- “Donat el caràcter i la finalitat exclusivament docent i eminentment il·lustrativa de les explicacions a classe d'aquesta presentació, l’autor s’acull a l’article 32 de la Llei de propietat intel·lectual vigent respecte de l'ús parcial d'obres alienes com ara imatges, gràfics o altre material contingudes en les diferents diapositives”
- “Dado el carácter y la finalidad exclusivamente docente y eminentemente ilustrativa de las explicaciones en clase de esta presentación, el autor se acoge al artículo 32 de la Ley de Propiedad Intelectual vigente respecto al uso parcial de obras ajenas como imágenes, gráficos u otro material contenidos en las diferentes diapositivas”.

1

UNIVERSITAT DE BARCELONA

Complex Networks

Albert Díaz Guilera
http://diaz-guilera.net
@anduviera

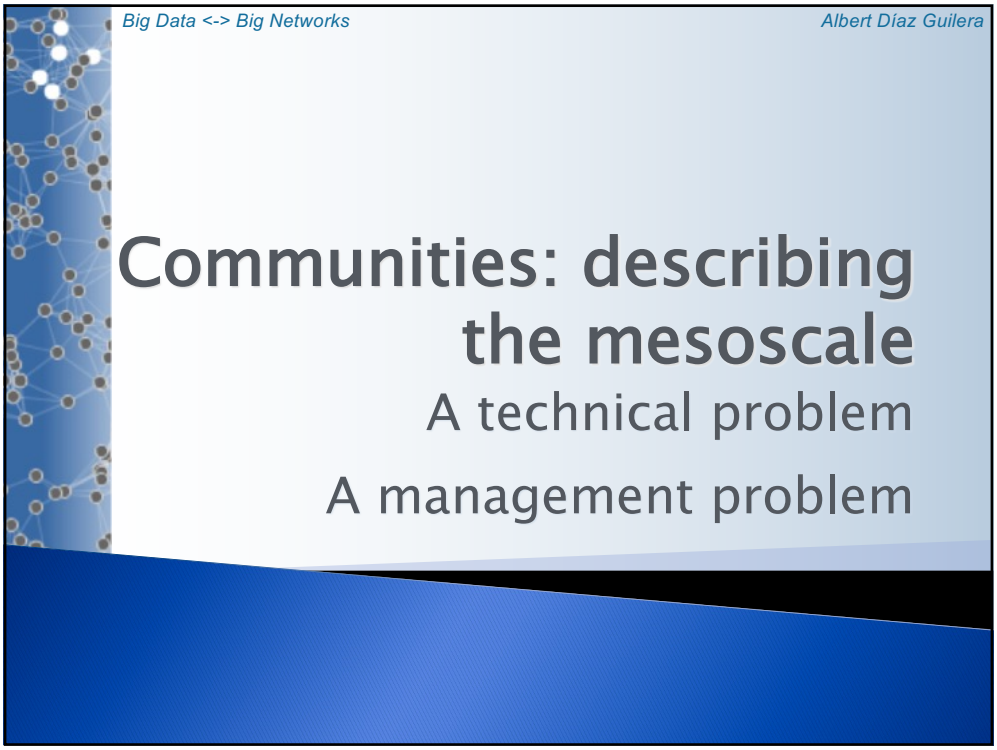
CLabB complexity lab barcelona

COMPLEXITAT

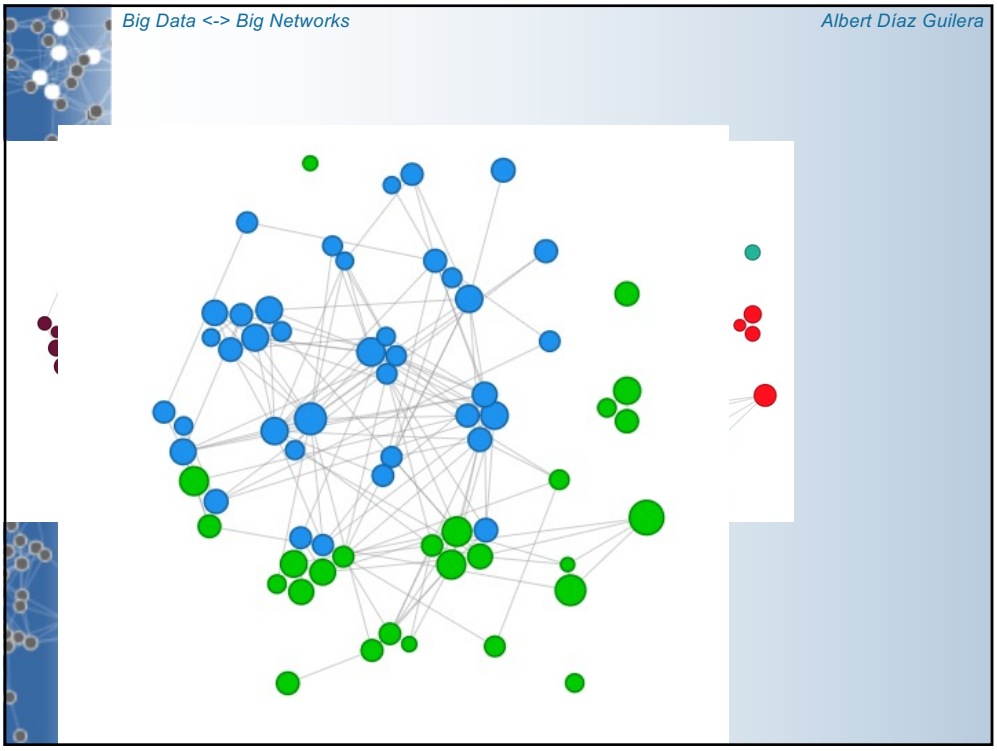
2


UNIVERSITAT DE BARCELONA
Communities
Albert Díaz Guilera
http://diaz-guilera.net
@anduviera
CLabB
complexity lab barcelona
COMPLEXITAT

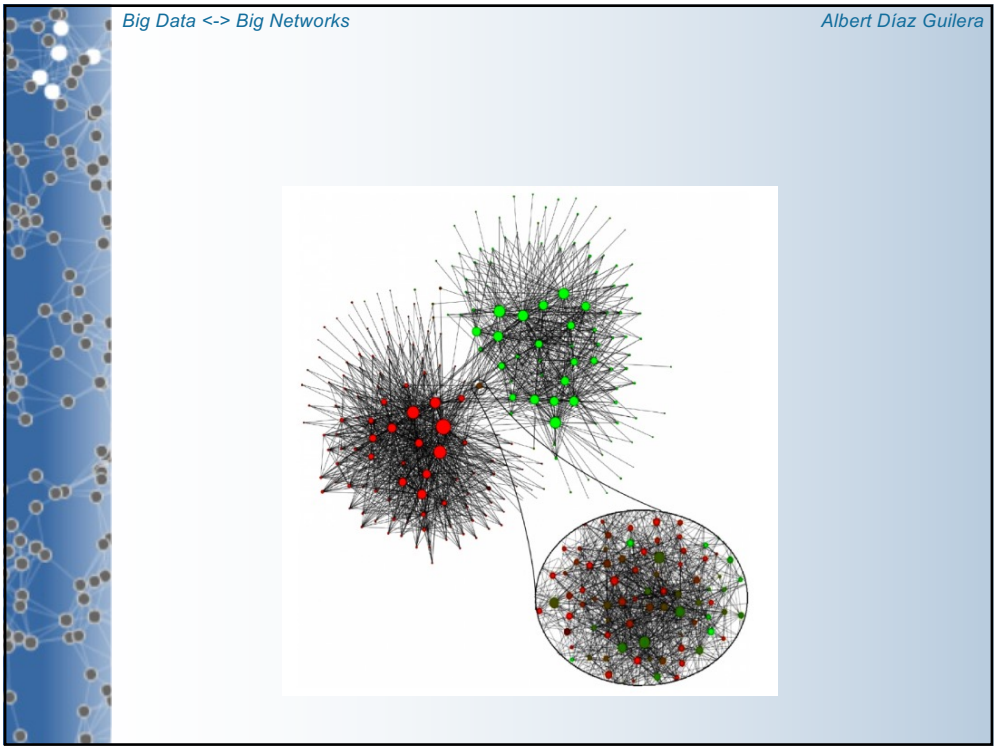
3


Big Data <-> Big Networks
Albert Díaz Guilera
Communities: describing the mesoscale
A technical problem
A management problem

6

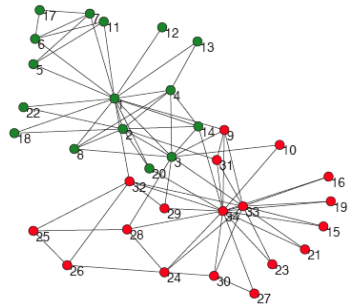

Big Data <-> Big Networks
Albert Díaz Guilera

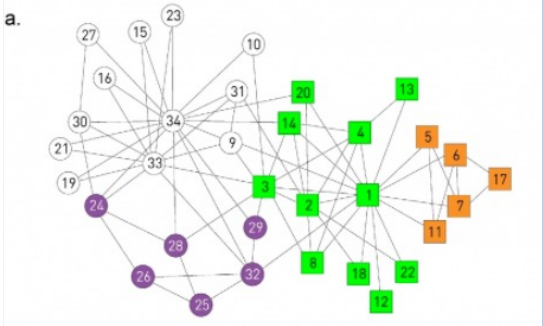
7

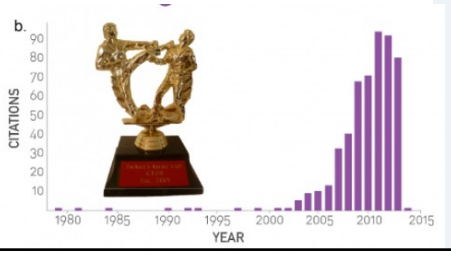

Big Data <-> Big Networks
Albert Díaz Guilera

8

Zackary's Karate Clubb





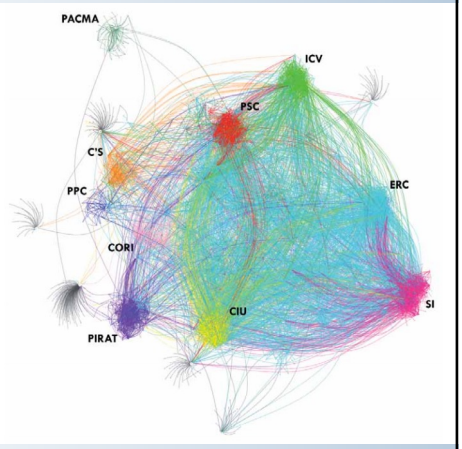


http://networkkarate.tumblr.com/

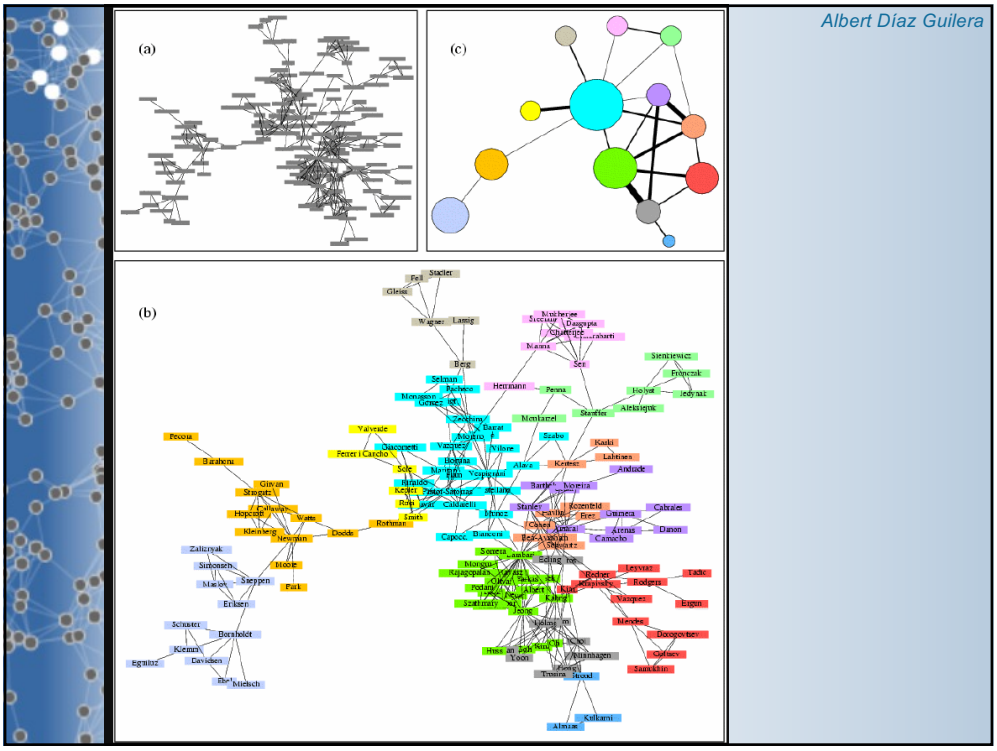
9

Communities

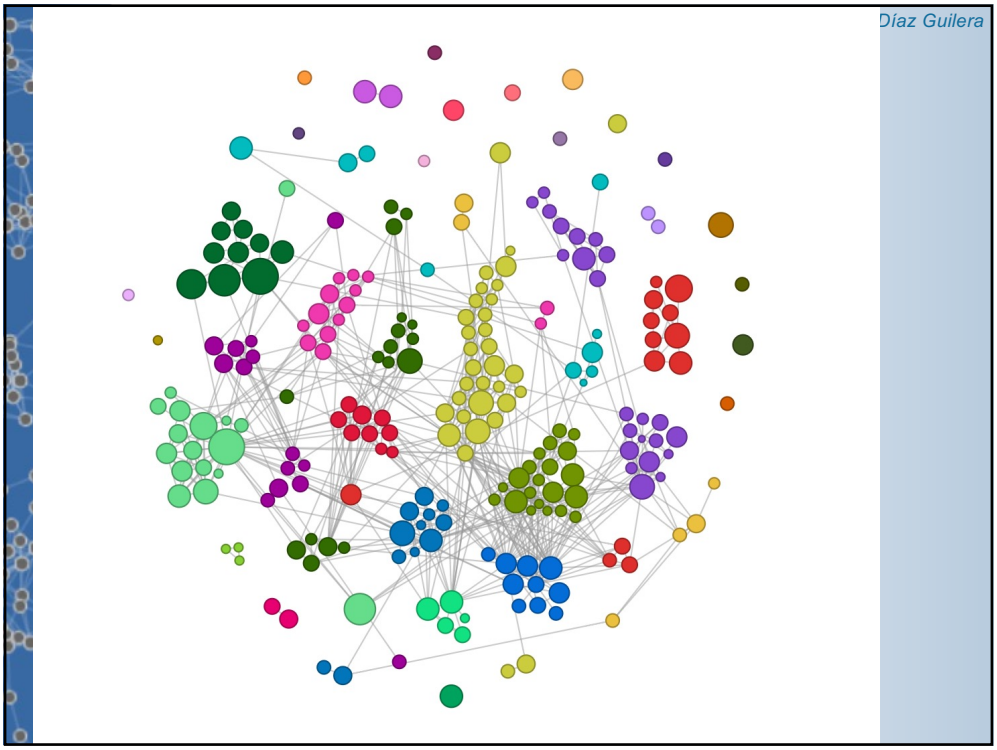
- Social networks are formed by communities
- According to:
  - Political reasons
  - Religion
  - Education
  - Scientific disciplines
  - ……..



10



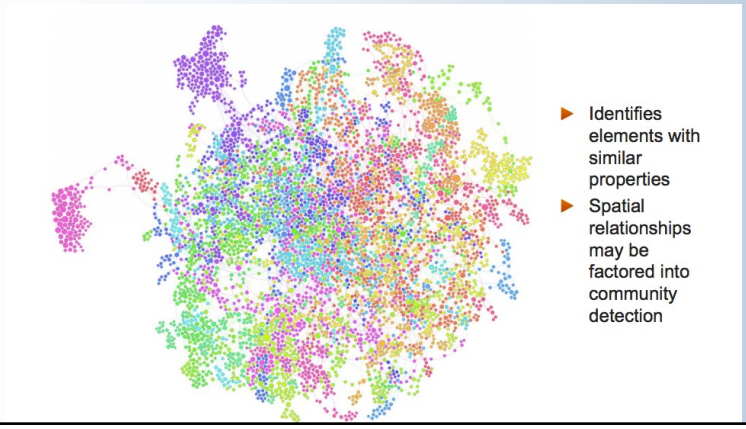
11



12

Technological Clusters

- Technological networks
  - Internet: connections according to geographical proximity
  - Power grids

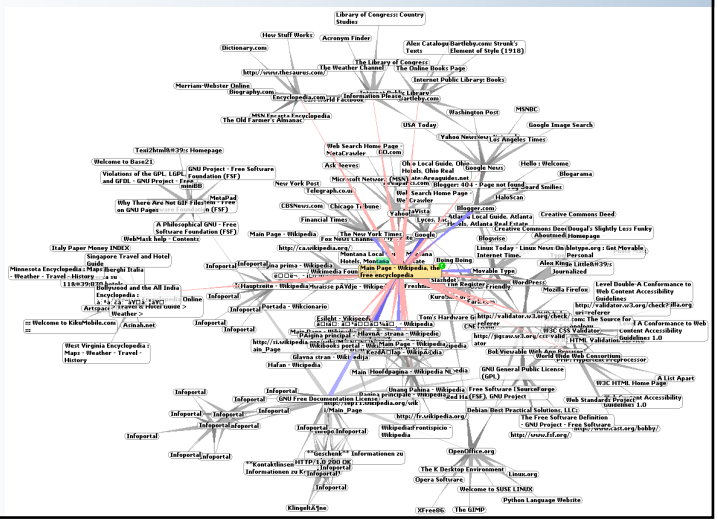


- Identifies elements with similar properties
- Spatial relationships may be factored into community detection

13

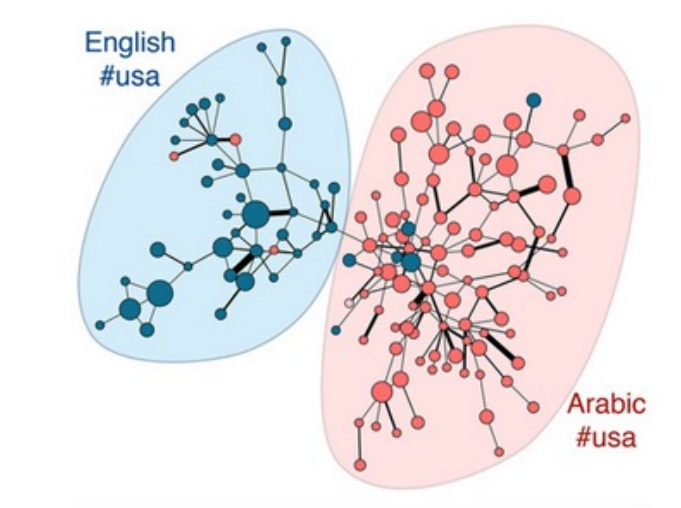
Thematic Clusters

- World-Wide Web: grouped by themes



14

Socio-technological networks



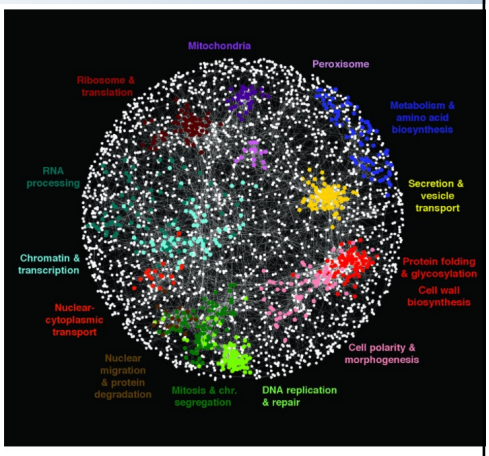
15

Communities and mobility

16

Modules

- Biological networks
- Gene regulatory networks: groups are functional modules

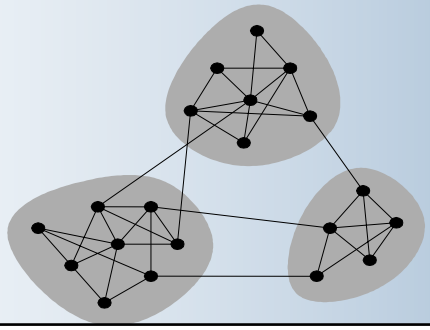


A GENETIC MAP SHOWING INTERACTIONS BETWEEN GENES IN A CEL

17

Objectives

- Existence of communities or modules in networks
- Technical issue: finding the best partition
- Management issue: finding meaningful partitions



18

Technical issue

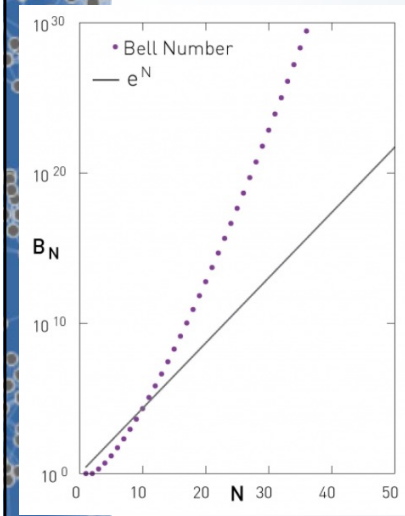
- We have to identify the communities
- How many possible partitions into communities?
- NP problem to find the best one

Bell number

From Wikipedia, the free encyclopedia

In combinatorial mathematics, the **Bell numbers** count the number of partitions of a set. These numbers have been studied by mathematicians since the 19th century, and their roots go back to medieval Japan, but they are named after Eric Temple Bell, who wrote about them in the 1930s.

19



Number of Partitions

The number of partitions of a network of size N is provided by the Bell number (9.6). The figure compares the Bell number to an exponential function, illustrating that the number of possible partitions grows faster than exponentially. Given that there are over 1040 partitions for a network of size N=50, brute-force approaches that aim to identify communities by inspecting all possible partitions are computationally infeasible.

10^{40}

Graph Partitioning

We can solve the graph bisection problem by inspecting all possible divisions into two groups and choosing the one with the smallest cut size. To determine the computational cost of this brute force approach we note that the number of distinct ways we can partition a network of N nodes into groups of N_1 and N_2 nodes is

$$\frac{N!}{N_1!N_2!} \qquad (9.3)$$

Using Stirling's formula

$$n! \simeq \sqrt{2\pi n}(n/e)^n$$

we can write (9.3) as

$$\frac{N!}{N_1!N_2!} \simeq \frac{\sqrt{2\pi N}(N/e)^N}{\sqrt{2\pi N_1}(N_1/e)^{N_1}\sqrt{2\pi N_2}(N_2/e)^{N_2}} \sim \frac{N^{N+1/2}}{N_1^{N_1+1/2}N_2^{N_2+1/2}} \qquad (9.4)$$

To simplify the problem let us set the goal of dividing the network into two equal sizes $N_1 = N_2 = N/2$. In this case (9.4) becomes

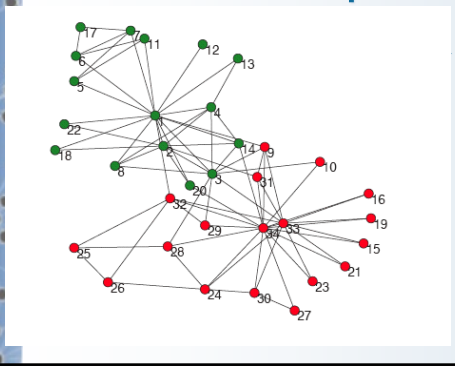
$$\frac{2^{N+1}}{\sqrt{N}} = e^{(N+1)\ln 2 - \frac{1}{2}\ln N} \qquad (9.5)$$

indicating that the number of bisections increases exponentially with the size of the network.

20

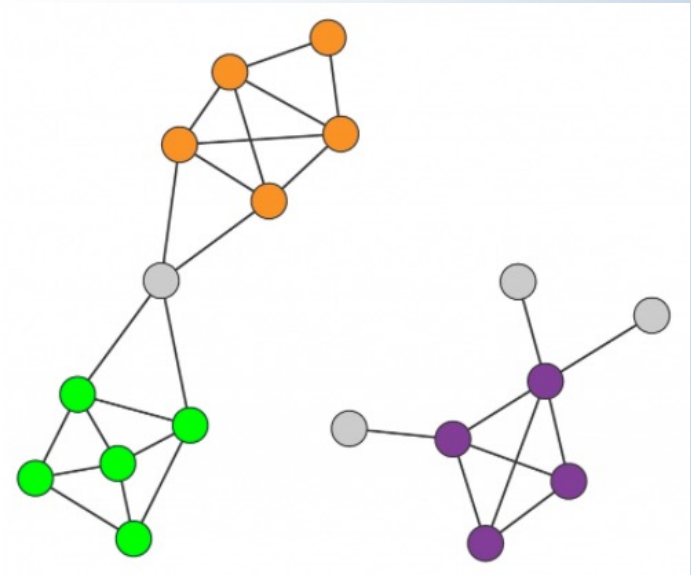
Communities: intuitive picture

- Definition: subsets of nodes that are more densely linked, when compared with the rest



Zachary's Karate club

21



A community is a locally dense connected subgraph in a network.

22

Textbook online

- Network Science by A.-L. Barabasi

23

Maximum cliques

- Define a community as group of individuals whose members all know each other. In graph theoretic terms this means that a community is a complete subgraph, or a **clique**.
- A clique is a connected subgraph with maximal link density.
- But:
  - Triangles define clustering and are quite common
  - Squares, ... very very rare
  - Too restrictive definition

24

Strong and weak

- ***Strong community***: C is a strong community if each node within C has more links within the community than with the rest of the graph
- ***Weak community***: C is a weak community if the total internal degree of a subgraph exceeds its total external degree

25

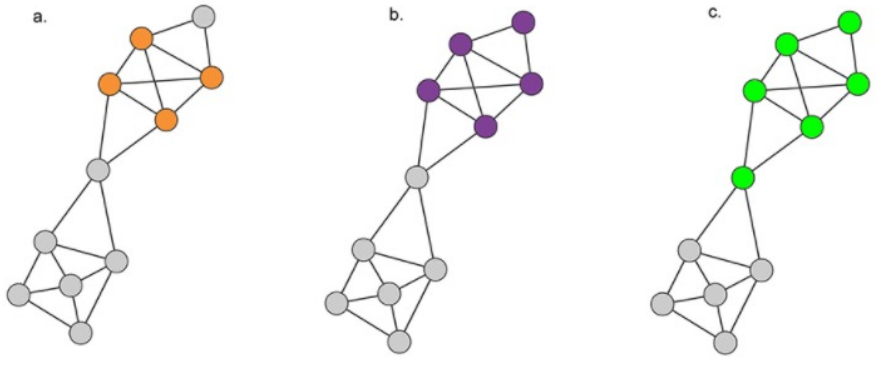


Image 9.5

Defining Communities

- **Cliques**

A *clique* corresponds to a complete subgraph. The highest order clique of this network is a square, shown in orange. There are several three-node cliques on this network. Can you find them?

- **Strong Communities**

A *strong community*, defined in (9.1), is a connected subgraph whose nodes have more links to other nodes in the same community that to nodes that belong to other communities. Such a strong community is shown in purple. There are additional strong communities on the graph - can you find at least two more?

- **Weak Communities**

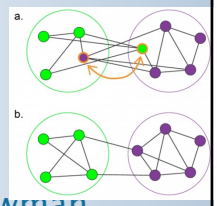
A weak community defined in (9.2) is a subgraph whose nodes' total internal degree exceeds their total external degree. The green nodes represent one of the several possible weak communities of this network.

We still have some freedom on how to define each community

26

Partition

- A partition is a division of the network into groups, communities or clusters
- The question is: Which of all possible partitions is the best?
- NP problem
- Community detection:
  - From computer scientists (graph partitioning, Kernighan-Lin)
  - To statistical physicists (Girvan-Newman, PNAS 99, 7821, 2002)



27

Quantifying a partition

- **Modularity:**

$$Q = \sum_i (e_{ii} - a_i^2)$$

- e_{ij}: **fraction of total links starting at a node in partition i and ending at a node in partition j**
- a_i: **fraction of links connected to i**
- a_i^2: **number of intracommunity links**

28

Modularity

Consider a network with N nodes and L links and a partition into n_c communities, each community having N_c nodes connected to each other by L_c links, where c=1,...,n_c. If L_c is larger than the expected number of links between the N_c nodes given the network's degree sequence, then the nodes of the subgraph C_c could indeed be part of a true community, as expected based on the Density Hypothesis H2 (Image 9.2). We therefore measure the difference between the network's real wiring diagram (A_{ij}) and the expected number of links between i and j if the network is randomly wired (p_{ij}),

$$M_c = \frac{1}{2L} \sum_{(i,j)\in C_c} (A_{ij} - p_{ij}) \qquad (9.9)$$

Here p_{ij} can be determined by randomizing the original network, while keeping the expected degree of each node unchanged. Using the degree preserving null model (7.1) we have

$$p_{ij} = \frac{k_i k_j}{2L} \qquad (9.10)$$

29

If M_c is positive, then the subgraph C_c has more links than expected by chance, hence it represents a potential community. If M_c is zero then the connectivity between the N_c nodes is random, fully explained by the degree distribution. Finally, if M_c is negative, then the nodes of C_c do not form a community.

Using (9.10) we can derive a simpler form for the modularity (9.9) (ADVANCED TOPICS 9.B)

$$M_c = \frac{L_c}{L} - \left(\frac{k_c}{2L}\right)^2 \qquad (9.11)$$

where L_c is the total number of links within the community C_c and k_c is the total degree of the nodes in this community.

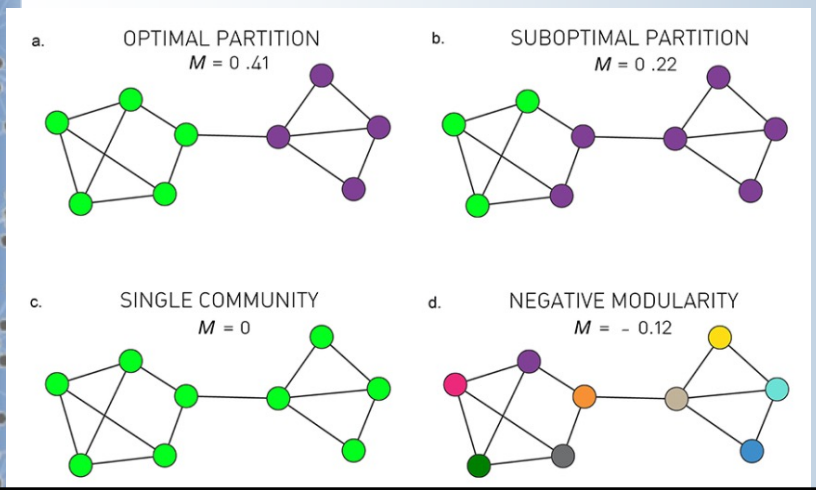
To generalize these ideas to a full network consider the complete partition that breaks the network into n_c communities. To see if the local link density of the subgraphs defined by this partition differs from the expected density in a randomly wired network, we define the partition's *modularity* by summing (9.11) over all n_c communities [23]

$$M = \sum_{c=1}^{n_c} \left[\frac{L_c}{L} - \left(\frac{k_c}{2L}\right)^2\right] \qquad (9.12)$$

30

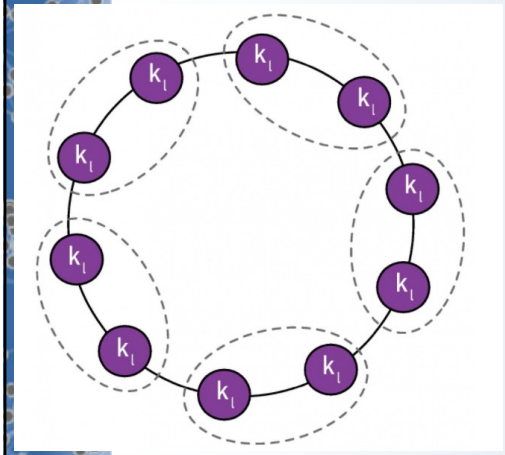
Modularity

- Higher modularity better partition
- Zero modularity (1 or N)



31

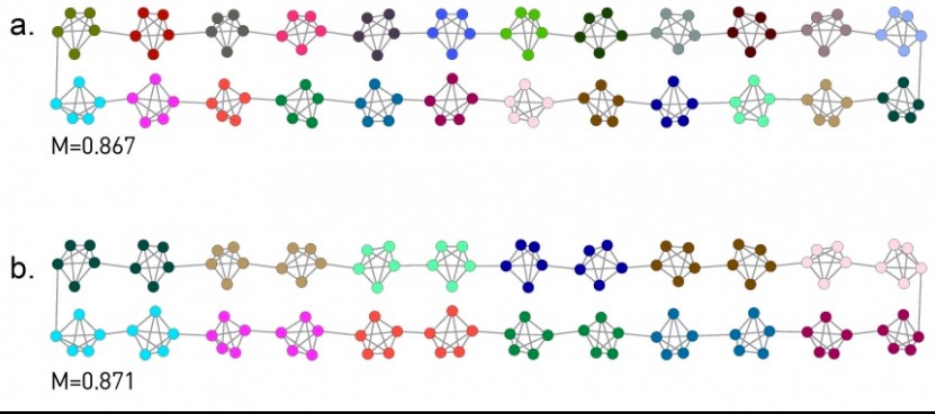
Resolution



- Modularity Resolution Limit

Consider a network consisting of a ring of n_c cliques, each clique having N_c nodes and $m(m-1)/2$ links. The neighboring cliques are connected by a single link (Image 9.34). The network has an obvious community structure, each community corresponding to a clique.

32



33

Methods of community identification

- L. Danon, J. Duch, A.D-G, A. Arenas J. Stat. Mech. (2005) P09008
  - Link removal methods
  - Agglomerative methods
  - Maximizing modularity
  - Spectral analysis methods
  - Based on physics: resistor networks, q-Potts model
- More recent reviews:
  - S. Fortunato, *Community detection in graphs* (Phys. Rep. 486, 75-174, 2010)

34

Computational costs

Reference	Alias	Order
(Newman and Girvan, 2004)	NG	$O(m^2n)$
(Girvan and Newman, 2002)	GN	$O(n^2m)$
(Fortunato et al., 2004)	FLM	$O(n^4)$
(Radicchi et al., 2004)	RCCLP	$O(n^2)$
(Newman, 2004b)	NF	$O(n\log^2 n)$
(Donetti and Muñoz, 2004),	DMSA	$O(n^3)$
(Donetti and Muñoz, 2004),	DMCA	$O(n^3)$
(Eckmann and Moses, 2002)	EM	$O(m\langle k^2\rangle)$
(Zhou and Lipowsky, 2005)	ZL	$O(n^3)$
(Reichardt and Bornholdt, 2004)	RB	unknown
(Bagrow and Bollt, 2004)	BB	$O(n^3)$
(Duch and Arenas, 2005)	DA	$O(n^2\log n)$
(Capocci et al., 2004)	CSCC	$O(n^2)$
(Wu and Huberman, 2004)	WH	$O(n+m)$

35

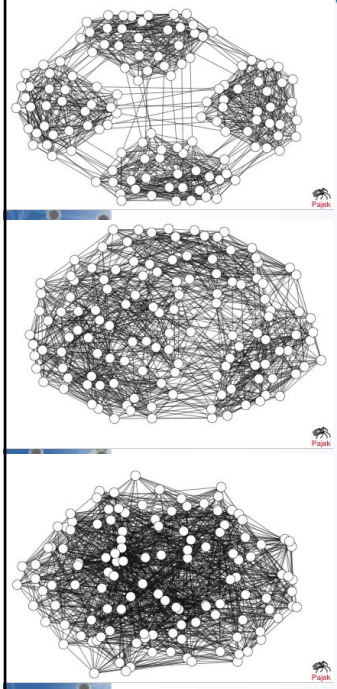
Name	Nature	Comp.	REF
Ravasz	Hierarchical Agglomerative	$O(N^2)$	[11]
Girvan-Newman	Hierarchical Divisive	$O(N^2)$	[9]
Greedy Modularity	Modularity Optimization	$O(N^2)$	[33]
Greedy Modularity (Optimized)	Modularity Optimization	$O(N\log^2 N)$	[35]
Louvain	Modularity Optimization	O(L)	[2]
Infomap	Flow Optimization	O(NlogN)	[44]
Clique Percolation (CFinder)	Overlapping Communities	Exp(N)	[48]
Link Clustering	Hierarchical Agglomerative; Overlapping Communities	$O(N^2)$	[51]

36

Comparing algorithms

- *ad-hoc* networks (Newman-Girvan, PRE 69, 026113, 2004)
  - 128 nodes
  - 4 communities of 32 nodes each
  - Each node has 16 links:
    - z_{in} internal links within the community
    - z_{out} links out of its community

37



$z_{in}=15$

$z_{in}=12$

$z_{in}=8$

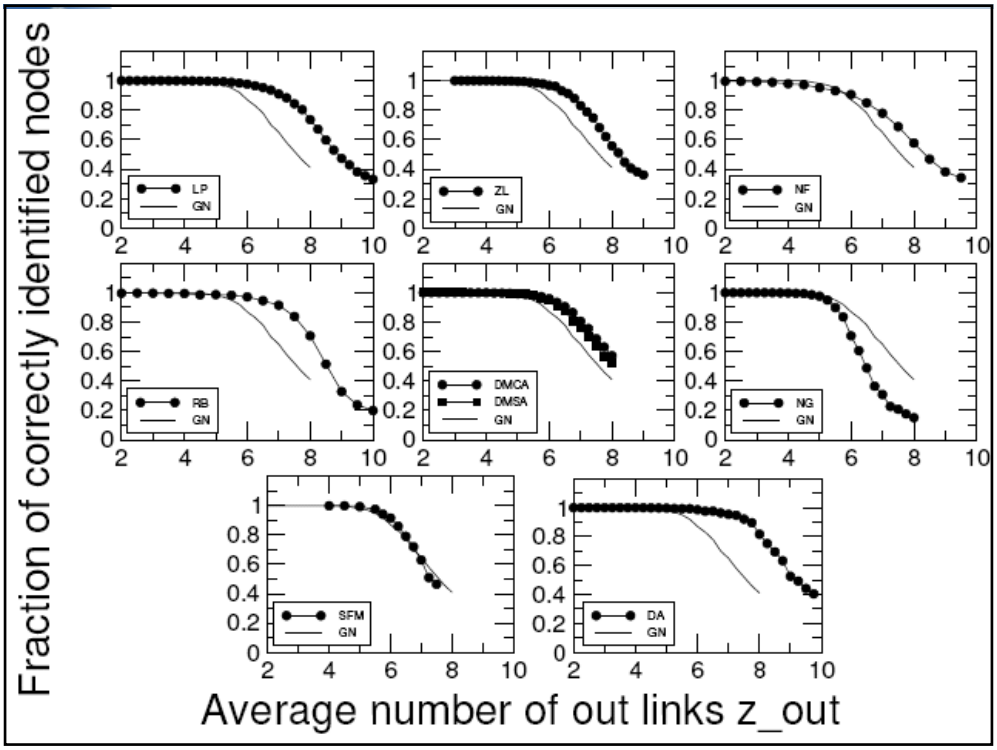
38

Journal of Statistical Mechanics: Theory and Experiment
An IOP and SISSA journal

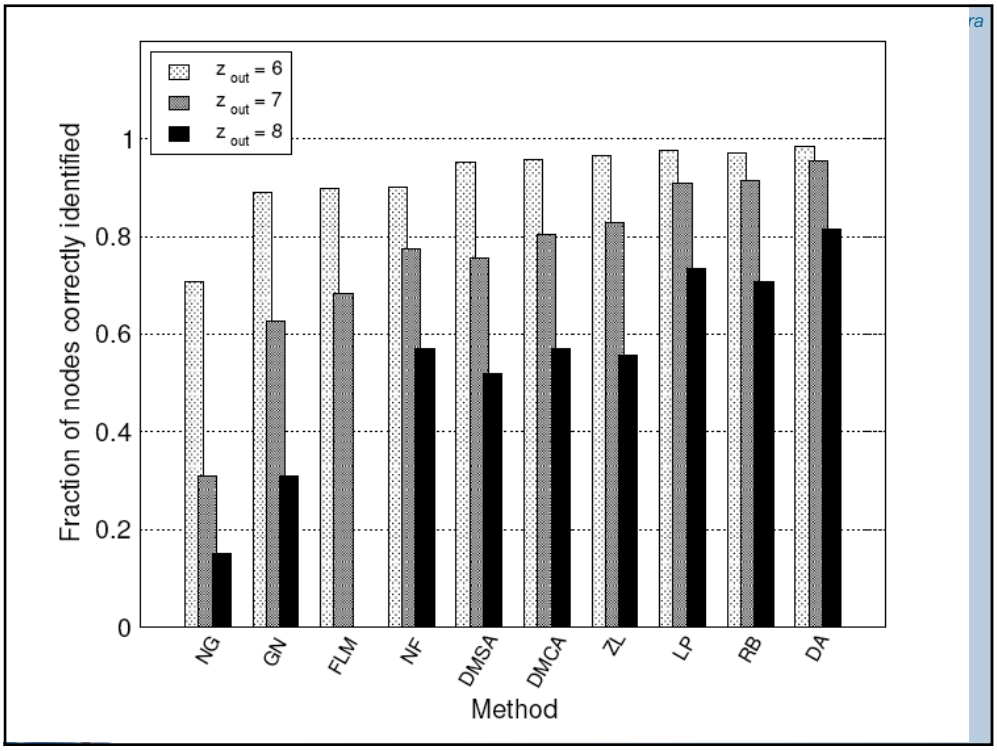
Comparing community structure identification

Leon Danon[1,2], Albert Díaz-Guilera[1], Jordi Duch[2] and Alex Arenas[2]

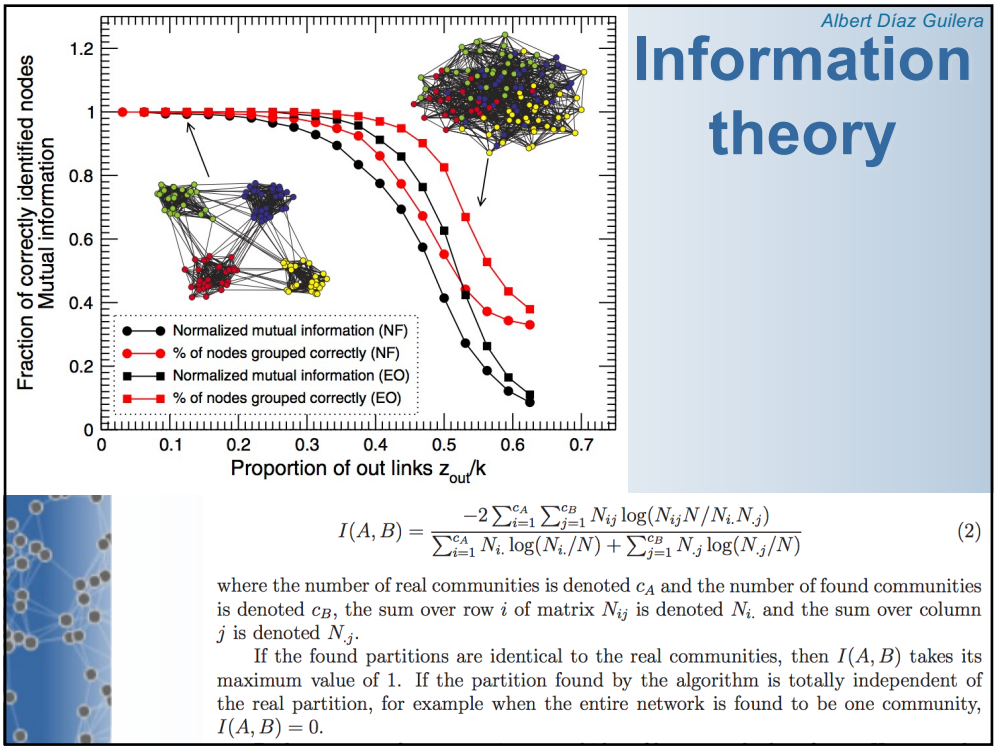
39



40



41



$$I(A,B) = \frac{-2\sum_{i=1}^{c_A}\sum_{j=1}^{c_B} N_{ij}\log(N_{ij}N/N_{i.}N_{.j})}{\sum_{i=1}^{c_A} N_{i.}\log(N_{i.}/N) + \sum_{j=1}^{c_B} N_{.j}\log(N_{.j}/N)} \quad (2)$$

where the number of real communities is denoted c_A and the number of found communities is denoted c_B, the sum over row i of matrix N_{ij} is denoted $N_{i.}$ and the sum over column j is denoted $N_{.j}$.

If the found partitions are identical to the real communities, then $I(A,B)$ takes its maximum value of 1. If the partition found by the algorithm is totally independent of the real partition, for example when the entire network is found to be one community, $I(A,B) = 0$.

42

Measuring Accuracy

To compare the predicted communities with those planted in the benchmark, consider an arbitrary partition into non-overlapping communities. In each step we randomly choose a node and record the label of the community it belongs to. The result is a random string of community labels that follow a $p(C)$ distribution, representing the probability that a randomly selected node belongs to the community C.

Consider two partitions of the same network, one being the benchmark (ground truth) and the other the partition predicted by a community finding algorithm. Each partition has its own $p(C_1)$ and $p(C_2)$ distribution. The joint distribution, $p(C_1, C_2)$, is the probability that a randomly chosen node belongs to community C_1 in the first partition and C_2 in the second. The similarity of the two partitions is captured by the normalized mutual information [38]

$$I_n = \frac{\sum_{C_1,C_2} p(C_1,C_2)\log_2 \frac{p(C_1,C_2)}{p(C_1)p(C_2)}}{\frac{1}{2}H(\{p(C_1)\})+\frac{1}{2}H(\{p(C_2)\})} \qquad (9.19)$$

43

The numerator of (9.19) is the *mutual information* I, measuring the information shared by the two community assignments: I=0 if C_1 and C_2 are independent of each other; I equals the maximal value $H(\{p(C_1)\}) = H(\{p(C_2)\})$ when the two partitions are identical and is the Shannon entropy.

$$H(\{p(C)\}) = -\sum_C p(C)\log_2 p(C) \qquad (9.20)$$

If all nodes belong to the same community, then we are certain about the next label and H=0, as we do not gain new information by inspecting the community to which the next node belongs to. H is maximal if $p(C)$ is the uniform distribution, as in this case we have no idea which community comes next and each new node provides H bits of new information.

In summary, I_n=1 if the benchmark and the detected partitions are identical, and I_n=0 if they are independent of each other. The utility of I_n is illustrated in Image 9.25b that shows the accuracy of the Ravasz algorithm for the Girvan-Newman benchmark. In Image 9.27 we use I_n to test the performance of each algorithm against the GN and LFR benchmarks. The results allow us to draw several conclusions:

- We have I_n=1 for $\mu < 0.5$. Consequently when the link density within communities is high compared to their surroundings, most algorithms accurately identify the planted communities. Beyond μ=0.5 the accuracy of each algorithm drops.
- The accuracy is benchmarks dependent. For the more realistic LFR benchmark the Louvain and the Ravasz methods offer the best performance and greedy modularity performs poorly.

44

Optimal?????

- For a given network the partition with **maximum modularity** corresponds to the **optimal** community structure.
- The hypothesis is supported by the inspection of small networks, for which the **maximum M agrees with the expected** communities

45

Greedy

Greedy algorithm

From Wikipedia, the free encyclopedia

A **greedy algorithm** is an algorithmic paradigm that follows the problem solving heuristic of making the locally optimal choice at each stage[1] with the hope of finding a global optimum. In many problems, a greedy strategy does not in general produce an optimal solution, but nonetheless a greedy heuristic may yield locally optimal solutions that approximate a global optimal solution in a reasonable time.

For example, a greedy strategy for the traveling salesman problem (which is of a high computational complexity) is the following heuristic: "At each stage visit an unvisited city nearest to the current city". This heuristic need not find a best solution, but terminates in a reasonable number of steps; finding an optimal solution typically requires unreasonably many steps. In mathematical optimization, greedy algorithms solve combinatorial problems having the properties of matroids.

46

- Search for local maxima of modularity
- Propose small changes and accept or reject them with some probability
  - Division
  - Aggregation

47

The effect of size heterogeneity on community identification in complex networks

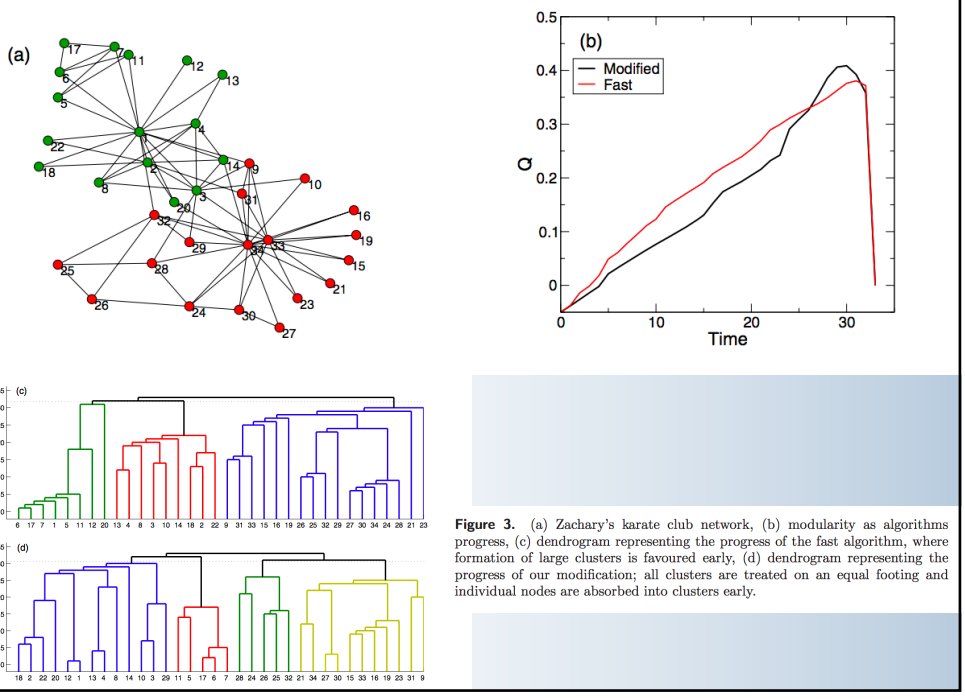
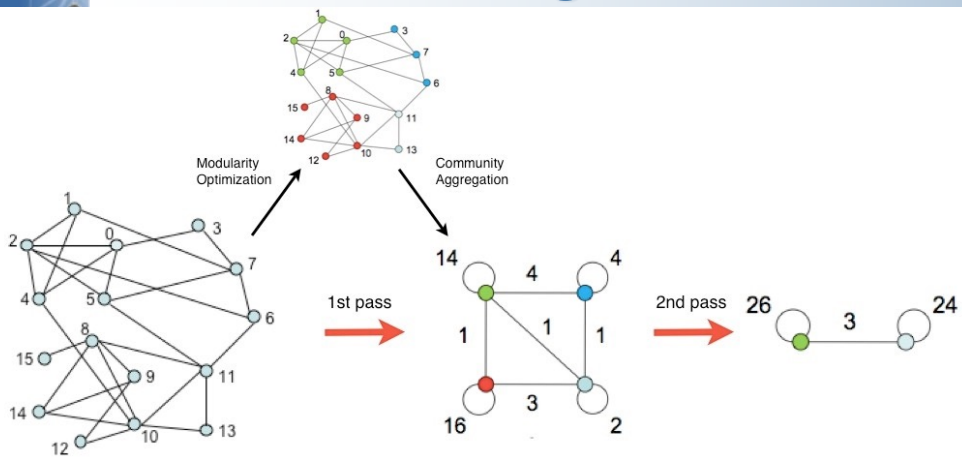


Figure 3. (a) Zachary's karate club network, (b) modularity as algorithms progress, (c) dendrogram representing the progress of the fast algorithm, where formation of large clusters is favoured early, (d) dendrogram representing the progress of our modification; all clusters are treated on an equal footing and individual nodes are absorbed into clusters early.

48

Louvain algorithm



49

First, each node in the network is assigned to its own community. Then for each node i, the change in modularity is calculated for removing i from its own community and moving it into the community of each neighbor j of i. This value is easily calculated by:[3]

$$\Delta Q = \left[\frac{\Sigma_{in} + k_{i,in}}{2m} - \left(\frac{\Sigma_{tot} + k_i}{2m} \right)^2 \right] - \left[\frac{\Sigma_{in}}{2m} - \left(\frac{\Sigma_{tot}}{2m} \right)^2 - \left(\frac{k_i}{2m} \right)^2 \right]$$

Where Σ_{in} is sum of all the weights of the links inside the community i is moving into, Σ_{tot} is the sum of all the weights of the links to nodes in the community, k_i is the weighted degree of i, $k_{i,in}$ is the sum of the weights of the links between i and other nodes in the community, and m is the sum of the weights of all links in the network. Then, once this value is calculated for all communities i is connected to, i is placed into the community that resulted in the greatest modularity increase. If no increase is possible, i remains in its original community. This process is applied repeatedly and sequentially to all nodes until no modularity increase can occur. Once this local maximum of modularity is hit, the first phase has ended.

50

Community detection for NetworkX's documentation

This module implements community detection.

It uses the louvain method described in Fast unfolding of communities in large networks, Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, Renaud Lefebvre, Journal of Statistical Mechanics: Theory and Experiment 2008(10), P10008 (12pp)

It depends on Networkx to handle graph operations : http://networkx.lanl.gov/

The program can be found in a repository where you can also report bugs :

https://bitbucket.org/taynaud/python-louvain

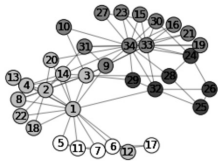
```
import community
import networkx as nx
import matplotlib.pyplot as plt

#better with karate_graph() as defined in networkx example.
#erdos renyi don't have true community structure
G = nx.erdos_renyi_graph(30, 0.05)

#first compute the best partition
partition = community.best_partition(G)

#drawing
size = float(len(set(partition.values())))
pos = nx.spring_layout(G)
count = 0.
for com in set(partition.values()) :
    count = count + 1.
    list_nodes = [nodes for nodes in partition.keys()
                                if partition[nodes] == com]
    nx.draw_networkx_nodes(G, pos, list_nodes, node_size = 20,
                                node_color = str(count / size)


nx.draw_networkx_edges(G,pos, alpha=0.5)
plt.show()
```

51

Community Detection with the Karate Club Network

In [25]:

```
import community

G=nx.read_edgelist("./data/karate.dat")

#first compute the best partition
partition = community.best_partition(G)

#plot the network
size = float(len(set(partition.values())))
pos = nx.spring_layout(G)
count = 0.
plt.axis('off')
for com in set(partition.values()) :
    count = count + 1.
    list_nodes = [nodes for nodes in partition.keys() \
                  if partition[nodes] == com]
    nx.draw_networkx_nodes(G, pos, list_nodes, node_size = 300, \
                           node_color = str(count / size))
    nx.draw_networkx_labels(G,pos)

nx.draw_networkx_edges(G,pos, alpha=0.5,width=1)
savefig('./data/karate_community.png',dpi=600)
```



52

Identifying communities

- Identifying what communities are
- Managerial point of view:
  - How a company is organized
  - How powerful is the formed informal chart

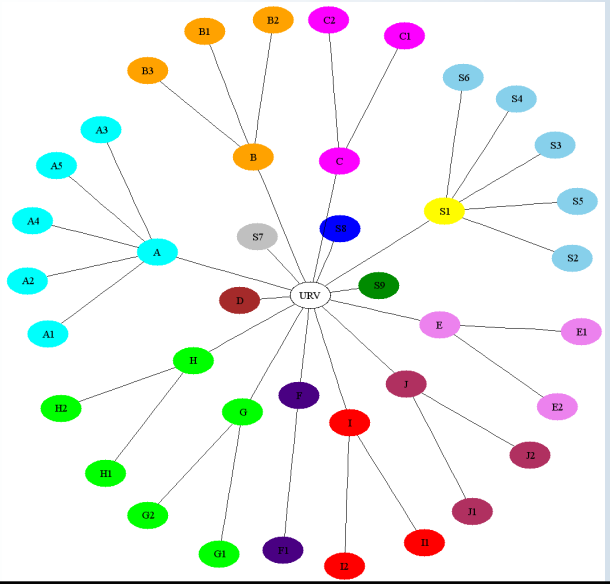
56

Two networks

- E-mail network at Universitat Rovira i Virgili
- FisEs

57

URV



58

Importance from management

Unravel the real (informal) organization behind the formal chart

"If the formal organization is the skeleton of a company, the informal is the central nervous system... Complex webs of social ties form every time colleagues communicate and solidify over time into surprisingly stable networks."

D. Krackhardt and J. R. Hanson, Harvard Business Review, 71, 104-113 (1993)

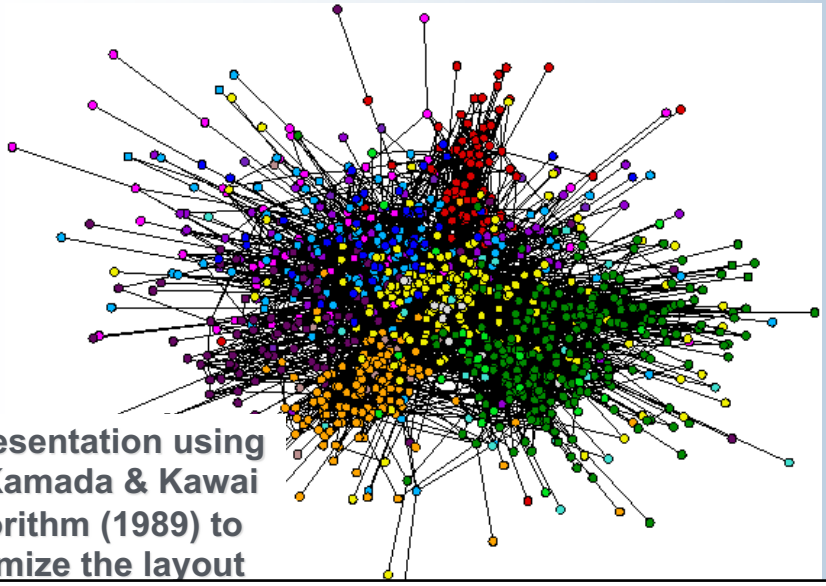
59

Data acquisition to construct the e-mail network of the URV

- Node => e-mail address
- Link => bidirectional e-mails between nodes (undirected graph)
- Number of users approx. 1700 (professors, technicians, administrators, graduate students)
- We consider only e-mails sent within the University during the first 3 months of 2002 (stable network)
- Non "spam" mail: (neglect >50 recipients)

60

Email at URV



Representation using the Kamada & Kawai algorithm (1989) to optimize the layout

61

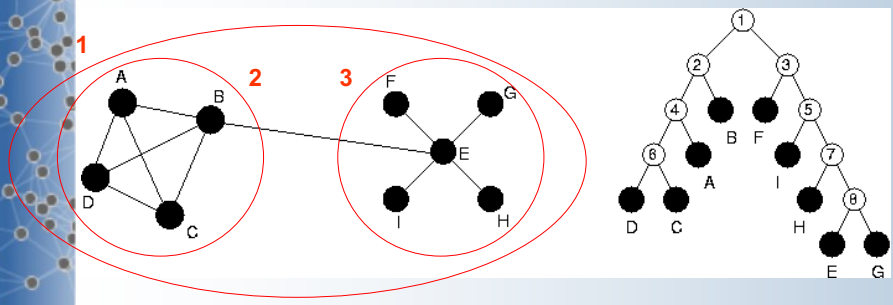
Community identification: the Girvan & Newman (GN) algorithm*

- Definition: Betweenness of a link = # minimum paths connecting pairs of nodes that go through that link
- Idea in GN algorithm: The links which connect highly clustered communities have a higher link betweenness. Then cut these links to separate communities.

*Girvan and Newman, PNAS USA 99, 7821-7826 (2002)

62

Communities



A network containing two clear communities linked by BE.

Since there is no more community structure, the rest of the nodes will be separated one by one generating a binary tree with two branches corresponding to the two communities.

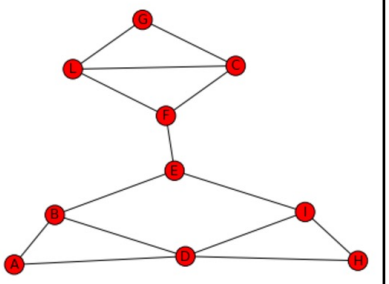
Leaders are at the tips of the branches

63

Guido's book GN

Code for the GN algorithm (networkx betweenness)

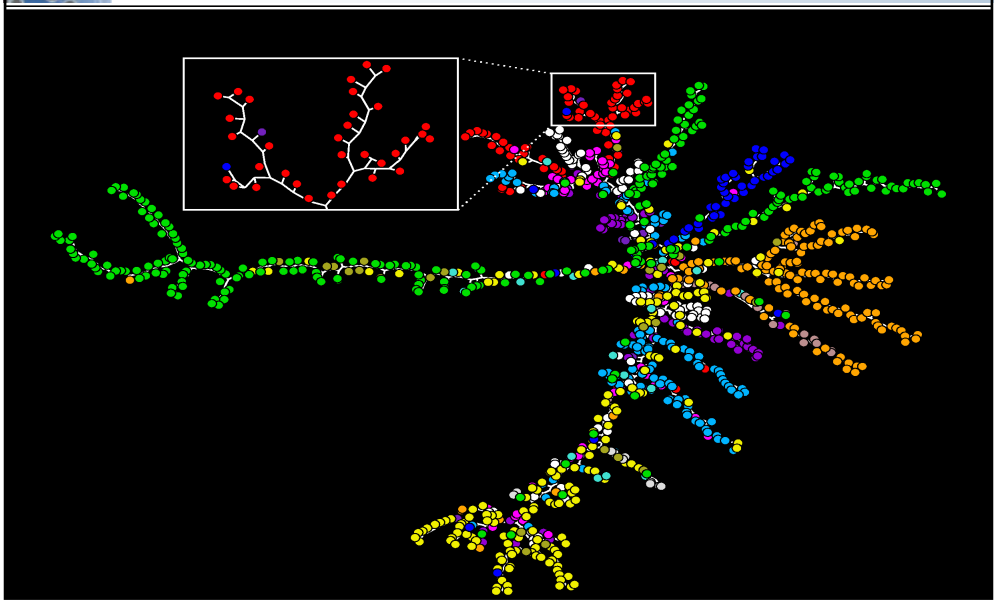
```
In [23]: G=nx.Graph()
G.add_edges_from([('A','B'),('A','D'),('B','D'),('B','E'),('E','I'),\
                  ('D','I'),('D','H'),('H','I'),('E','F'),('F','C'),\
                  ('F','L'),('C','L'),('C','G'),('G','L')])

pos=nx.graphviz_layout(G,prog='neato')

nx.draw(G, pos,with_labels=True)

#NOTE: THE ORDER OF EDGES IS DIFFERENT FOR THE FACT THAT MANY
#OF THEM HAVE THE SAME BETWEENNESS VALUE...

sorted_bc=[1]
actual_number_components=1
while not sorted_bc==[]:
    d_edge=nx.edge_betweenness_centrality(G)
    sorted_bc = sorted(d_edge.items(), key=operator.itemgetter(1))
    e=sorted_bc.pop()
    print "deleting edge:", e[0],
    G.remove_edge(*e[0])
    num_comp=nx.number_connected_components(G)
    print "...we have now ",num_comp," components"
    if num_comp>actual_number_components:
        actual_number_components=num_comp
```



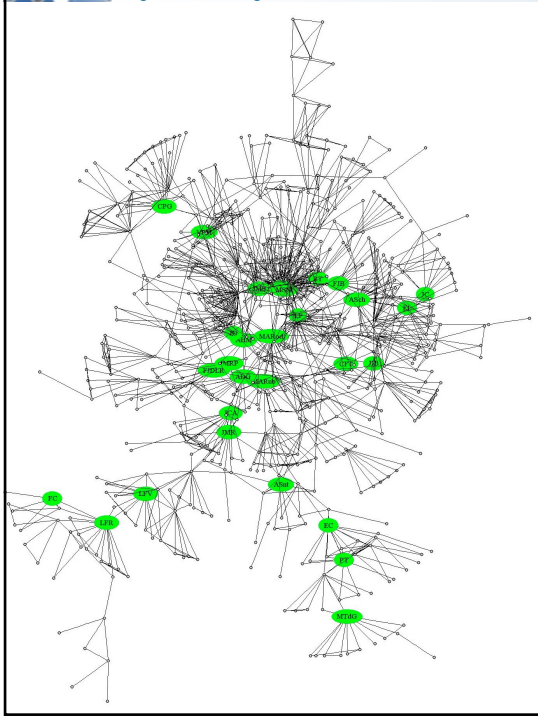

```
deleting edge: ('E', 'F') ...we have now  2  components
deleting edge: ('B', 'E') ...we have now  2  components
deleting edge: ('D', 'I') ...we have now  2  components
deleting edge: ('D', 'H') ...we have now  3  components
deleting edge: ('I', 'H') ...we have now  4  components
deleting edge: ('F', 'L') ...we have now  4  components
deleting edge: ('C', 'F') ...we have now  5  components
deleting edge: ('B', 'D') ...we have now  5  components
deleting edge: ('A', 'B') ...we have now  6  components
deleting edge: ('G', 'L') ...we have now  6  components
deleting edge: ('C', 'G') ...we have now  7  components
deleting edge: ('A', 'D') ...we have now  8  components
deleting edge: ('C', 'L') ...we have now  9  components
deleting edge: ('E', 'I') ...we have now  10  components
```

64

Communities in URV



65



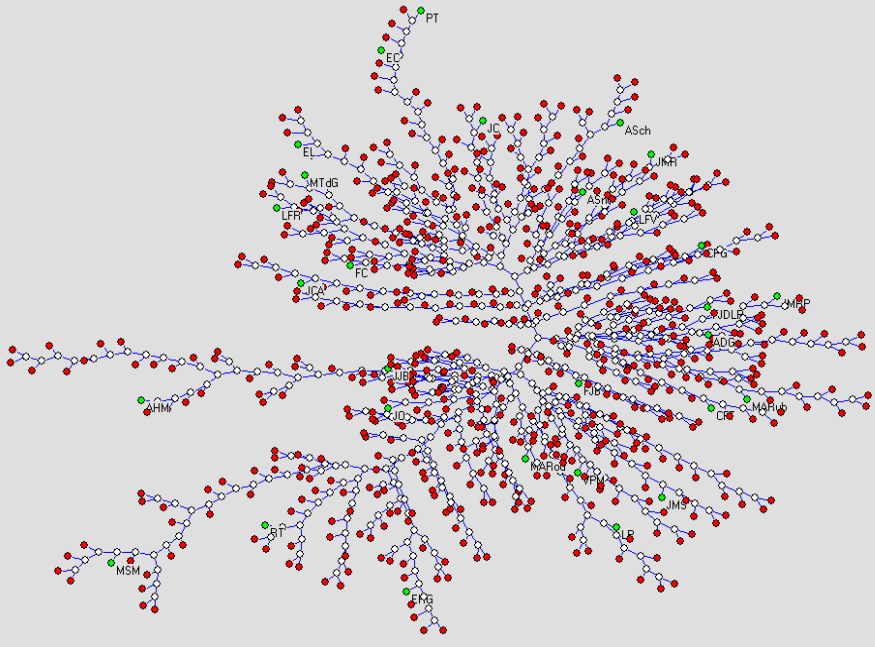
FisEs network

(principal component)

785 total
655 pc (84%)

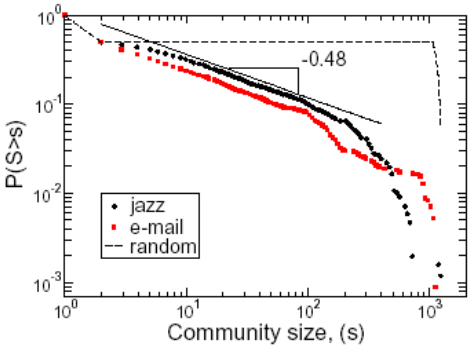
66

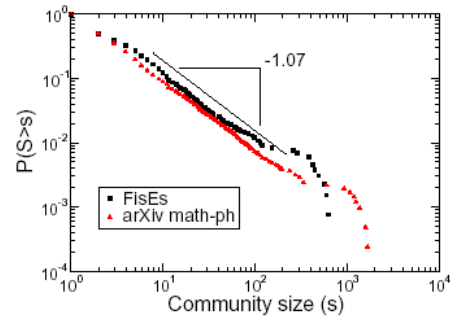
Community structure



67

Self-similarity





68

CNPhysics

- Project Complex Networks & Physics

69

Labelling communities

We identify the communities with the titles of the articles published in the community C. With these we follow the next steps to identify a community C:

1. A list of all the papers that published the members of C is created to make statistics.
2. The list is split in its words and he common English words (a, the, for, this...) are eliminated.
3. We find the frequency and the inverse document frequency (idf) of each word.
4. And finally the community is named with the top four words with the highest product of idf and frequency.

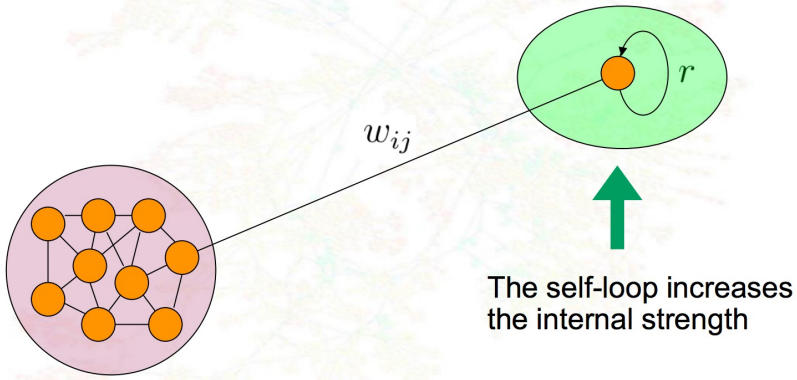
The idf is a measure of how a word w is relevant in a set of documents D:

$$\mathrm{idf(w)} = -\log\left(\frac{\mathrm{n}}{\mathrm{Card(D)}}\right) \quad (5)$$

70

Communities at all levels (self-loops)

- Tune the *resistance* of nodes to join communities, adding *self-loops*



71

- Multiple resolution method
  (Arenas, Fernandez & Gómez (2008) New J Phys **10**, 053039)
  - Add a common resistance (self-loop) to all nodes

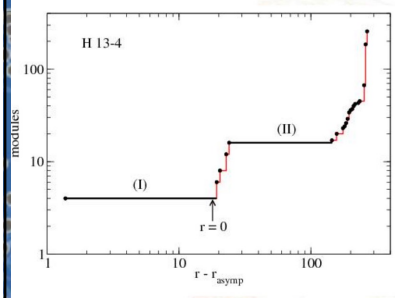
$$w'_{ij} = \begin{cases} w_{ij} & \text{if } i \neq j \\ r & \text{if } i = j \end{cases} \qquad \begin{aligned} w'_i &= w_i + r \\ 2w' &= 2w + Nr \end{aligned}$$

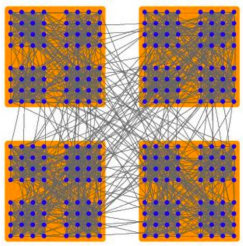
  - Optimize modularity

$$Q_r = \frac{1}{2w'} \sum_i \sum_j \left(w'_{ij} - \frac{w'_i w'_j}{2w'} \right) \delta(C_i, C_j)$$

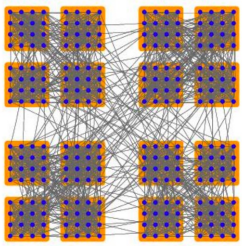
72

- Homogeneous with two hierarchical levels
  (Arenas, Diaz-Guilera & Perez-Vicente (2006) Phys Rev Lett **96**, 114102)





(I)

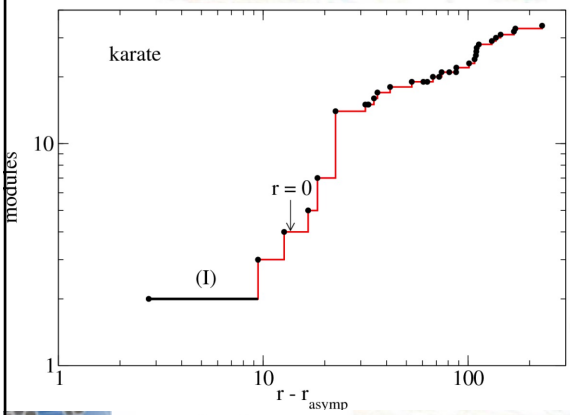


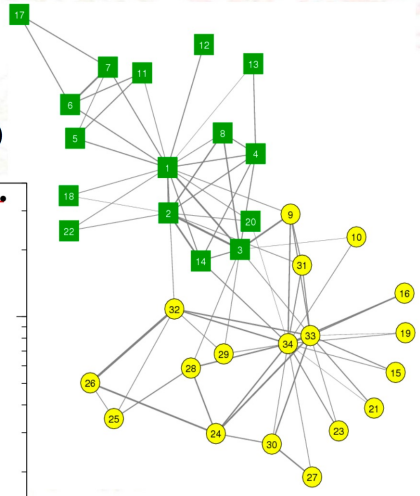
(II)

73

Social networks

- Zachary karate club
  (Zachary (1977) J Anthropol Res **33**, 452)

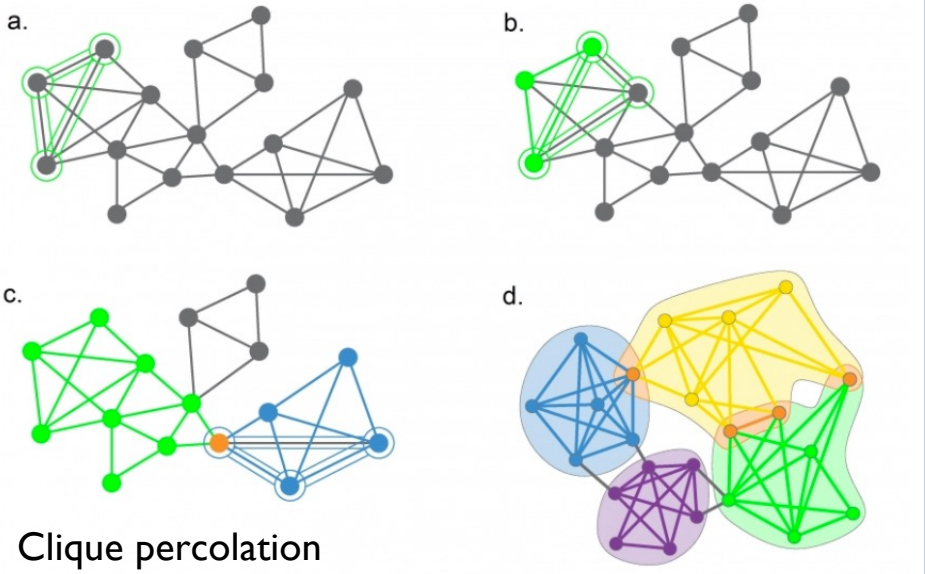




- Software: Radatools
  - http://deim.urv.cat/~sgomez/radatools.php

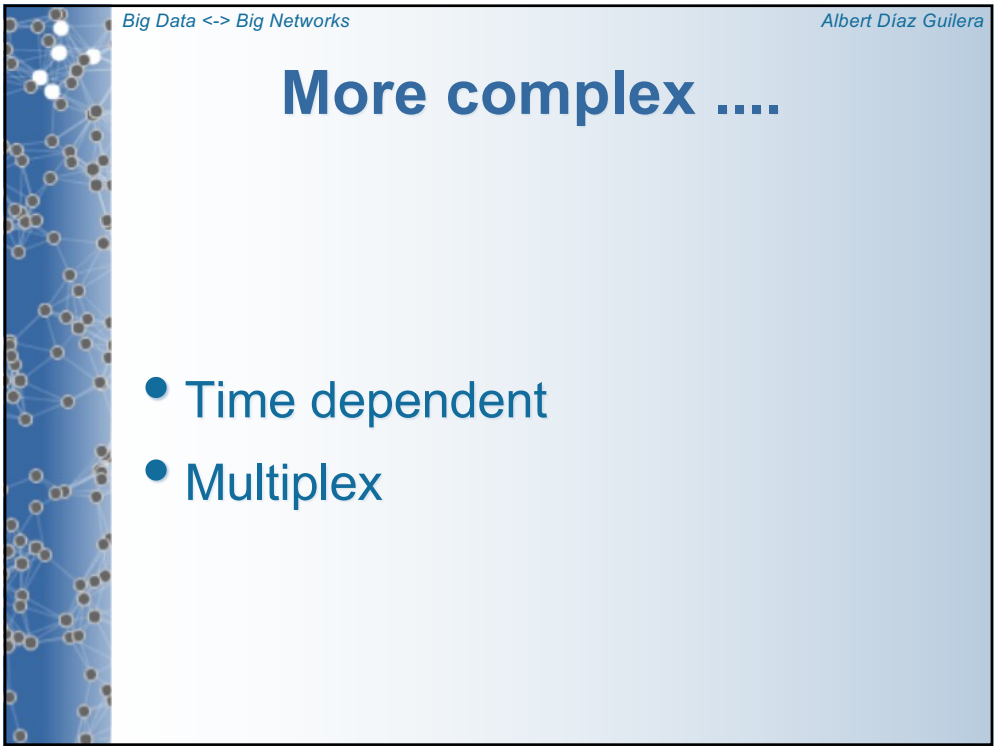
74

Overlapping communities



Clique percolation

75


Big Data <-> Big Networks
Albert Díaz Guilera
More complex ....
Time dependent
Multiplex

76