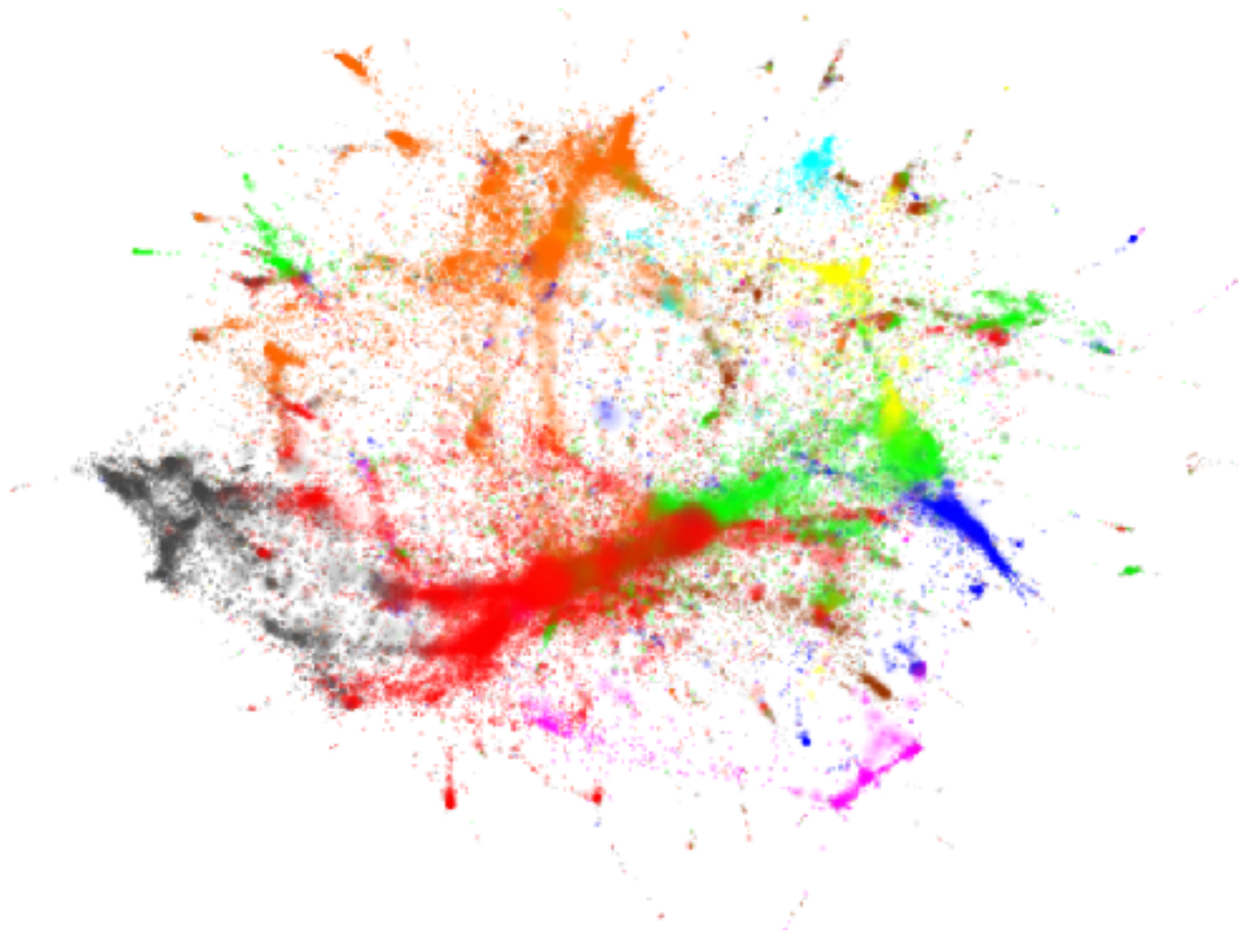
Complex Networks

Motifs and communities



Dr. Márton Karsai
ENS Lyon 2016

Practical matters

Course web page:

- perso.ens-lyon.fr/marton.karsai/Marton_Karsai/complexnet.html

Slides:

- http://perso.ens-lyon.fr/marton.karsai/protected/complexnets/
- login: complexnet
- psw: cnet123

Lectures (upcoming):

- 03/11/2016 - 10:15-12:15 - Amphi B
- 10/11/2016 - 10:15-12:15 - Amphi B
- 17/11/2016 - No lecture (but TD)
- 24/11/2016 - 10:15-12:15 - **B1**
- 01/12/2016 - 10:15-12:15 - Amphi B
- 08/12/2016 - No lecture (research schools)
- 15/12/2016 - 10:15-12:15 - Amphi B
- 05/01/2017 - 10:15-12:15 - Amphi B

Tutorials (upcoming):

- 17/11/2016 - 10:15-12:15 - Amphi B

Exam:

- 12/01/2017 - 10:15-12:15 - Amphi B (TBC)

Outline

1. Network types and characteristics
2. Fundamental network models
3. Motifs and communities
4. Temporal and dynamical networks
5. Planar and multiplex networks
6. Network sampling and statistical analysis
7. Application of networks

Todays schedule

1. Subgraphs and motifs
2. Communities - basics
3. Vertex similarity measures
4. Partitioning
5. Modularity
6. Hierarchical clustering
7. The Girvan-Newman method
8. The Louvain method
9. The Infomap method
10. The Clique percolation method
11. Benchmarks and testing algorithms

Literature

Physics Reports 486 (2010) 75–174

Contents lists available at ScienceDirect

Physics Reports

journal homepage: www.elsevier.com/locate/physrep

Community detection in graphs

Santo Fortunato *

Complex Networks and Systems Lagrange Laboratory, ISI Foundation, Viale S. Severo 65, 10133, Torino, I, Italy

ARTICLE INFO

Article history:
Accepted 5 November 2009
Available online 4 December 2009
editor: I. Procaccia

Keywords:
Graphs
Clusters
Statistical physics

ABSTRACT

The modern science of networks has brought significant advances to our understanding of complex systems. One of the most relevant features of graphs representing real systems is community structure, or clustering, i.e. the organization of vertices in clusters, with many edges joining vertices of the same cluster and comparatively few edges joining vertices of different clusters. Such clusters, or communities, can be considered as fairly independent compartments of a graph, playing a similar role like, e.g., the tissues or the organs in the human body. Detecting communities is of great importance in sociology, biology and computer science, disciplines where systems are often represented as graphs. This problem is very hard and not yet satisfactorily solved, despite the huge effort of a large interdisciplinary community of scientists working on it over the past few years. We will attempt a thorough exposition of the topic, from the definition of the main elements of the problem, to the presentation of most methods developed, with a special focus on techniques designed by statistical physicists, from the discussion of crucial issues like the significance of clustering and how methods should be tested and compared against each other, to the description of applications to real networks.

Contents

1. Introduction 76
2. Communities in real-world networks 78
3. Elements of community detection 82
   3.1. Computational complexity 83
   3.2. Communities 83
      3.2.1. Basics 83
      3.2.2. Local definitions 84

Science
AAAS

Network Motifs: Simple Building Blocks of Complex Networks
R. Milo *et al.*
Science **298**, 824 (2002);
DOI: 10.1126/science.298.5594.824

Community structure in social and biological networks

M. Girvan*†‡ and M. E. J. Newman*§

*Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, NM 87501; †Department of Physics, Cornell University, Clark Hall, Ithaca, NY 14853-2501; and §Department of Physics, University of Michigan, Ann Arbor, MI 48109-1120

Edited by Lawrence A. Shepp, Rutgers, State University of New Jersey–New Brunswick, Piscataway, NJ, and approved April 6, 2002 (received for review December 6, 2001)

PHYSICAL REVIEW E **69**, 026113 (2004)

Finding and evaluating community structure in networks

M. E. J. Newman[1,2] and M. Girvan[2,3]
[1]*Department of Physics and Center for the Study of Complex Systems, University of Michigan, Ann Arbor, Michigan 48109-1120, USA*
[2]*Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, New Mexico 87501, USA*
[3]*Department of Physics, Cornell University, Ithaca, New York 14853-2501, USA*
(Received 19 August 2003; published 26 February 2004)

We propose and study a set of algorithms for discovering community structure in networks—natural divi-

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

Fast unfolding of communities in large networks

Vincent D Blondel[1], Jean-Loup Guillaume[1,2], Renaud Lambiotte[1,3] and Etienne Lefebvre[1]

[1] Department of Mathematical Engineering, Université Catholique de Louvain,

PNAS

Maps of random walks on complex networks reveal community structure

Martin Rosvall*† and Carl T. Bergstrom*‡

*Department of Biology, University of Washington, Seattle, WA 98195-1800; and ‡Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, NM 87501

Edited by Brian Skyrms, University of California, Irvine, CA, and approved December 10, 2007 (received for review July 21, 2007)

To comprehend the multipartite organization of large-scale biological and social systems, we introduce an information theoretic approach that reveals community structure in weighted and di-

the links between modules capture the avenues of in flow between those modules.

Succinctly describing information flow is a coding

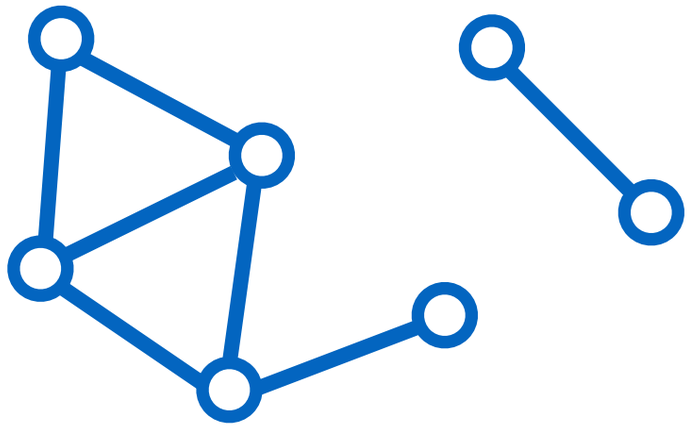
nature Vol 435|9 June 2005|doi:10.1038/nature03607

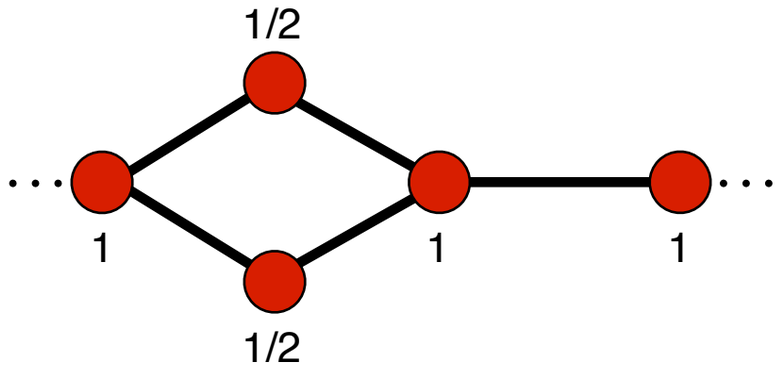
LETTERS

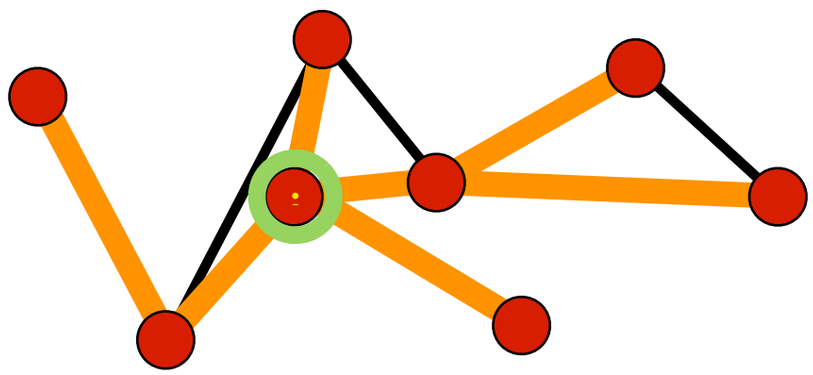
Uncovering the overlapping community structure of complex networks in nature and society

Gergely Palla[1,2], Imre Derényi[2], Illés Farkas[1] & Tamás Vicsek[1,2]

Complex networks - Microscopic view







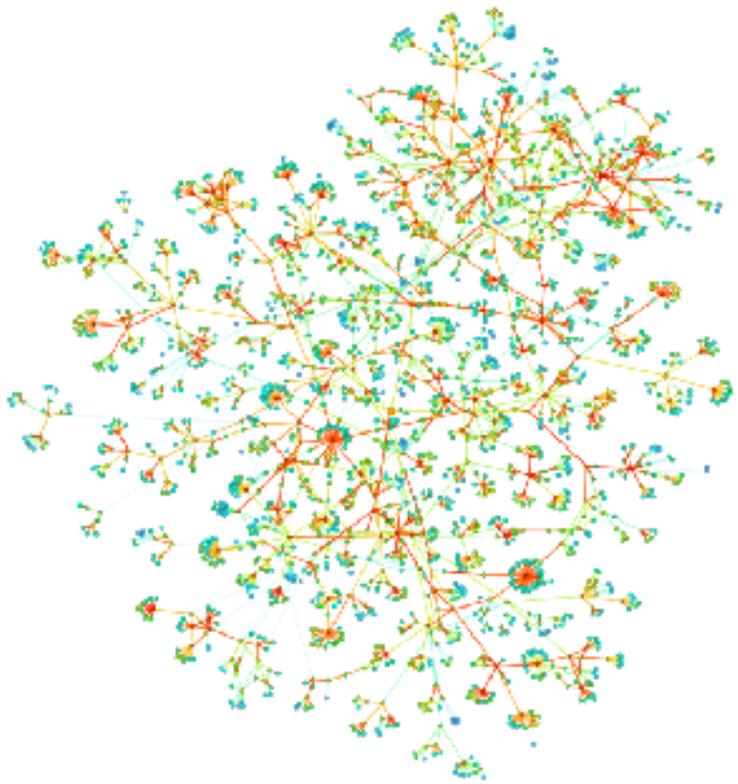
Node properties

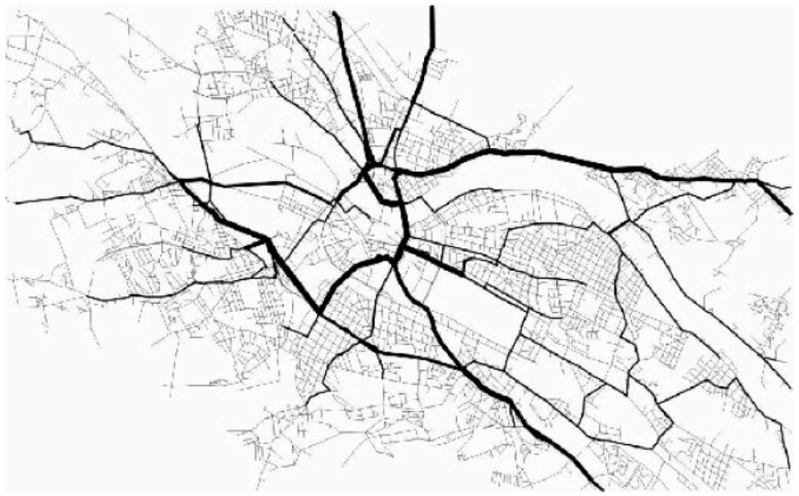
- Node degree - k_i
- Node strength - s_i
- Clustering coefficient - C_i
- Path length
- Closeness centrality
- Betweenness centrality
- Eigenvalue centrality

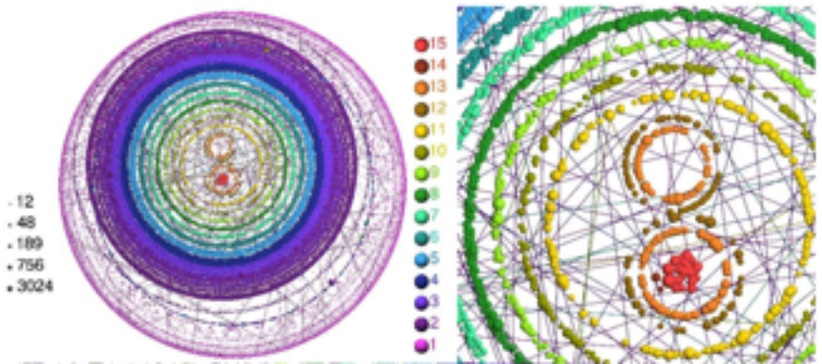
Link properties

- Edge weight - w_{ij}
- Edge centrality
- …

Complex networks - Macroscopic view







Statistical description

- Degree distribution - $P(k)$, $\langle k \rangle$
- Strength distribution - $P(s)$, $\langle s \rangle$
- Average clustering - C
- Average path length - $\langle l \rangle$
- Network density - ρ
- Degree correlations $P(klk')$
- k-core and k-shell decomposition

WHAT IS MISSING?

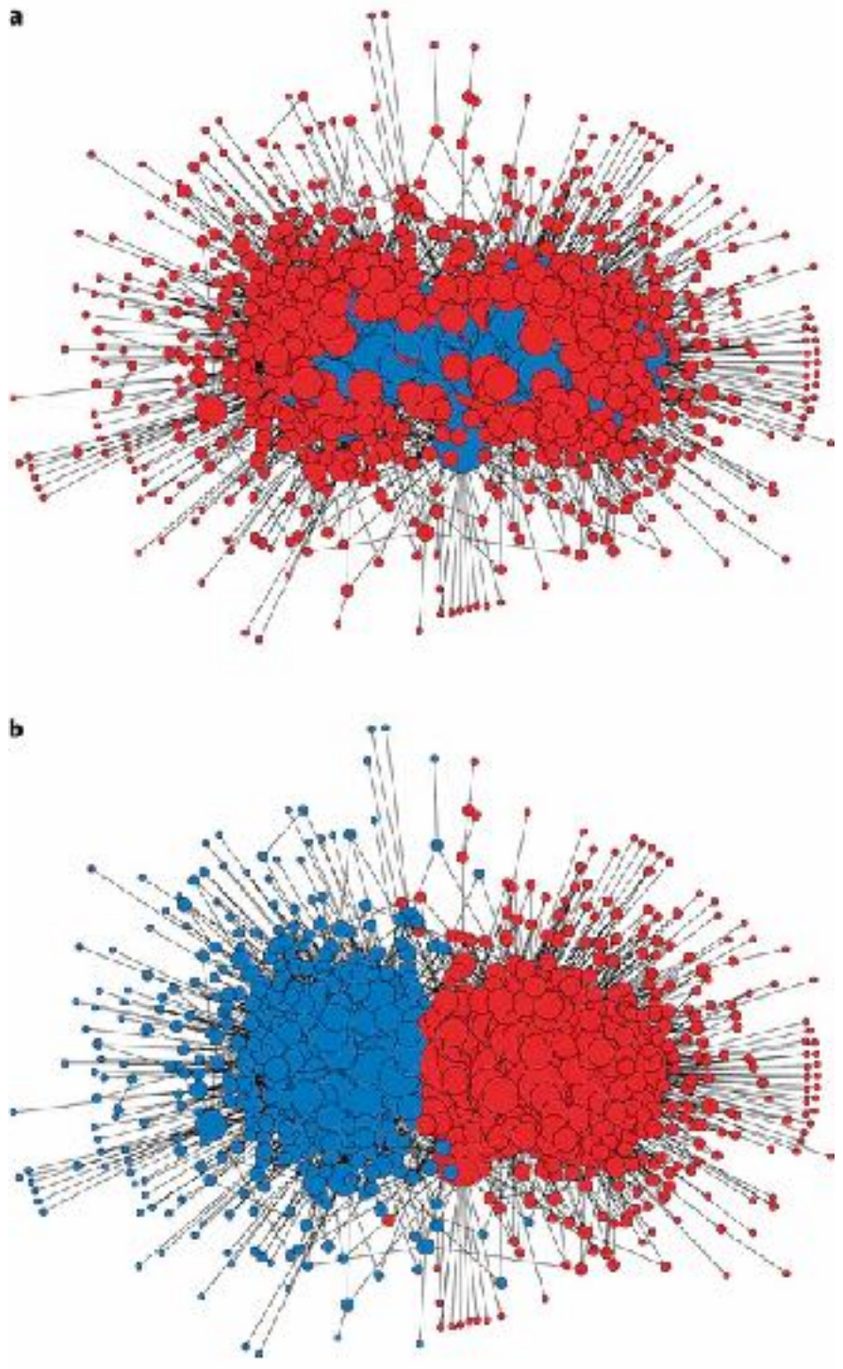
Complex networks - Mesoscopic view

Mesoscopic structures

- Motifs
- Partitions
- Modules
- Communities

Questions:

- Methods to find
- Measures to quantify
- Structure and frequency
- Applications
  - Visualization
  - Recommendation systems
  - Unknown functionality
  - …

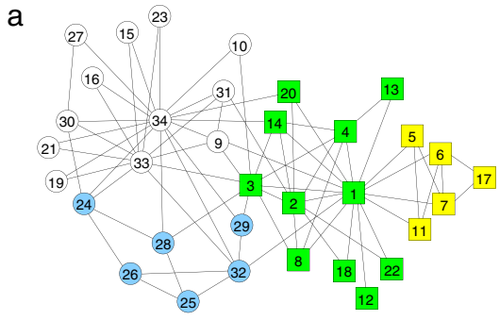


Newman, Nature Physics (2012)

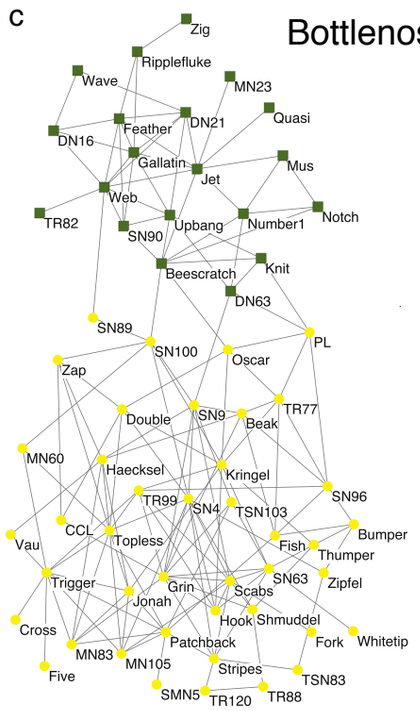
Communities in real world

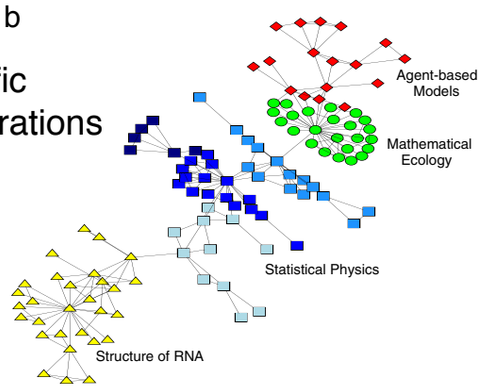
Communities in social networks

Zachary's karate club



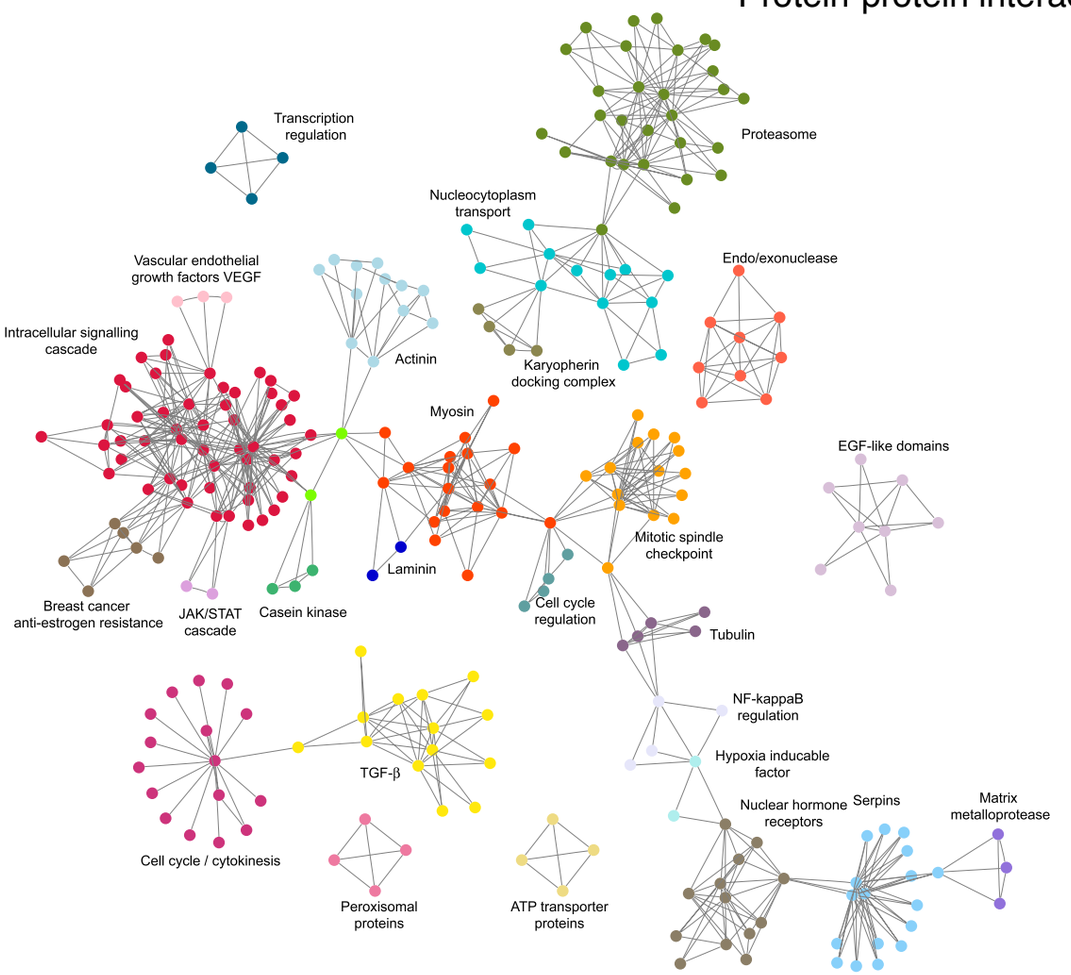
Bottlenose dolphins





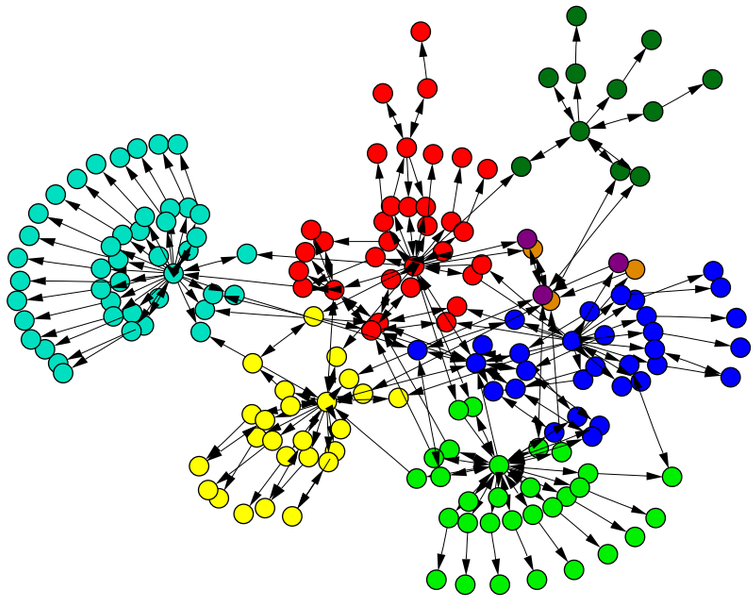
Communities in biological networks

Protein-protein interactions



Communities in information networks

www



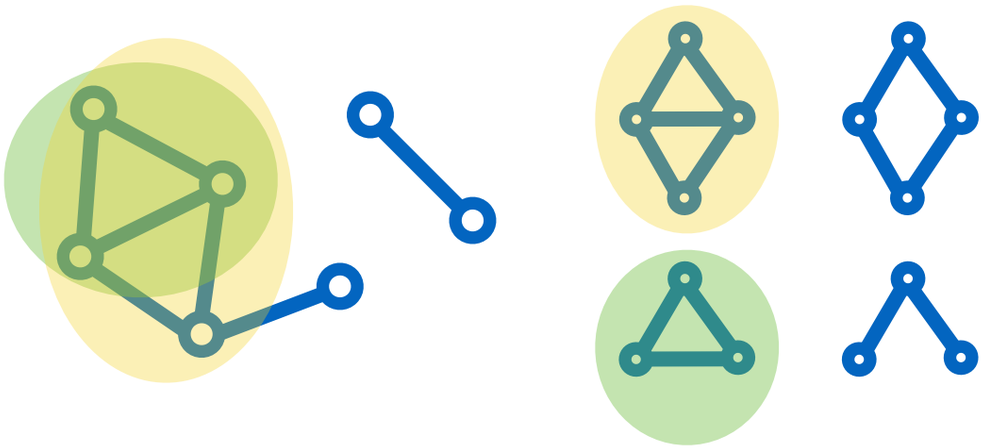
etc.

Fortunato (2010)

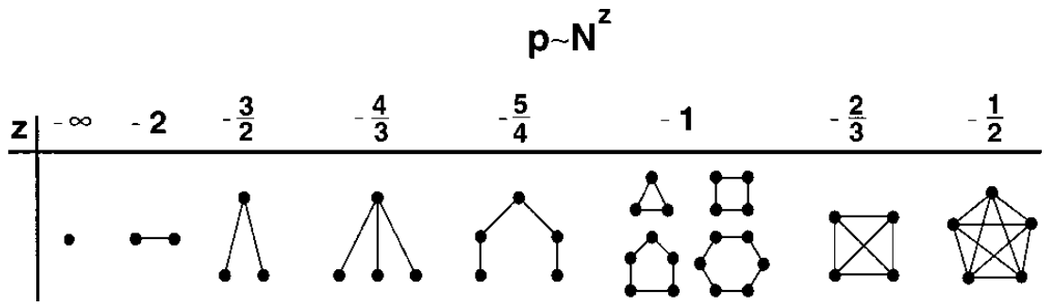
Subgraphs and motifs

Subgraphs

- $G'=(V',E')$ is a subgraphs of $G=(V,E)$ if $G' \subseteq G$ and $E' \subseteq E$



- In random graphs the appearance of different subgraphs is a function of the system size and the p probability

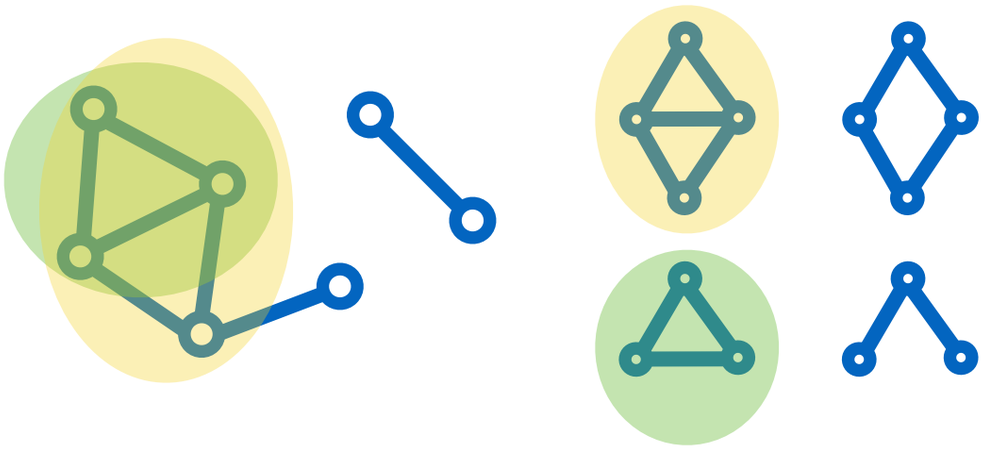


Albert et al., (2002)

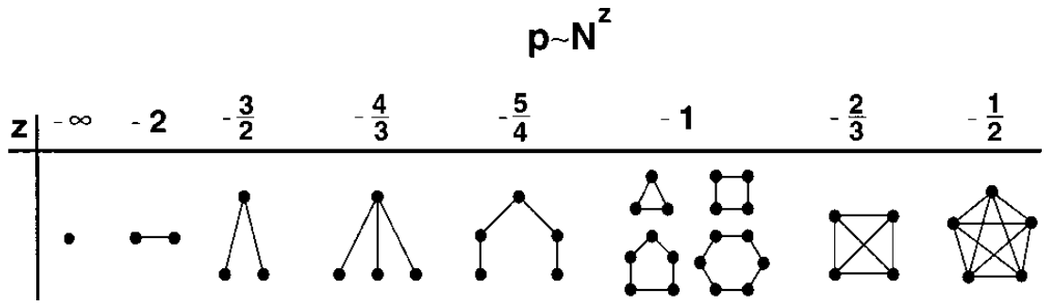
Subgraphs and motifs

Subgraphs

- $G'=(V',E')$ is a subgraphs of $G=(V,E)$ if $G' \subseteq G$ and $E' \subseteq E$



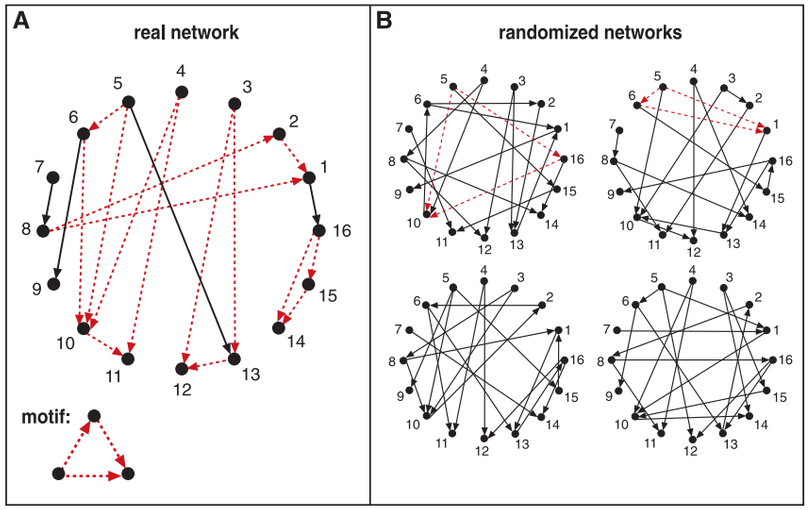
- In random graphs the appearance of different subgraphs is a function of the system size and the p probability



Albert et al., (2002)

Motifs

- Sub-graphs that appear with a significantly higher frequency in the real network than in the randomised version of the studied network



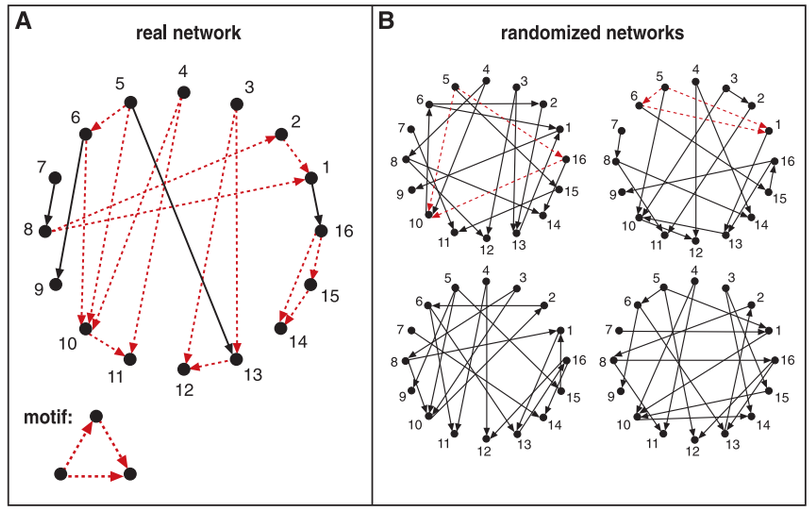
R. Milo et al., Science 298, 824 (2002)

- Randomised networks: Ensemble of maximally random networks preserving the degree distribution of the original network
- Algorithms: mfinder, FPF, ESU, …

Subgraphs and motifs

Z-score

1. Count subgraphs in the original network
2. Repeat:
   (i) rewire original network with configuration model
   (ii) count subgraphs in the rewired network
3. Compare original count to the reference ensemble as:



number of times the subgraph M occurs in the empirical network

average number of times the subgraph M occurs in the randomised reference network

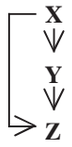
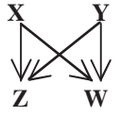
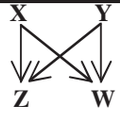
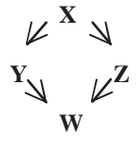
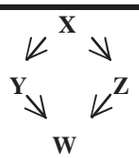
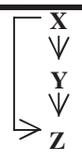
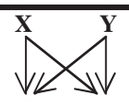
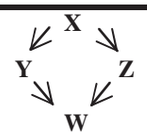
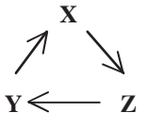
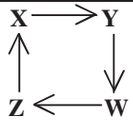
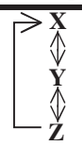
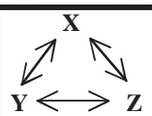
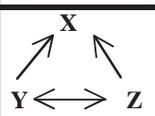
$$Z_M = \frac{n_M - \langle n_M^{rand} \rangle}{\sigma_{n_M}^{rand}}$$

standard deviation of n_M in the reference system

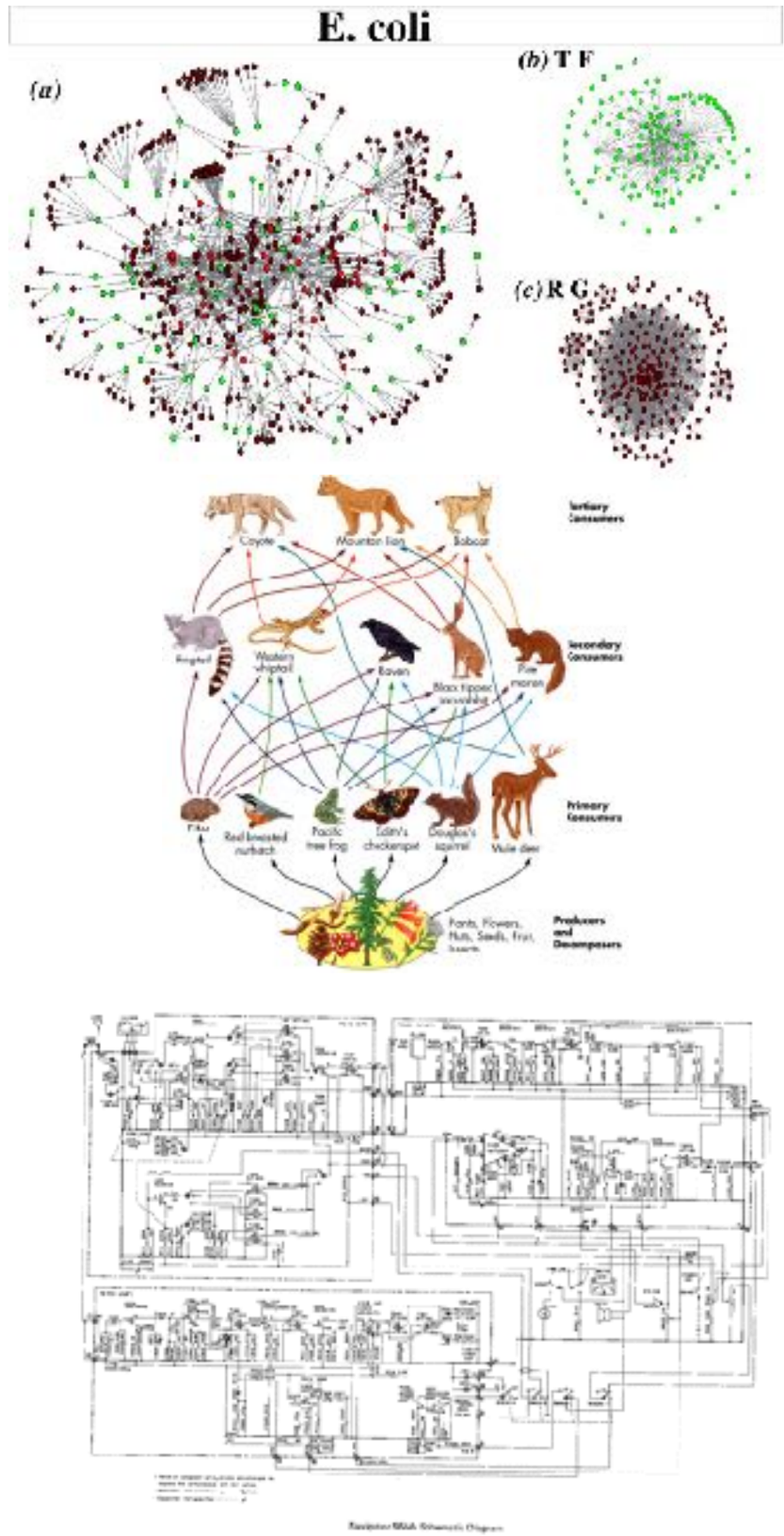
- Subgraph frequency in the configuration model depends on the system size
- To compare different networks Z-scores need to be normalised as:

$$\hat{Z}_i = \frac{z_i}{\sqrt{\sum_i z_i^2}}$$

Motifs in real networks

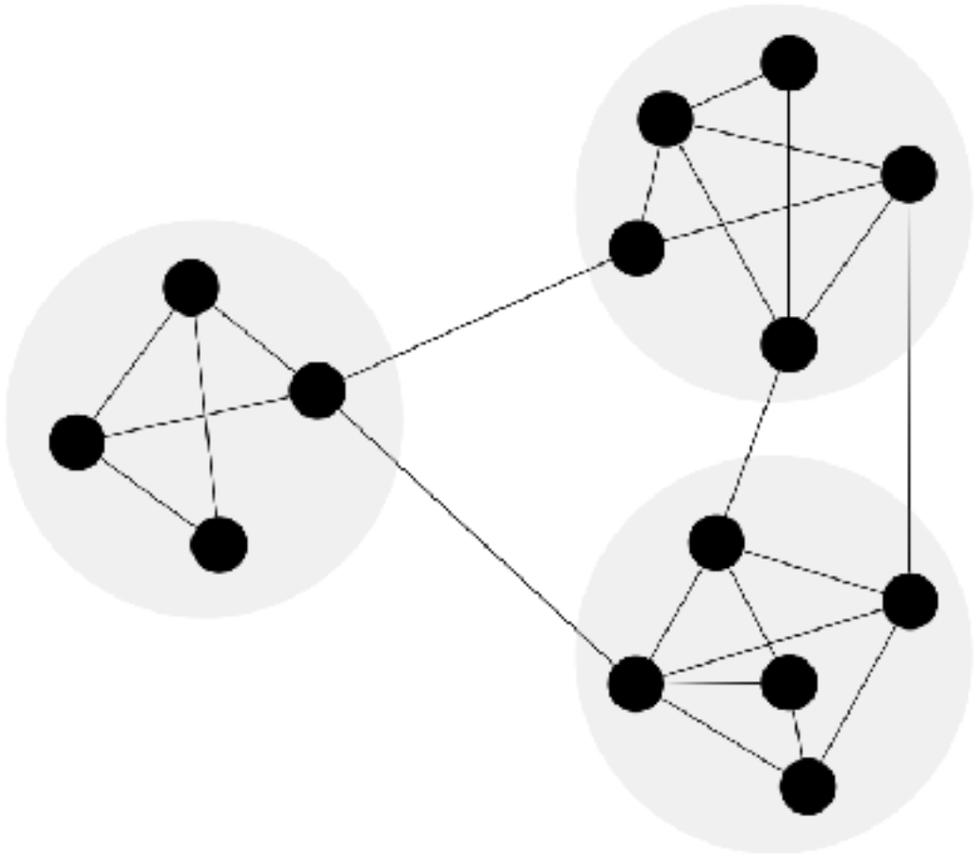
Network	Nodes	Edges	N_{real}	$N_{rand} \pm SD$	Z score	N_{real}	$N_{rand} \pm SD$	Z score	N_{real}	$N_{rand} \pm SD$	Z score
Gene regulation (transcription)					**Feed-forward loop**			**Bi-fan**			
E. coli	424	519	40	7 ± 3	10	203	47 ± 12	13			
*S. cerevisiae**	685	1,052	70	11 ± 4	14	1812	300 ± 40	41			
Neurons					**Feed-forward loop**			**Bi-fan**			**Bi-parallel**
C. elegans†	252	509	125	90 ± 10	3.7	127	55 ± 13	5.3	227	35 ± 10	20
Food webs					**Three chain**			**Bi-parallel**			
Little Rock	92	984	3219	3120 ± 50	2.1	7295	2220 ± 210	25			
Ythan	83	391	1182	1020 ± 20	7.2	1357	230 ± 50	23			
St. Martin	42	205	469	450 ± 10	NS	382	130 ± 20	12			
Chesapeake	31	67	80	82 ± 4	NS	26	5 ± 2	8			
Coachella	29	243	279	235 ± 12	3.6	181	80 ± 20	5			
Skipwith	25	189	184	150 ± 7	5.5	397	80 ± 25	13			
B. Brook	25	104	181	130 ± 7	7.4	267	30 ± 7	32			
Electronic circuits (forward logic chips)					**Feed-forward loop**			**Bi-fan**			**Bi-parallel**
s15850	10,383	14,240	424	2 ± 2	285	1040	1 ± 1	1200	480	2 ± 1	335
s38584	20,717	34,204	413	10 ± 3	120	1739	6 ± 2	800	711	9 ± 2	320
s38417	23,843	33,661	612	3 ± 2	400	2404	1 ± 1	2550	531	2 ± 2	340
s9234	5,844	8,197	211	2 ± 1	140	754	1 ± 1	1050	209	1 ± 1	200
s13207	8,651	11,831	403	2 ± 1	225	4445	1 ± 1	4950	264	2 ± 1	200
Electronic circuits (digital fractional multipliers)					**Three-node feedback loop**			**Bi-fan**			**Four-node feedback loop**
s208	122	189	10	1 ± 1	9	4	1 ± 1	3.8	5	1 ± 1	5
s420	252	399	20	1 ± 1	18	10	1 ± 1	10	11	1 ± 1	11
s838‡	512	819	40	1 ± 1	38	22	1 ± 1	20	23	1 ± 1	25
World Wide Web					**Feedback with two mutual dyads**			**Fully connected triad**			**Uplinked mutual dyad**
nd.edu§	325,729	1.46e6	1.1e5	2e3 ± 1e2	800	6.8e6	5e4±4e2	15,000	1.2e6	1e4 ± 2e2	5000

R. Milo et al., Science 298, 824 (2002)



MOTIFS ARE NOT COMMUNITIES!

Communities are subgraphs which are denser connected inside than to the rest of the network



Communities - basics

General conditions

- Community structure inferred only from structural informations, relations with actual groups is unclear
- The number of m edges of the graph is of the order of the n number of vertices

  otherwise the problem becomes similar to data clustering

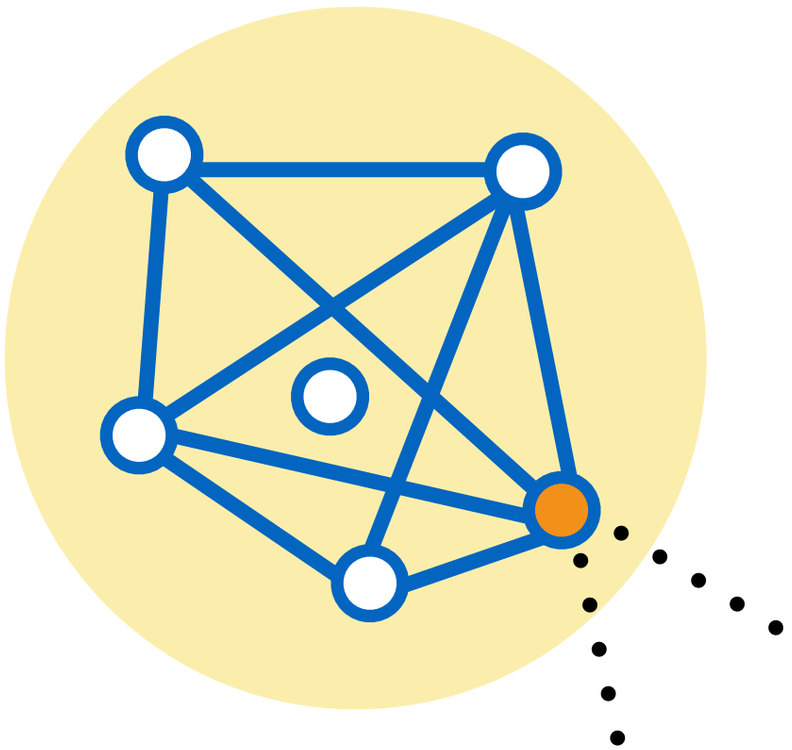
Important features

- Computational complexity: how a method performs on very large networks
- Possible network structure: bipartite, large, dense, temporal, …
- Possible community features: disjoint, overlapping, temporal, …
- Possible validation: other methods, benchmarks, meta informations

Limitations

- Communities are usually implicitly defined by the specific algorithm adopted, without an explicit definition!
- The practical definition may depend on the specific system/application

Communities - basics



k_v^{int} - internal degree

$k_{int}^C = \sum_{v \in C} k_v^{int}$ - internal degree of community C

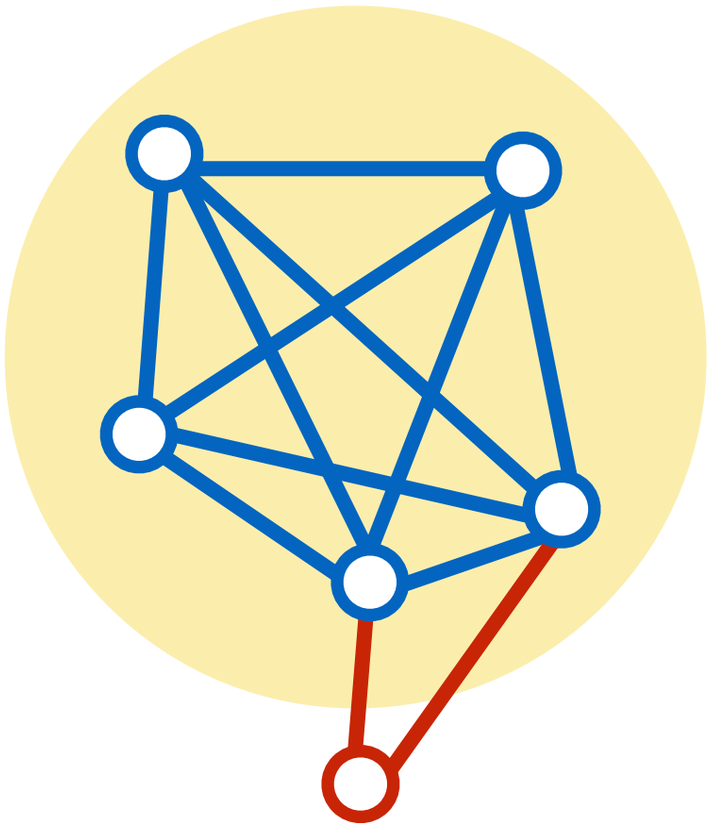
k_v^{ext} - external degree

$k_{ext}^C = \sum_{v \in C} k_v^{ext}$ - external degree of community C

$$\delta_{int}(C) = \frac{\#\text{internal edges of } C}{n_c(n_c - 1)/2}$$ - Intra-cluster edge density

$$\delta_{ext}(C) = \frac{\#\text{external edges of } C}{n_c(n - n_c)/2}$$ - Inter-cluster edge density

Communities - local definitions



Principle

- Concentrate only on specific subgraph and neglect the rest of the network

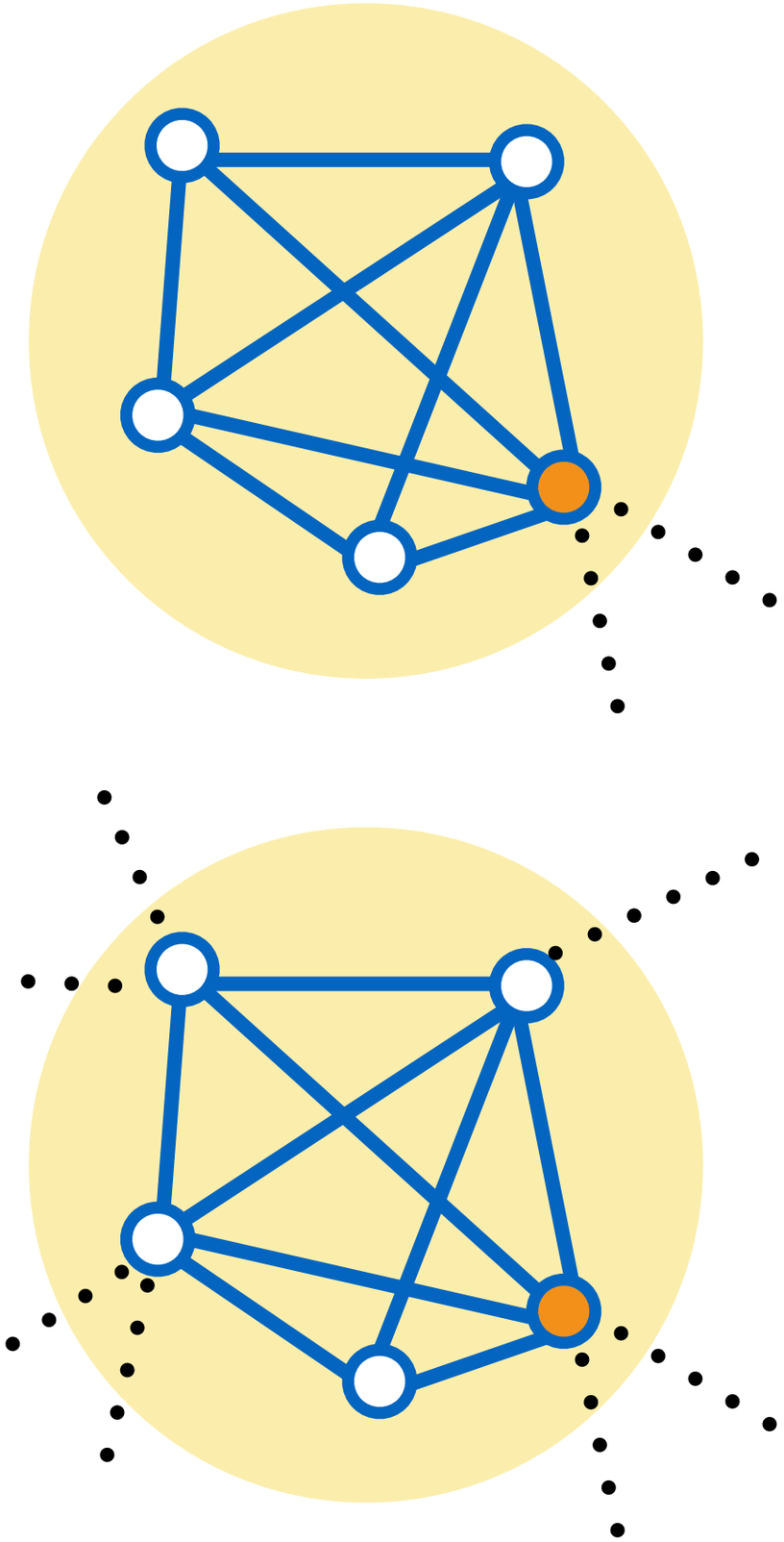
Examples

- Cliques - complete subgraphs
- n-cliques - subgraph such that the distance between each pair of vertices does not exceed n (Luce, 1950)

Problems

- Too strict condition
- All vertices are symmetric, while in reality communities usually have different roles
- Cliques are hard to find (NP complete)

Communities - local definitions



Principle

- Comparison between the internal and external cohesion of a subgraph

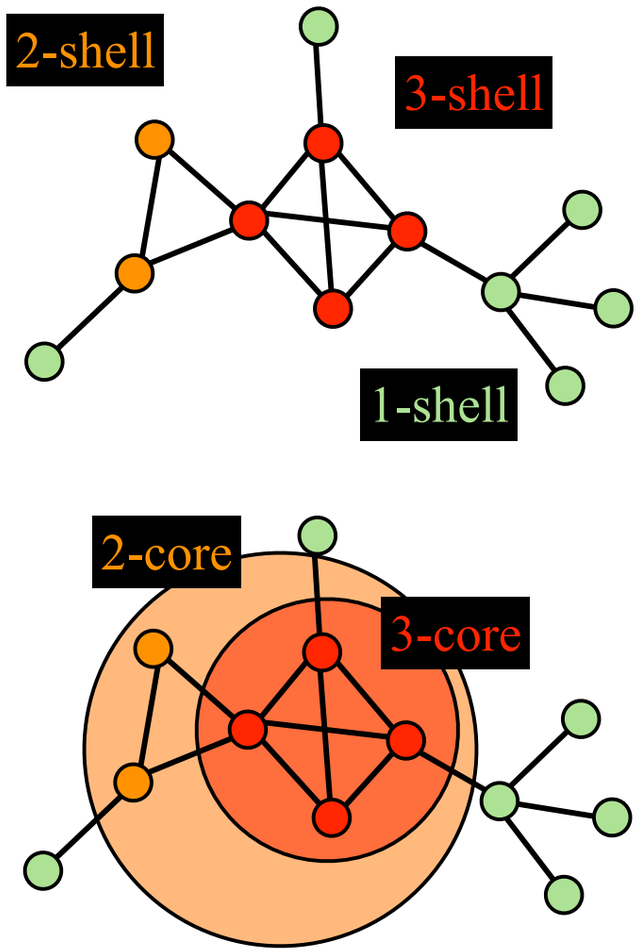
Examples

- LS-set or strong community: subgraph such that the internal degree of each vertex is greater than its external degree (Luccio & Sami, 1969)
- Weak communities: subgraph such that the internal degree of the subgraph is greater than its external degree (Radicchi et al., 2004)

Problems

- Too strict condition
- Unrealistic in practical cases

Communities - local definitions



Principle

- Cohesion through vertex adjacency

Examples

- k-plex: maximal subgraph such that each vertex is adjacent to all other vertices of the subgraph except at most k of them (Seidman & Foster, 1978)
- k-core: maximal subgraph such that each vertex is adjacent to at least k other vertices of the subgraph (Seidman, 1983)
- p-quasi complete subgraph: maximal subgraph such that the degree of each vertex is larger than p(k-1), where p∈[0,1] and k is the order of the subgraph (Matsuda et al., 1999)]

Communities - vertex similarity

Principle

- Communities are subgraphs of vertices, which are “similar” to each other
- Measures of similarity are needed!

Measures for graphs embedded in space

- There is a distance measure defined between nodes to quantify their similarity or dissimilarity
- They run on pairs of vertices $A=(a_1,a_2,\ldots,a_n)$, $B=(b_1,b_2,\ldots,b_n)$

Euclidean distance

$$d_{AB}^{E} = \sum_{k=1}^{n} \sqrt{(a_k - b_k)^2}$$

Manhattan distance

$$d_{AB}^{M} = \sum_{k=1}^{n} |a_k - b_k|$$

etc.

Communities - vertex similarity

Measures for graphs not embedded in space

- Only information: Adjacency matrix

Structural equivalence dissimilarity

$$d_{ij} = \sqrt{\sum_{k \neq i,j} (A_{ik} - A_{jk})^2}$$

Neighbourhoods overlap

$$\omega_{ij} = \frac{|\Gamma(i) \cap \Gamma j|}{|\Gamma(i) \cup \Gamma j|}$$

Pearson correlation coefficient

(between columns and rows of the adjacency matrix)

$$C_{ij} = \frac{\sum_k (A_{ik} - \mu_i)(A_{jk} - \mu_j)}{n \sigma_i \sigma_j}$$

Measures based on paths: Number of edge- (or vertex-) independent paths, Total number of paths

Measures based on random walks: Commute-time, Average first passage time, Escape probability

etc.

Partitioning

Graph partition: division of a graph as a union of non-overlapping and non-empty subgraphs

- Number of possible partitions of a graph with n vertices into k clusters is given by the S(n,k) Stirling number of the second kind

$$S(n,k) = \frac{1}{k!}\sum_{j=0}^{k}(-1)^{k-j}\binom{k}{j}j^n$$

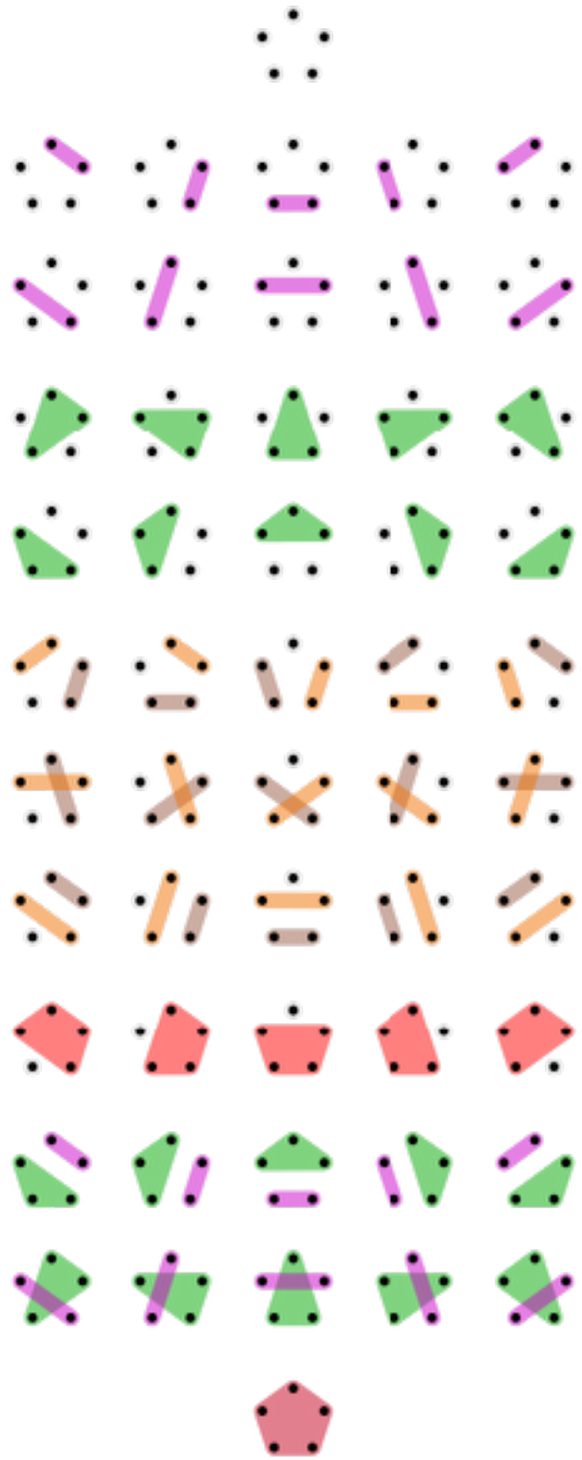
- Total number of possible partitions: Bell-number

$$B_n = \sum_{k=0}^{n} S(n,k)$$

- In the large n limit:

$$B_n \sim \frac{1}{\sqrt{n}}[e^{W(n)}]^{n+1/2}e^{e^{W(n)}-n-1}, \quad W(n)\text{: Lambert function}$$

- It is a double exponential ➡ very large number



The 52 partitions of a set with 5 elements

Partitioning - quality function

Quality function: assigns a score to each partition of a graph

- It is a measure proportional to the goodness of the partition

Examples:

Performance: number of “correctly” interpreted vertices

Number of links which connects nodes in the same cluster

Number of node pairs which are not placed in the same cluster and not connected by an edge

$$P(\mathcal{P}) = \frac{|\{(i,j) \in E, C_i = C_j\}| + |\{(i,j) \notin E, C_i \neq C_j\}|}{n(n-1)/2}$$

By definition: $0 \leq P(\mathcal{P}) \leq 1$

Coverage: ratio between the number of intra-community edges and the total number of edges

$$C(P) = \frac{|(i,j) \in E, C_i = C_j|}{m}$$

Modularity

Newman & Girvan, 2004

Principle: Random graphs have no community structure

Method: comparing the edge density in each cluster with the edge density of the cluster in a randomised version of the graph

- It is the fraction of the edges that fall within the given groups minus the expected such fraction if edges were distributed at random

Null model:

- Principally it is arbitrary
- ER (Bernoulli) random graph
- Random graph preserving the original node degree sequence (generated by a Configuration Model process)

Modularity

Take a network $G=(V,E)$ with n nodes, m links, and A_{ij} adjacency matrix

Assume C communities:

- assume $i \in V$ is in community C_i and $j \in V$ is in community C_j
- membership: $\delta(C_i, C_j) = 1$ if C_i=C_j, and 0 otherwise

Modularity: fraction of edges fall within communities, minus the expected fraction of such edges in a reference model

Expected fraction of edges in a reference model (using a configuration network model):

- It keeps the degrees of nodes unchanged
- It cuts each link in two stubs and rewire links randomly

Total number of stubs: $l_b = \sum_i k_i = 2m$

Expected number of connected edges between i and j: $\frac{k_i k_j}{l_b} = \frac{k_i k_j}{2m}$

Modularity

Adjacency matrix of the original graph

Expected number of edges between nodes with degree k_i and k_j in the configuration model network

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(C_i, C_j)$$

Total number of stubs, number of possible rewiring of a link ($n \to \infty$)

δ function: 1 if both nodes are in the same module $C_i = C_j$, 0 otherwise

Modularity

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(C_i, C_j)$$

Adjacency matrix of the original graph

Expected number of edges between nodes with degree k_i and k_j in the configuration model network

Total number of stubs, number of possible rewiring of a link ($n \to \infty$)

δ function: 1 if both nodes are in the same module $C_i = C_j$, 0 otherwise

$$Q = \sum_{c=1}^{n_c} \left(\frac{l_c}{m} - \left(\frac{d_c}{2m} \right)^2 \right)$$

Number of modules

l_c: number of edges inside module c

d_c: total degree of module c

$\frac{d}{(2m)} \frac{d}{(2m)}$: expected number of links in module c

Probability that a link is in module c

Modularity

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(C_i, C_j)$$

Features:

- Q < 1
- Q = 0 for a partition where the whole graph is a single cluster
- -1/2 ≤ Q: negative for multipartite structures
- Depends on the network size - resolution limit problem
  - small communities in large graphs are merged
  - expected number of edges between two groups of nodes decreases if the network size increases
  - Modularity may overestimate importance of intracommunity links and assign them inside communities Fortunato Barthalémy PNAS (2006)
- High modularity value does not necessarily assign good partitioning - random graphs can have high partition as well…

Partitioning

Divide the graph in n parts, such that the number of links between them (cut size) is minimal

Problems:

- Number of partitions must be specified in advance
- Size of clusters must be specified in advance

Traditional methods:

- Graph bi-sectioning
- Kerninghan-Lin algrorithm
- Spectral partitioning
- Partitional clustering
- K-means clustering
- …

One would like methods that can predict the number and the size of the partition and indicate a subset of “good” partitions

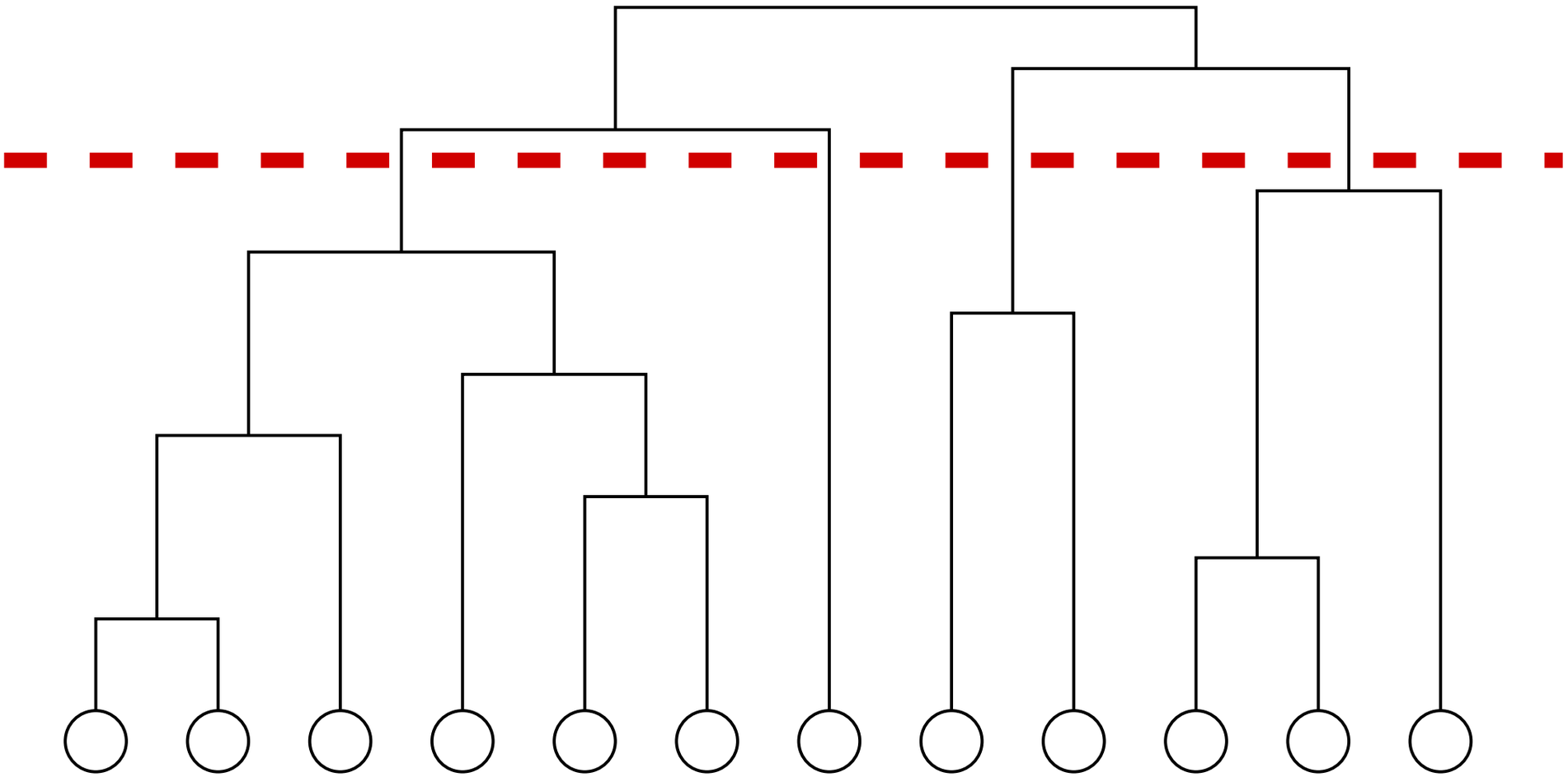
Hierarchical clustering

- Very common in social network analysis
- Two methods: agglomerative (bottom-up approach), divisive (top-down approach)

General algorithm:

1. A criterion is introduced to compare nodes based on their similarity
2. A similarity matrix X_{ij} is constructed: the similarity of nodes i and j is X_{ij}
3. (Agglomerative) Starting from the individual nodes, larger groups are built by joining groups of nodes based on their similarity
4. (Divisive) Starting from the graph as a single cluster, separate the most dissimilar parts, etc.

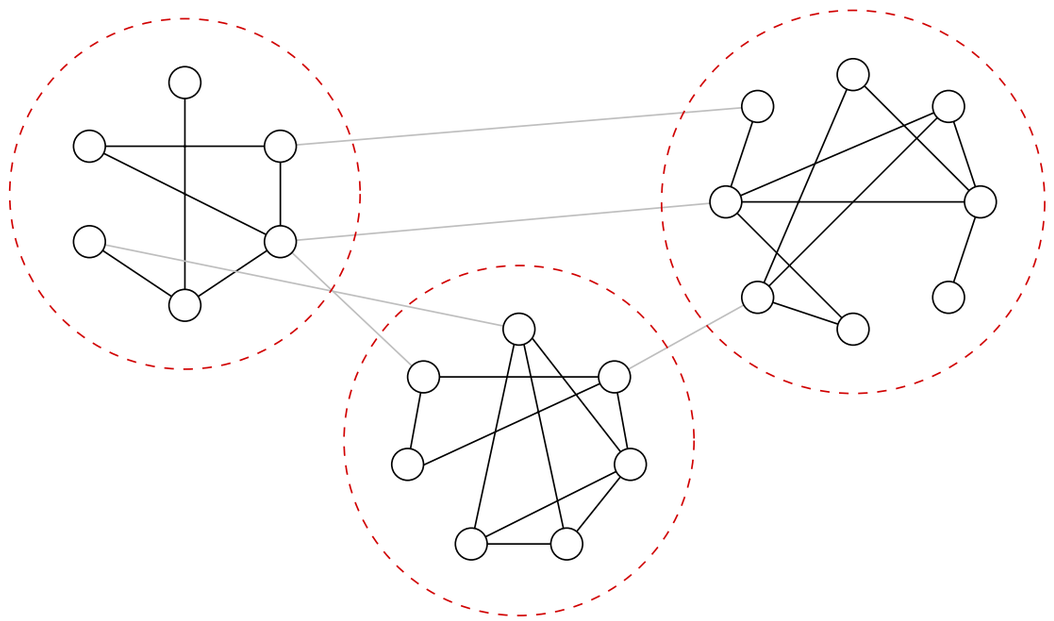
Hierarchical clustering - dendogram



Newman (2003)

Girvan-Newman method

- Divisive method: one removes the links that connect the clusters, until the clusters are isolated
- To identify inter-community links it uses edge betweenness measure
- If there are more geodesic paths between the same pair of vertices crossing the edge one divides the contribution of each path by their multiplicity
- Computable with algorithms based on breadth-first-search algorithm, with complexity $O(mn)$ (Brandes, 2001)



Newman (2003)

Girvan-Newman method

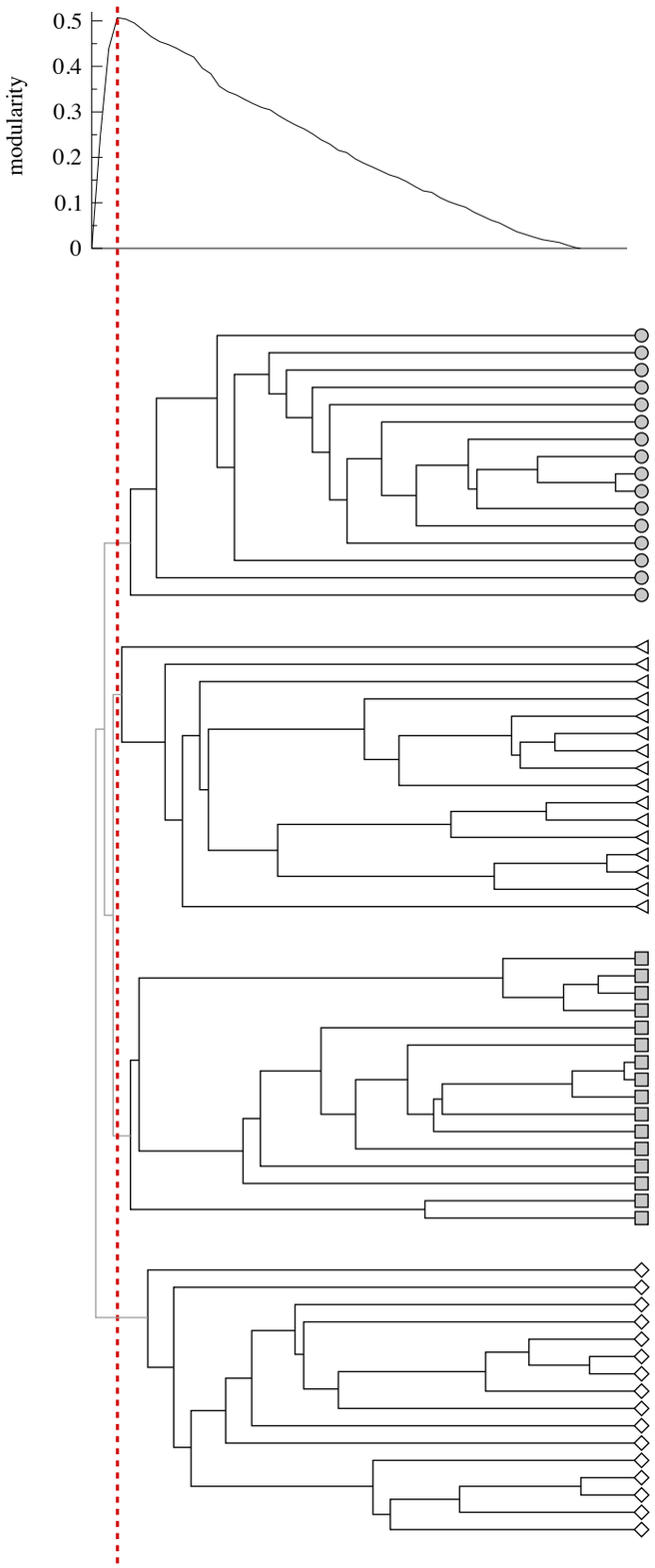
Algorithm

1. Calculate the betweenness of all edge
2. Remove the one with the highest betweenness
3. Recalculate the betweenness of the remaining edges
4. Repeat from step 2

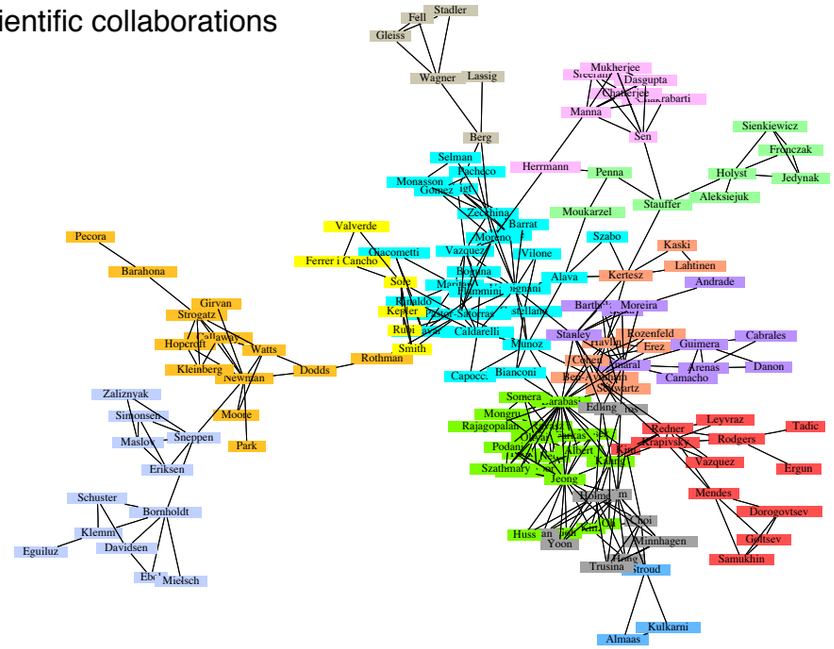
Features

- Complexity: $O(m^2n)$ (or $O(n^3)$ on sparse graphs and may be lowered by calculating step 3 only for a sample of node pairs)
- It delivers a hierarchy of partitions! Which one is the best?
- Girvan&Newman (2004): the best partition is the one corresponding to the highest modularity

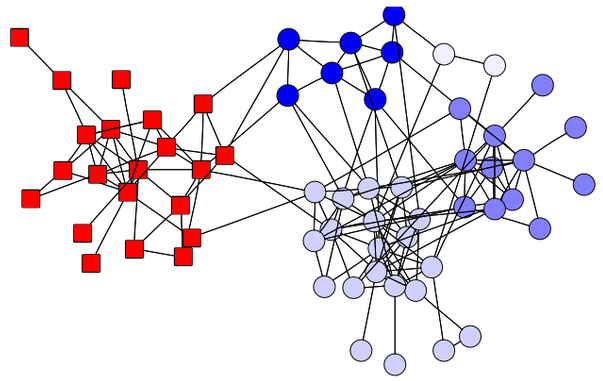
Girvan-Newman method



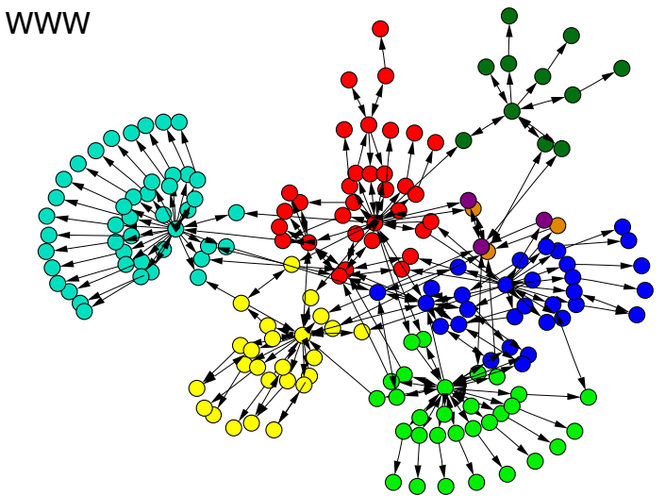
Scientific collaborations



Dolphins



WWW



Newman 2003

Modularity optimisation

$$Q = \frac{1}{m} \sum_{c=1}^{n_c} \left(l_c - \frac{d_c^2}{4m} \right)$$

Goal: find the maximum of Q over all possible network partitions

- Problem: NP-complete (Brandes et.al., 2007)

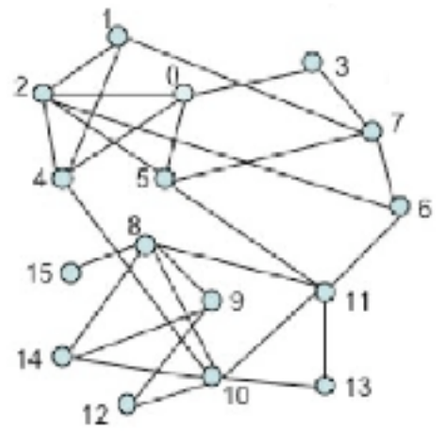
Strategies:

- Greedy algorithms
- Simulated annealing
- Extremal optimisation
- Spectral optimisation
- …

The Louvain method - a greedy algorithm

Algorithm (agglomerative method)

1. Take a (un)weighted network and put each node in a separated community



2. Repeat until Q is not maximal (Modularity optimisation)

   - For each node i consider each first neighbour j
     - Replace i to the C community of that neighbour j which maximally increases the modularity Q

Sum of link weights inside C

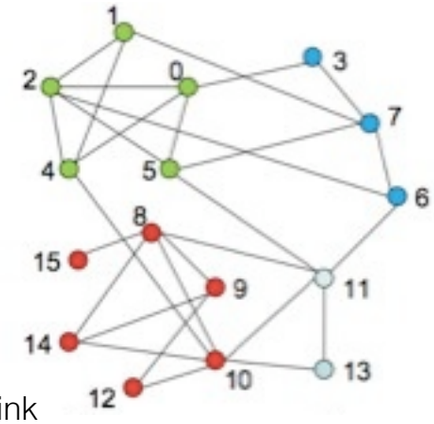
Sum of link weights connected node i to nodes in C

Sum of link weights connected to the nodes in C

Sum of link weights connected node i

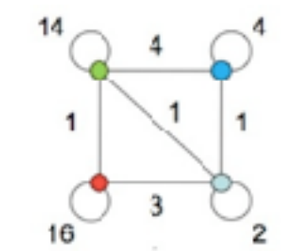
$$\Delta Q = \left[\frac{\sum_{\text{in}} + 2k_{i,\text{in}}}{2m} - \left(\frac{\sum_{\text{tot}} + k_i}{2m}\right)^2\right] - \left[\frac{\sum_{\text{in}}}{2m} - \left(\frac{\sum_{\text{tot}}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right]$$

Sum of all link weights in the network



3. For each node in the actual network (Community aggregation)

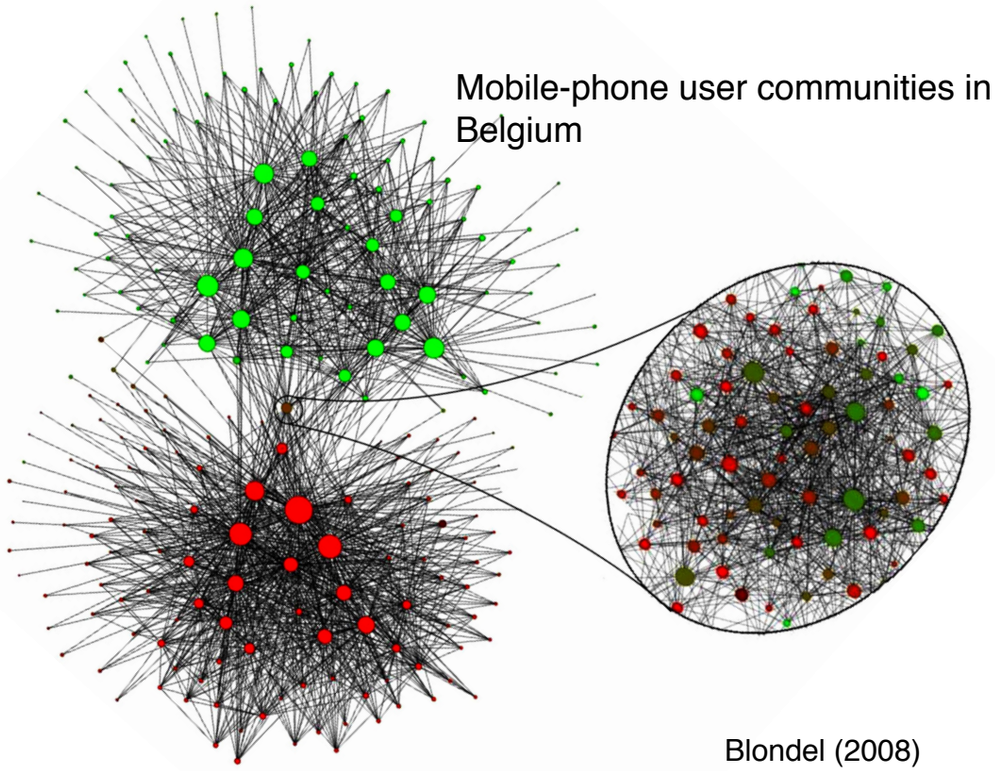
   - Merge nodes of step 2 into super nodes
   - Calculate the links and weights between super nodes as the sum of link weights connecting the merged nodes in the two super nodes



The Louvain method - a greedy algorithm

Features

- Greedy: always chooses the local maxima of modularity
- Non-deterministic: The order of nodes are taken is arbitrary
  - It does not affect the maximal modularity considerably
  - It effects the running time of the algorithm (better heuristics can be found)
- Change of modularity can be calculated locally ➜ fast performance
- Assigns hierarchical structure of the network
- Performs well on large networks



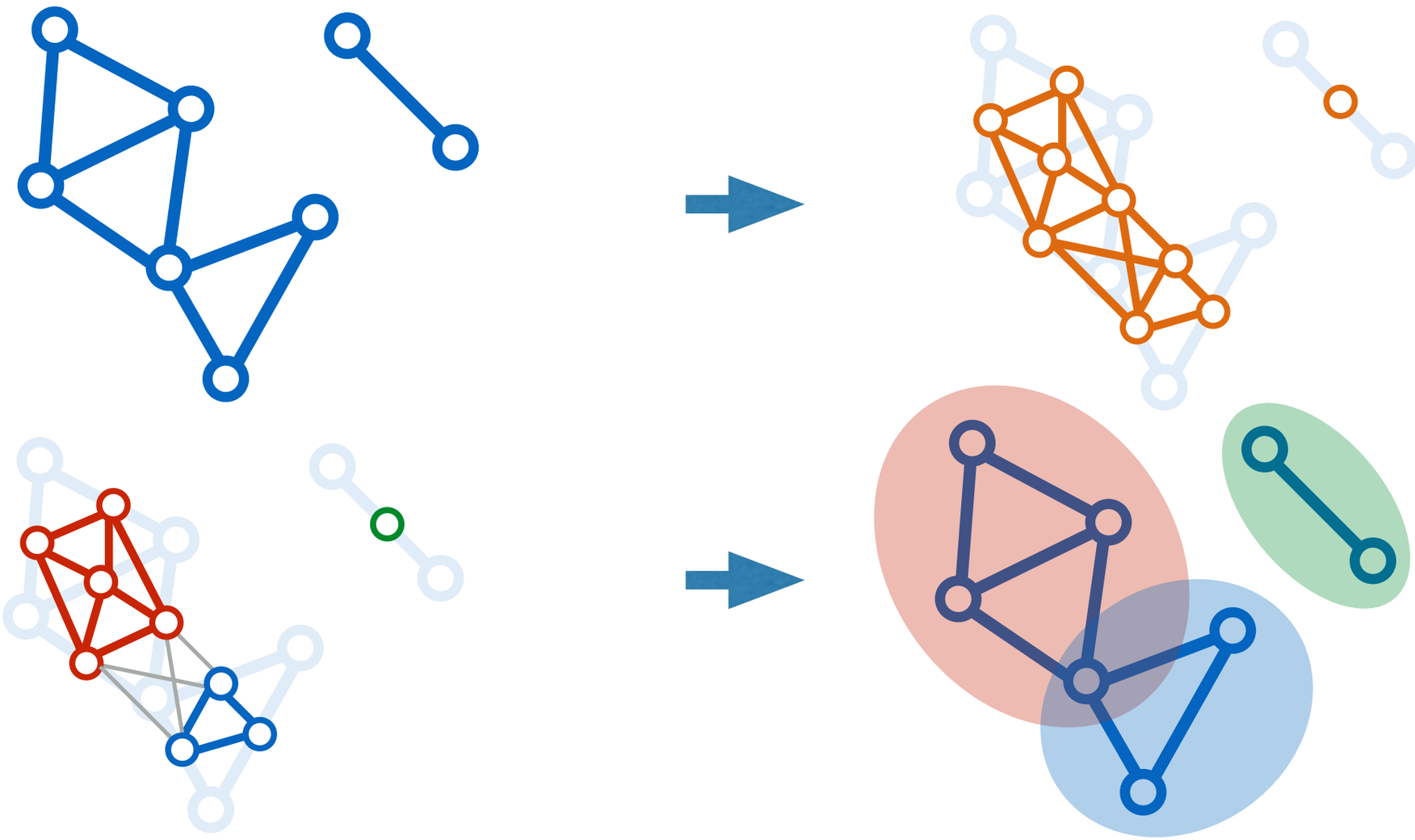
Mobile-phone user communities in Belgium

Blondel (2008)

Link clustering - overlapping communities

Link graphs

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



- Community detection on link graphs allows for overlapping communities

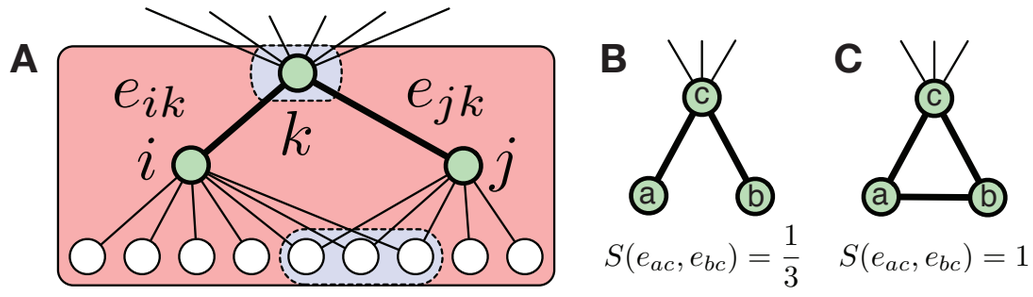
Link clustering - overlapping communities

Link communities Ahn et. al. (2010)

- Hierarchical clustering method
- Similarity measure: Jaccard index for links

$$S(e_{ik}, e_{jk}) = \frac{|n_+(i) \cap n_+(j)|}{|n_+(i) \cup n_+(j)|}$$

Set of node i and its first neighbours



- Quality function: partition density

$$D = \frac{2}{M} \sum_c m_c \frac{m_c - (n_c - 1)}{(n_c - 2)(n_c - 1)}$$

Number of links in cluster C

Number of nodes in cluster C

Total number of links

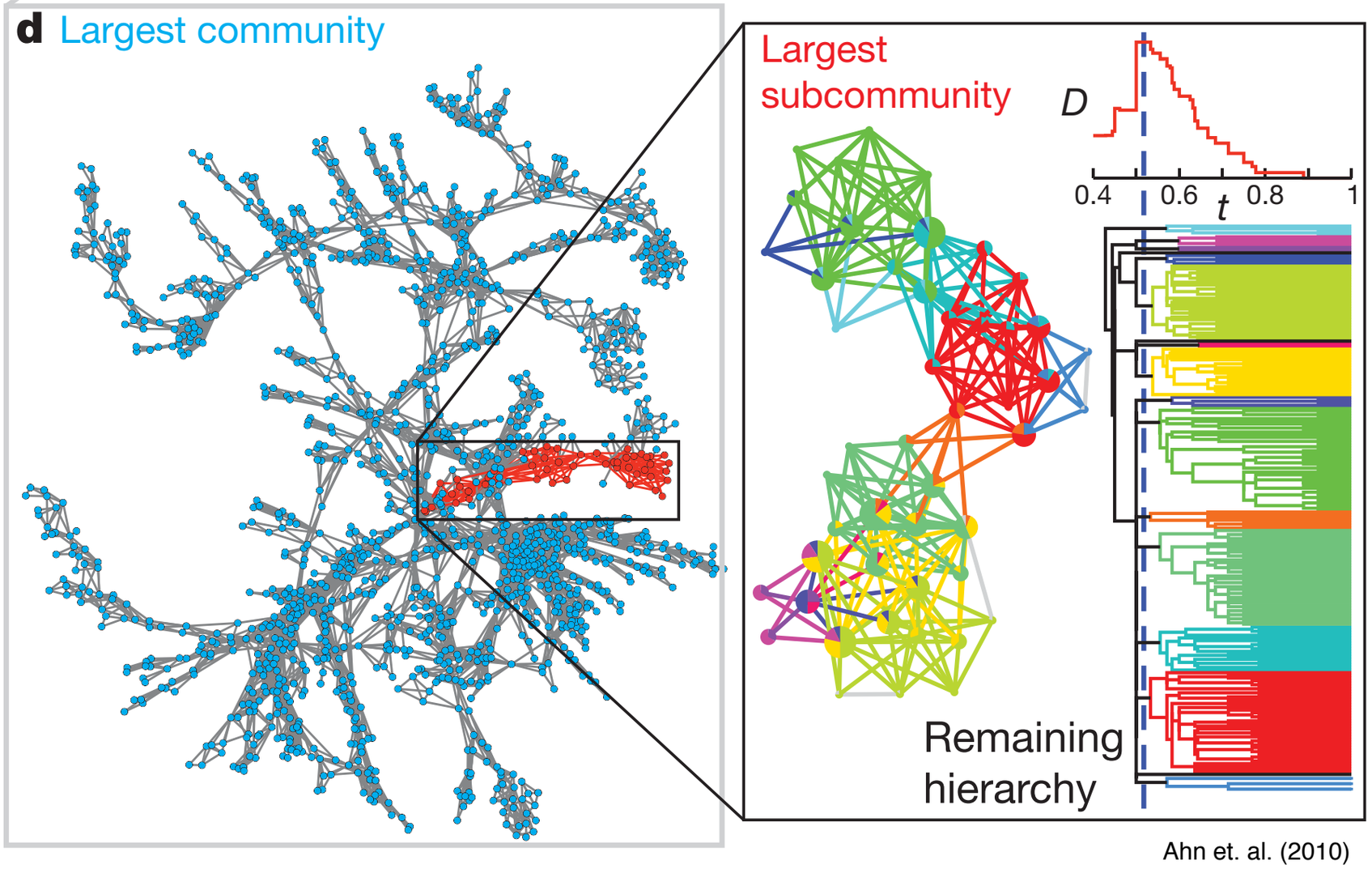
- Average link density weighted by the fraction of present links in a partition C

Link communities

Algorithm

- Calculate $S(e_{ik},e_{jk})$ for each link pairs connected in the line graph
- Merge all edge pairs in descending order of S for $S \geq t$
- Calculate D for the each t threshold

Take the t threshold and the corresponding link communities when D is maximal



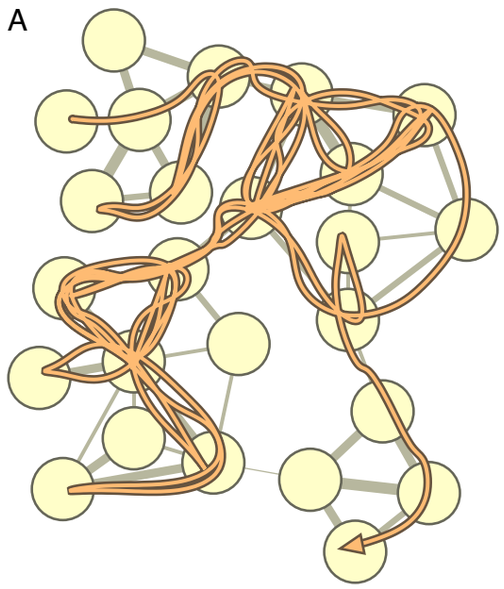
Ahn et. al. (2010)

The Infomap method

Rosvall&Bergstrom (2008)

Finding a compressed description of a random walk taking place on a graph

Rosvall et. al. (2008)



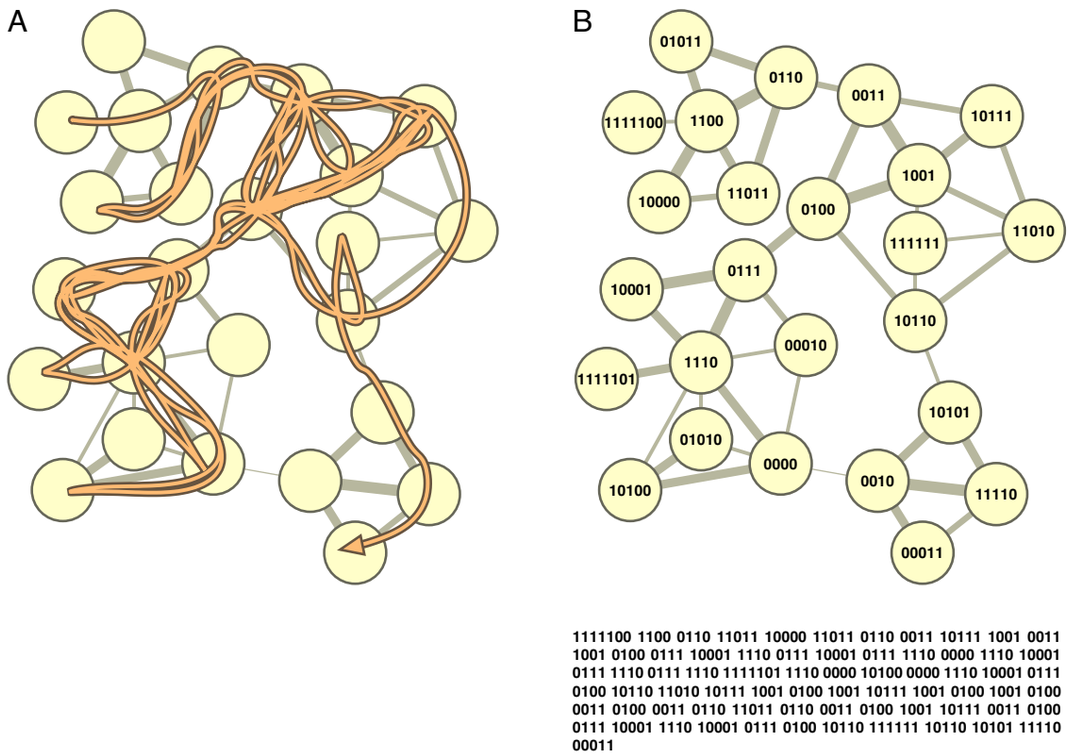
- Take a random walker on a (weighted, directed) graph

The Infomap method

Rosvall&Bergstrom (2008)

Finding a compressed description of a random walk taking place on a graph

Rosvall et. al. (2008)



1111100 1100 0110 11011 10000 11011 0110 0011 10111 1001 0011 1001 0100 0111 10001 1110 0111 10001 0111 1110 0000 1110 10001 0111 1110 0111 1110 1111101 1110 0000 10100 0000 1110 10001 0111 0100 10110 11010 10111 1001 0100 1001 10111 1001 0100 1001 0100 0011 0100 0011 0110 11011 0110 0011 0100 1001 10111 0011 0100 0111 10001 1110 10001 0111 0100 10110 111111 10110 10101 11110 00011

- Take a random walker on a (weighted, directed) graph

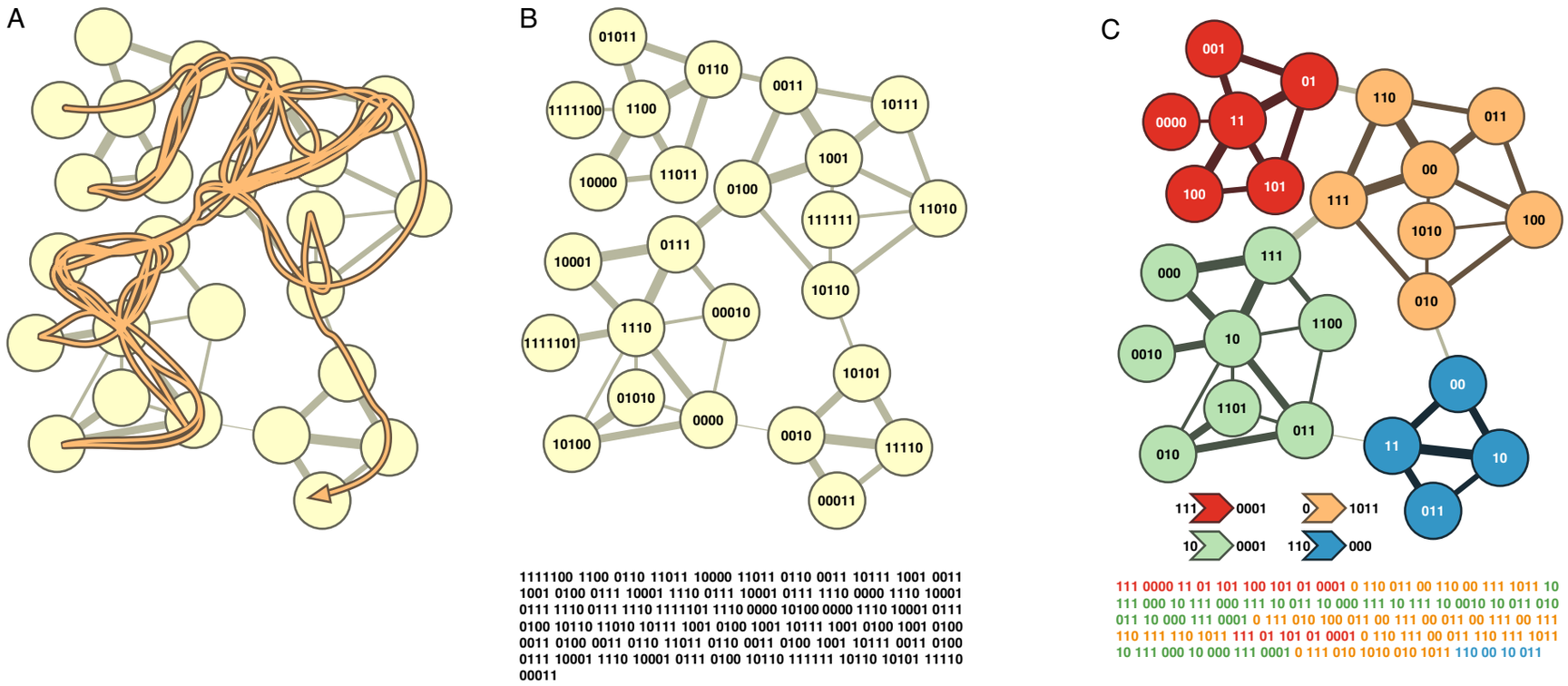
- Assign code words to each node with length proportional to the frequency of visit (Huffman code)
- The length of codeword is bounded below by the $H(P)$ entropy of the RW (Shannon's source coding theorem)

The Infomap method

Rosvall&Bergstrom (2008)

Finding a compressed description of a random walk taking place on a graph

Rosvall et. al. (2008)



- Take a random walker on a (weighted, directed) graph

- Assign code words to each node with length proportional to the frequency of visit (Huffman code)
- The length of codeword is bounded below by the $H(P)$ entropy of the RW (Shannon's source coding theorem)

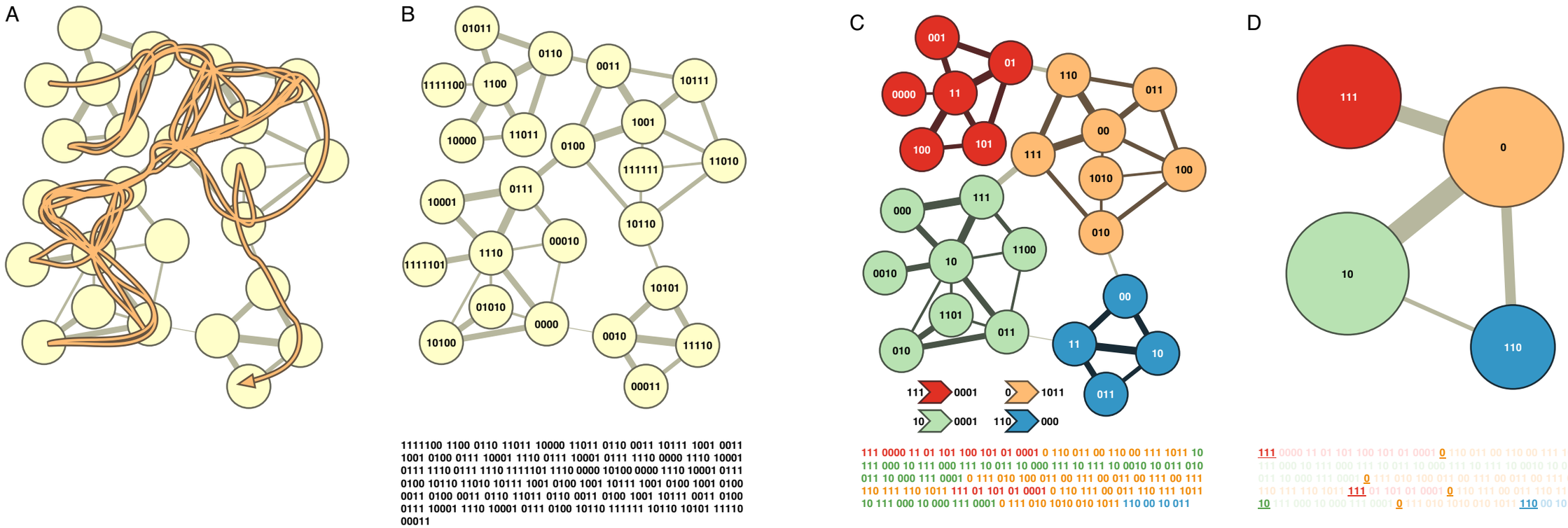
- Define codewords for clusters and recycle names inside (just like in geographic maps)
- With the recycling procedure we are able to shorten the description

The Infomap method

Rosvall&Bergstrom (2008)

Finding a compressed description of a random walk taking place on a graph

Rosvall et. al. (2008)



- Take a random walker on a (weighted, directed) graph

- Assign code words to each node with length proportional to the frequency of visit (Huffman code)
- The length of codeword is bounded below by the $H(P)$ entropy of the RW (Shannon's source coding theorem)

- Define codewords for clusters and recycle names inside (just like in geographic maps)
- With the recycling procedure we are able to shorten the description

- Exit code assigns (0001) each time a walker leaves a cluster
- Enter code assigns the new cluster where the walker enter to

The Infomap method

Finding the optimal partition M:

- Minimise the expected description length of the random walk

probability of between modules movements

probability of within modules movements

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

Expected decryption length of partition M

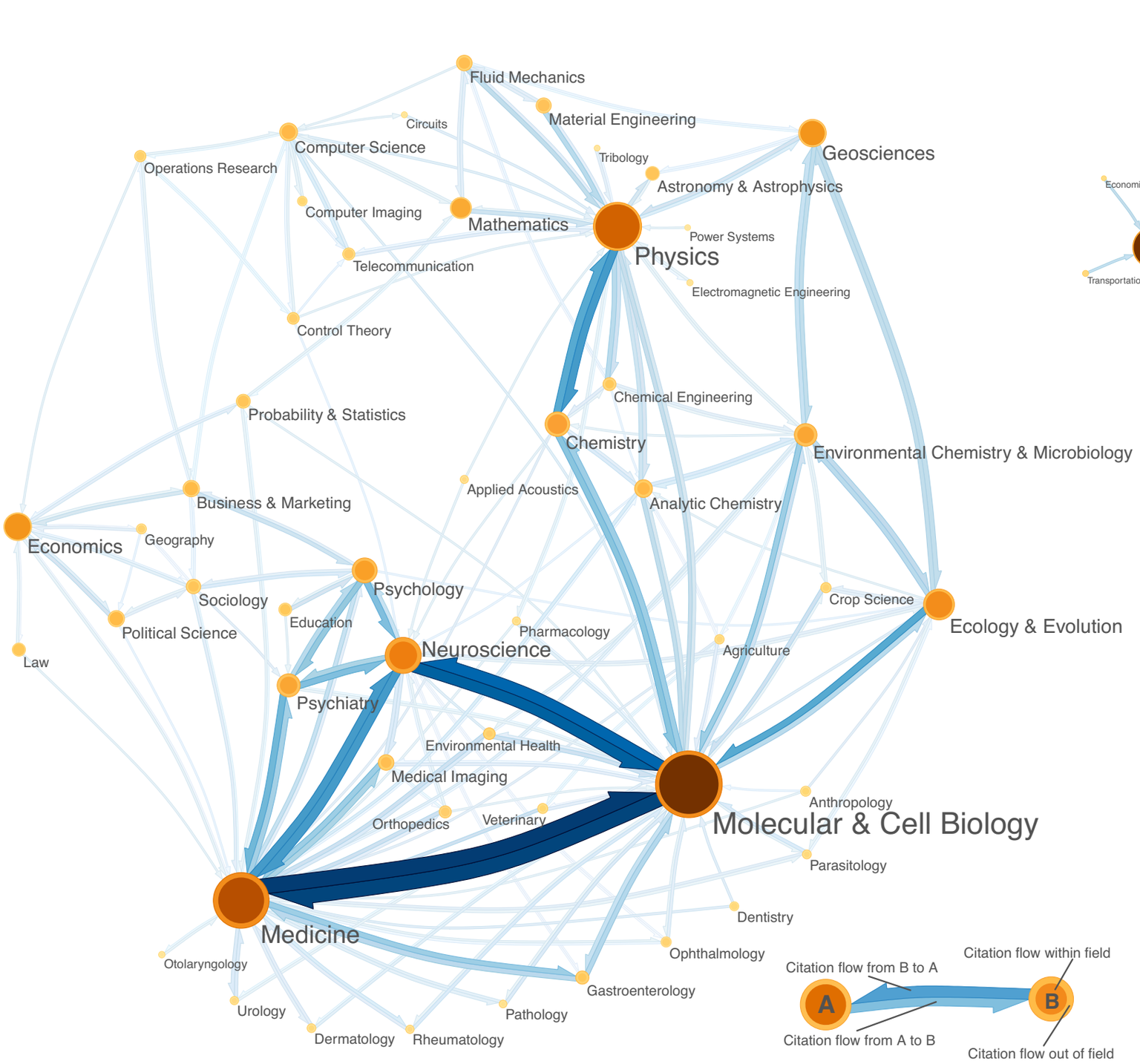
Entropy of movement between modules

Entropy of movement inside modules

Algorithm

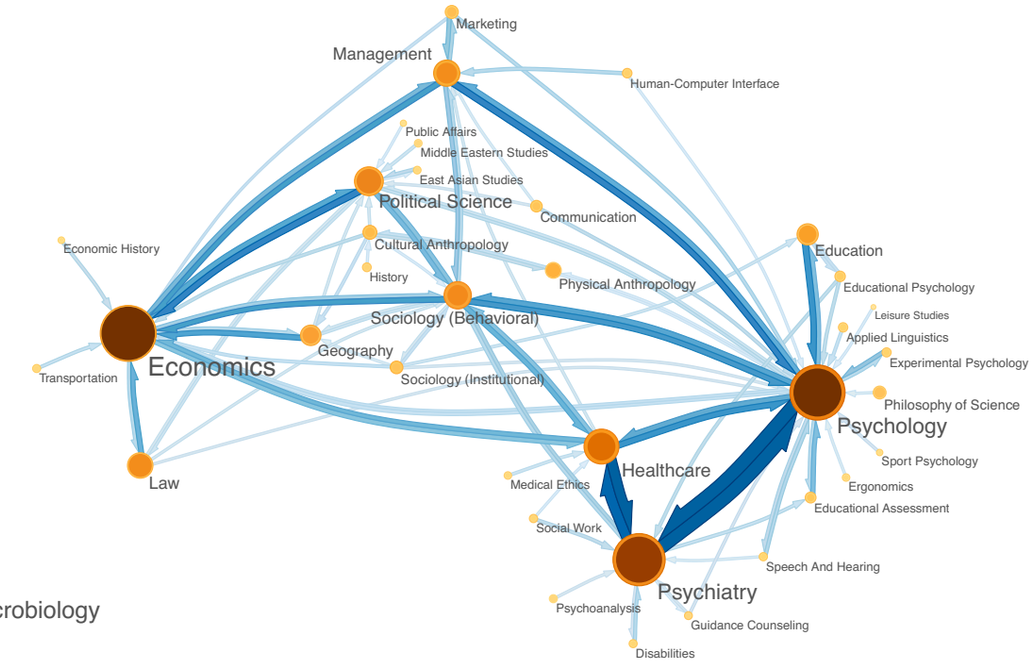
1. Compute the fraction of time each node is visited by the random walker (Power-method)
2. Explore the space of possible partitions (deterministic greedy search algorithm)
3. Refine the results with simulated annealing (heat-bath algorithm)

The Infomap method



Rosvall et. al. (2008)

Map of science based on citation patterns



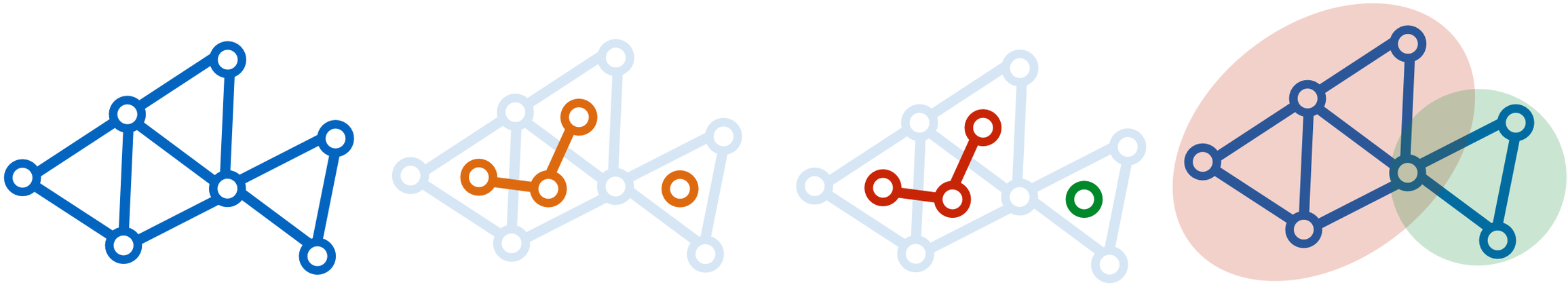
Map of social sciences

The Clique percolation method

Palla et.al. (2005)

Finding overlapping communities by detecting percolating cliques in graphs

- A community consists of several complete subgraphs (cliques), which tend to share many of their nodes
- Define a k-clique community, as a union of all k-cliques (complete subgraphs of size k) that can be reached from each other through a series of adjacent k-cliques (where adjacency means sharing $k\text{-}1$ nodes)
- A single node can belong to several k-clique communities



- The observed community structure is depending on the choice of k-clique size
- Percolating k-clique clusters are including k'-clique clusters if $k' \leqq k$

The Clique percolation method

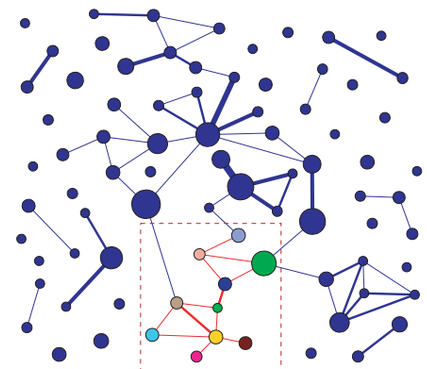
- Finding cliques is an NP-complete problem but on real networks it can be done faster

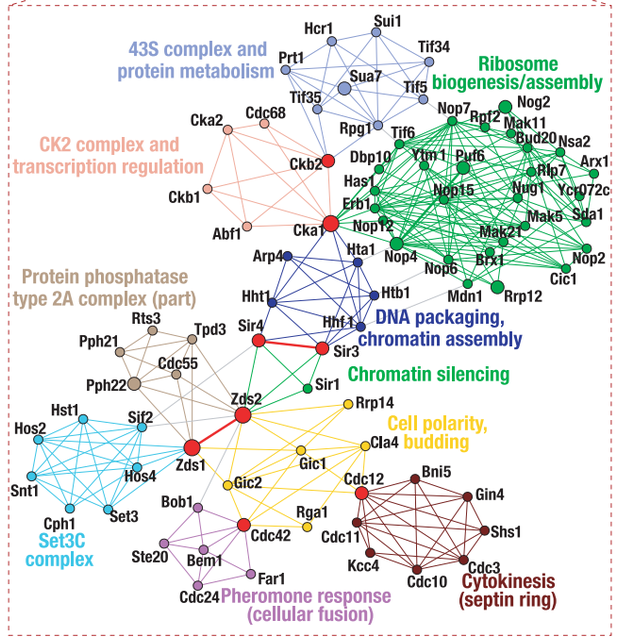
Algorithm

1. Locate all cliques of a graph

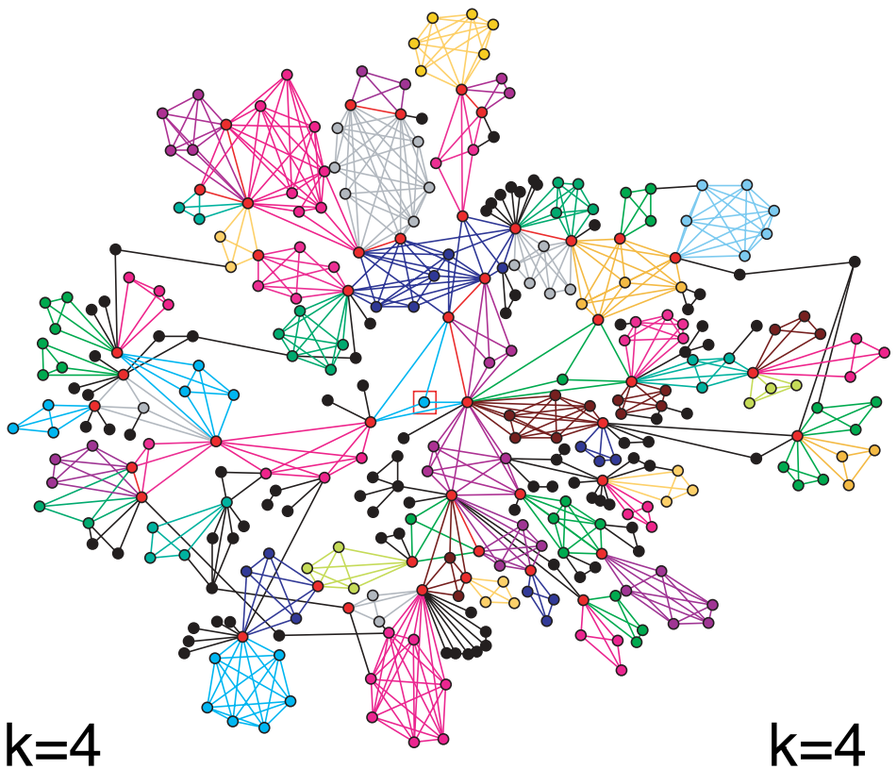
2. Identifying the communities by standard component analysis of the clique–clique overlap matrix

Protein-protein interactions



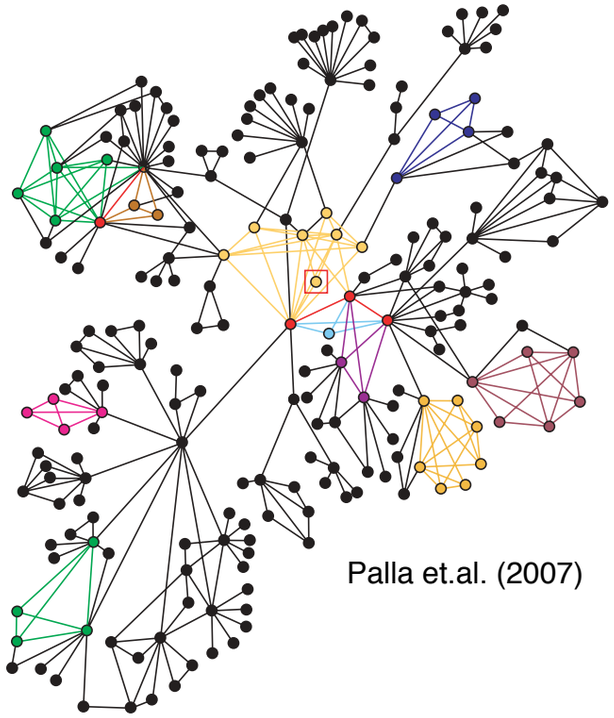


a Co-authorship



k=4

b Phone call



Palla et.al. (2007)

k=4

Palla et.al. (2005)

Testing algorithms

Question: How to test clustering algorithms?

Answer:

- Checking whether they are able to recover known community structure of benchmark graphs
- Checking whether they recover groups assigned from meta-data (location, qualified node groups, gender groups, interest groups, etc.)

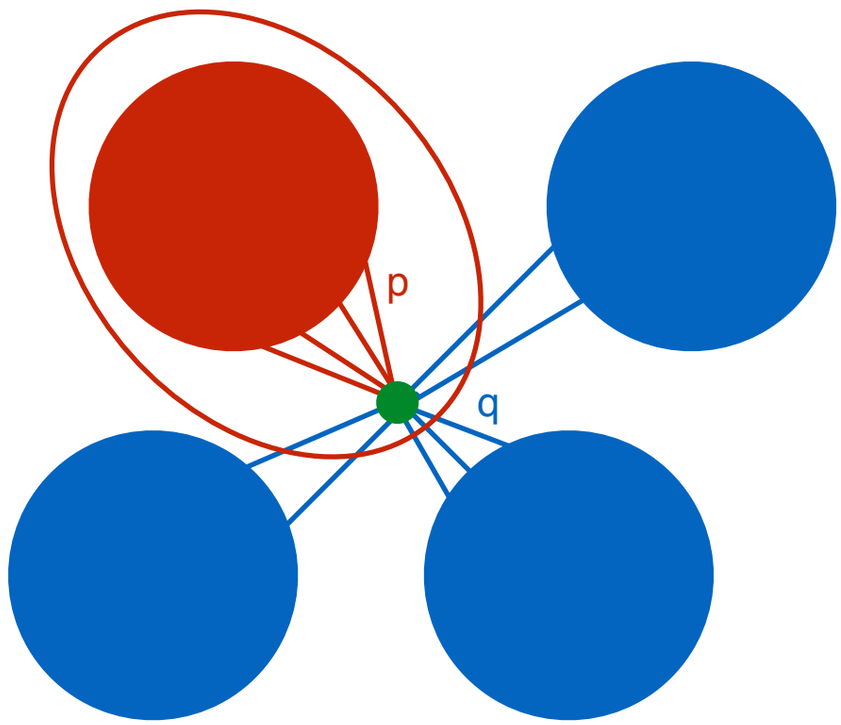
- Warning: the definition of community in the benchmark and in the real graphs should be consistent!

Planted l-partition model

Ingredients

- Graph with n vertices divided into l equal size partitions with $g=n/l$ nodes each
- p - probability that nodes of the same cluster are joined
- q - probability that nodes of different clusters are joined
- If $p>q$ the resulted groups are “communities”

$$\langle k \rangle = p(g-1) + qg(l-1)$$

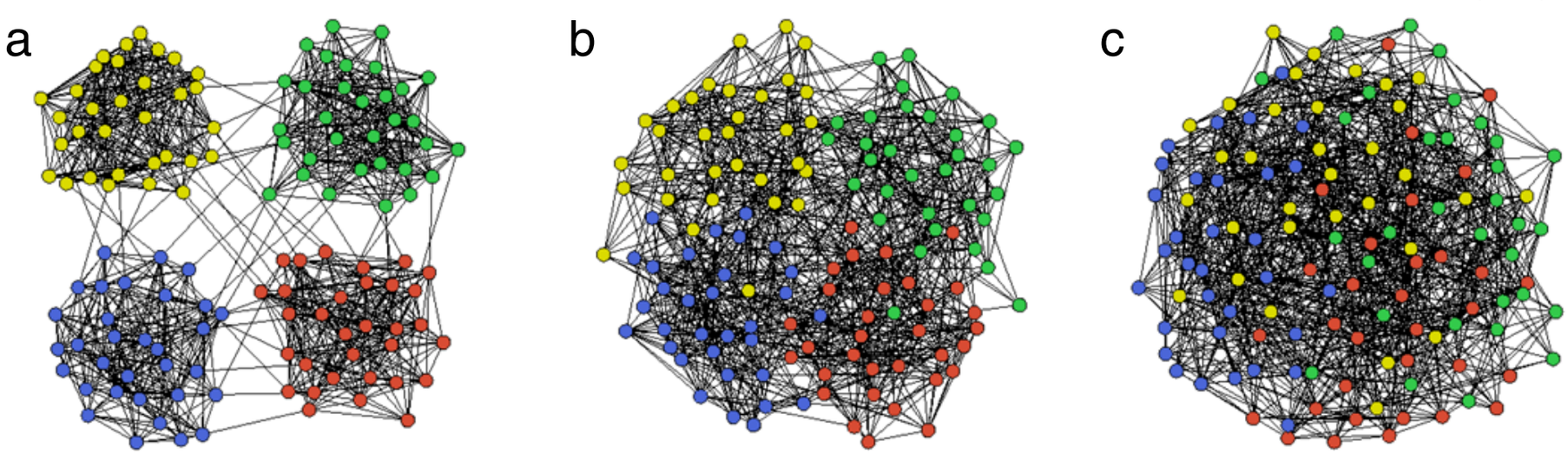


Girvan-Newman benchmark

Special case of planted l-partition model

- $n=128$, $l=4$, $g=32$
- $k_{in} = p(g-1) \sim pg$, $k_{out} = qg(l-1) \sim pg$, $k_{in}+k_{out}=16$
- $p \geq q$: $k_{in} \geq 4$, $k_{out} \leq 12$

Girvan & Newman (2005)



$k_{in}=15$ $k_{in}=11$ $k_{in}=8$

Problems:

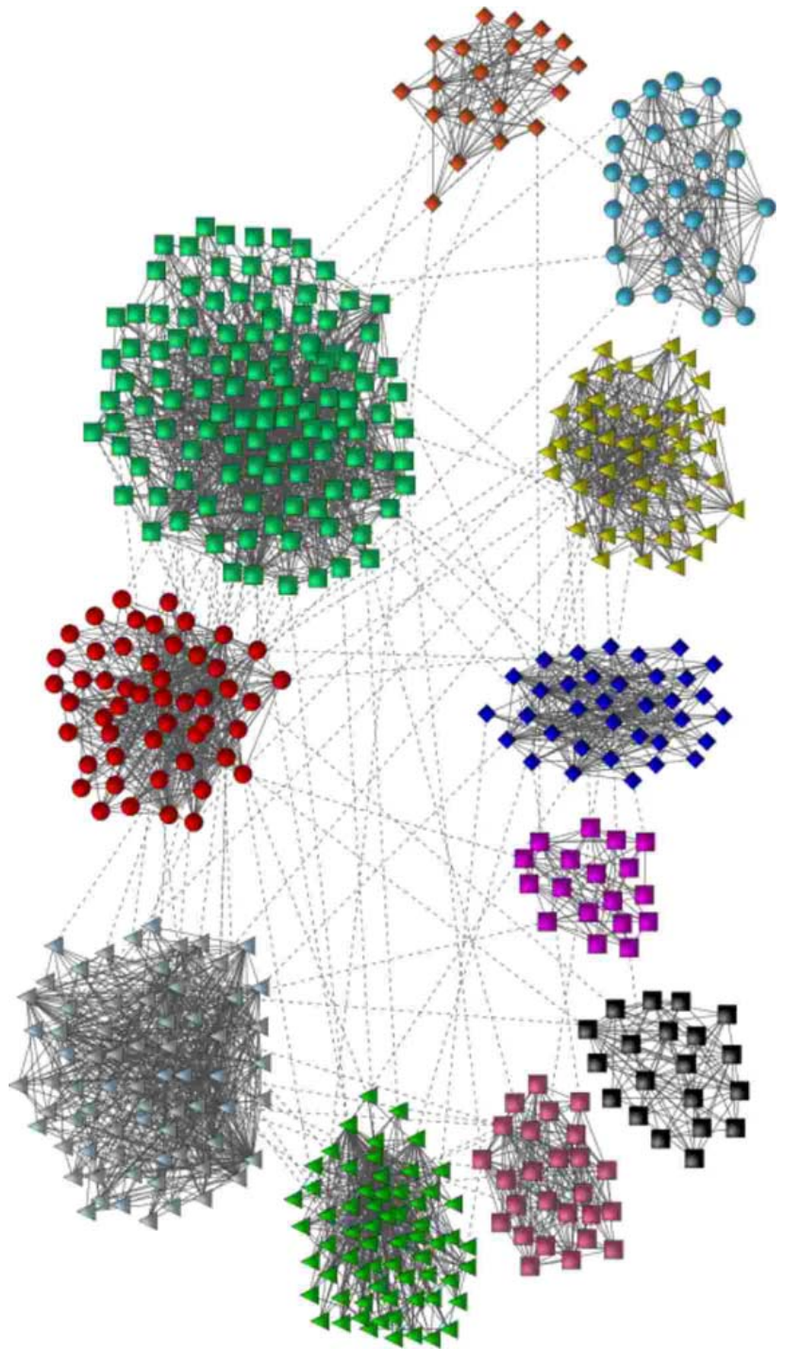
- each node has the same degree
- each community has the same size

LFR benchmark

Lancichinetti, Fortunato, Radicchi (2008)

Features

- Power-law degree distribution with exponent τ_1
- Power-law community size distribution with exponent τ_2
- Mixing parameter μ sets the ratio between external and total degree of each node
  - $k_i(1-\mu)$ number of internal links randomly wired inside the community
  - μk_i number of external links randomly attached to nodes in other communities



Lancichinetti et. al. (2008)

Testing algorithms

Similarity measures between partitions

Example: Normalised mutual information

- x_i and y_i assigns communities where node i was partitioned in the benchmark and by the actual method
- assuming x and y to be random variables: $P(X=x) = n_x/n$, $P(Y=y) = n_y/n$
- Shannon entropy for random variable x: $H(X) = -\sum_x P(x) log P(x)$
- Shannon conditional entropy: $H(X|Y) = -\sum_{xy} P(x,y) log P(x|y)$
- Mutual information: $I(X,Y) = H(X) - H(X|Y)$
- Normalised mutual information: $I_{norm}(X,Y) = \frac{2I(X,Y)}{H(X)+H(Y)}$

  mutual information identical for all Y subpartitions of X

$I_{norm}(X,Y) \in [0,1]$

Independent — Perfect matching

Many other methods:

Author	Label	Order
Girvan & Newman	GN	$O(nm^2)$
Clauset et al.	Clauset et al.	$O(n \log^2 n)$
Blondel et al.	Blondel et al.	$O(m)$
Guimerà et al.	Sim. Ann.	parameter dependent
Radicchi et al.	Radicchi et al.	$O(m^4/n^2)$
Palla et al.	Cfinder	$O(\exp(n))$
Van Dongen	MCL	$O(nk^2)$, $k < n$ parameter
Rosvall & Bergstrom	Infomod	parameter dependent
Rosvall & Bergstrom	Infomap	$O(m)$
Donetti & Muñoz	DM	$O(n^3)$
Newman & Leicht	EM	parameter dependent
Ronhovde & Nussinov	RN	$O(n^\beta)$, $\beta \sim 1$

Fortunato (2010)