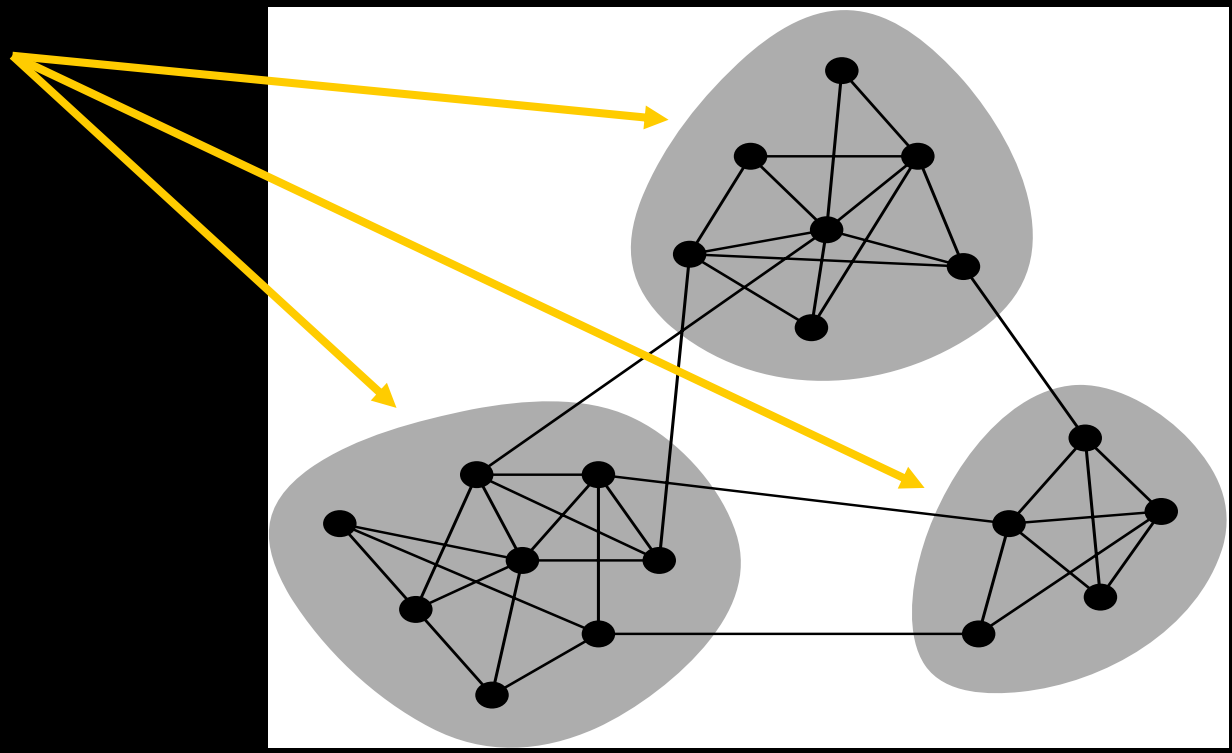
A new fast algorithm to detect communities in networks

Santo Fortunato
Andrea Lancichinetti
Janos Kertész

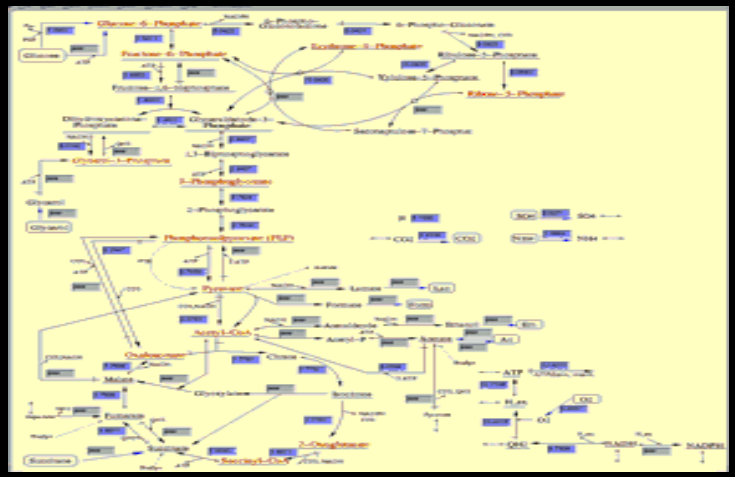
“Communities”



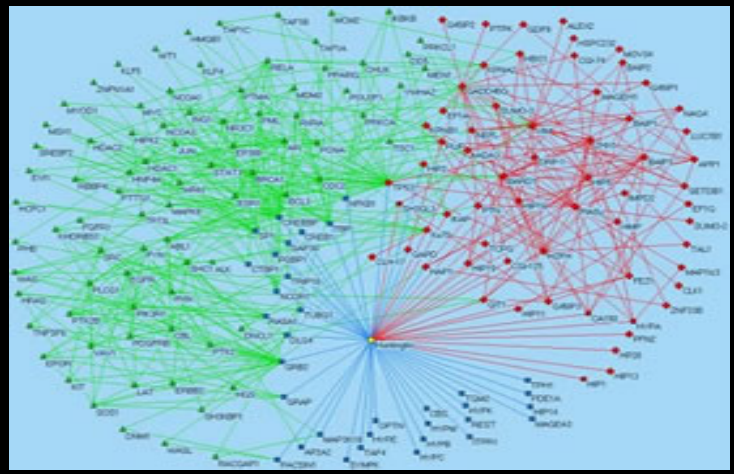
More links “inside” than “outside”

Graphs are “sparse”

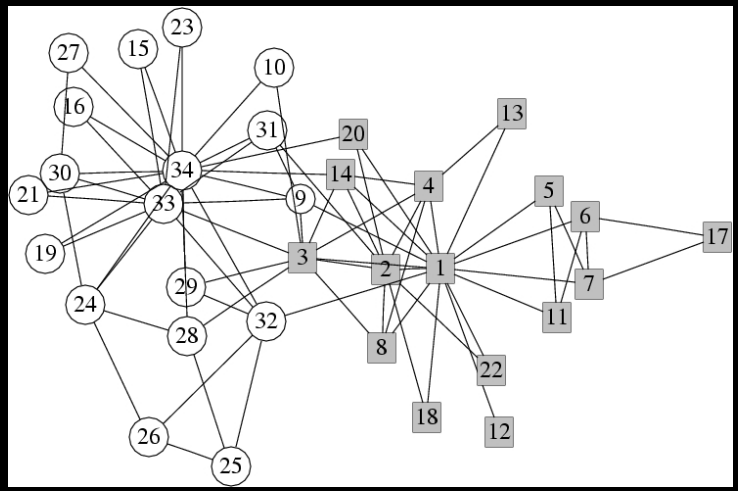
Metabolic



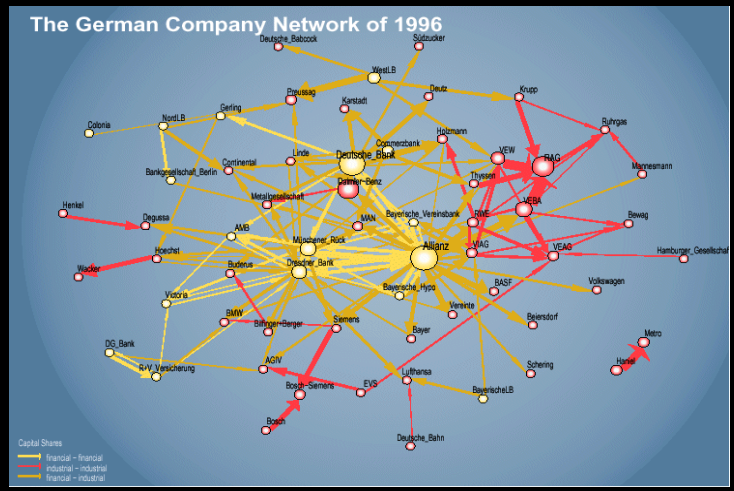
Protein-protein



Social



Economical



History

- 1970s: Graph partitioning in computer science
- Hierarchical clustering in social sciences
- 2002: Girvan and Newman, PNAS 99, 7821-7826
- 2002-onward: methods of “new generation”

Null hypothesis

The relations between nodes can be inferred from the topology, i.e.

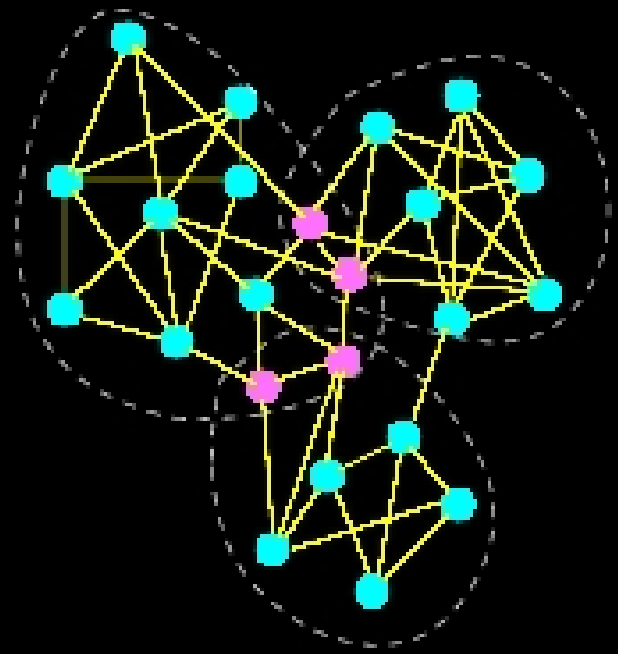
Real communities = **Topological communities**

Limits of current methods

- Overlapping communities
- Hierarchies
- Computer time

Overlapping communities

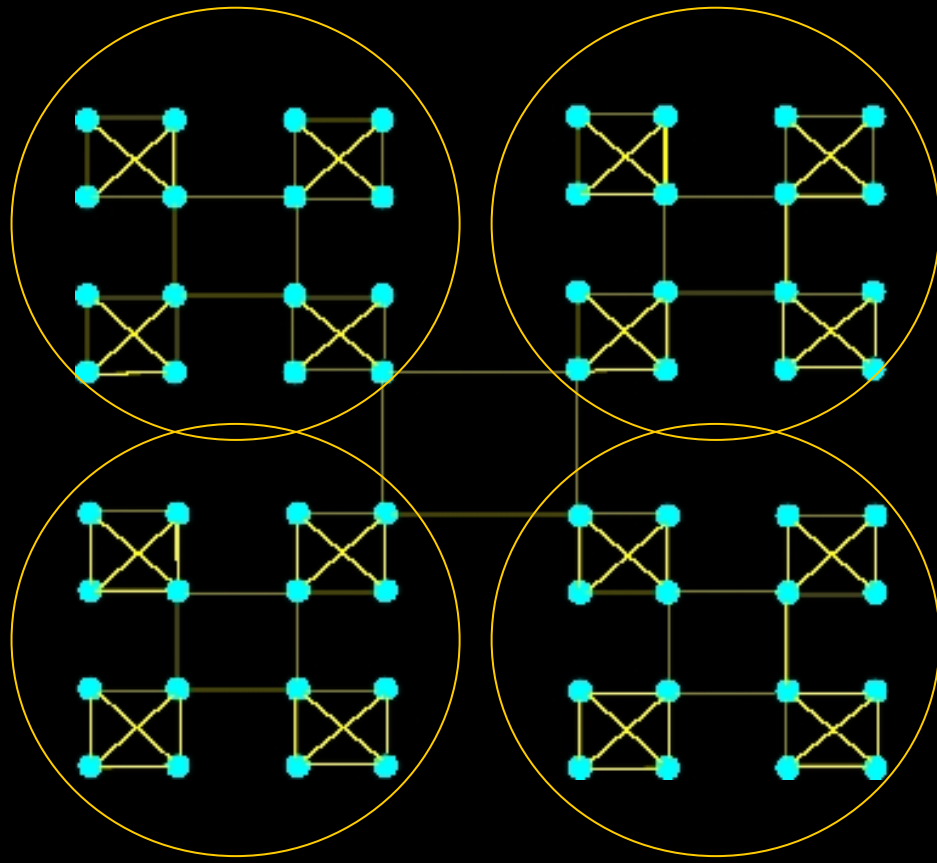
In real networks, vertices may belong to different modules



G. Palla, I. Derényi, I. Farkas, T. Vicsek, Nature 435, 814, 2005

Hierarchies

Modules may embed smaller modules, yielding different organizational levels



A. Clauset, C. Moore, M.E.J. Newman, LNCS 4503, 1, 2007

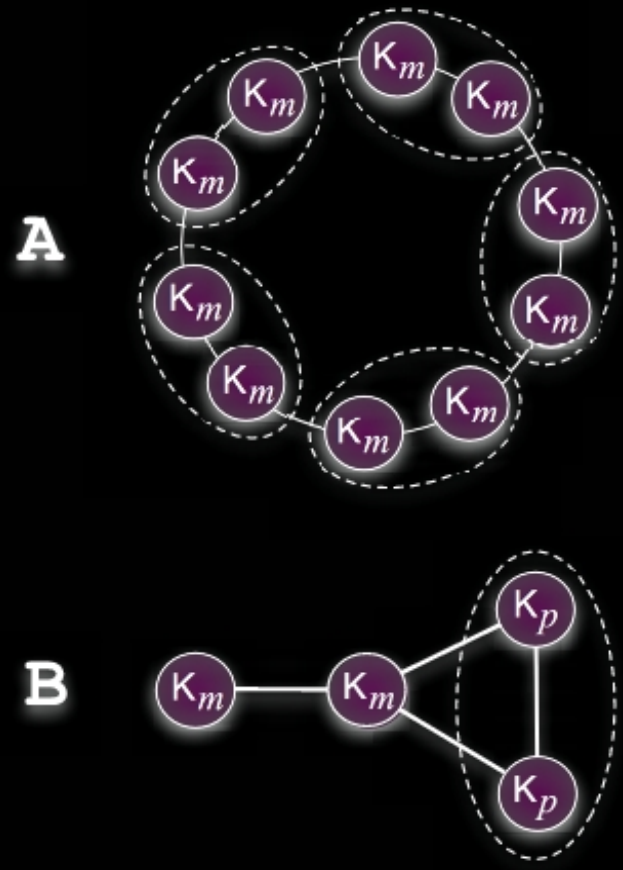
Computer time

Good algorithms run in a time $O(n^2)$

Some methods run in almost linear time!

- **Greedy modularity optimization (Clauset, Newman, Moore, PRE 70, 066111, 2004)**
- **Wu-Huberman method (EPJB 38, 331, 2004)**

The resolution limit of modularity optimization



S.F. & M. Barthélemy, PNAS 104, 36 (2007)

Goal

Designing a FAST algorithm that accounts both for overlapping communities AND for hierarchies

Global or local?

“Global” community: a cluster of nodes with some property relative to the whole network

“Local” community: a cluster of nodes with a property relative to the nodes themselves and (possibly) their neighbors

Global:

- **Girvan-Newman algorithm**
- **modularity optimization**
- **random walks**

Local:

- **clique percolation**
- **L-shell method**
- **edge clustering method**

The method

Basic rule: finding local communities about individual nodes

A local community is built by maximizing a *fitness function*

The fitness function depends on a parameter that tunes the size of the communities

The fitness function

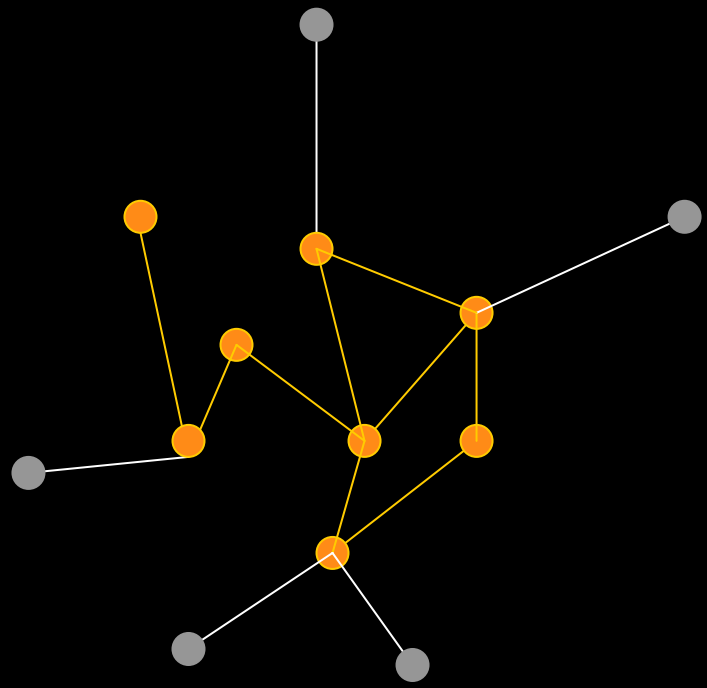
Several options

$$f_i = \frac{k_{in}^i}{(k_{in}^i + k_{out}^i)^\alpha}$$

Resolution parameter **α>0**

Inspired by weak definition (α=1)

(Radicchi, Castellano, Cecconi, Loreto & Parisi, PNAS 101, 2658, 2004)



Node fitness

Node A, cluster i

$$f_i^A = f_{i \cup A} - f_{i-A}$$

Positive fitness if the fitness of cluster i increases due to the addition of node A

Steps of the algorithm

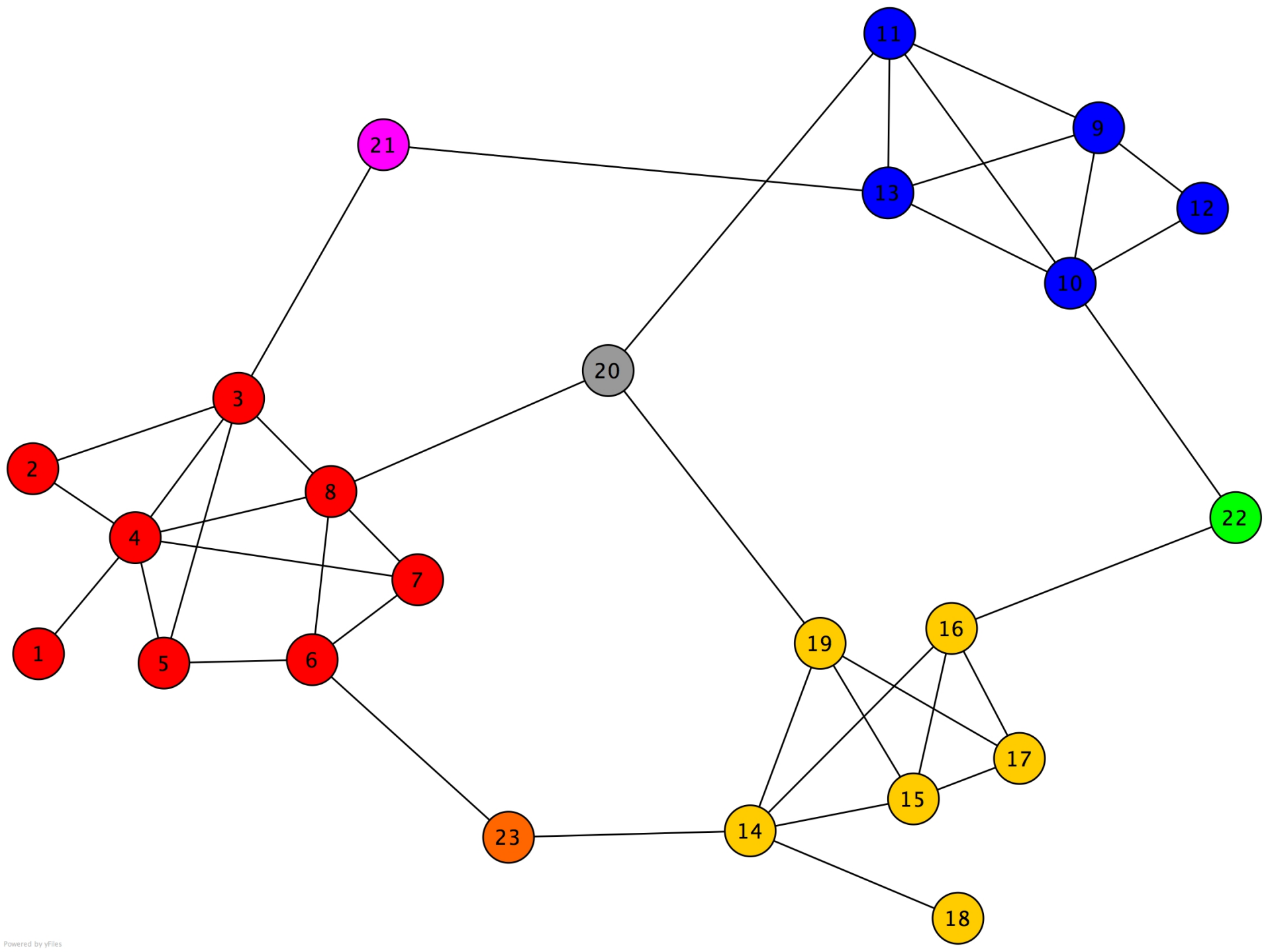
α is fixed

1. **Take a node A at random**
2. **Look for community of A**
3. **Pick a node B at random not yet assigned to a community; the community of node B** *may overlap with the others*
4. **Repeat from 2**

Building a node's community

Cluster with s nodes

- The neighboring node with the largest (positive) fitness is added to the group
- If a node is added, the fitness of all nodes of the group is recalculated
- Nodes with negative fitness are removed
- The process is repeated until all neighboring nodes have negative fitness (maximal cluster)



Computer time

The time to "close" a community with s nodes goes (about) as $O(s^2)$

The average CPU time is of the order of $O(ns_{Max})$

The worst-case time scales as $O(n^2)$

Resolution & hierarchies

Different values of the resolution parameter α yield partitions with different cluster sizes

α small → large communities

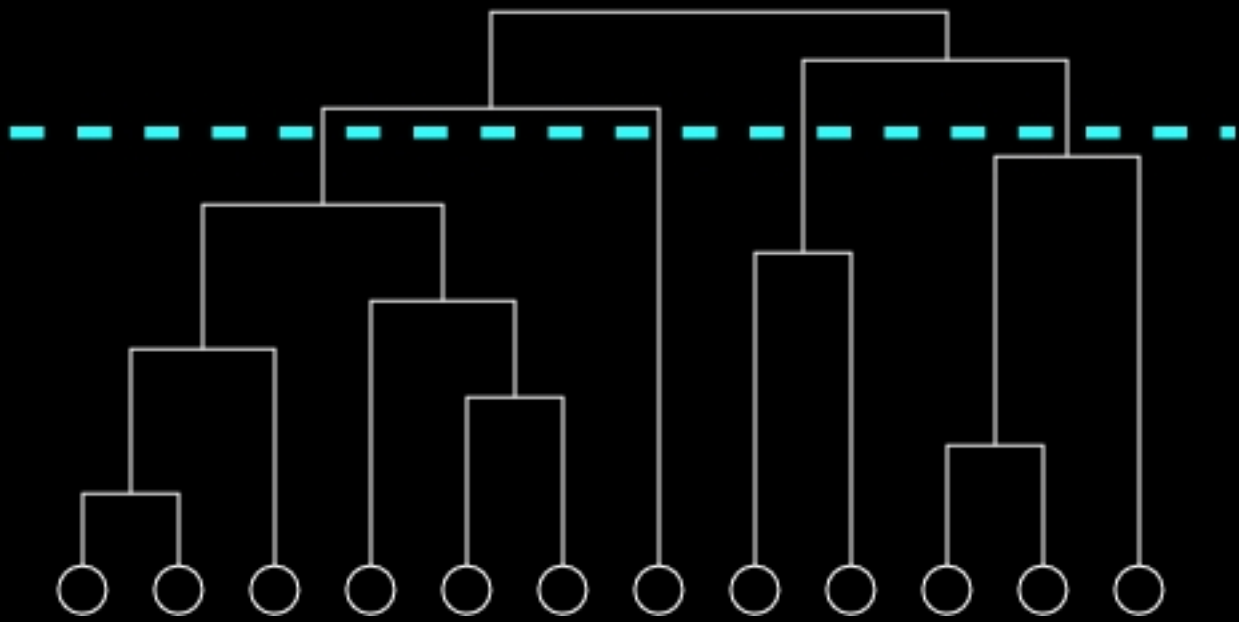
α large → small communities

By varying α hierarchical structure can be recovered

α_{min} α_0 α_{max}

$n_c=1$ $n_c=n_0$ $n_c=n_M$

If $1<n_0<n_{max}$ split the two subintervals



For hierarchical networks, the depth of the dendrogram varies as log n → the number of α-values is of the order of log n

Quality of partitions

The method delivers many partitions: which one(s) is the best?

Answer: the best partition is the *most stable* in the range of α

$$F(\alpha = 1) = \frac{1}{n_c} \sum_{i=1}^{n_c} \frac{k_{in}^i}{k_{in}^i + k_{out}^i}$$

Stable partitions appear as long plateaus of F vs α

Further stability index: overlapping nodes

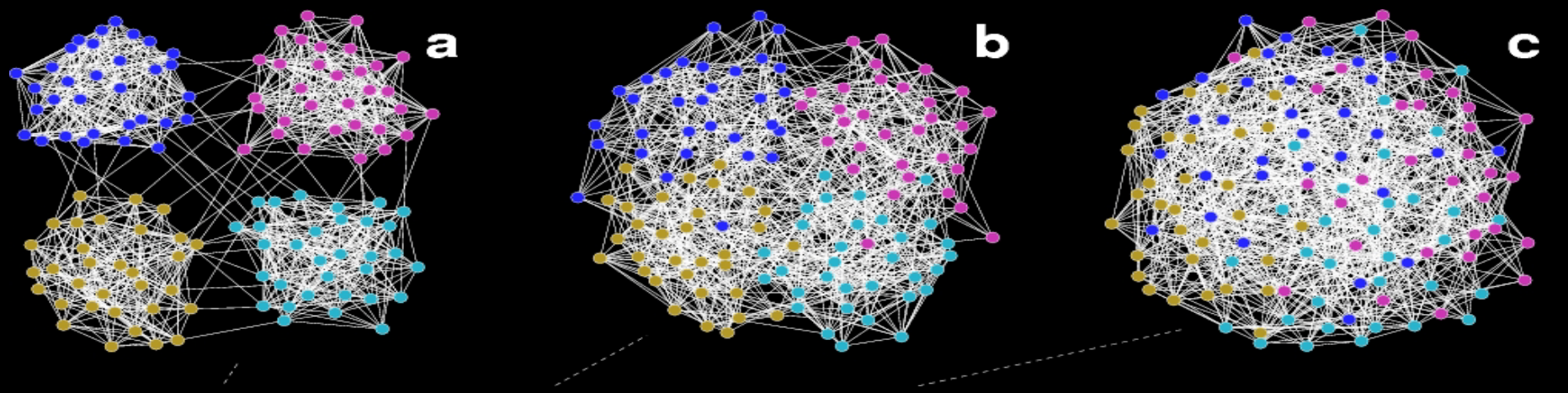
r = 1 - (fraction of overlapping nodes)

Principle: the more overlapping the communities, the less well they are defined

Recipe

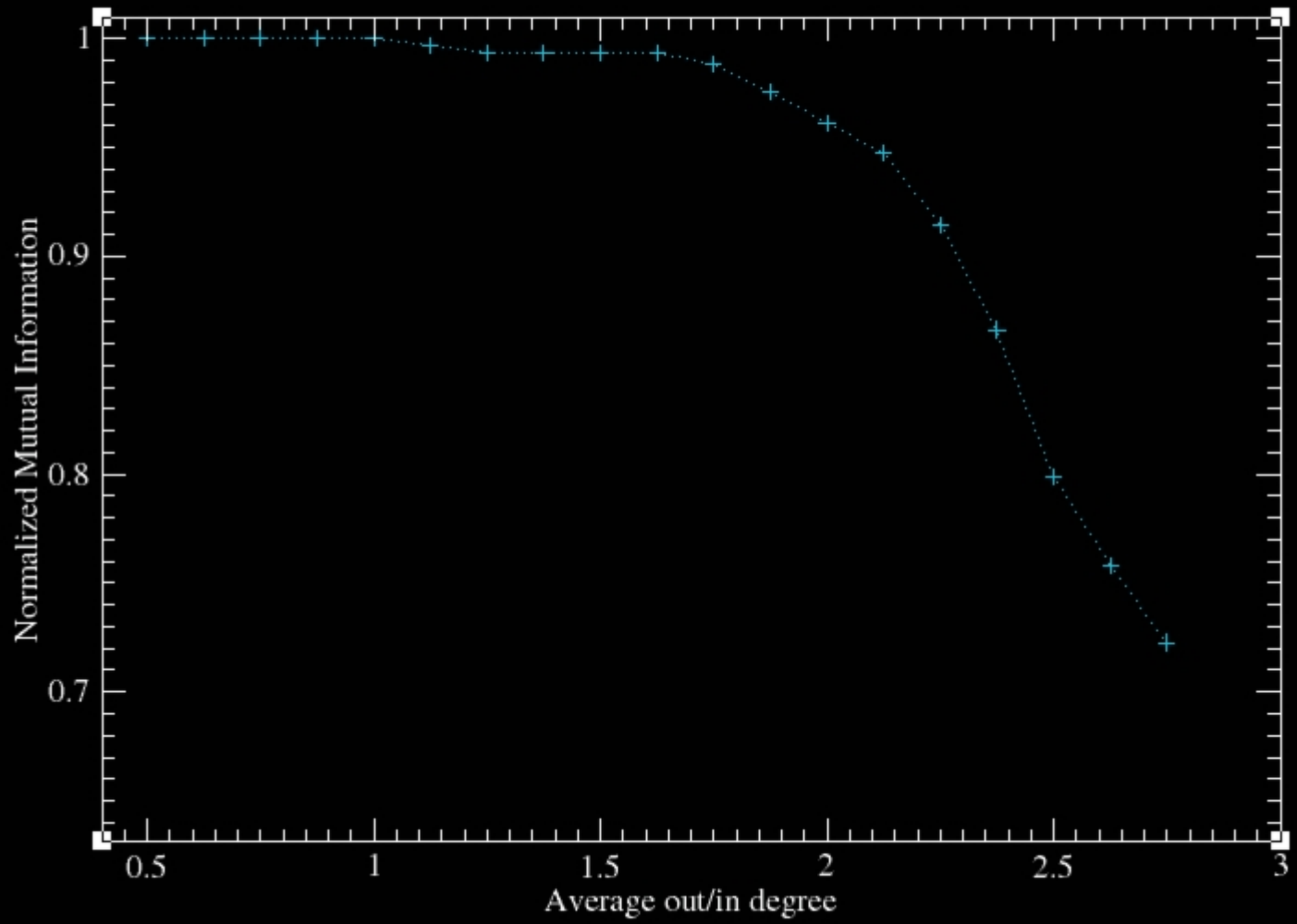
- **The “best” partition corresponds to the longest plateau of F vs α !**
- **Hierarchical levels are determined by partitions at lower (higher) α produced by complete splitting or merging of clusters of the best partition**

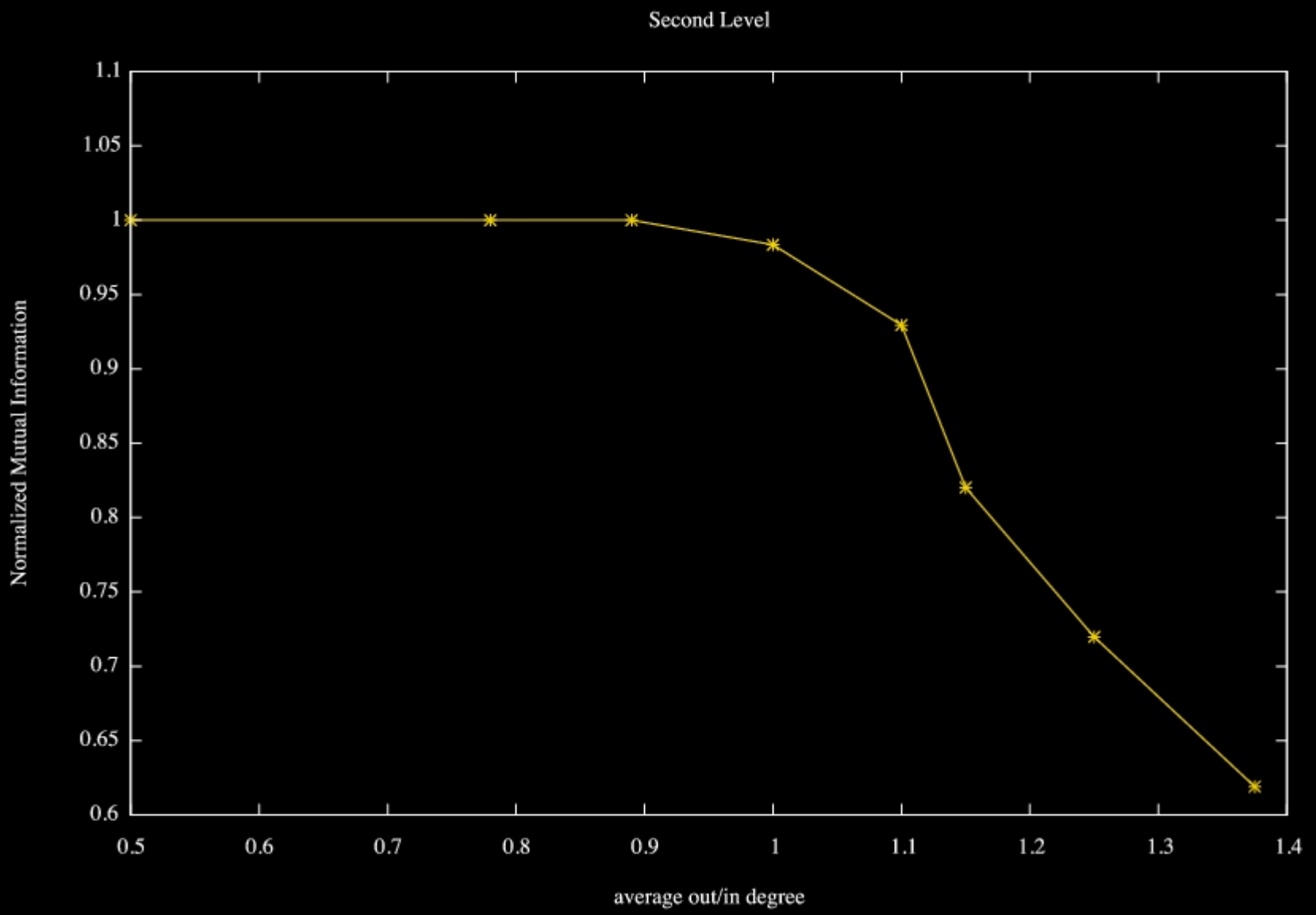
Hierarchical benchmark



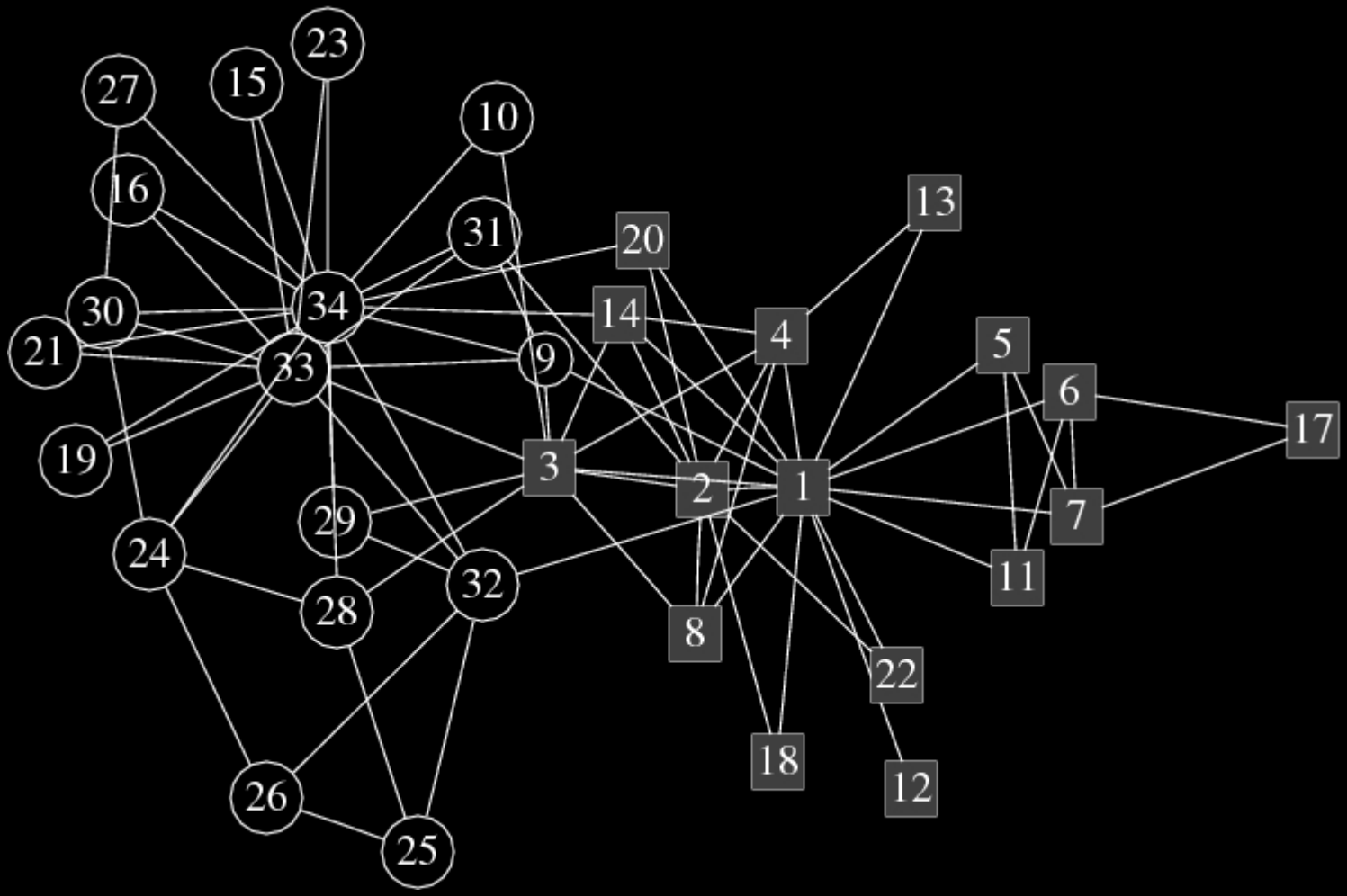
Two levels: 4 communities of 128 nodes, each including 4 communities of 32

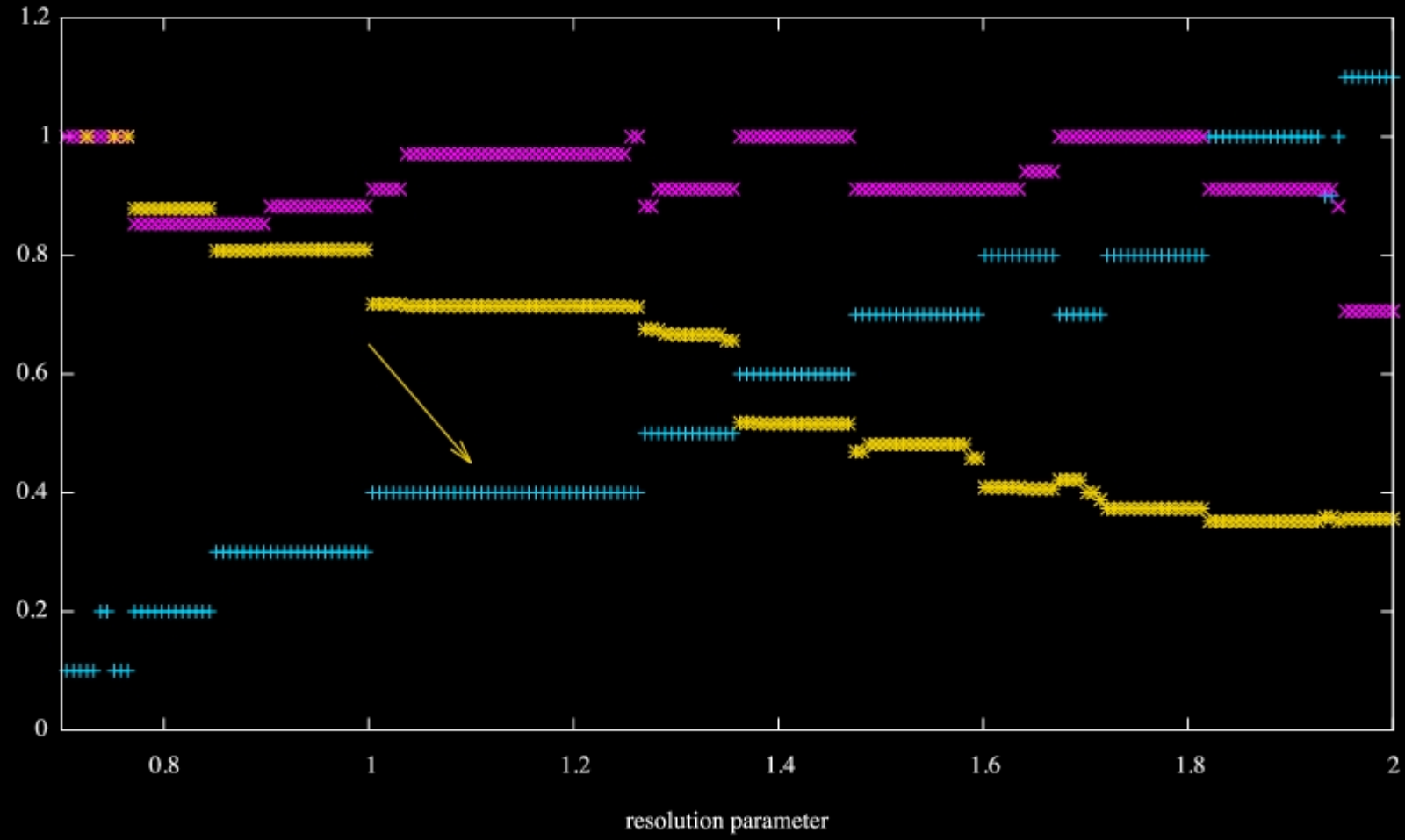
First Level


Normalized Mutual Information
1
0.9
0.8
0.7
0.5
1
1.5
2
2.5
3
Average out/in degree


Second Level
Normalized Mutual Information
1.1
1.05
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.5
0.6
0.7
0.8
0.9
1
1.1
1.2
1.3
1.4
average out/in degree

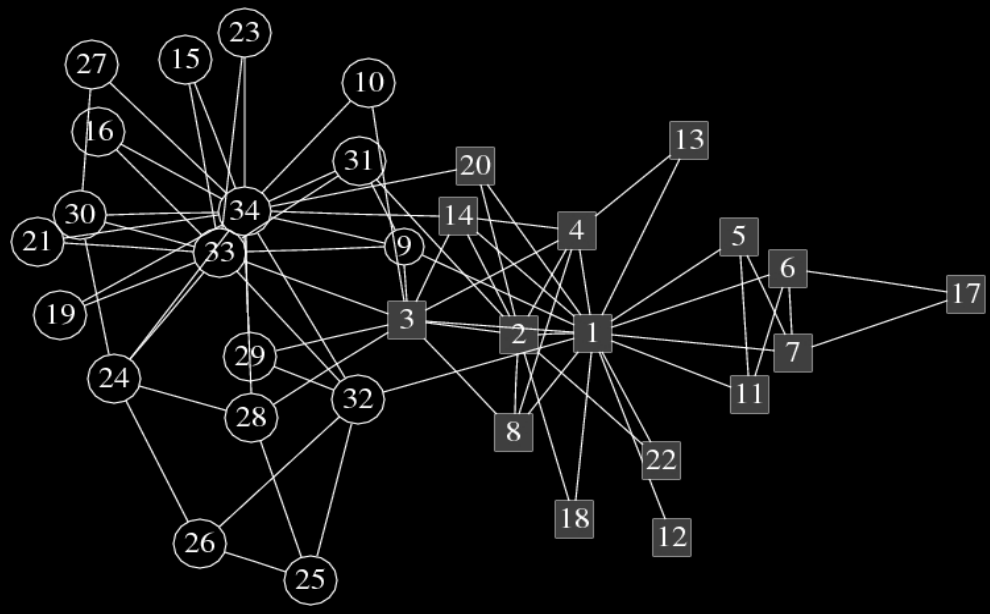
Zachary's karate club




1.2
1
0.8
0.6
0.4
0.2
0
0.8
1
1.2
1.4
1.6
1.8
2
resolution parameter

Best partition in 4 clusters: natural partition in 2 corresponds to the higher hierarchy

Overlapping nodes: 3, 9, 10, 14, 31

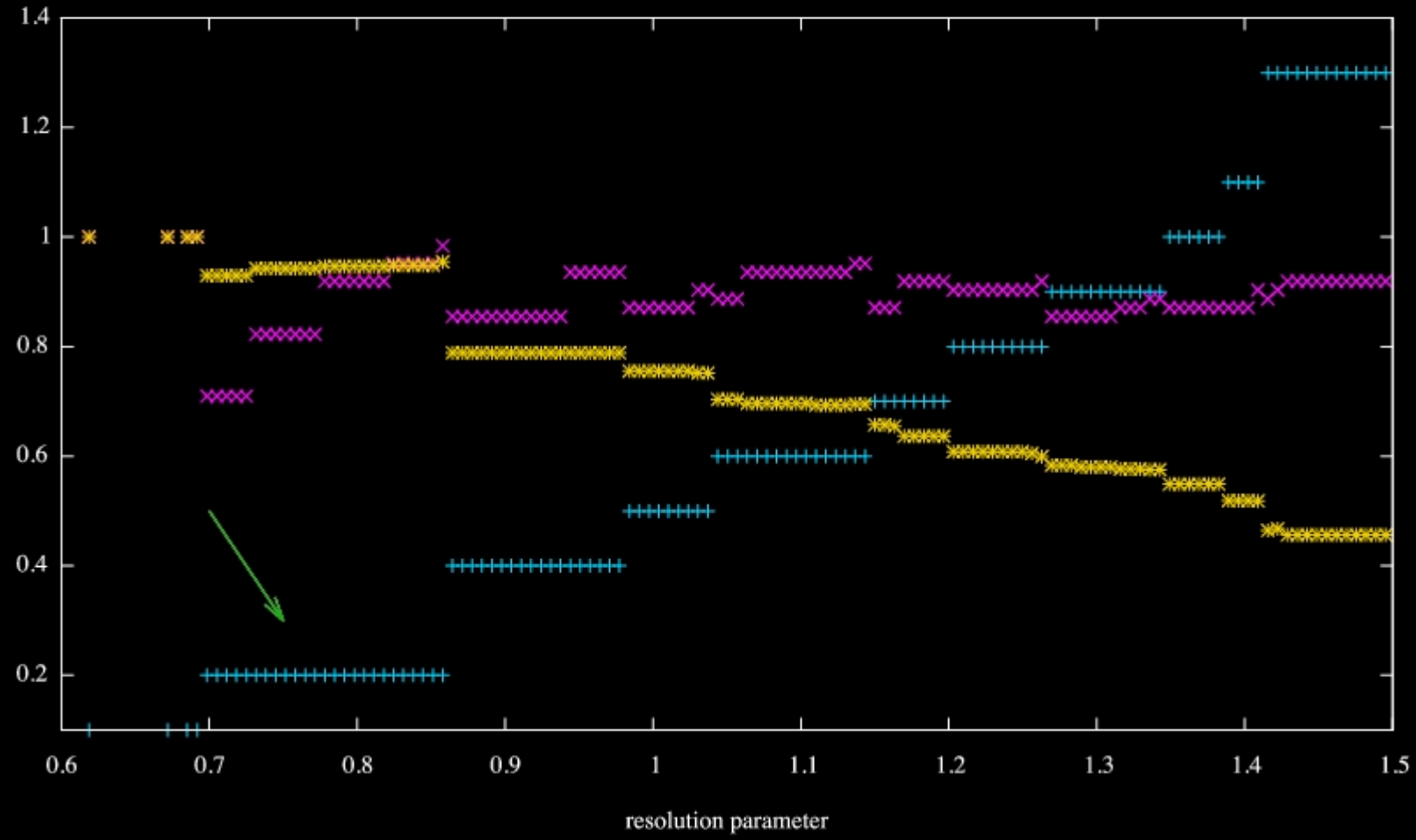


Dolphins' network

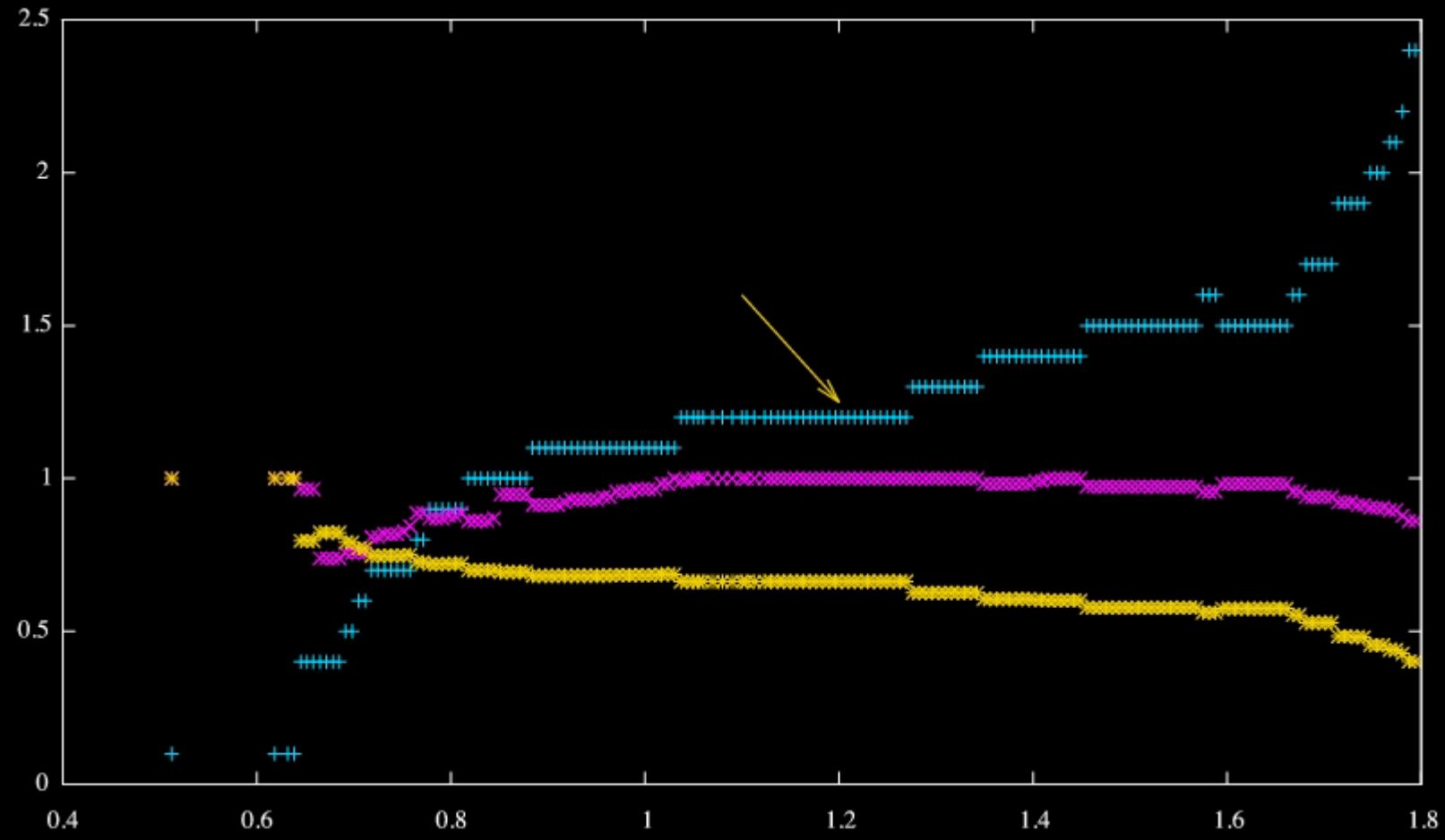
Studied by Lusseau (2003)

62 nodes, two "social" communities

Best split exactly matches the natural partition

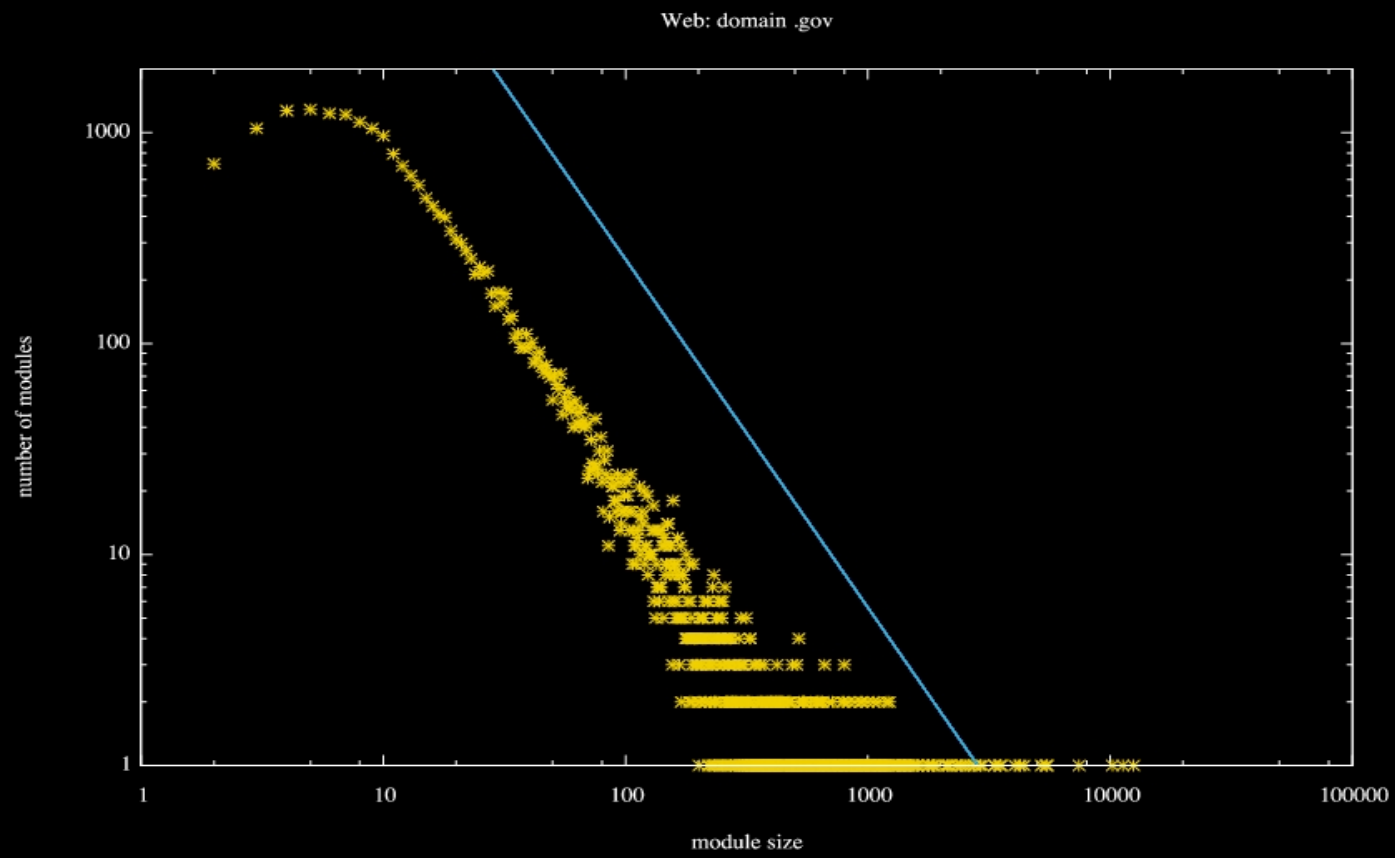

1.4
1.2
1
0.8
0.6
0.4
0.2
0.6
0.7
0.8
0.9
1
1.1
1.2
1.3
1.4
1.5
resolution parameter

College football



Web graph: domain .gov

774908 URLs, 4711340 links



Summary

Our method is:

- **Fast**
- **Easy to implement**
- **It finds overlapping nodes**
- **It finds hierarchies**
- **Tests on artificial and real networks give excellent results**

So use it!