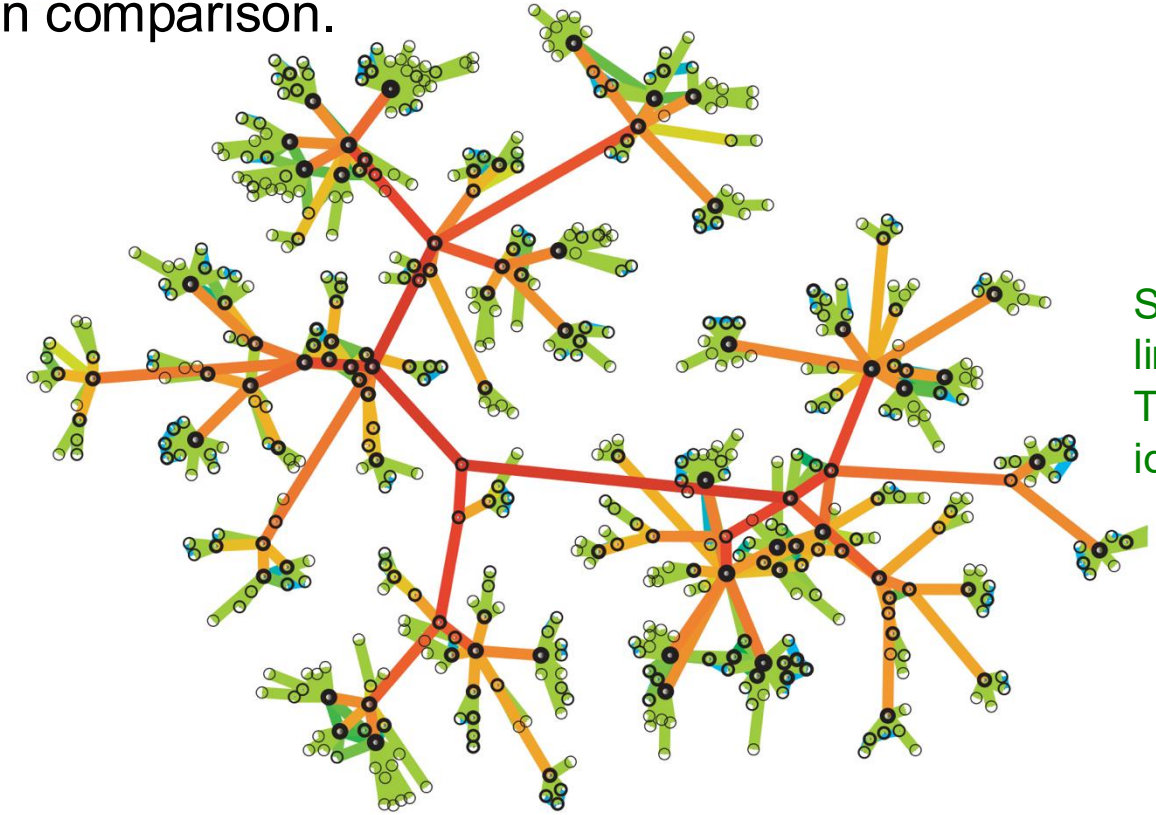


Mobile phone network

Good illustration of Granovetter's "Strength of Weak Ties"

Final remarks: Communities and link weights

Biological and Technological Networks. If the link weights are driven by the need to transport information or materials, as it is often the case in technological and biological systems, strong ties are between communities. In contrast links within communities are weaker in comparison.

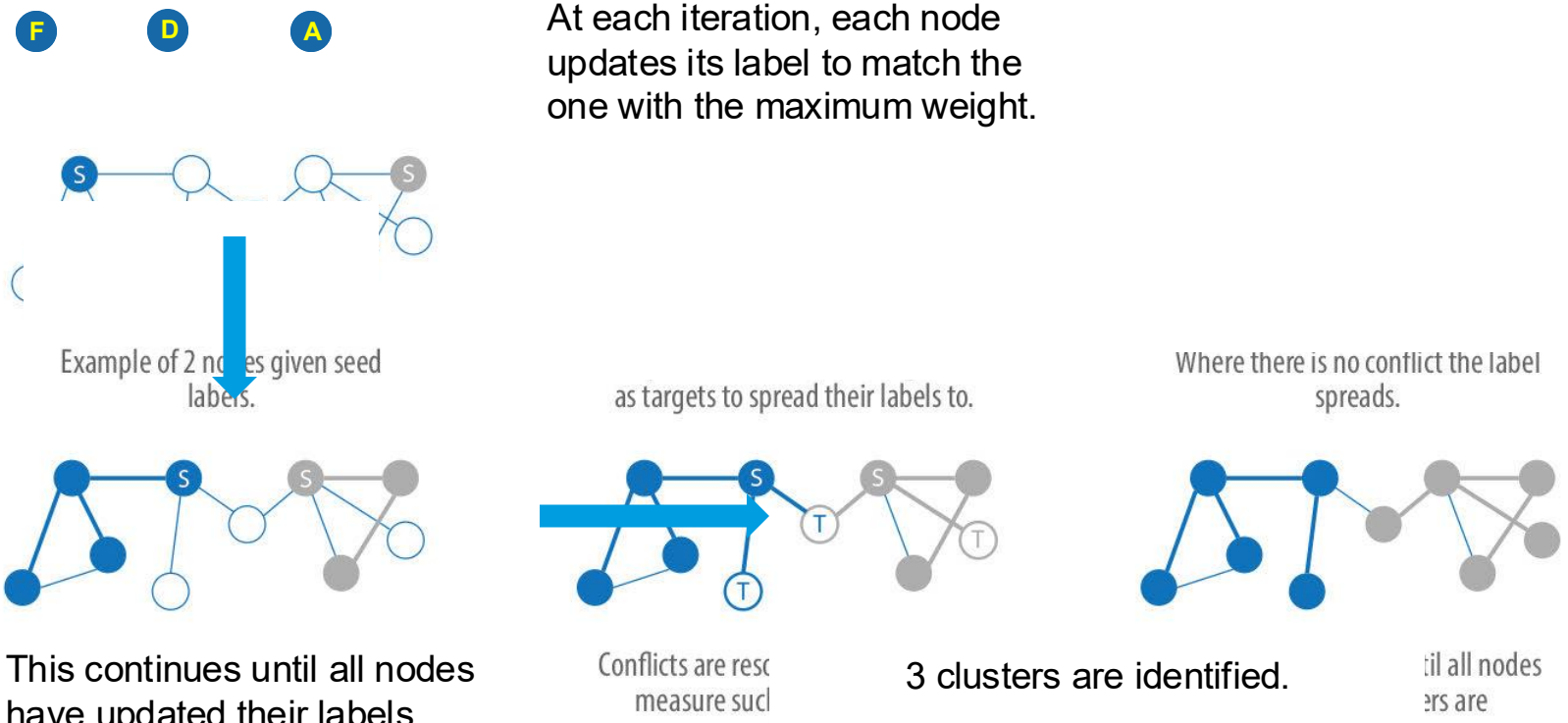


Same network, yet highlighting link's betweeness centrality. This is the idea of Granovetter's idea of the *strength of weak ties*.

LPA – Label Propagation Algorithm

U. N. Raghavan, R. Albert, and S. Kumara (Phys Rev E 2007)

Typical steps of the Label Propagation Algorithm:



At each iteration, each node updates its label to match the one with the maximum weight.

3 clusters are identified.

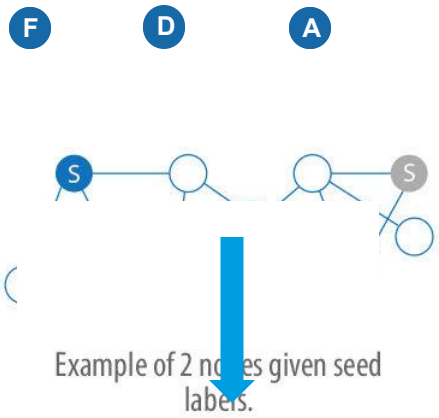
NetworkX Network Analysis in Python

`label_propagation_communities(G)` [source]

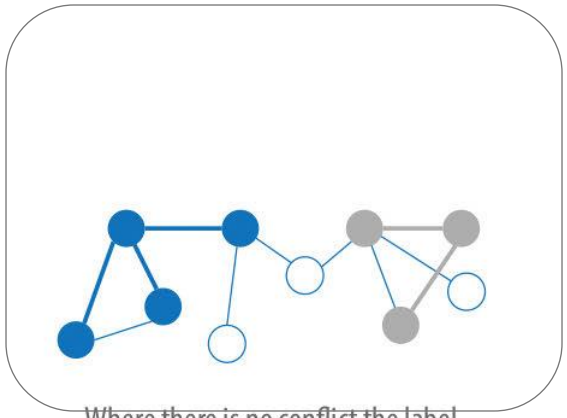
LPA – Label Propagation Algorithm

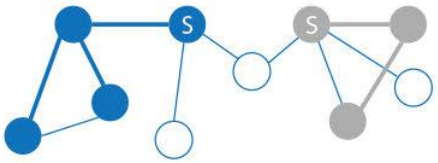
U. N. Raghavan, R. Albert, and S. Kumara (Phys Rev E 2007)

Typical steps of the Label Propagation Algorithm:

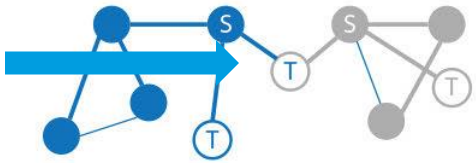


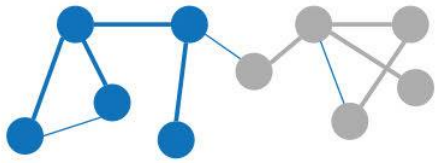
At each iteration, each node updates its label to match the one with the maximum weight.





This continues until all nodes have updated their labels





3 clusters are identified.

til all nodes
ers are

NetworkX
Network Analysis in Python

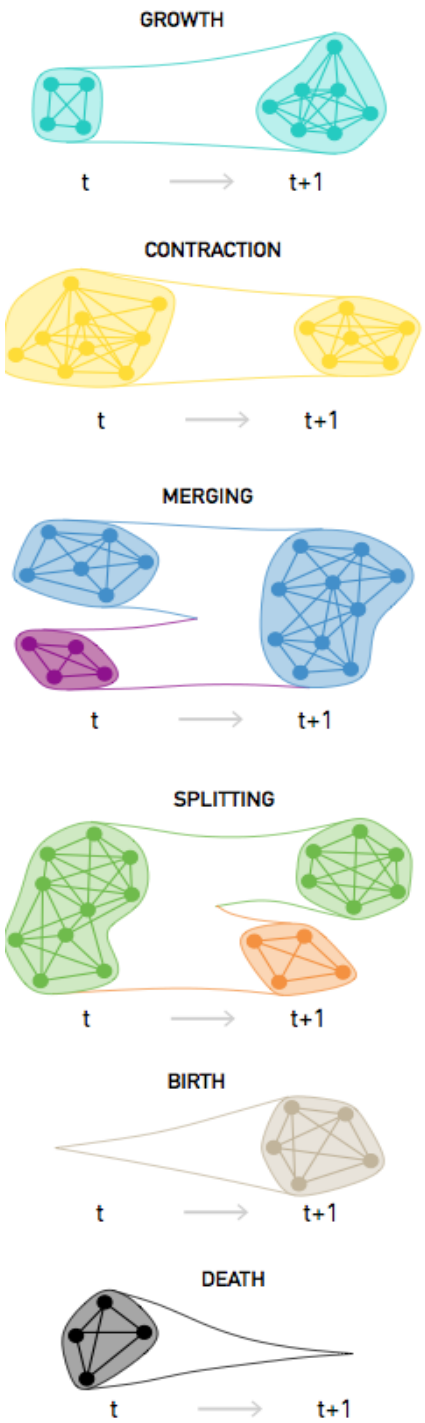
`label_propagation_communities(G)` [source]

Final remarks: Temporal networks

Changes in a network's wiring diagram can have multiple consequences for communities: birth of new communities, growth and contraction, communities can merge with each other or split into several smaller communities, etc.

Studies focusing on social and communication networks offer several insights into the changes communities experience.

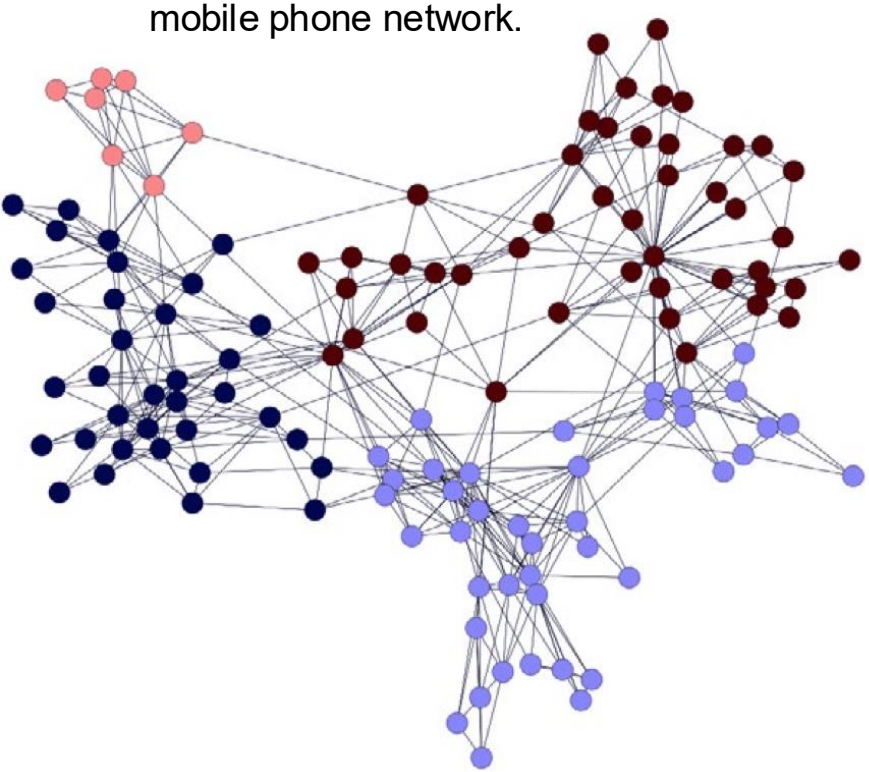
For instance, membership of large communities changes faster with time than the membership of smaller communities.



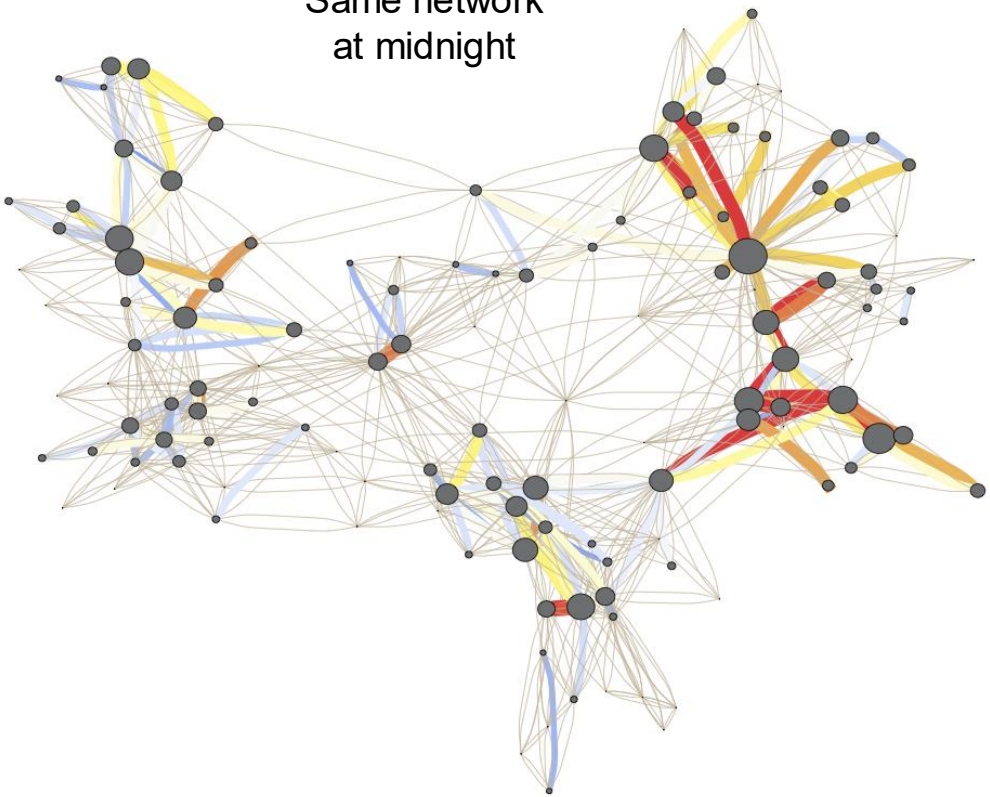
Final remarks: Temporal networks

Also, communities may be cyclic and completely change in time. Depending on how information is aggregated with may get completely different results. Example: Call patterns

Four communities in a sample of a mobile phone network.



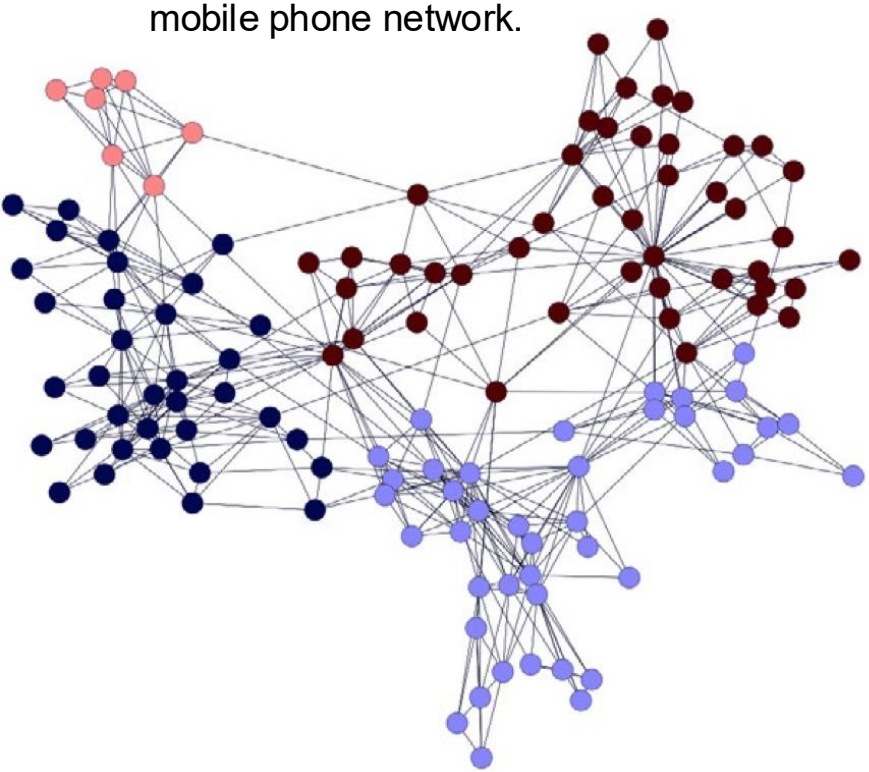
Same network at midnight



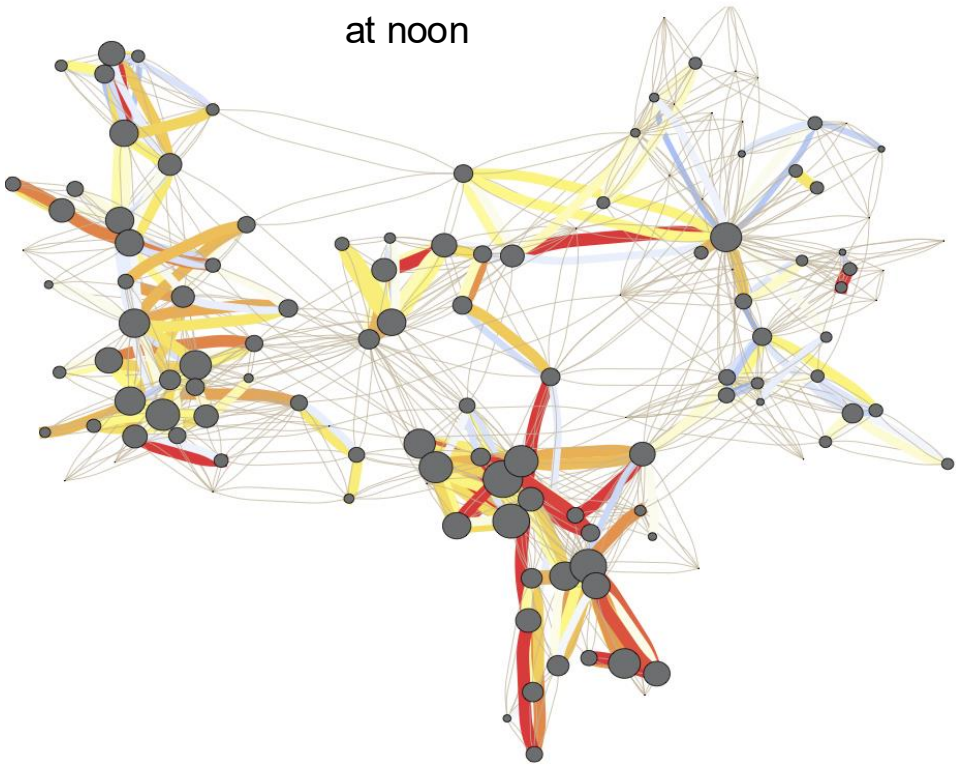
Final remarks: Temporal networks

Also, communities may be cyclic and completely change in time. Depending on how information is aggregated with may get completely different results. Example: Call patterns

Four communities in a sample of a mobile phone network.



Same network at noon



Final remarks: Other questions

- **Do We Really Have Communities?**

Community finding algorithms are designed to identify communities, whether they are there or not.

- **Must all Nodes Belong to Communities?**

Current algorithms force all nodes into communities.

- **Can we cope with dense networks?**

Most networks are sparse. Yet, with improvements in data collection, many real network maps will likely gain numerous links and overlapping communities, calling for new algorithms.

Final remarks: Other questions

- **Community vs Function?**

Can we correlate communities and functions in biological networks.

- **How much we will gain from parallelizing these algorithms?**

Must current algorithms do not take this possibility into account (note: it can be difficult!).

- **Problem specific algorithms**

community finding algorithms run behind many social networks sites, like Facebook, Twitter, or LinkedIn, demanding specific requirements and modularity measures.

Understanding of community organization continues to develop rapidly, offering increasingly accurate tools to diagnose the local structure of large networks.