NATIONAL RESEARCH UNIVERSITY

# Network communities

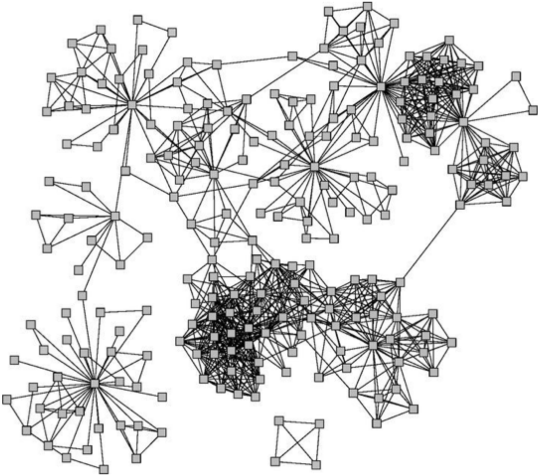
## Network Science
## Lecture 8

Leonid Zhukov

lzhukov@hse.ru
www.leonidzhukov.net/hse/2022/networks

National Research University Higher School of Economics
School of Data Analysis and Artificial Intelligence, Department of Computer Science

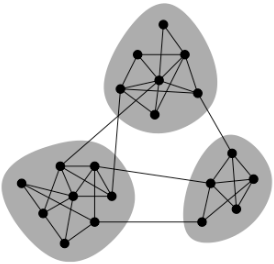
# Network structure



# Network communities

### Definition

*Network communities* are groups of vertices such that vertices inside the group connected with many more edges than between groups.



- Community detection is an assignment of vertices to communities.

# Overlapping and non-overlapping communitites

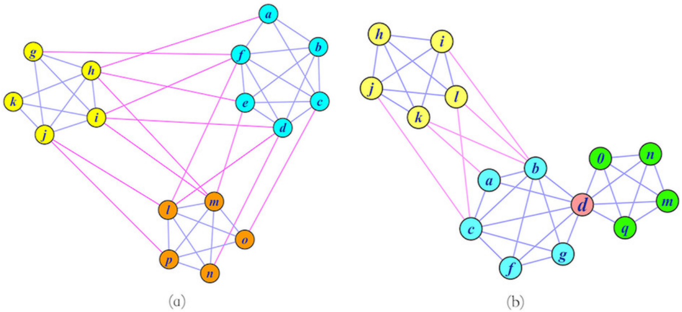



image from W. Liu , 2014

# What makes a community?

Community is a cohesive subgroup:

- Mutuality of ties. Almost everyone in the group has ties (edges) to one another
- Compactness. Closeness or reachability of group members in small number of steps, not necessarily adjacency
- Density of edges. High frequency of ties within the group
- Separation. Higher frequency of ties among group members compared to non-members

Wasserman and Faust

# Community density

- Graph $G(V, E), n = |V|, m = |E|$
- Community - set of nodes $S$
  $n_s$-number of nodes in $S$, $m_s$ - number of edges in $S$
- Graph density

$$\rho = \frac{m}{n(n-1)/2}$$

- community internal density

$$\delta_{int} = \frac{m_s}{n_s(n_s-1)/2}$$

- external edges density

$$\delta_{ext} = \frac{m_{ext}}{n_s(n-n_s)}$$

- community (cluster): $\delta_{int} > \rho$, $\delta_{ext} < \rho$

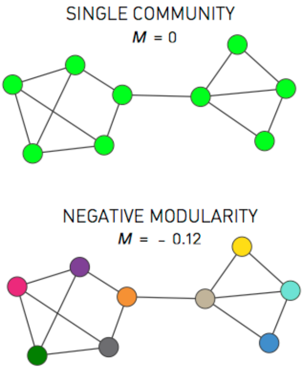
# Modularity

- Compare fraction of edges within the cluster to expected fraction in random graph with identical degree sequence
- Modularity score

$$Q = \frac{1}{2m} \sum_{ij} \left( A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j), = \sum_u \left( \frac{m_u}{m} - \left( \frac{k_u}{2m} \right)^2 \right)$$

  $m_u$ - number of internal edges in a community $u$,
  $k_u$ - sum of node degrees within a community
- Modularity score range $Q \in [-1/2, 1)$, single community $Q = 0$

# Modularity

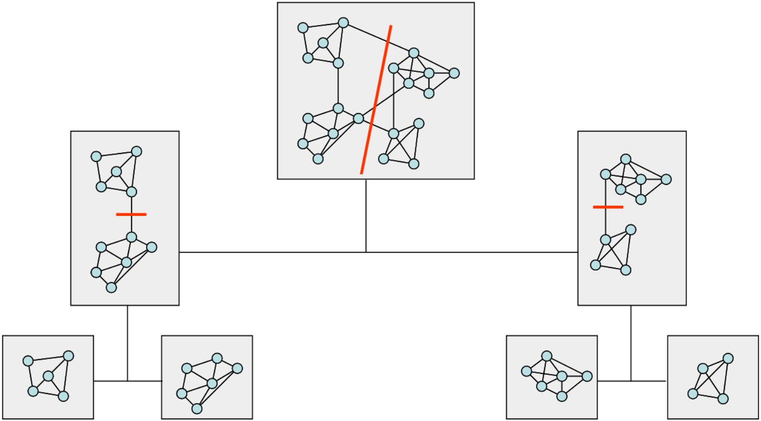



- The higher the modularity score - the better are communities

from A.L. Barabasi 2016

# Community detection by graph partitioning

Recursive graph partitioning



# Finding optimal partition

Graph $G(E, V)$ partition: $V = V_1 + V_2$

- Graph cut

$$Q = cut(V_1, V_2) = \sum_{i \in V_1, j \in V_2} e_{ij}$$

- Ratio cut:

$$Q = \frac{cut(V_1, V_2)}{||V_1||} + \frac{cut(V_1, V_2)}{||V_2||}$$

- Normalized cut:

$$Q = \frac{cut(V_1, V_2)}{Vol(V_1)} + \frac{cut(V_1, V_2)}{Vol(V_2)}$$

- Quotient cut (conductance):

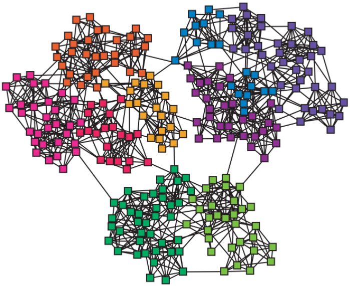
$$Q = \frac{cut(V_1, V_2)}{\min\left(Vol(V_1), Vol(V_2)\right)}$$

where: $Vol(V_1) = \sum_{i \in V_1, j \in V} e_{ij} = \sum_{i \in V_1} k_i$

# Heuristic approach

Edge betweenness -number of shortest paths $\sigma_{st}(e)$ going through edge $e$

$$C_B(e) = \sum_{s \neq t} \frac{\sigma_{st}(e)}{\sigma_{st}}$$



Recover communities by progressively removing edges

# Edge betweenness

Newman-Girvan, 2004

**Algorithm:** Edge Betweenness

**Input:** graph G(V,E)

**Output:** Dendrogram

**repeat**

For all $e \in E$ compute edge betweenness $C_B(e)$;

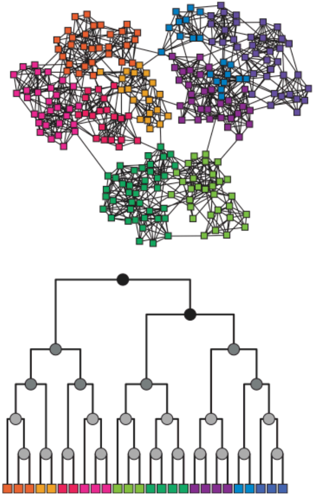
remove edge $e_i$ with largest $C_B(e_i)$ ;

**until** *edges left*;

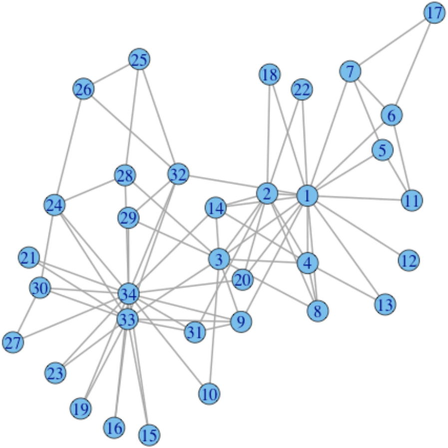
If bi-partition, then stop when graph splits in two components (check for connectedness)

# Edge betweenness

Hierarchical algorithm, dendrogram

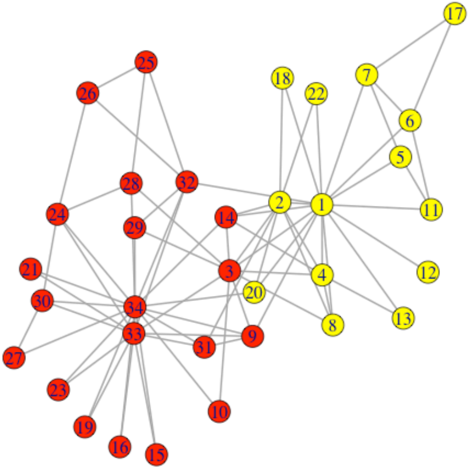


Zachary karate club



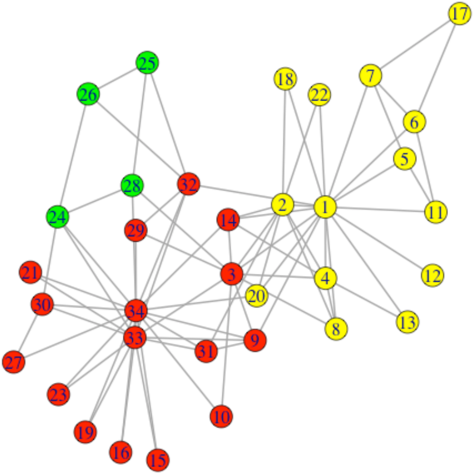
# Edge betweenness

Zachary karate club

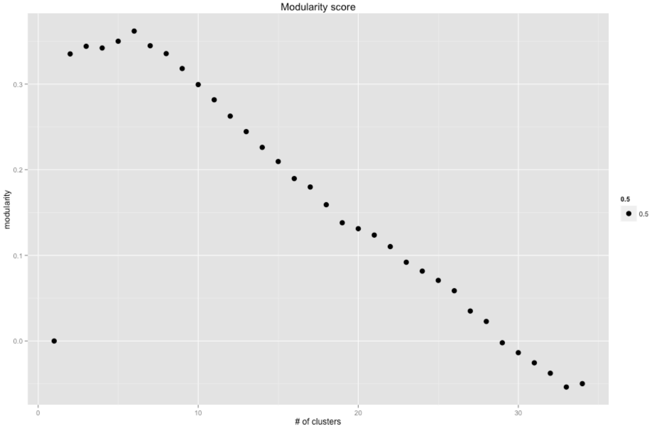


# Edge betweenness

Zachary karate club

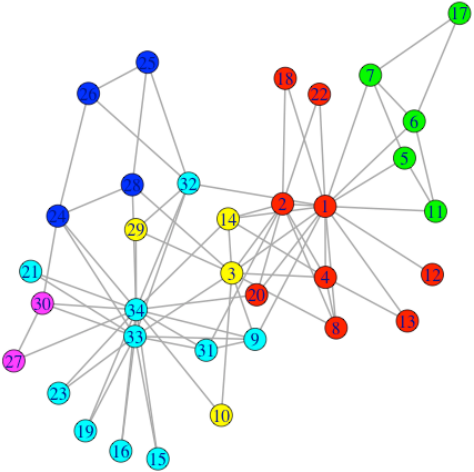


# Edge betweenness



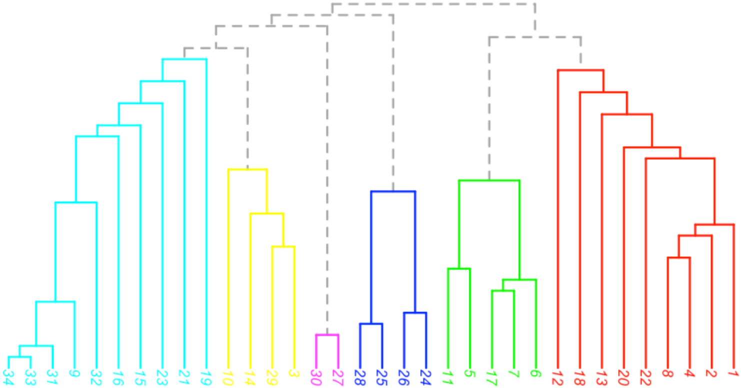
# Edge betweenness

best: clusters = 6, modularity = $0.345$



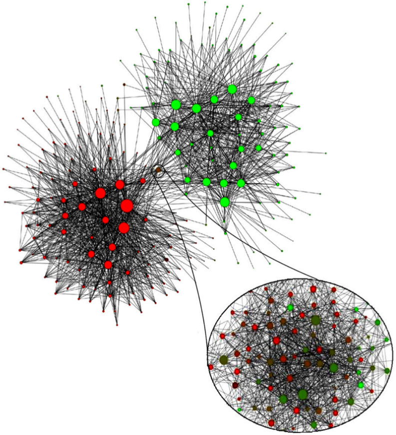
# Edge betweenness

Zachary karate club



# Fast community unfolding

Multi-resolution scalable method



$2$ mln mobile phone network
V. Blondel et.al., 2008

# Fast community unfolding

"The Louvain method"

- Heuristic method for greedy modularity optimization
- Find partitions with high modularity
- Multi-level (multi-resolution) hierarchical scheme
- Scalable

Modularity:

$$Q = \frac{1}{2m} \sum_{i,j} \left( A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j)$$

V. Blondel et.al., 2008

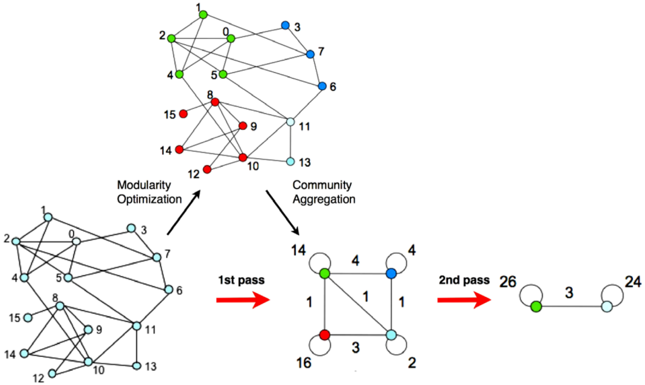
# Fast community unfolding

Algorithm

- Assign every node to its own community
- Phase I
  - For every node evaluate modularity gain from removing node from its community and placing it in the community of its neighbor
  - Place node in the community maximizing modularity gain
  - repeat until no more improvement (local max of modularity)
- Phase II
  - Nodes from communities merged into "super nodes"
  - Weight on the links added up
- Repeat until no more changes (max modularity)

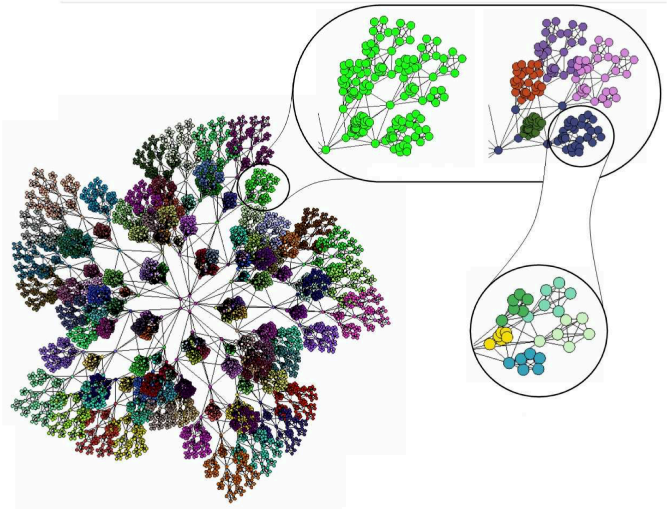
V. Blondel et.al., 2008

# Fast community unfolding




V. Blondel et.al., 2008

# Fast community unfolding



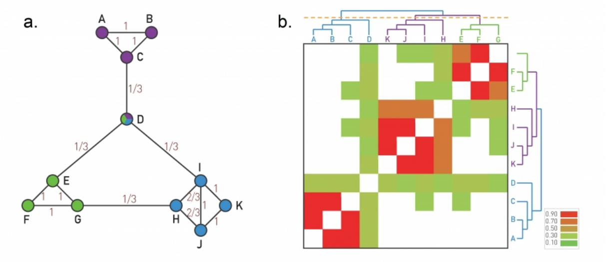
V. Blondel et.al., 2008

# Agglomerative hierarchical clustering

Similarity matrix

$$x_{ij} = \frac{N(i,j)}{\min(k_i, k_j) + 1 - \theta(A_{i,j})}$$

- $N_{ij}$ - number of common neighbors
- $\theta()$ - step function, $0$ for $x \leq 0$ and $1$ for $x > 0$




E. Ravasz et.al., 2002

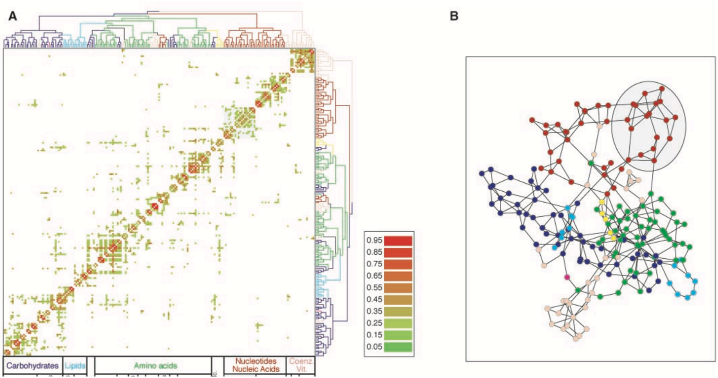
# Agglomerative hierarchical clustering

The Ravasz algorithm:

- Assign each node to a community of its own and evaluate $x_{ij}$ for all node pairs.
- Find the node (community) pair with the highest similarity and merge them into a single community.
- Calculate the similarity between the new community and all other communities.
- Repeat Steps 2 and 3 until all nodes form a single community.
- Find the optimal cut of the dendrogram

E. Ravasz et.al., 2002

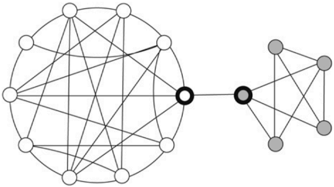
# Agglomerative hierarchical clustering




Reduced E.coli metabolic network

E. Ravasz et.al., 2002

# Communities and random walks



- Random walks on a graph tend to get trapped into densely connected parts corresponding to communities.

# Walktrap community

Walktrap

- Consider random walk on graph
- At each time step walk moves to NN uniformly at random
  $P_{ij} = \frac{A_{ij}}{d(i)}, P = D^{-1}A, D_{ii} = diag(d(i))$
- $P^t_{ij}$ - probability to get from $i$ to $j$ in $t$ steps, $t \ll t_{mixing}$
- Assumptions: for two $i$ and $j$ in the same community $P^t_{ij}$ is high
- if $i$ and $j$ are in the same community, then $\forall k, P^t_{ik} \approx P^t_{jk}$
- Distance between nodes:

$$r_{ij}(t) = \sqrt{\sum_{k=1}^{n} \frac{(P^t_{ik} - P^t_{jk})^2}{d(k)}} = ||D^{-1/2}P^t_i - D^{-1/2}P^t_j||$$

P. Pons and M. Latapy, 2006

# Walktrap

Computing node distance $r_{ij}$

- Direct (exact) computation: $P^t_{ij} = (P^t)_{ij}$ or $P^t_i = P^t p^0_i, p^0_i(k) = \delta_{ik}$
- Approximate computation (simulation):
  – Compute $K$ random walks of length $t$ starting form node $i$
  – Approximate $P^t_{ik} \approx \frac{N_{ik}}{K}$, number of walks end up on $k$

Distance between communities:

$$P^t_{Cj} = \frac{1}{|C|} \sum_{i \in C} P^t_{ij}$$

$$r_{C_1 C_2}(t) = \sqrt{\sum_{k=1}^{n} \frac{(P^t_{C_1 k} - P^t_{C_2 k})^2}{d(k)}} = ||D^{-1/2} P^t_{C_1} - D^{-1/2} P^t_{C_2}||$$

P. Pons and M. Latapy, 2006

# Walktrap

Algorithm (hierarchical clustering)

- Assign each vertex to its own community $S_1 = \{\{v\}, v \in V\}$
- Compute distance between all adjacent communities $r_{C_i C_j}$
- Choose two "closest" communities that minimizes (Ward's methods):

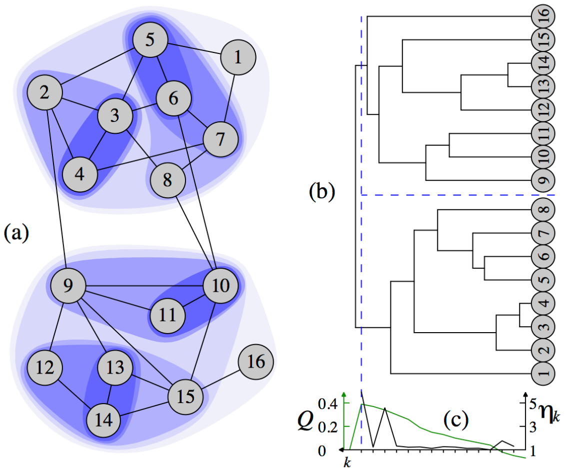
$$\Delta\sigma(C_1, C_2) = \frac{1}{n}\left(\sum_{i \in C_3} r^2_{iC_3} - \sum_{i \in C_1} r^2_{iC_1} - \sum_{i \in C_2} r^2_{iC_2}\right)$$

  and merge them $S_{k+1} = (S_k \backslash \{C_1, C_2\}) \cup C_3$, $C_3 = C_1 \cup C_2$
- update distance between communities

After $n - 1$ steps finish with one community $S_n = \{V\}$

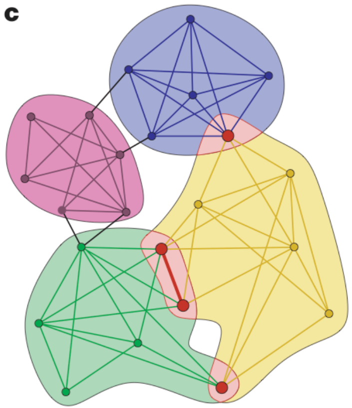
P. Pons and M. Latapy, 2006

# Walktrap



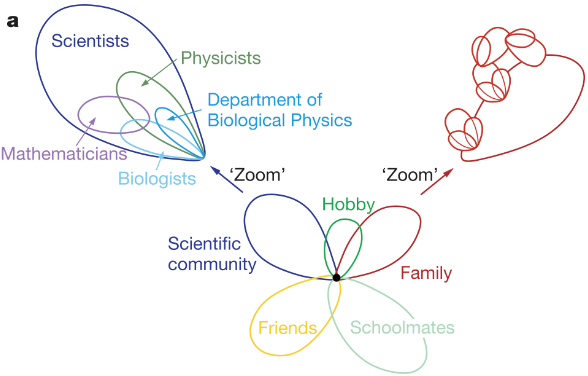

P. Pons and M. Latapy, 2006

# Overlapping communities




Palla, 2005

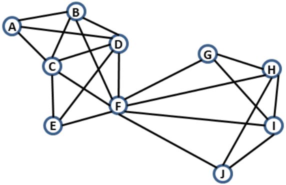
# Overlapping communities

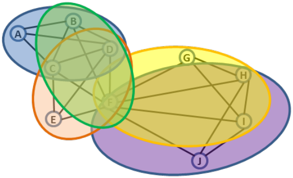



Palla, 2005

# $k$-clique community

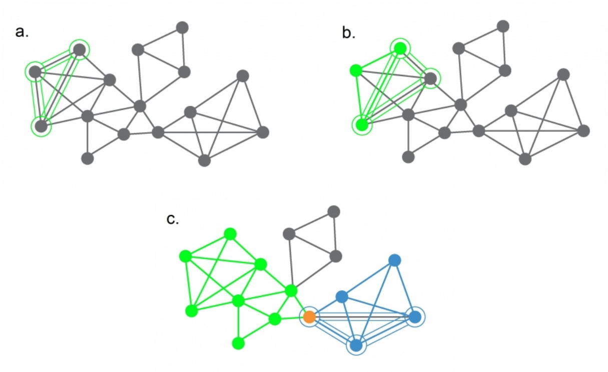
- $k$-clique is a clique (complete subgraph) with $k$ nodes
- $k$-clique community a union of all $k$-cliques that can be reached from each other through a series of adjacent $k$-cliques
- two k-cliques are said to be adjacent if they share $k-1$ nodes.




Adjacent $4$-cliques

# $k$-clique cumminity



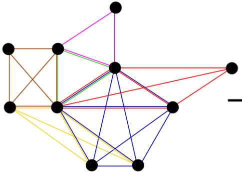

Palla, 2005

# Algorithm CFinder

- Find all maximal cliques
- Create clique overlap matrix
- Threshold matrix at value $k-1$
- Communities = connected components

Palla, 2005

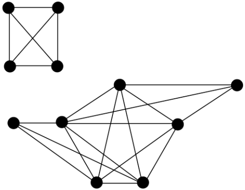
# *k*-clique percolation



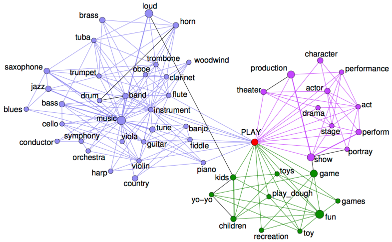
| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | | | |
| | 5 | 3 | 2 | 1 | 3 | 1 |
| | 3 | 4 | 2 | 1 | 1 | 1 |
| | 2 | 2 | 3 | 2 | 1 | 2 |
| | 1 | 1 | 2 | 3 | 0 | 1 |
| | 3 | 1 | 1 | 0 | 4 | 2 |
| | 1 | 1 | 2 | 1 | 2 | 4 |

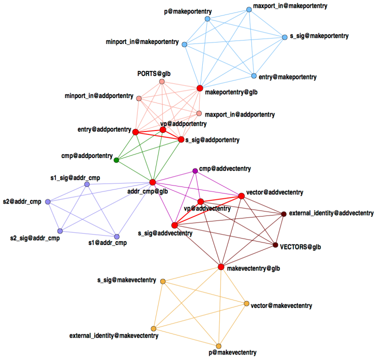
k=4

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | | | |
| | 1 | 1 | 0 | 0 | 1 | 0 |
| | 1 | 1 | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0 | 0 | 0 |
| | 1 | 0 | 0 | 0 | 1 | 0 |
| | 0 | 0 | 0 | 0 | 0 | 1 |



# $k$-clique percolation

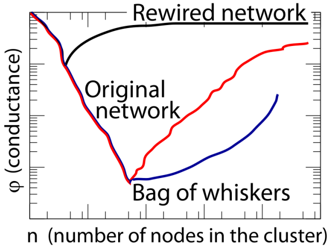


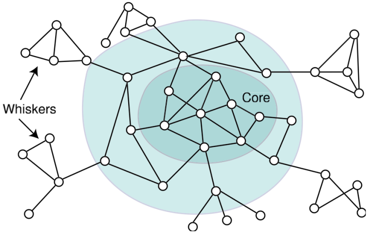

$k = 4$ $\qquad\qquad$ $k = 5$

Palla, 2005

# Real world communities




(a) Typical NCP plot




(b) Caricature of network structure

Best conductance of a vertex set $S$ of size $k$:

$$\Phi(k) = \min_{S \in V, |S| = k} \phi(S), \quad \phi(S) = \frac{cut(S, V \backslash S)}{\min(vol(S), vol(S \backslash V))}$$

where $vol(S) = \sum_{i \in S} k_i$ - sum of all node degrees in the set

J. Leskovec, K. Lang, 2010

# Community detection algorithms

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

| Author | Ref. | Label | Order |
|---|---|---|---|
| Girvan & Newman | (Girvan and Newman, 2002; Newman and Girvan, 2004) | GN | $O(nm^2)$ |
| Clauset et al. | (Clauset *et al.*, 2004) | Clauset et al. | $O(n\log^2 n)$ |
| Blondel et al. | (Blondel *et al.*, 2008) | Blondel et al. | $O(m)$ |
| Guimerà et al. | (Guimerà and Amaral, 2005; Guimerà *et al.*, 2004) | Sim. Ann. | parameter dependent |
| Radicchi et al. | (Radicchi *et al.*, 2004) | Radicchi et al. | $O(m^4/n^2)$ |
| Palla et al. | (Palla *et al.*, 2005) | Cfinder | $O(\exp(n))$ |
| Van Dongen | (Dongen, 2000a) | MCL | $O(nk^2)$, $k<n$ parameter |
| Rosvall & Bergstrom | (Rosvall and Bergstrom, 2007) | Infomod | parameter dependent |
| Rosvall & Bergstrom | (Rosvall and Bergstrom, 2008) | Infomap | $O(m)$ |
| Donetti & Muñoz | (Donetti and Muñoz, 2004, 2005) | DM | $O(n^3)$ |
| Newman & Leicht | (Newman and Leicht, 2007) | EM | parameter dependent |
| Ronhovde & Nussinov | (Ronhovde and Nussinov, 2009) | RN | $O(m^\beta\log n)$, $\beta\sim1.3$ |

# References


- M.A Porter, J-P Onella, P.J. Mucha. Communities in Networks, Notices of the American Mathematical Society, Vol. 56, No. 9, 2009
- Finding and evaluating community structure in networks, M.E.J. Newman, M. Girvan, Phys. Rev E, 69, 2004
- Modularity and community structure in networks, M.E.J. Newman, PNAS, vol 103, no 26, pp 8577-8582, 2006
- S. E. Schaeffer. Graph clustering. Computer Science Review, 1(1):27–64, 2007.
- S. Fortunato. Community detection in graphs, Physics Reports, Vol. 486, Iss. 3–5, pp 75-174, 2010

# References


- G. Palla, I. Derenyi, I. Farkas, T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, Nature 435 (2005) 814?818.
- P. Pons and M. Latapy, Computing communities in large networks using random walks, Journal of Graph Algorithms and Applications, 10 (2006), 191-218.
- V.D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech. P10008 (2008).
- J. Leskovec, K.J. Lang, A. Dasgupta, and M.W. Mahoney. Statistical properties of community structure in large social and information networks. In WWW 08: Procs. of the 17th Int. Conf. on World Wide Web, pages 695-704, 2008.