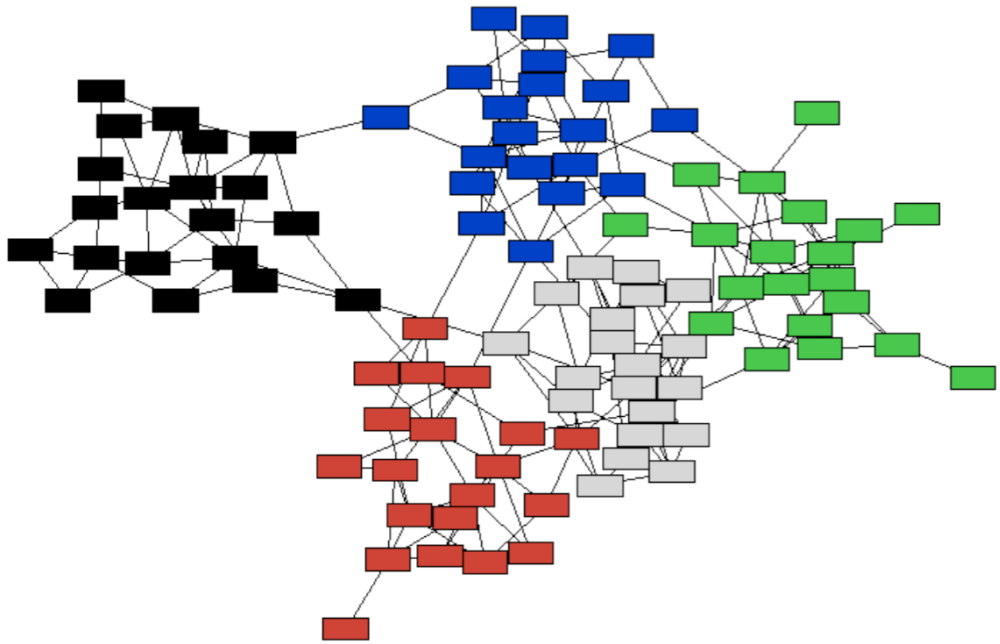
Beyond the degree distribution

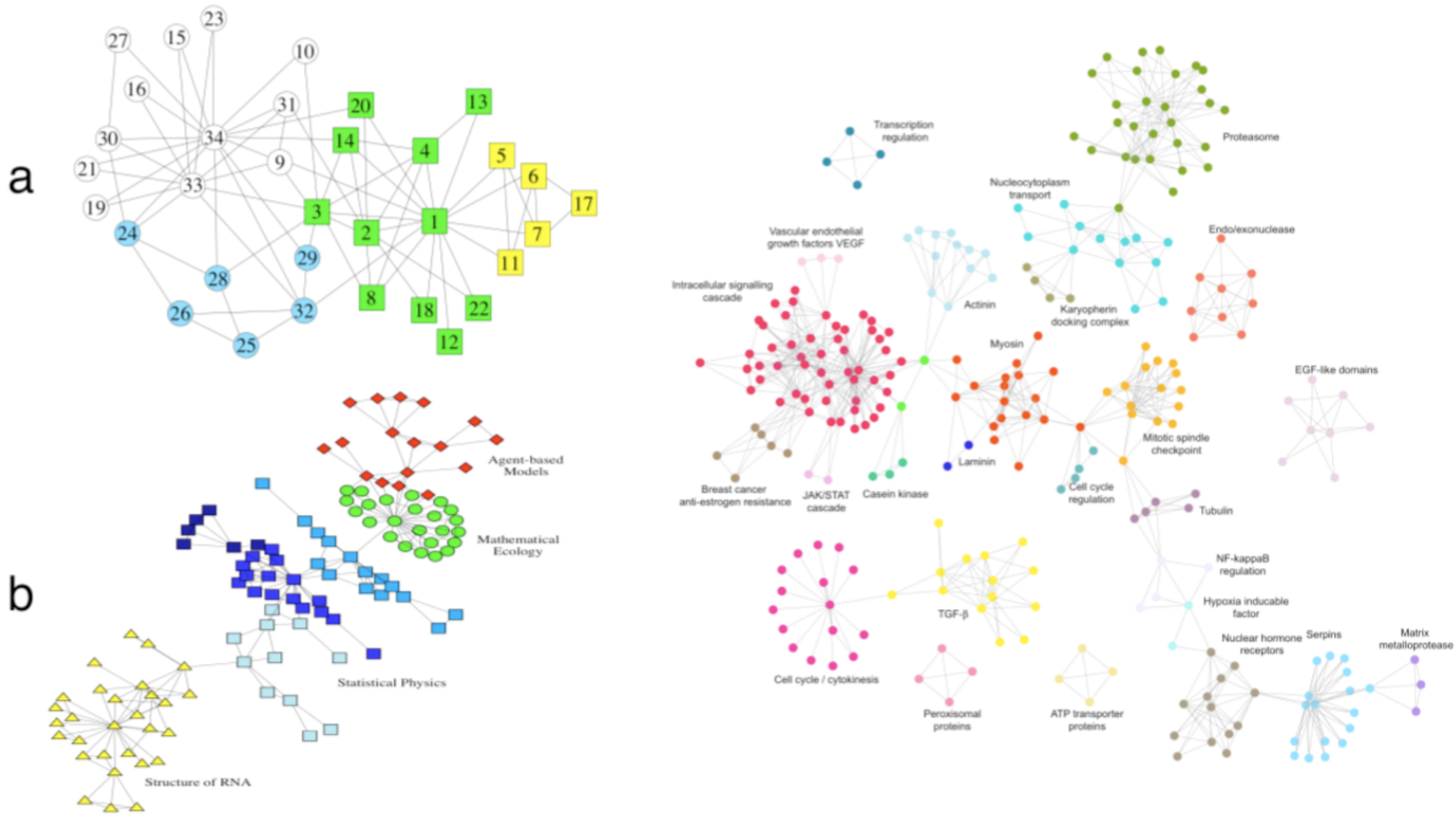
clustering coefficient
density of cliques
motifs
k-core
degree-degree correlations

Modularity:
Many networks are inhomogeneous and are made of modules: many links within modules and a few links between different modules



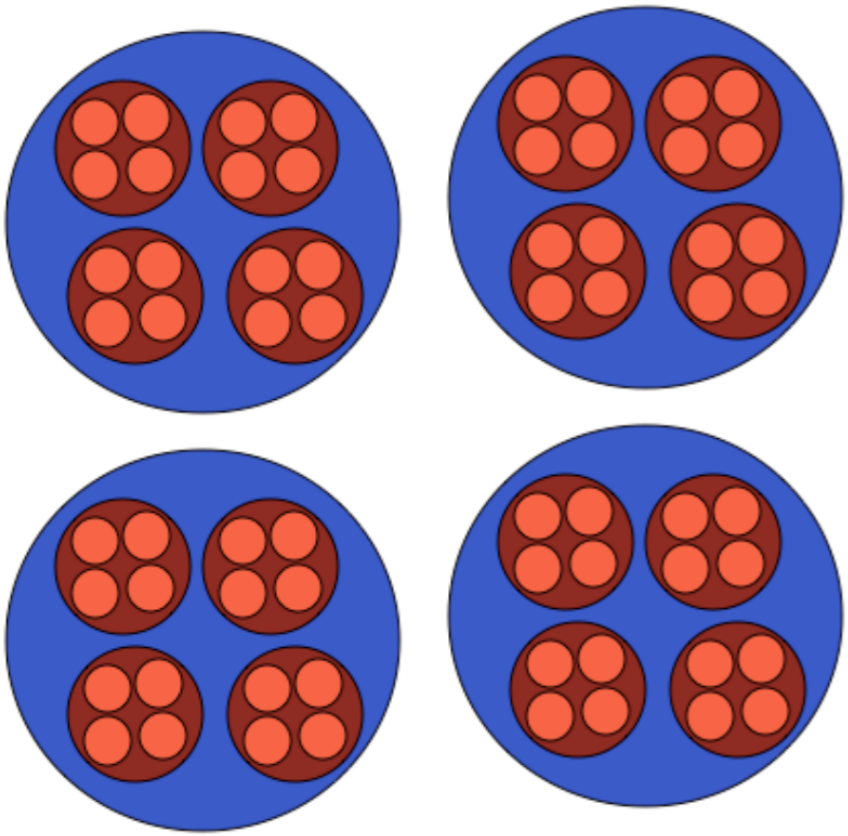
Properties: Modularity

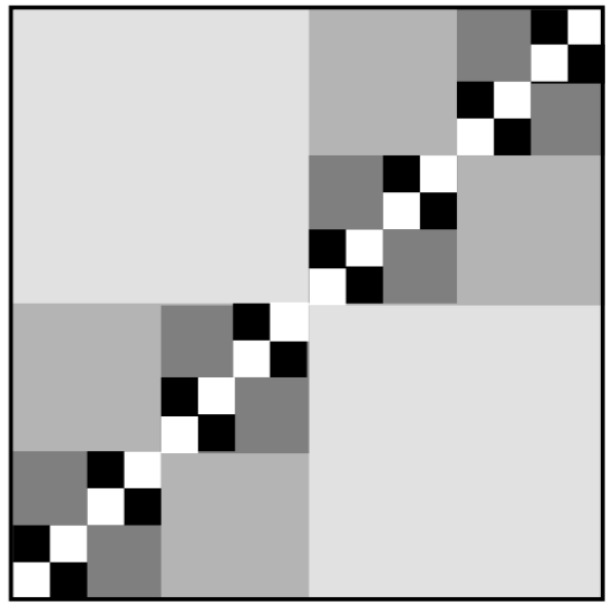
Observed in social, biological and information networks



Properties: Multi-level modularity (hierarchy)

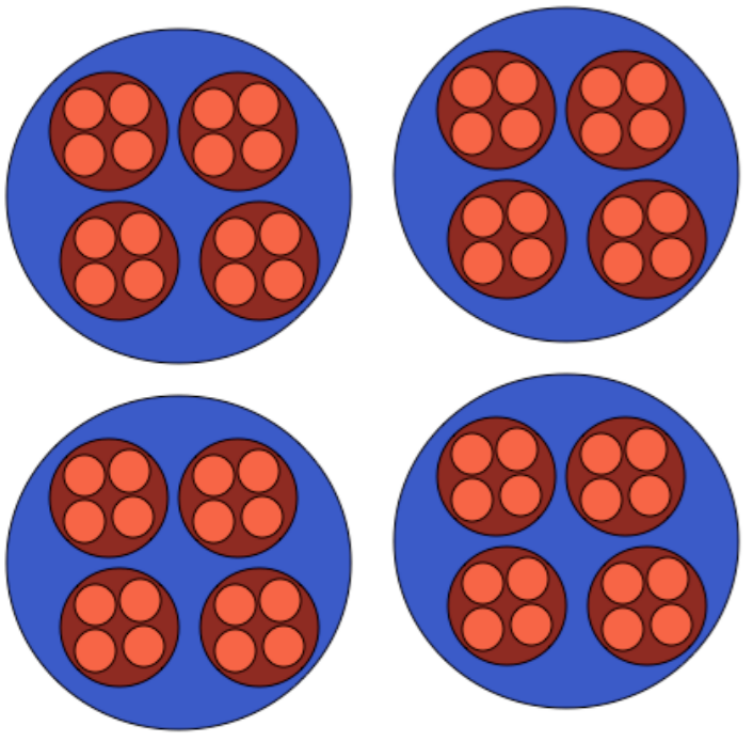
Networks have a hierarchical/multi-scale structure: modules within modules
Nested organization



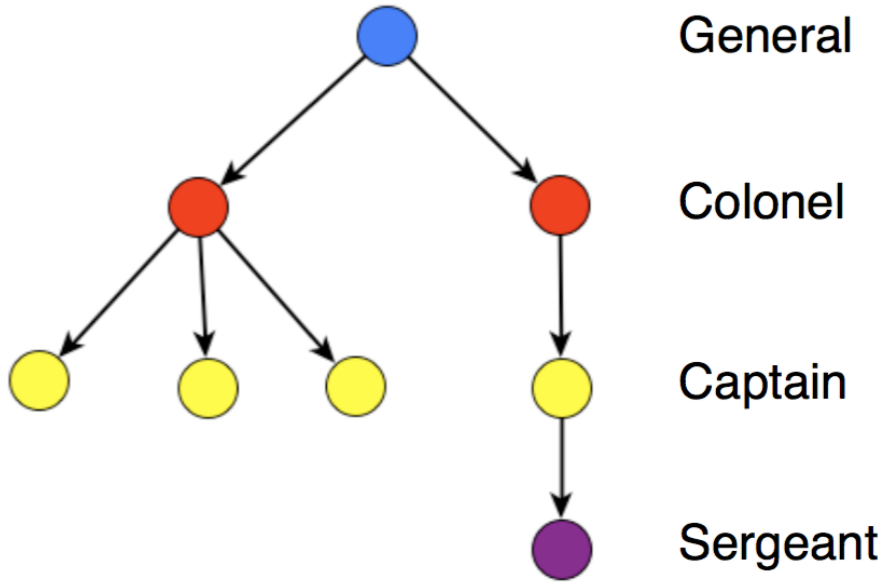


nb. Different notions of hierarchy

Hierarchy = multi-scale structure: modules within modules

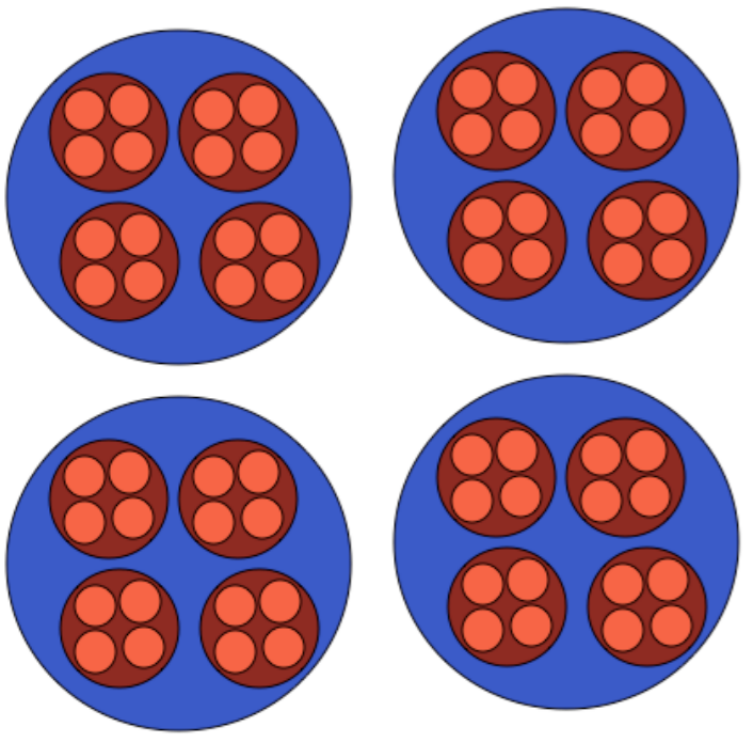


Hierarchy = subordination

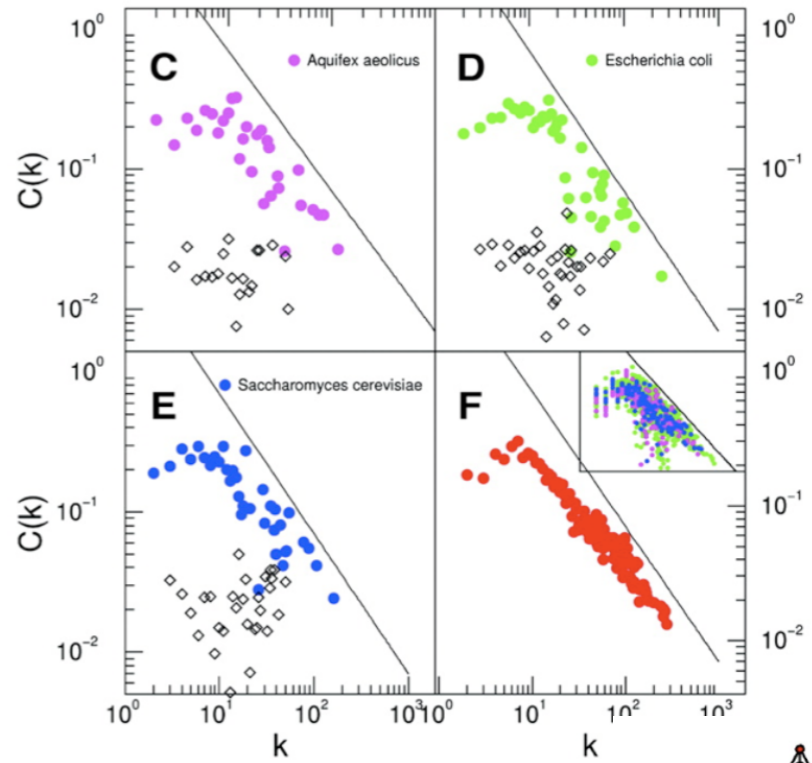


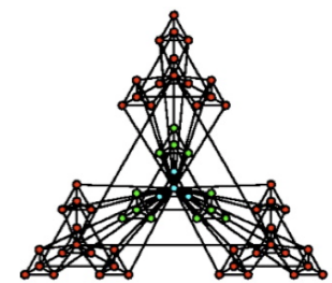
nb. Different notions of hierarchy

Hierarchy = multi-scale structure: modules within modules



Hierarchy of nodes with different degrees of “modularity” (clustering)





Hierarchical Organization of Modularity in Metabolic Networks, E. Ravasz et al.
http://barabasi.com/f/108.pdf

Random networks with communities

Generalization to arbitrary number of communities, distribution of community size and degree distribution

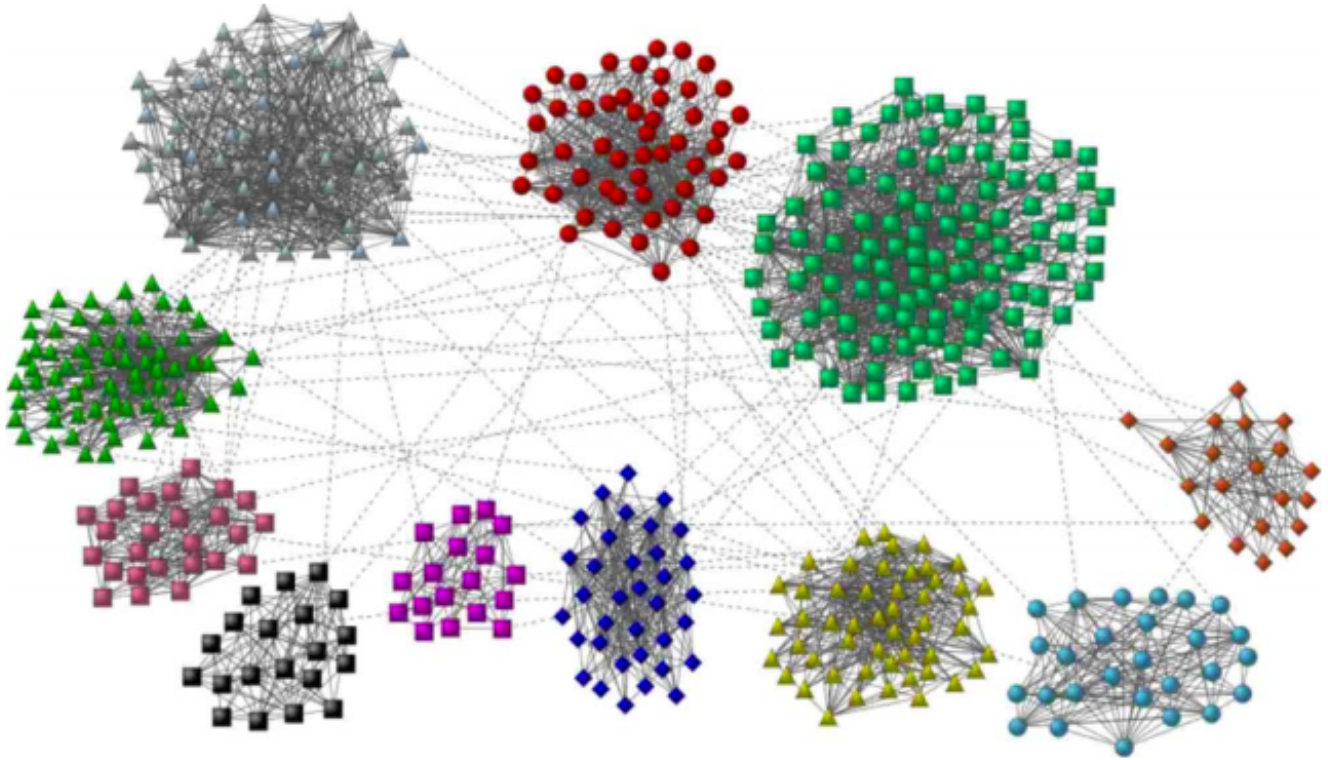
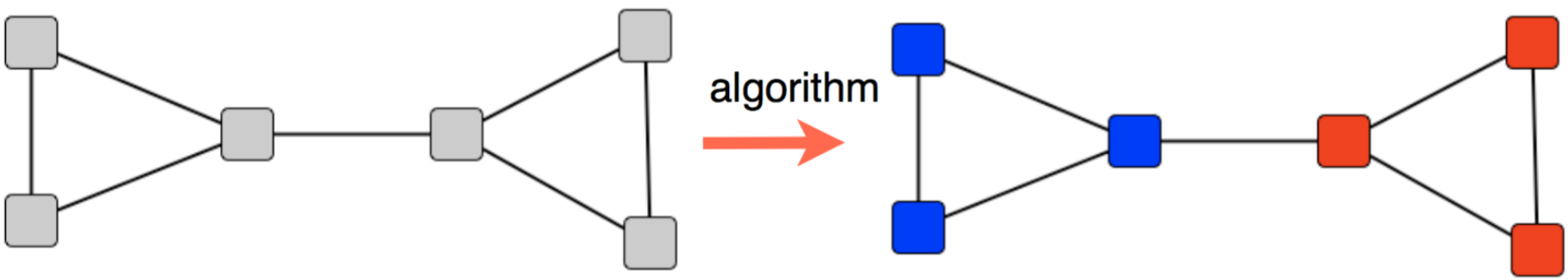


FIG. 1. (Color online) A realization of the new benchmark, with 500 nodes.

Benchmark graphs for testing community detection algorithms, Andrea Lancichinetti et al.
https://6c131308-a-62cb3a1a-s-sites.googlegroups.com/site/andrealancichinetti/benchmark.pdf

What is Community Detection?

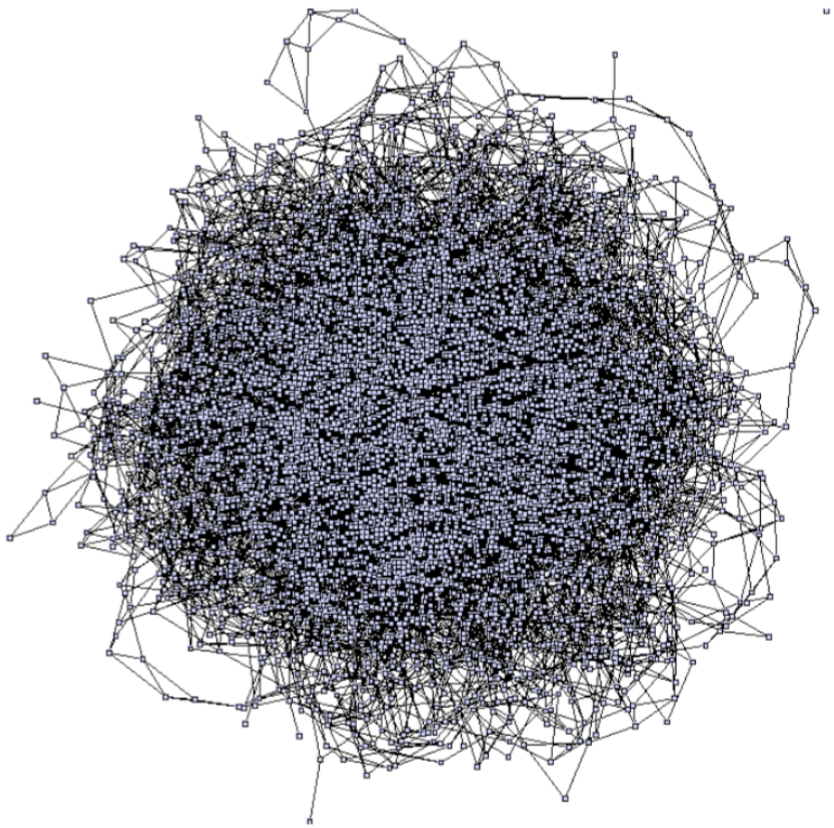
Is it possible to uncover the (multi-scale) modular organisation of networks in an automated fashion? And please avoid false positives.
Given a graph, we look for an algorithm able to uncover its modules without specifying their number nor their size
The method should be scalable to accomodate very large networks, as often observed in the real-world.



Why community detection?

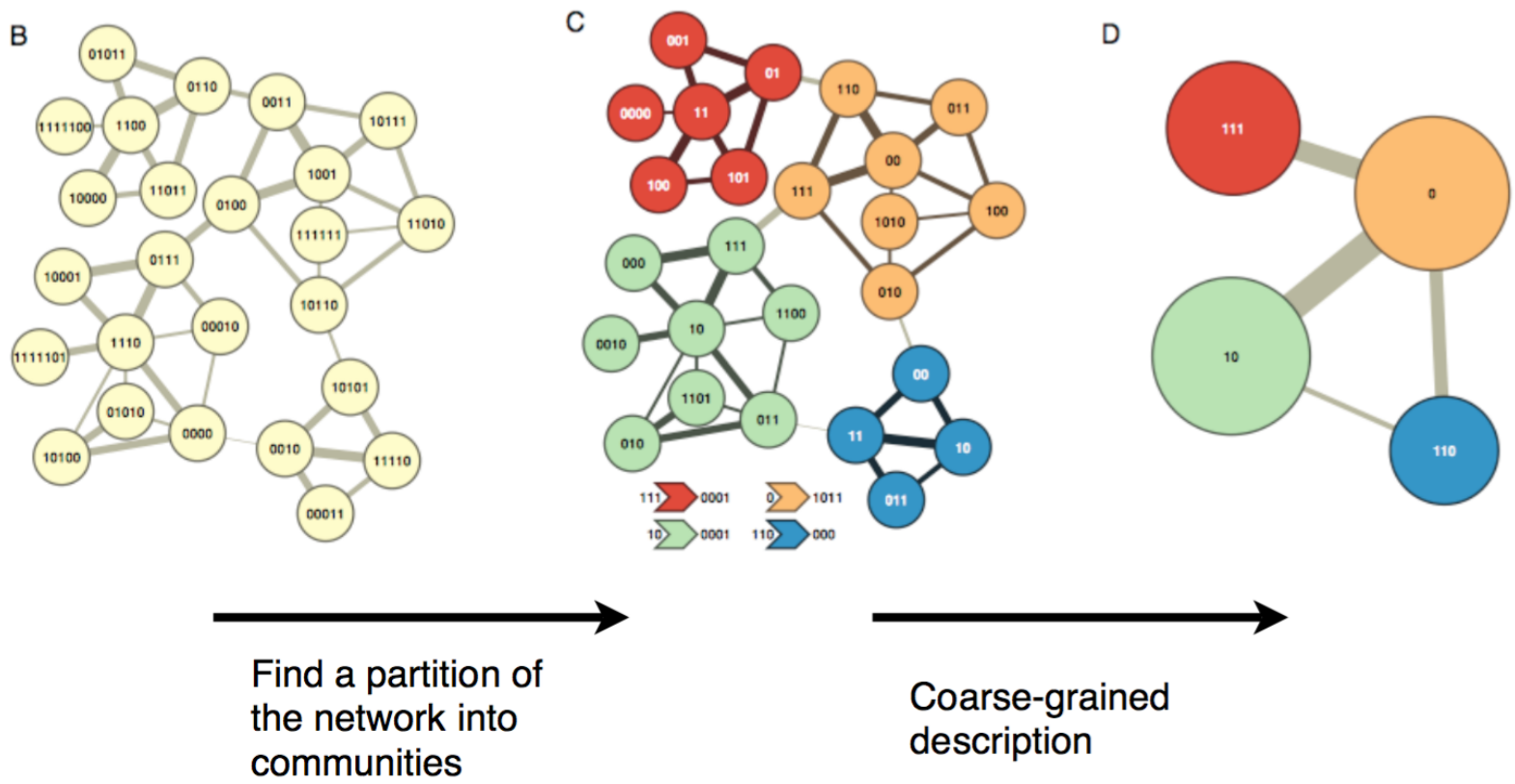
Graphs help us to comprehend in a visual way the global organisation of the system. This works extremely well when the graph is small but, as soon as the system is made of hundreds or thousands of nodes, a brute force representation typically leads to a meaningless cloud of nodes.

FIG. 2.—The direct relationship structure at Jefferson High



Why community detection?

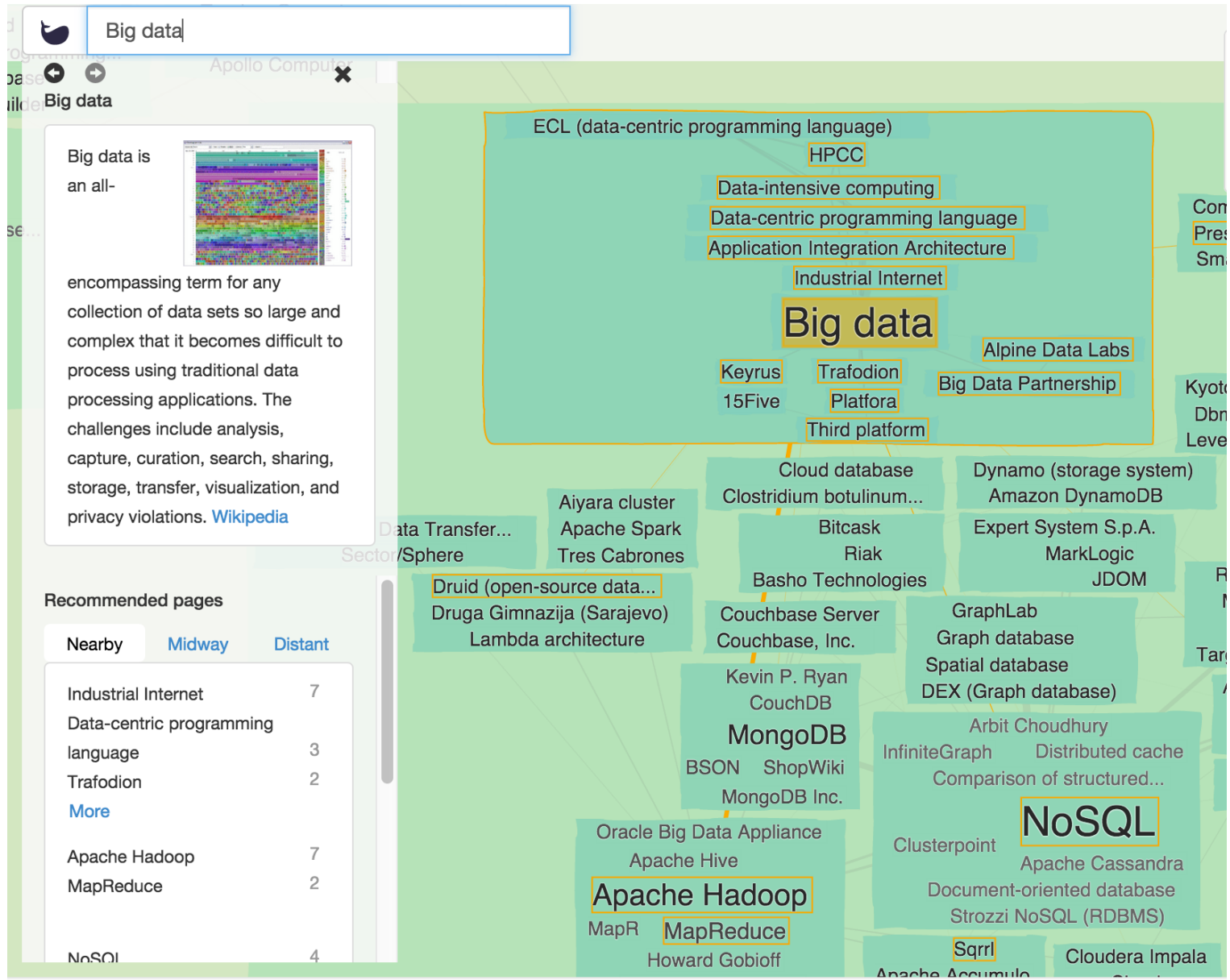
Uncovering communities/modules helps to change the resolution of the representation and to draw a readable map of the network



Martin Rosvall and Carl T. Bergstrom, PNAS 105, 1118 –1123 (2008)

Why community detection?

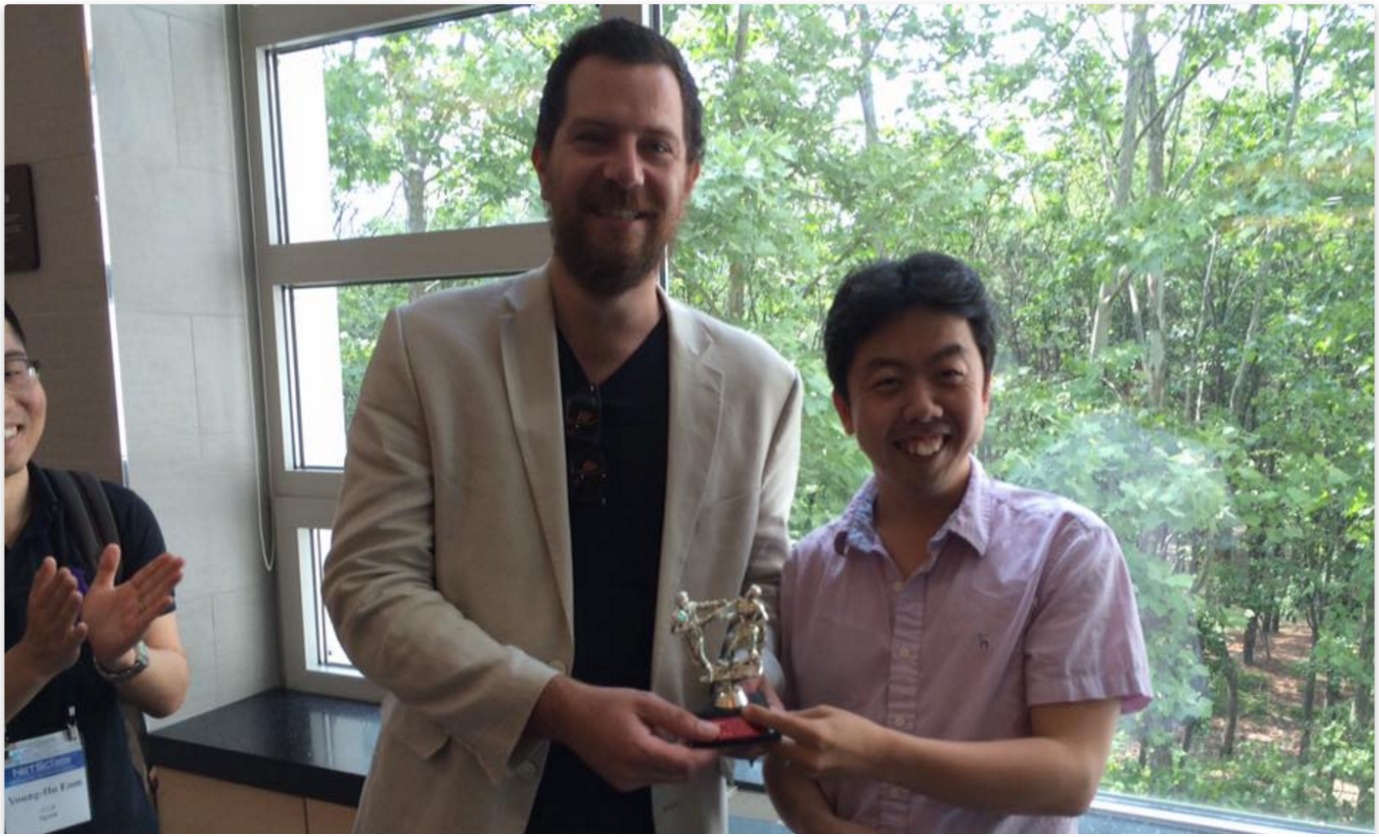
Uncovering communities/modules helps to change the resolution of the representation and to draw a readable map of the network



InfoBaleen

Why community detection?

Network Scientists with Karate Trophies



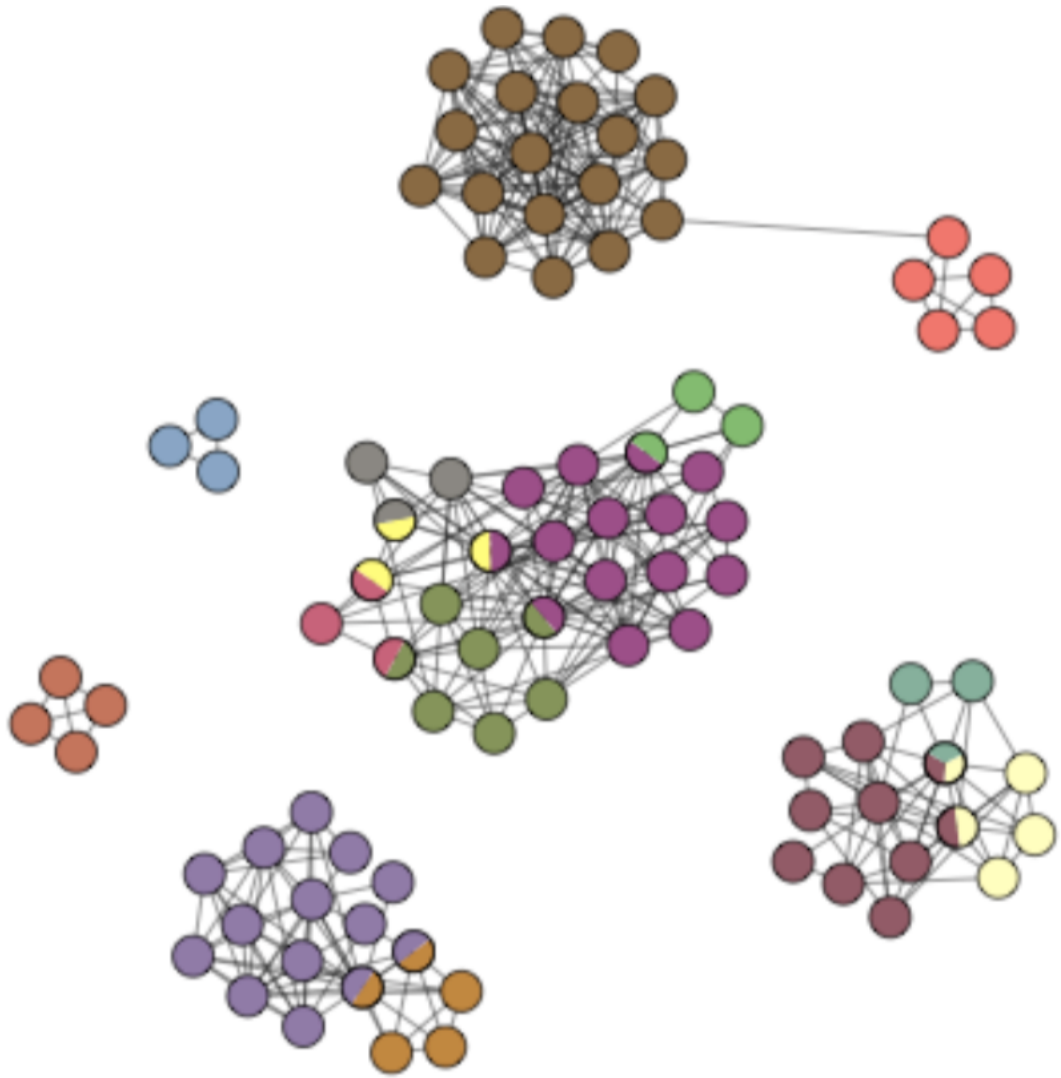
The first scientist at any conference on networks who uses Zachary's karate club as an example is inducted into the Zachary Karate Club Club, and awarded a prize. This tumblr records those moments.

RSS

ARCHIVE

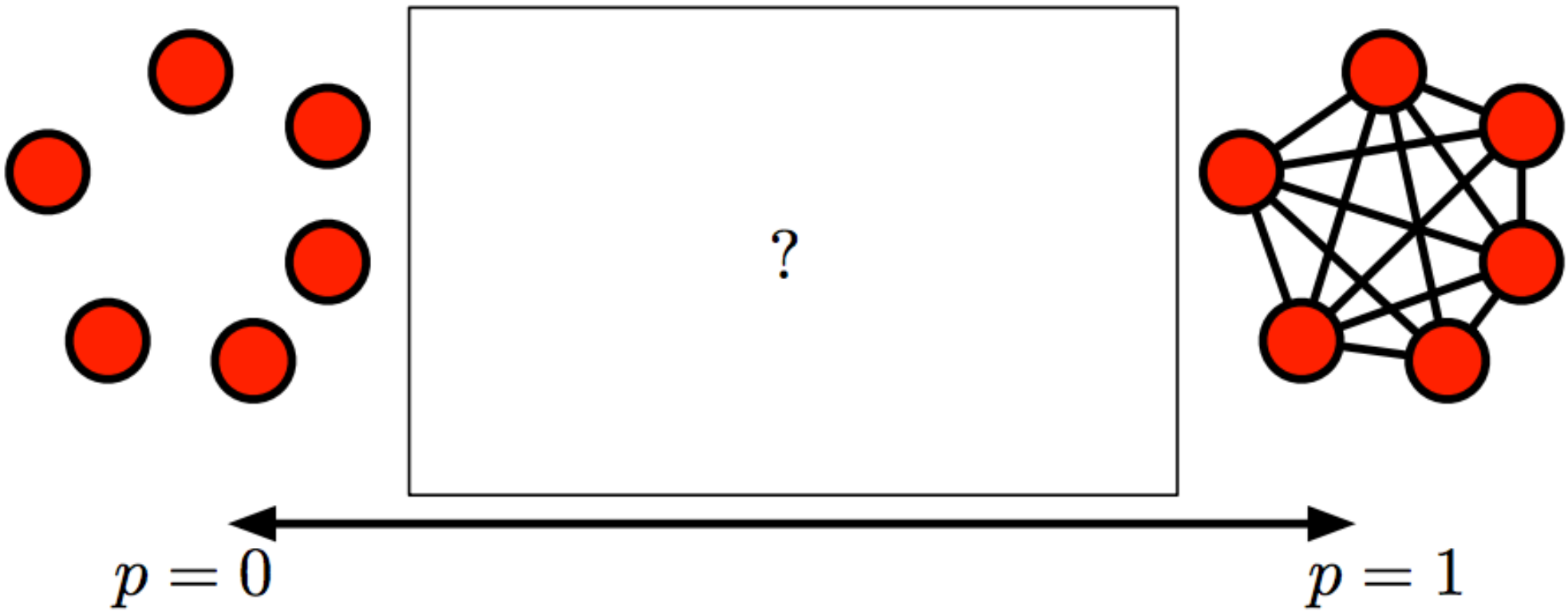
What is a “good” community?

A connected component is certainly a good community, in case of several components

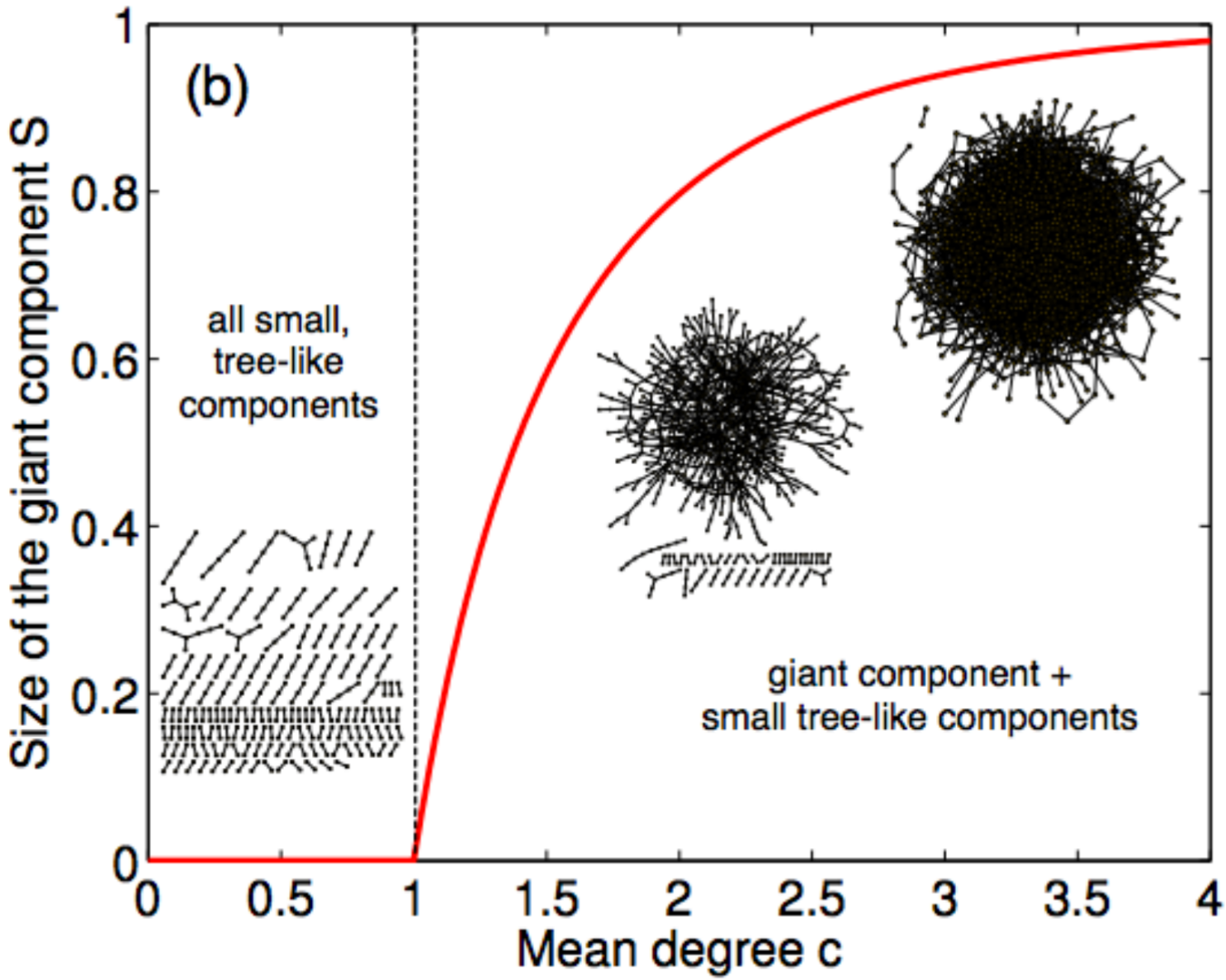


Percolation as a phase transition

Take the Erdos-Renyi network: how many disconnected components shall we expect depending on p?



Percolation as a phase transition



Community detection versus network partitioning

Both terms refer to the division of a network into dense groups

Graph partitioning: the number and size of the groups is **fixed** by the user. For instance, in a typical bisection problem: what is the best division of a network into 2 groups of equal size such that the number of links between the groups is minimised.
Application parallel computing: minimise inter-processor communication. Divides the system into the required number of groups, whatever the organisation of the system.

Community detection: the number and size of the groups are unspecified, but determined by the organisation of the network. Ideally, the method should be able to uncover a mixture of groups of different size in the same system. It should divide a network only when a good subdivision exists and leave it undivided otherwise.

Community detection versus network partitioning

Both terms refer to the division of a network into dense groups

Graph partitioning: the number and size of the groups is **fixed** by the user. For instance, in a typical bisection problem: what is the best division of a network into 2 groups of equal size such that the number of links between the groups is minimised.
Application parallel computing: minimise inter-processor communication. Divides the system into the required number of groups, whatever the organisation of the system.

Community detection: the number and size of the groups are unspecified, but determined by the organisation of the network. Ideally, the method should be able to uncover a mixture of groups of different size in the same system. It should divide a network only when a good subdivision exists and leave it undivided otherwise.

Different but similar methods
In both cases:
1) How to formalise the problem? **Definition of a good community**
2) How to solve it in practice? **Optimisation techniques to find it**

Graph bipartition

Definition of the problem:
Find the best division of a network into 2 groups of size n1 and n2 such that the cut size is minimal, where the cut size is the total number of links between different groups.

Solving the problem:
Looking through all bi-partitions and choose the one with the smallest cut size?
Impossible in practice, as the exhaustive search is extremely costly in terms of computer time

E.g. The number of ways to divide a network of 2n nodes into two groups of n and n nodes is:

$$\frac{(2n)!}{n!n!} \quad \xrightarrow{\text{Stirling}} \quad \frac{2^{n+1}}{\sqrt{n}}$$

No method solving exactly the problem in polynomial time for all networks…
But several existing heuristics allow to find approximate solutions in non-prohibitive times.

Spectral methods

$$R = \frac{1}{2} \sum_{ij \text{ in different groups}} A_{ij}$$

Let us denote by $s_i = \pm 1$ the assignment of node i

$$R = \frac{1}{2} \sum_{ij} A_{ij}(1 - s_i s_j) = \frac{1}{4} \sum_{ij} L_{ij} s_i s_j$$

By performing a spectral decomposition of the Laplacian matrix, one finds:

$$L_{ij} = \sum_{\alpha=2}^{N} \lambda_\alpha v_{\alpha,i} v_{\alpha,j}$$

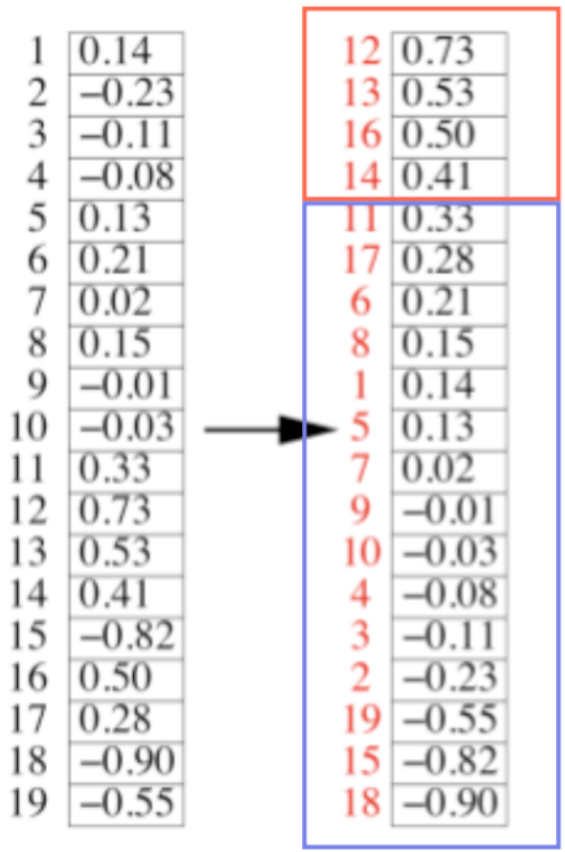
If there is no condition on si, the optimal solution would be $s_i = v_{2,i}$

$$R = \frac{1}{4} \lambda_2$$

But this solution is not a partition (except in extremely trivial situations) and it probably does not satisfy the required size of the groups.

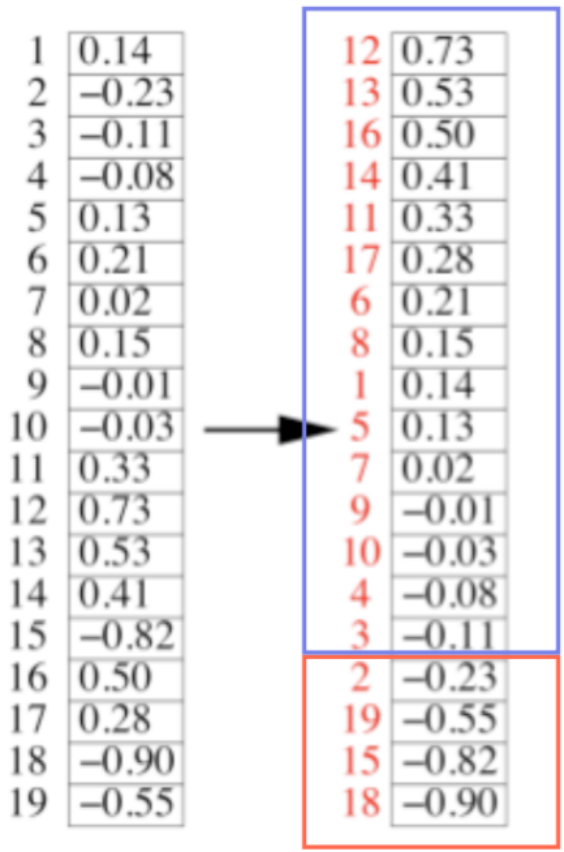
Spectral methods

Approximation: If one wants a split into n1 and n2=n-n1 vertices, one orders the components of the Fiedler vector from the largest positive to the smallest negative and picks the n1 largest (smallest) components of the Fiedler vector



1	0.14
2	−0.23
3	−0.11
4	−0.08
5	0.13
6	0.21
7	0.02
8	0.15
9	−0.01
10	−0.03
11	0.33
12	0.73
13	0.53
14	0.41
15	−0.82
16	0.50
17	0.28
18	−0.90
19	−0.55

12	0.73
13	0.53
16	0.50
14	0.41
11	0.33
17	0.28
6	0.21
8	0.15
1	0.14
5	0.13
7	0.02
9	−0.01
10	−0.03
4	−0.08
3	−0.11
2	−0.23
19	−0.55
15	−0.82
18	−0.90



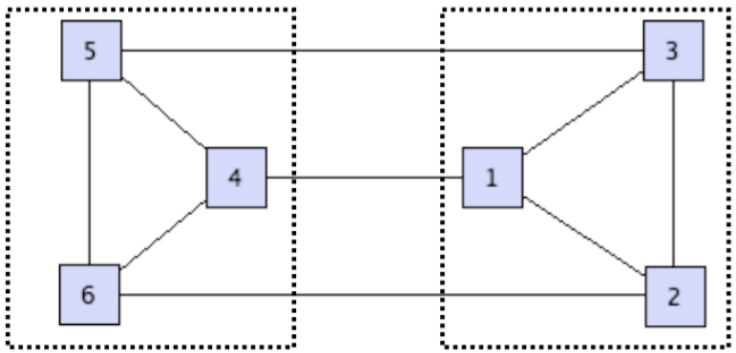
1	0.14
2	−0.23
3	−0.11
4	−0.08
5	0.13
6	0.21
7	0.02
8	0.15
9	−0.01
10	−0.03
11	0.33
12	0.73
13	0.53
14	0.41
15	−0.82
16	0.50
17	0.28
18	−0.90
19	−0.55

12	0.73
13	0.53
16	0.50
14	0.41
11	0.33
17	0.28
6	0.21
8	0.15
1	0.14
5	0.13
7	0.02
9	−0.01
10	−0.03
4	−0.08
3	−0.11
2	−0.23
19	−0.55
15	−0.82
18	−0.90

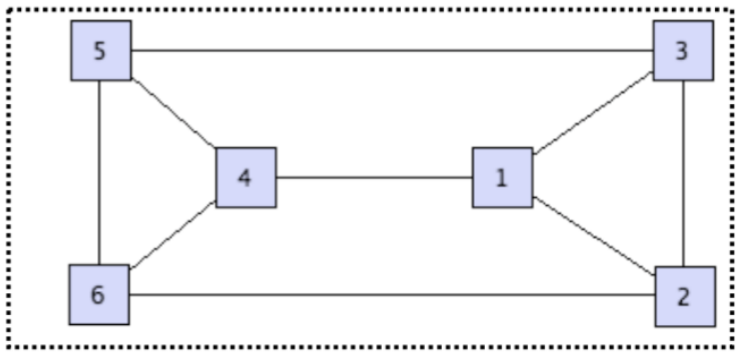
Complexity: O(N^2) on sparse networks

Community detection

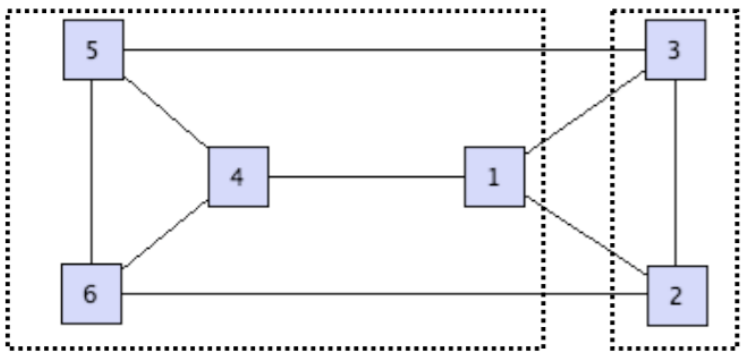
What is the best partition of a network into modules?
How do we rank the quality of partitions of different sizes?



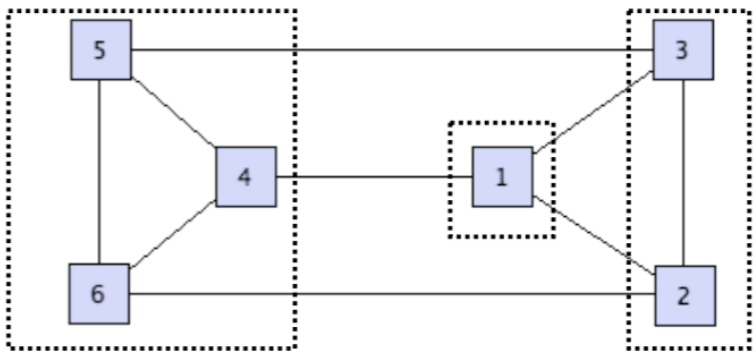
Q1



Q2



Q3



Q4

.....

Newman-Girvan Modularity

Q = fraction of edges within communities - expected fraction of such edges

Let us attribute each node i to a community ci

$$Q = \frac{1}{2m} \sum_{i,j} \Big[A_{ij} - P_{ij} \Big] \delta(c_i, c_j)$$

$P_{ij} = \frac{k_i k_j}{2m}$ expected number of links between i and j

$$Q_C = \frac{1}{2m} \sum_{i,j} \Big[A_{ij} - k_i k_j / 2m \Big] \delta(c_i, c_j) \qquad Q_C \in [-1/2, 1]$$

Allows to compare partitions made of different numbers of modules

M.E.J. Newman and M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E, 69, 026113, 2004.

Note on the null model

Random network with constrained degrees $P_{ij} = \frac{k_i k_j}{2m}$

What if one has extra information about the nodes?

Directed networks -> $P_{ij} = k_i^{\text{in}} k_j^{\text{out}} / m$

Spatially-embedded networks -> $P_{ij}^{\text{Spa}} = N_i N_j f(d_{ij})$

Or if the information on the degrees is expected to be irrelevant:

$$P_{ij} = \langle k \rangle^2 / 2m = \langle k \rangle / N$$

$$\sum_{ij} P_{ij} = \sum_{ij} A_{ij} = 2m$$

Modularity

Property 1 *A partition where all the vertices are grouped into the same community has a modularity equal to zero. This proves to be simply shown from the definition of the null model* $\frac{k_i k_j}{2m}$ *for which* $\sum_{ij} \frac{k_i k_j}{2m} = 2m$ *and from the expression of modularity in this particular case*

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] = 0 \ . \tag{3}$$

This property implies that any partition with a positive modularity is better than this trivial one, but also that it is always possible to find a partition such that $Q \geq 0$.

Modularity

Property 2 *If a partition contains a disconnected community, it is always preferable (in terms of modularity) to split this community into connected communities. Let us consider, for the sake of simplicty, the case of a disconnected community C_1 formed by two connected subgraphs C_{11}, C_{12}. In this case, modularity is given by*

$$
\begin{aligned}
Q = & \frac{1}{2m}\left[\sum_{C\neq C_1}\sum_{i,j\in C}(A_{ij}-\frac{k_ik_j}{2m})+\sum_{i,j\in C_1}(A_{ij}-\frac{k_ik_j}{2m})\right] \\
= & \frac{1}{2m}\left[\sum_{C\neq C_1}\sum_{i,j\in C}(A_{ij}-\frac{k_ik_j}{2m})+\sum_{i,j\in C_{11}}(A_{ij}-\frac{k_ik_j}{2m})\right. \\
& \left.+\sum_{i,j\in C_{12}}(A_{ij}-\frac{k_ik_j}{2m})+2\sum_{i\in C_{11},j\in C_{12}}(A_{ij}-\frac{k_ik_j}{2m})\right. .
\end{aligned}
\tag{4}
$$

Given that $A_{ij} = 0$ if $i \in C_{11}, j \in C_{12}$, the sum $\sum_{i\in C_{11},j\in C_{12}}$ is composed uniquely of negative terms and it is thus preferable to split the community into two subcommunities.

This property implies that any partition made of disconnected communities is sub-optimal and that the optimal partition of a graph is only made of connected communities.

Modularity Optimization

Optimization of modularity is an NP-complete problem

Need for efficient heuristics

Optimization: Spectral methods

Similar method to one for minimizing the cut, based on the spectral properties of the modularity matrix Q

$$Q_{ij} = A_{ij} - \frac{k_i k_j}{2m}$$

Let us first focus on the best division of the network into 2 communities.

Let us denote by $s_i = \pm 1$ the assignment of node i $\delta(c_i, c_j) = \frac{1}{2}(s_i s_j + 1)$

$$Q = \frac{1}{2m} \sum_{ij} Q_{ij} \delta(c_i, c_j) = \frac{1}{4m} \sum_{ij} Q_{ij} s_i s_j$$

By performing a spectral decomposition of the modularity matrix, one finds:

$$Q_{ij} = \sum_{\alpha=1}^{N} \lambda_\alpha v_{\alpha,i} v_{\alpha,j}$$

si is chosen to be as similar to the dominant eigenvector

$$s_i = 1 \text{ if } v_{N,i} > 0$$
$$s_i = -1 \text{ if } v_{N,i} < 0$$

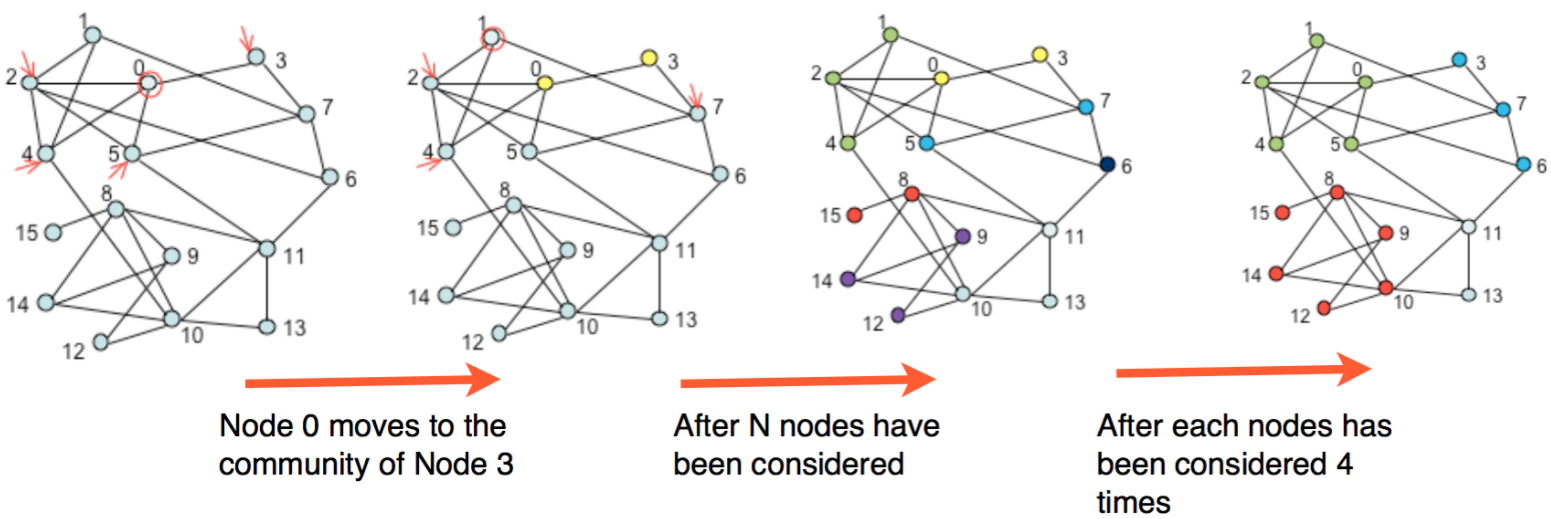
M.E.J. Newman, Finding community structure in networks using the eigenvectors of matrices, Phys. Rev. E, vol. 74, 036104, 2006.

Optmization: Greedy optimization

Louvain: multi-scale, agglomerative and greedy

The algorithm is based on two steps that are repeated iteratively. **First phase**: Find a local maximum

1) Give an order to the nodes (0,1,2,3,...., N-1)
2) Initially, each node belongs to its own community (N nodes and N communities)
3) One looks through all the nodes (from 0 to N-1) in an ordered way. The selected node looks among its neighbours and adopt the community of the neighbour for which the increase of modularity is maximum (and positive).
4)This step is performed iteratively until a local maximum of modularity is reached (each node may be considered several times).

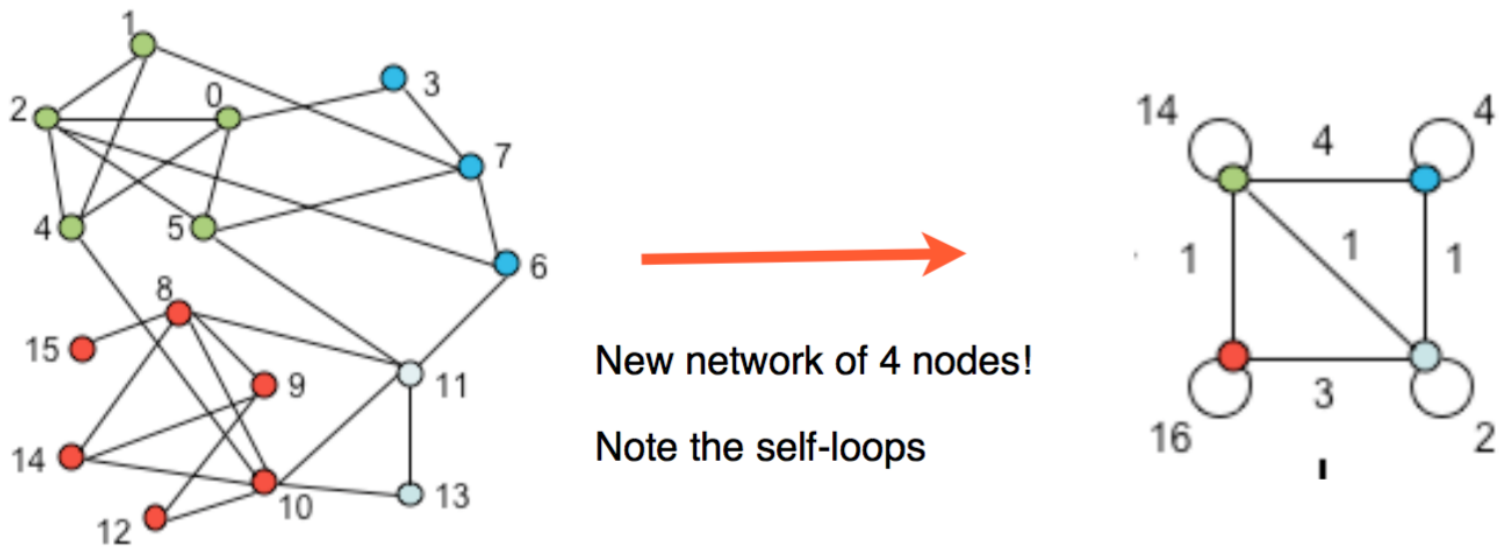


V.D. Blondel, J.-L. Guillaume, R. Lambiotte and E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech., P10008, 2008.

Optmization: Greedy optimization

Louvain: multi-scale, agglomerative and greedy

Once a local maximum has been attained, **second phase**:
We build a new network whose nodes are the communities. The weight of the links between communities is the total weight of the links between the nodes of these communities.



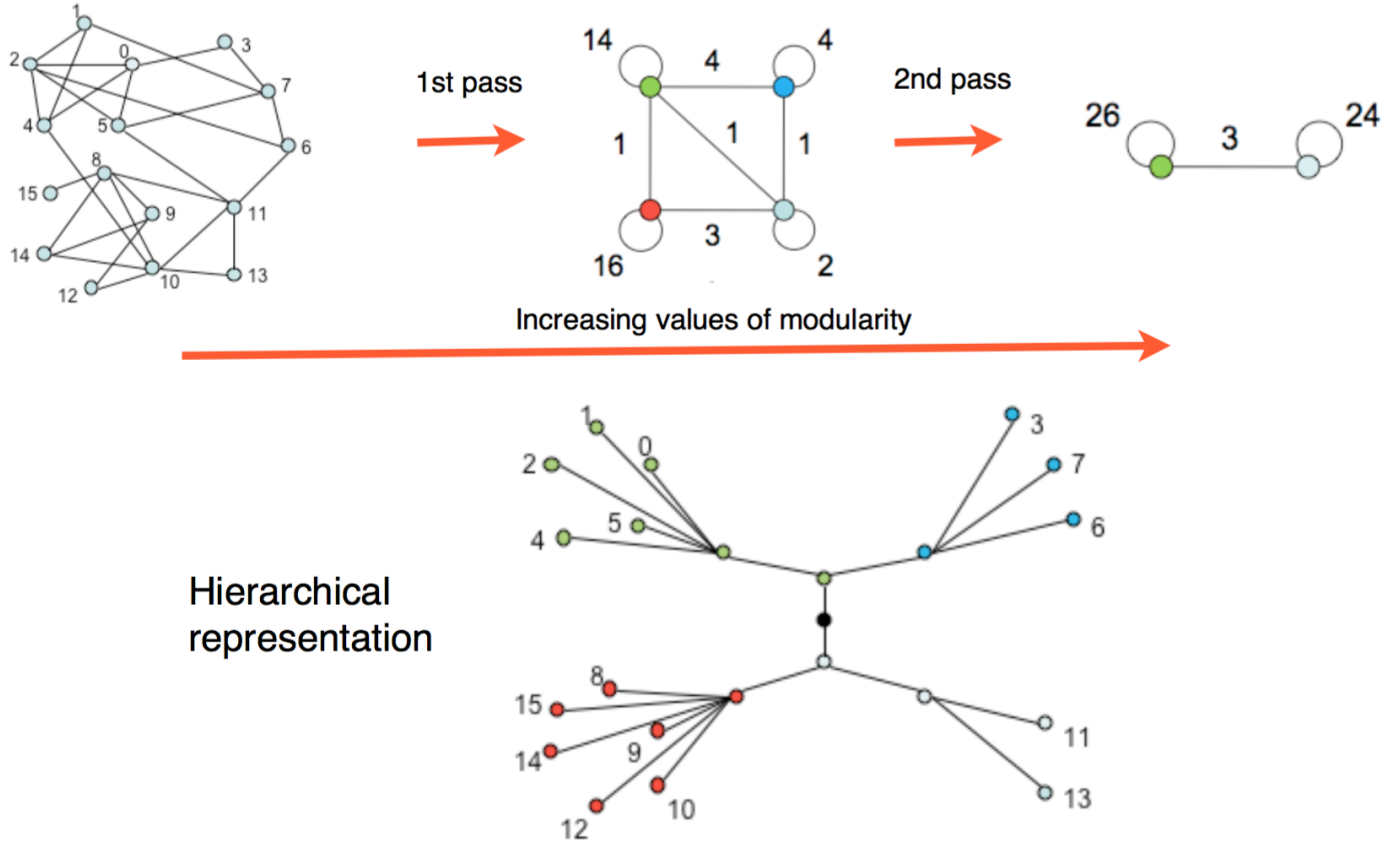
In typical realizations, the number of nodes diminishes drastically at this step.

V.D. Blondel, J.-L. Guillaume, R. Lambiotte and E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech., P10008, 2008.

Optmization: Greedy optimization

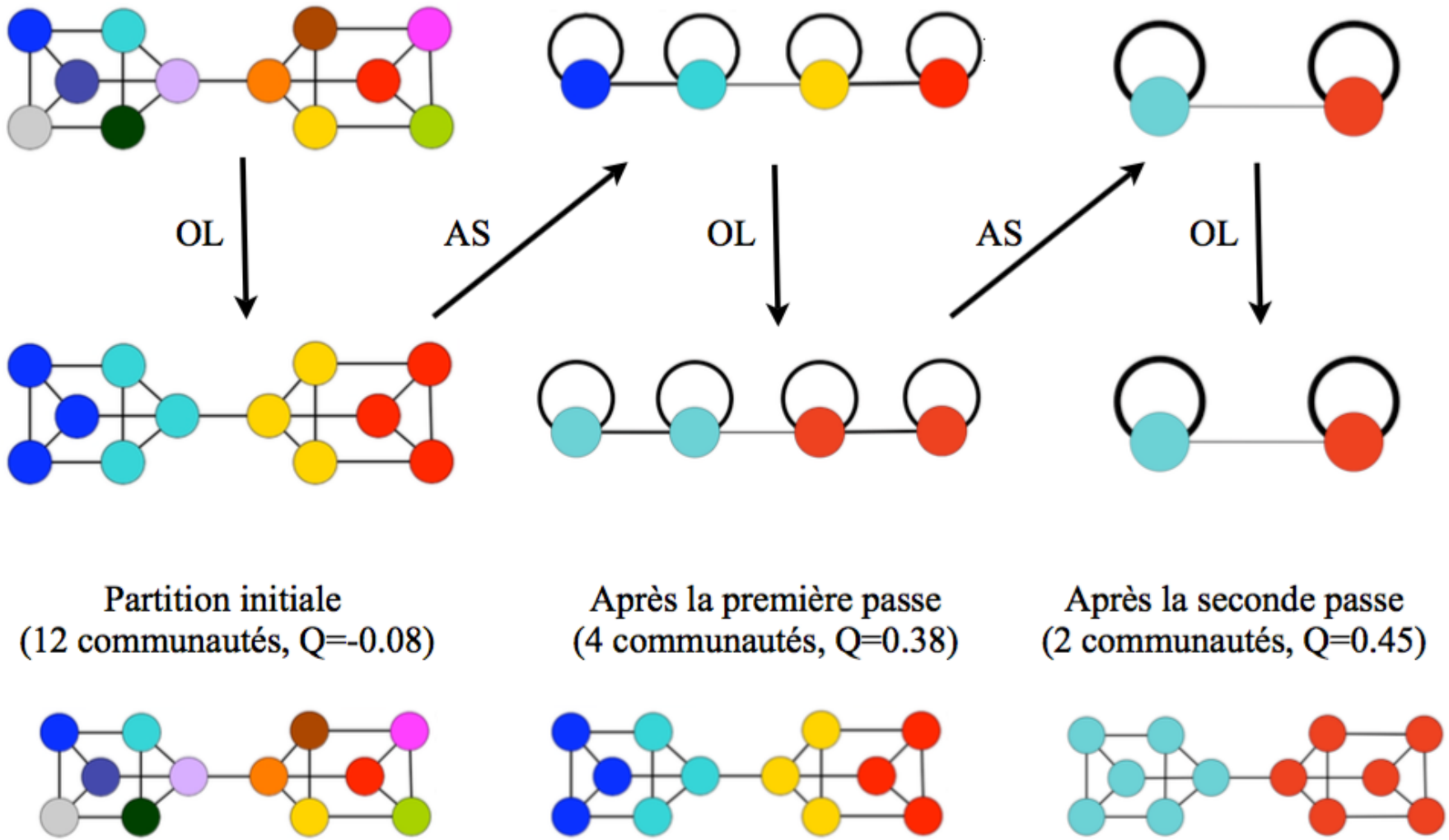
Louvain: multi-scale, agglomerative and greedy

The two steps are repeated iteratively, thereby leading to a hierarchical decomposition of the network.
Multi-scale optimisation: local search first among neighbours, then among neighbouring communities, etc.



V.D. Blondel, J.-L. Guillaume, R. Lambiotte and E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech., P10008, 2008.

Optimization: Greedy optimization



V.D. Blondel, J.-L. Guillaume, R. Lambiotte and E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech., P10008, 2008.

Louvain

Algorithm 1 Pseudo-code of the community detection algorithm.

```
1: Community detectionG initial graph
2: repeat
3:    Place each vertex of G into a single community
4:    Save the modularity of this decomposition
5:    while there are moved vertices do
6:       for all vertex n of G do
7:          c ← neighboring community maximizing the modularity increase
8:          if c results in a strictly positive increase then
9:             move n from its community to c
10:         end if
11:      end for
12:   end while
13:   if the modularity reached is higher than the initial modularity, then
14:      end ← false
15:      Display the partition found
16:      Transform G into the graph between communities
17:   else
18:      end ← true
19:   end if
20: until end
```

Louvain

The efficiency of the algorithm partly resides in the fact that the variation of modularity Δ_{ij} obtained by moving a vertex i from its community to the community of one of its neighbors j can be calculated with only local information. In practice, the variation of modularity is calculated by removing i from its community $\Delta_{remove;i}$ (this is only done once) then inserting it into the community of j $\Delta_{insert;ij}$ for each neighbor j of i. The variation is therefore: $\Delta_{ij} = \Delta_{remove;i} + \Delta_{insert;ij}$.

Optimization: Greedy optimization

Louvain: multi-scale, agglomerative and greedy

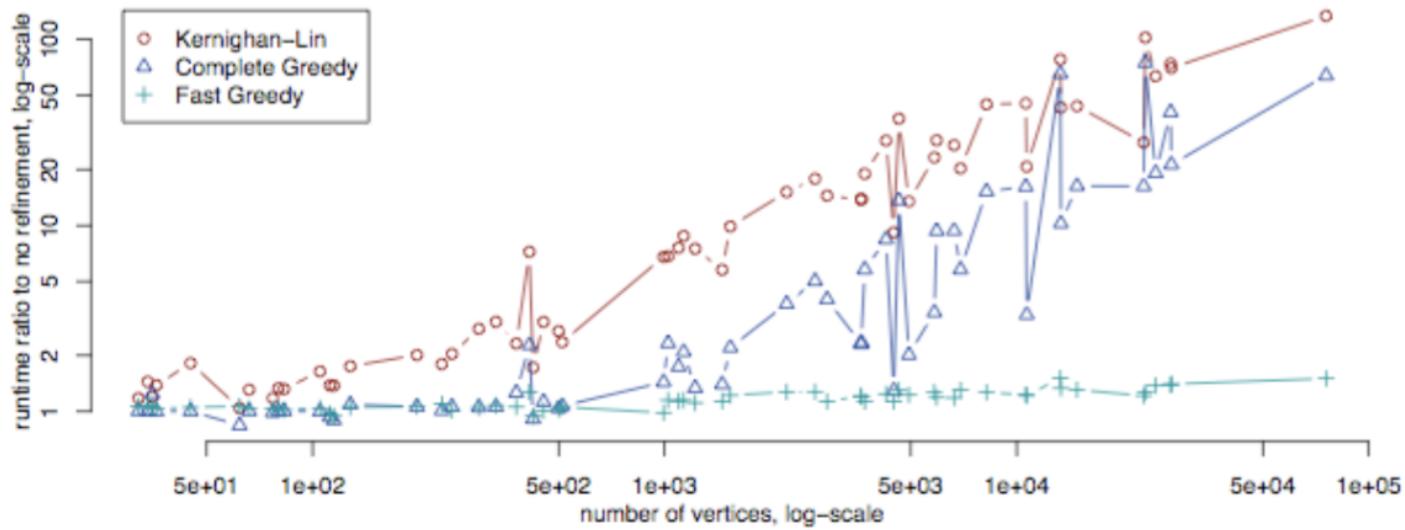
Very fast: O(N) in practice. The only limitation being the storage of the network in main memory
Good accuracy (among greedy methods)

	Karate	Arxiv	Internet	Web nd.edu	Phone	Web uk-2005	Web WebBase 2001
Nodes/links	34/77	9k/24k	70k/351k	325k/1M	2.04M/5.4M	39M/783M	118M/1B
CNM	.38/0s	.772/3.6s	.692/799s	.927/5034s	-/-	-/-	-/-
PL	.42/0s	.757/3.3s	.729/575s	.895/6666s	-/-	-/-	-/-
WT	.42/0s	.761/0.7s	.667/62s	.898/248s	.553/367s	-/-	-/-
Our algorithm	.42/0s	.813/0s	.781/1s	.935/3s	.76/44s	.979/738s	.984/152mn

V.D. Blondel, J.-L. Guillaume, R. Lambiotte and E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech., P10008, 2008.

How to test the methods?

Test the heuristics: what is the value of Q obtained for different algorithms? Time complexity?



graph	size	subdivision		coarsening		local search		math prog		SS+ML
karate [42]	34	[29]	.419	[41]	.4198	[12]	.4188	[1]	.4197	.4197
dolphins [22]	62	[29]	*.4893*	[31]	*.5171*	[33]	*.5285*	[40]	.5285	.5276
polBooks [21]	105	[29]	*.3992*	[37]	*.5269*	[4]	*.5204*	[1]	.5272	.5269
afootball [14]	115	[39]	.602	[41]	.605	[4]	*.6045*	[1]	.6046	.6002
jazz [15]	198	[29]	.442	[9]	.4409	[12]	.4452	[1]	.445	.4446
celeg_metab [12]	453	[29]	.435	[36]	.450	[12]	.4342	[1]	.450	.4452
email [17]	1133	[29]	.572	[9]	.5569	[12]	.5738	[1]	.579	.5774
Erdos02 [16]	6927	[29]	*.5969*	[32]	.6817	[33]	*.7094*			.7162
PGP_main [5]	11k	[29]	.855	[9]	.7462	[12]	.8459			.8841
cmat03_main [25]	28k	[29]	.723	[41]	.761	[12]	.6790			.8146
ND_edu [2]	325k			[7]	.927	[4]	.935			.9509

How to test the methods?

Comparison with real-world data: do modules reveal nodes having similar meta-data?

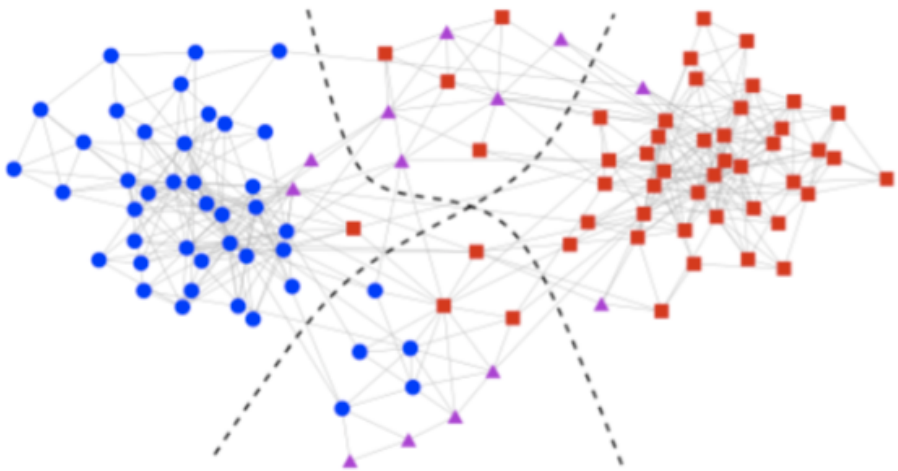
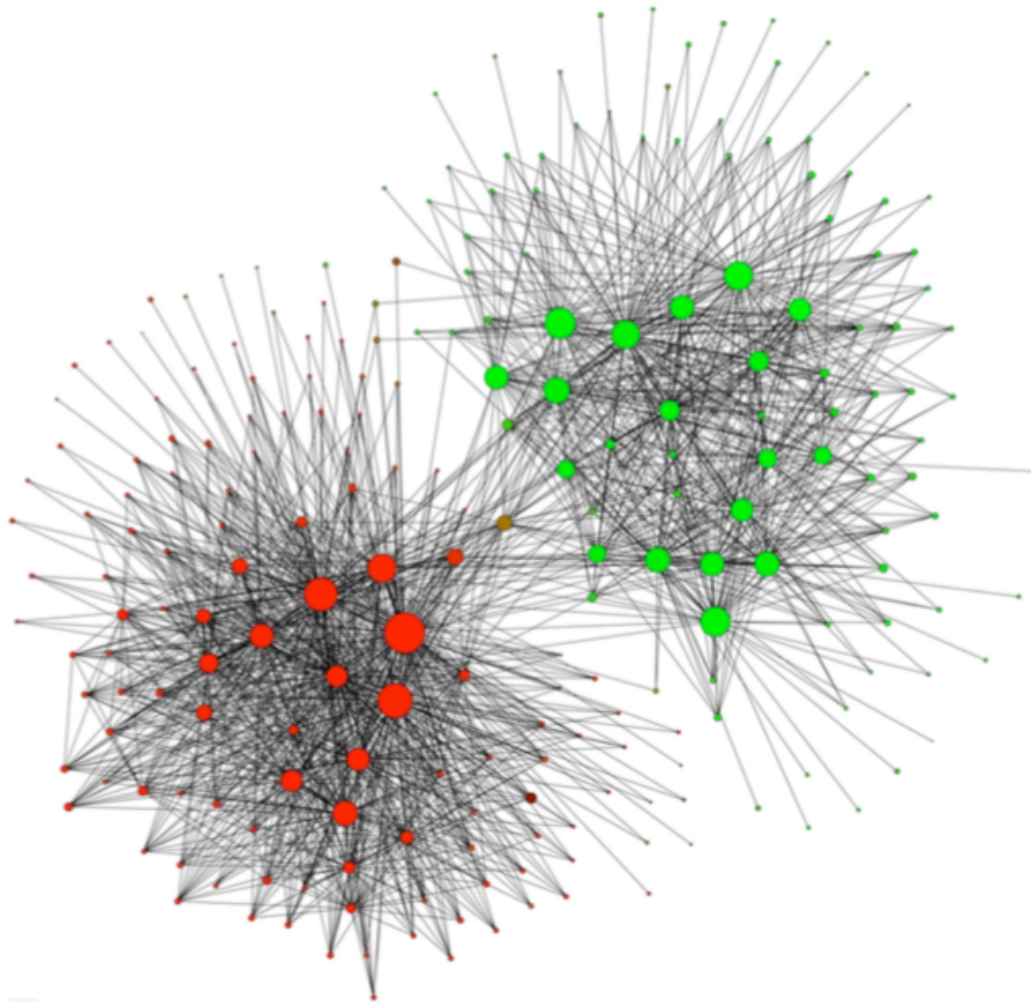


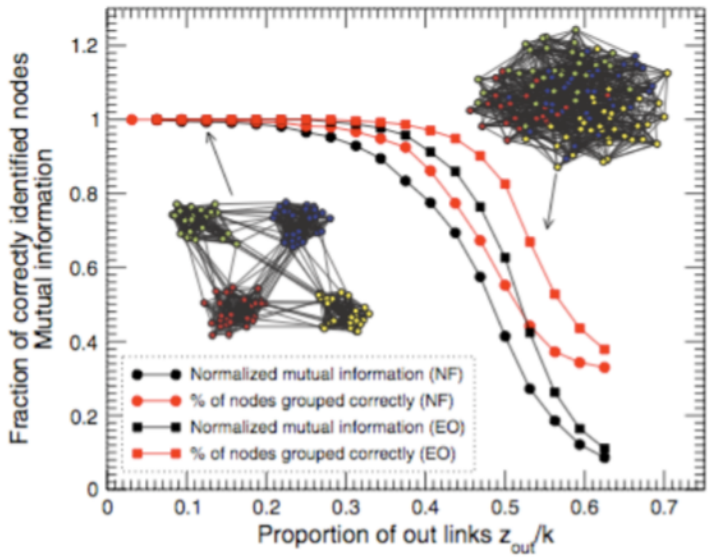
FIG. 3: Krebs' network of books on American politics. Vertices represent books and edges join books frequently purchased by the same readers. Dotted lines divide the four communities found by our algorithm and shapes represent the political alignment of the books: circles (blue) are liberal, squares (red) are conservative, triangles (purple) are centrist or unaligned.

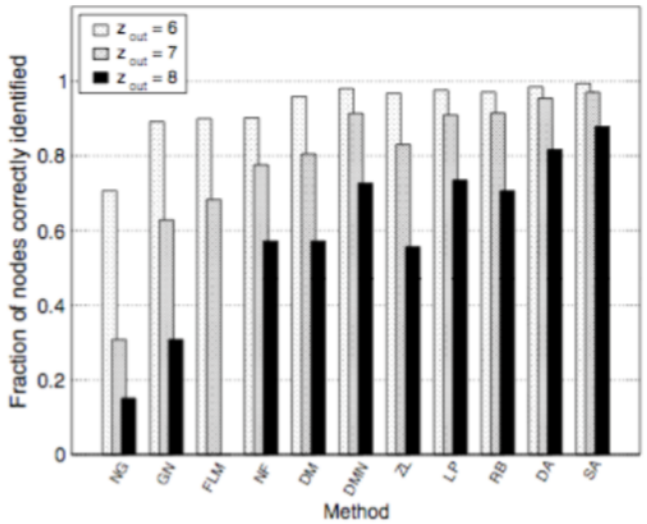


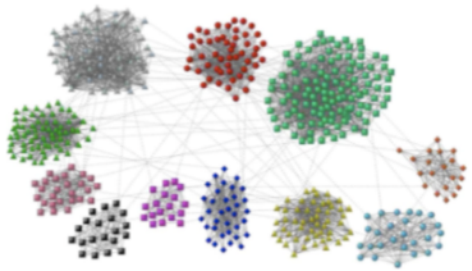
But: meta-data are often unknown. No insurance that modular organization coincides with semantic/cultural organisation

How to test the methods?

Benchmarks: artificial networks with known community structure.





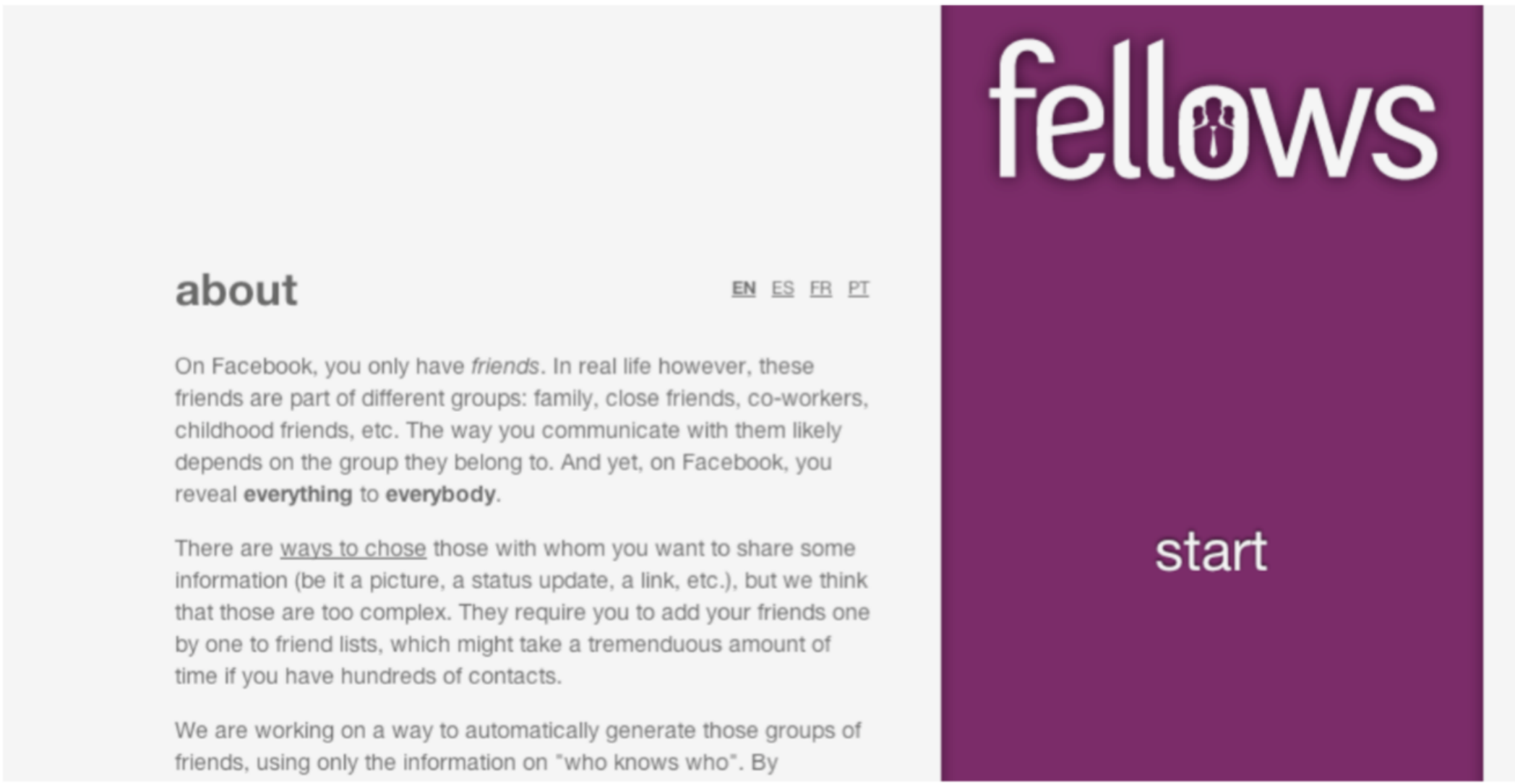


But: random networks (their structure is quite different from real-world networks).
In the way the benchmark is built, there is a (hidden) choice for what good partitions should be

Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi, Phys. Rev. E 78, 046110 (2008)

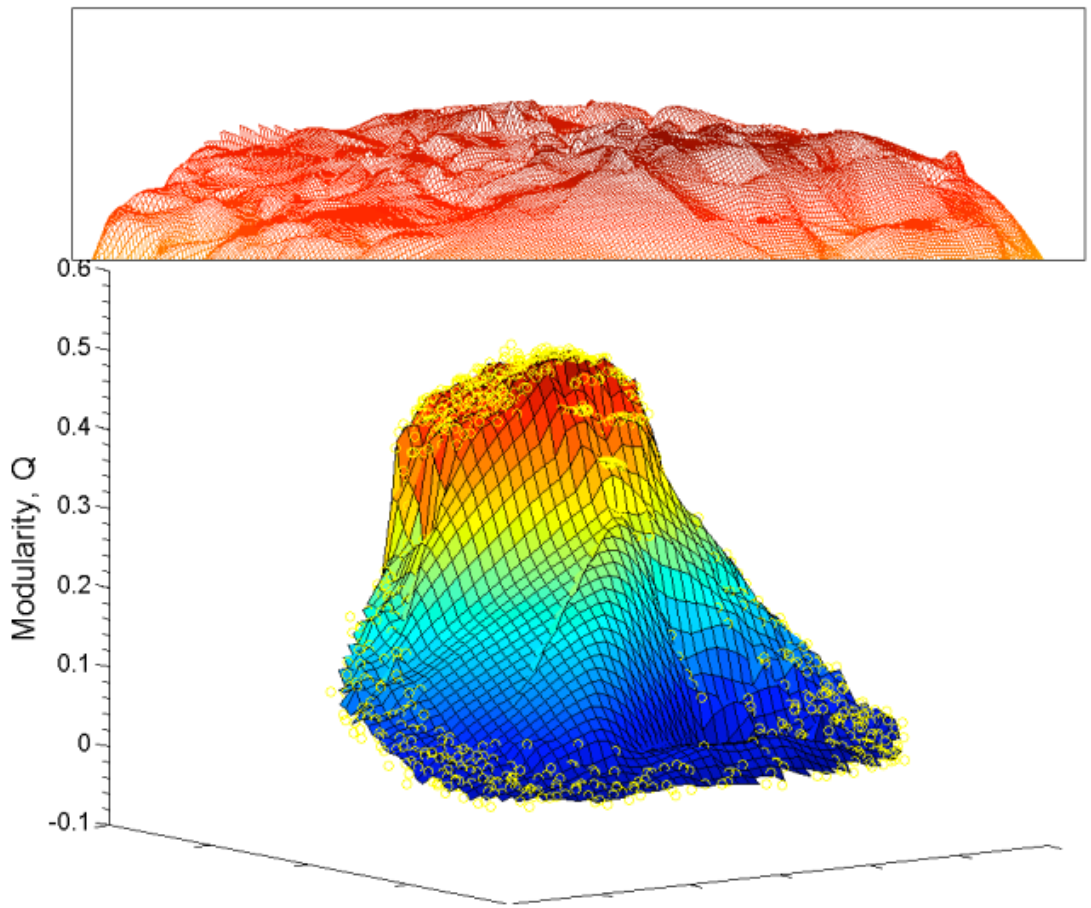
How to test the methods?

Ajk the people!



Limitations of modularity (1)

The modularity landscape tends to be very rugged, with many partitions, possibly very different, having similar value of modularity.



The performance of modularity maximization in practical contexts, Benjamin H. Good et al.
http://arxiv.org/pdf/0910.0165.pdf

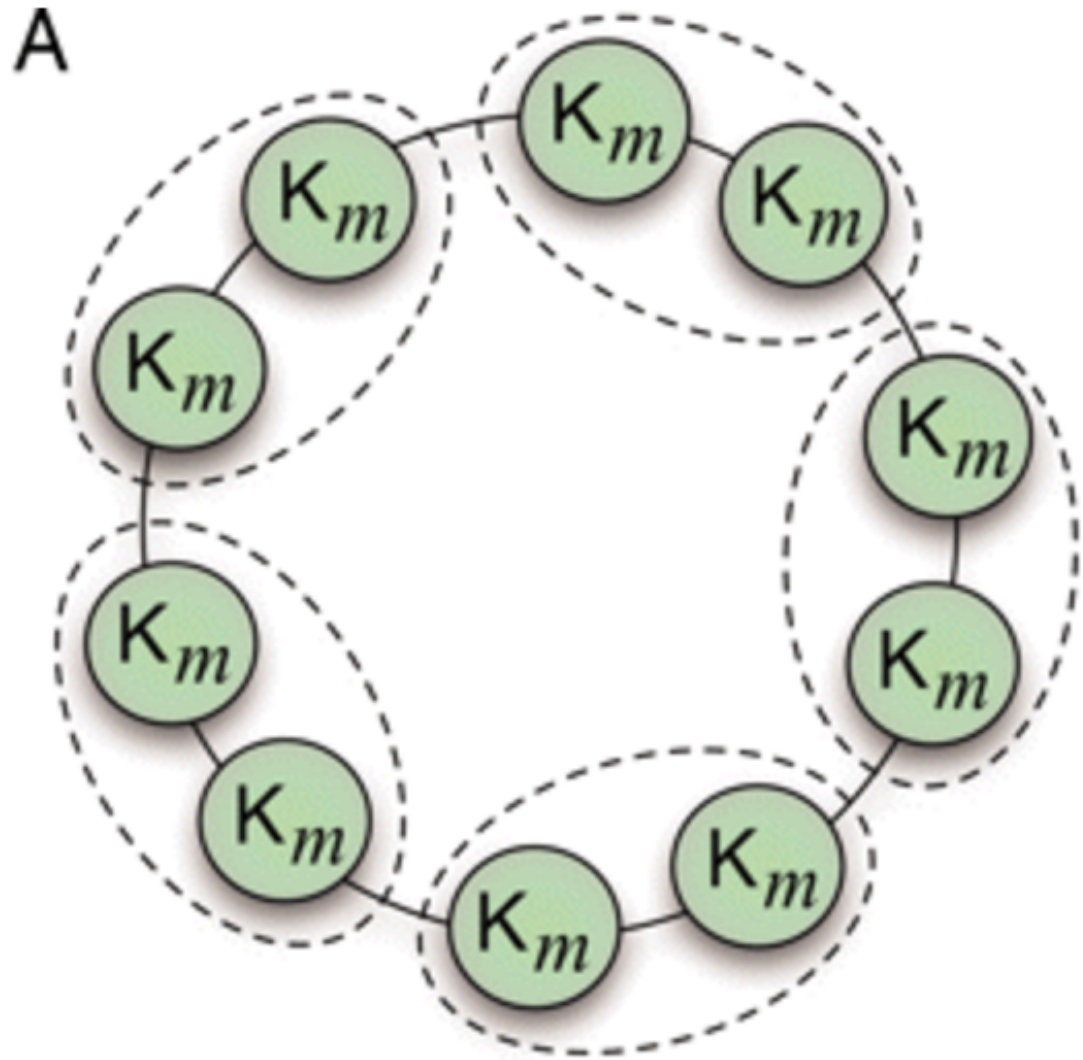
Limitations of modularity (2)

Second, Q exhibits a resolution limit, because using Q it is impossible to detect dense clusters of nodes that are smaller than a certain scale [Fortunato and Barthélemy (2007)]. The resolution limit originates from the dependency of the null model on $2M$. The dependency decreases when the number of links, M, is increased. Then, modularity maximisation tends to favour larger communities. In the limit $M \to \infty$, the null model is neglected and modularity optimisation simply uncovers the connected components. Modularity-based methods implicitly favour communities having a certain size, depending on the size of the entire network, not only on its internal structure.

$$Q_{ij} = A_{ij} - \frac{k_i k_j}{2m}$$

Resolution limit in community detection Santo Fortunato and Marc Barthélemy
http://www.pnas.org/content/104/1/36.full

Limitations of modularity (2)



Resolution limit in community detection Santo Fortunato and Marc Barthélemy
http://www.pnas.org/content/104/1/36.full

Limitations of modularity (3)

Finally, although modularity allows us to compare partitions of the same network, it is by no means intended to compare modularity values of different networks. Therefore, Q should not be used as a measure of the modularity of a network. For instance, the modularity of the best partition of a random network tends to $Q = 1$ when the network is sufficiently large, whereas this network is by no means modular [Guimerà *et al.* (2004)].

Resolution limit in community detection Santo Fortunato and Marc Barthélemy
http://www.pnas.org/content/104/1/36.full