School of Information
University of Michigan

Community structure (lab)

Outline

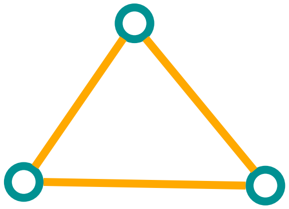
- finding a motif (Pajek)
- FANMOD
- doing a triad census (Pajek)
- hierarchical clustering (Pajek)
- betweenness clustering (Guess)
- getting an m-slice

Finding motifs (cliques and subgraphs) in Pajek

- Create a second network that is the subgraph you are looking for

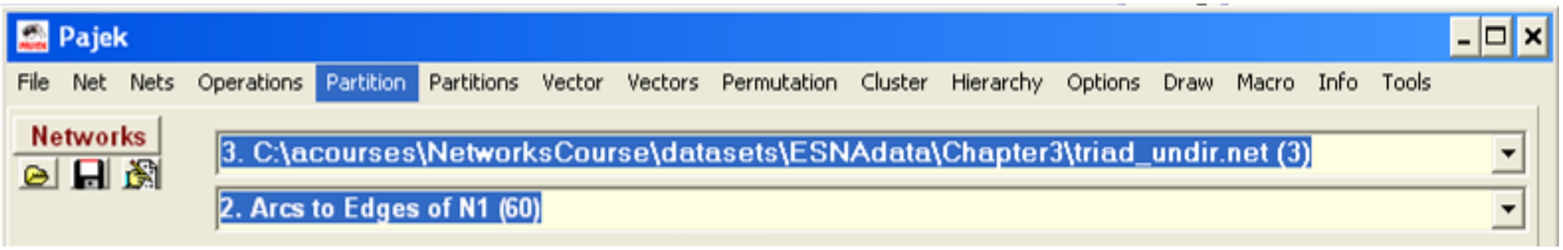
e.g. an undirected triad

```
*Vertices     3
    1 "v1"
    2 "v2"
    3 "v3"
*Arcs
*Edges
    2     3     1
    1     2     1
    1     3     1
```



finding motifs with Pajek

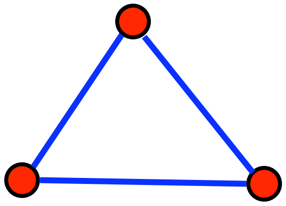
- Use the two drop down menus in the ‘networks’ list to specify two networks:



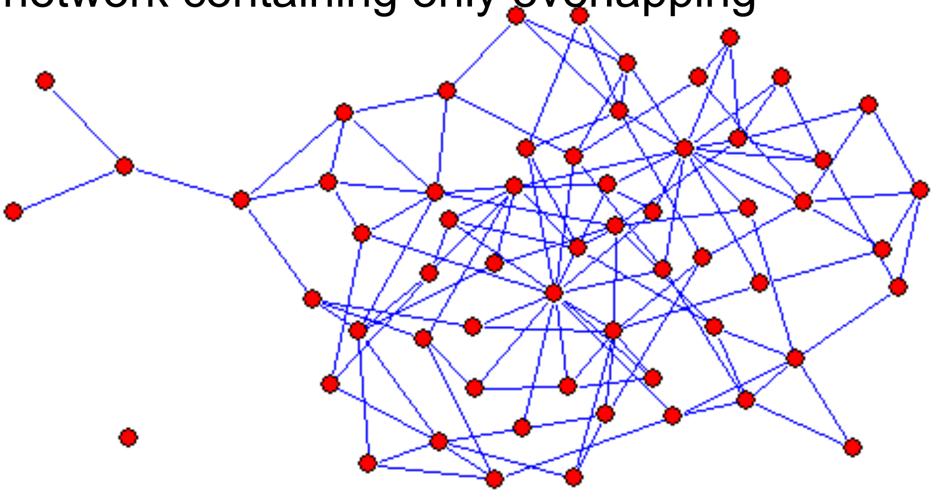
3. C:\acourses\NetworksCourse\datasets\ESNAdata\Chapter3\triad_undir.net (3)

2. Arcs to Edges of N1 (60)

- Then run Nets>Fragment (1 in 2)>Find
  - under Nets>Fragment (1 in 2)>Options
    - can select ‘induced’ subnetwork containing only overlapping fragments

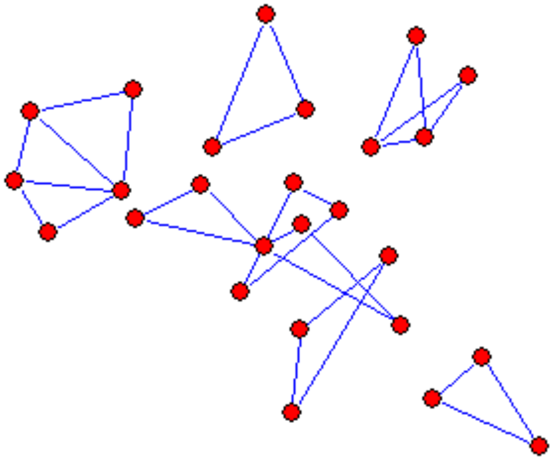


in



finding motifs with Pajek (cont'd)

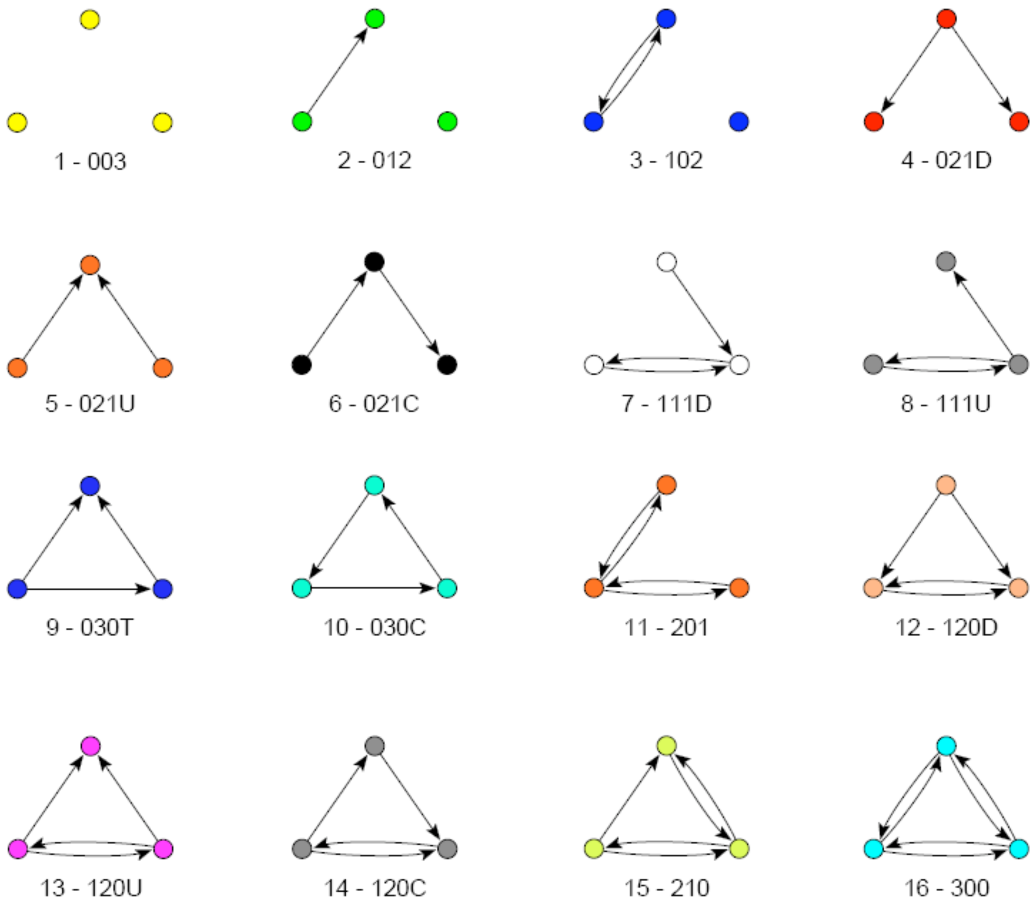
- Now we have just the triads:



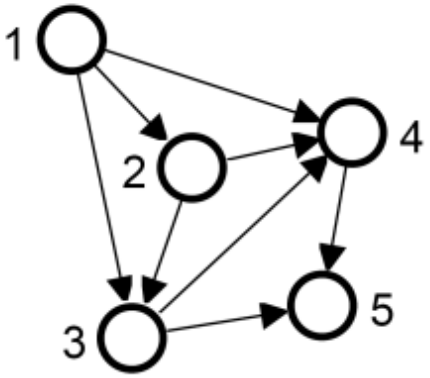
- Creates a hierarchy object with the membership of each triad listed

Triadic census in Pajek

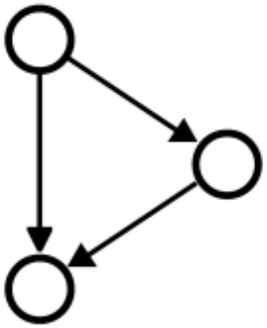
- Info > Network > Triadic Census



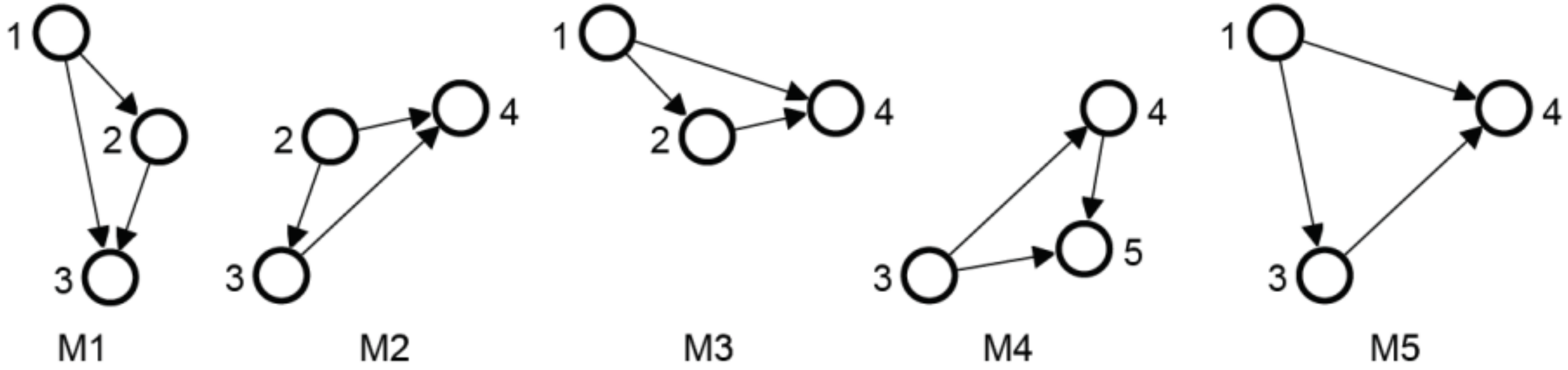
Finding “motifs” in the network



graph



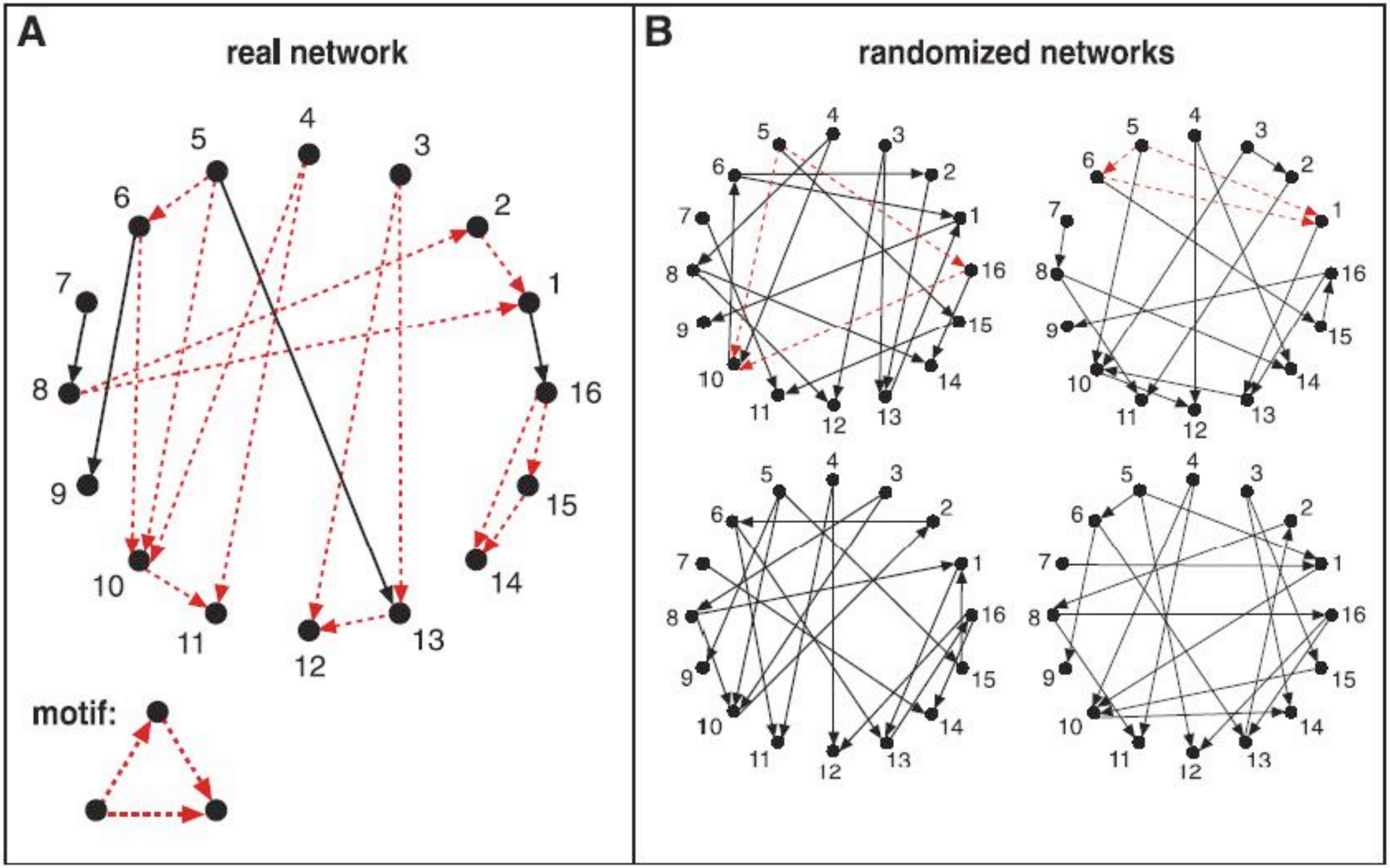
motif to be found



motif matches in the target graph

http://mavisto.ipk-gatersleben.de/frequency_concepts.html

Schematic view of network motif detection



source: Milo et al., Network motifs: Simple building blocks of complex networks, Science 298:824-827, 2002

Network motif detection

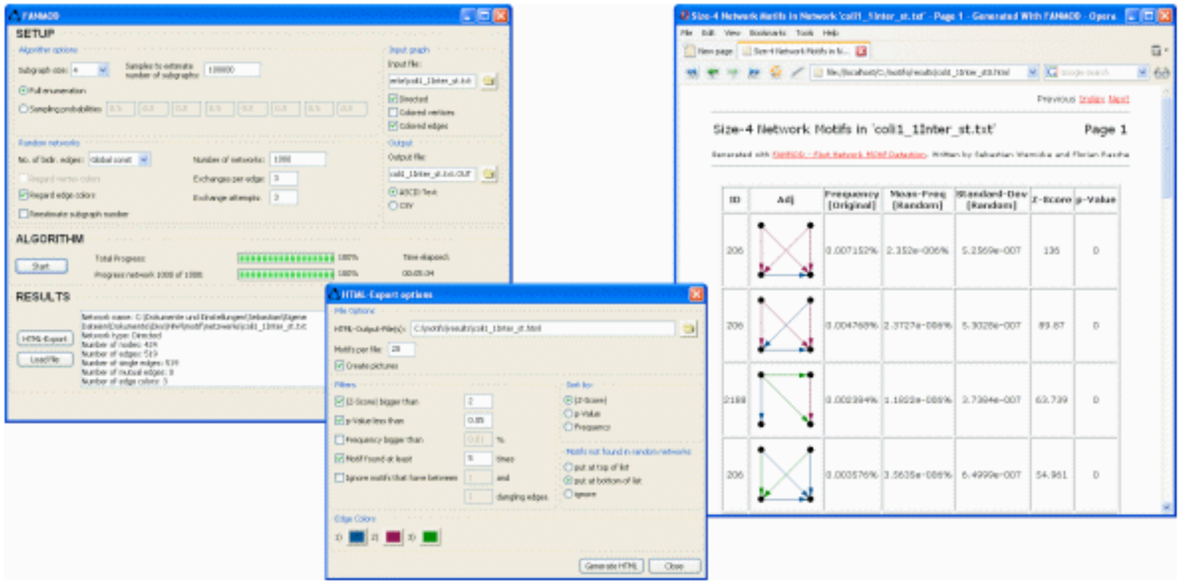
- Some motifs will occur more often in real world networks than random networks
- Technique:
  - construct many random graphs with the same number of nodes and edges (same node degree distribution?)
  - count the number of motifs in those graphs
  - calculate the Z score: the probability that the given number of motifs in the real world network could have occurred by chance
- Software available:
  - http://www.weizmann.ac.il/mcb/UriAlon/ (the original)
  - http://theinf1.informatik.uni-jena.de/~wernicke/motifs/index.html

  (faster and more user friendly)

FANMOD

- http://theinf1.informatik.uni-jena.de/~wernicke/motifs/index.html

FANMOD a tool for fast network motif detection



FANMOD

SETUP

Algorithm options

Subgraph size: 4

Samples to estimate number of subgraphs: 100000

● Full enumeration

○ Sampling probabilities: 0.5 0.8 0.8 0.5 0.8 0.8 0.5 0.8

Input graph

Input file: .../coli1_1Inter_st.txt

☑ Directed

□ Colored vertices

☑ Colored edges

Random networks

No. of bidir. edges: Global const

Number of networks: 1000

Exchanges per edge: 3

Exchange attempts: 3

□ Regard vertex colors

☑ Regard edge colors

□ Reestimate subgraph number

Output

Output file: coli1_1Inter_st.txt.OUT

● ASCII Text

○ CSV

ALGORITHM

Total Progress: 100%

Progress network 1000 of 1000: 100%

Time elapsed: 00:05:04

RESULTS

Network name: C:\Dokumente und Einstellungen\Sebastian\Eigene Dateien\Dokumente\Diss\HiWi\motif\networks\coli1_1Inter_st.txt
Network type: Directed
Number of nodes: 424
Number of edges: 519
Number of single edges: 519
Number of mutual edges: 0
Number of edge colors: 3

HTML-Export options

File Options

HTML-Output-File(s): C:\motifs\results\coli1_1Inter_st.html

Motifs per file: 20

☑ Create pictures

Filters

☑ |Z-Score| bigger than 2

☑ p-Value less than 0.05

□ Frequency bigger than 0.01 %

☑ Motif found at least 5 times

□ Ignore motifs that have between 1 and 1 dangling edges

Sort by

● |Z-Score|

○ p-Value

○ Frequency

Motifs not found in random networks

○ put at top of list

● put at bottom of list

○ ignore

Edge Colors

1) 🟦 2) 🟥 3) 🟩

Size-4 Network Motifs in Network 'coli1_1Inter_st.txt' - Page 1 - Generated With FANMOD - Opera

Previous Index Next!

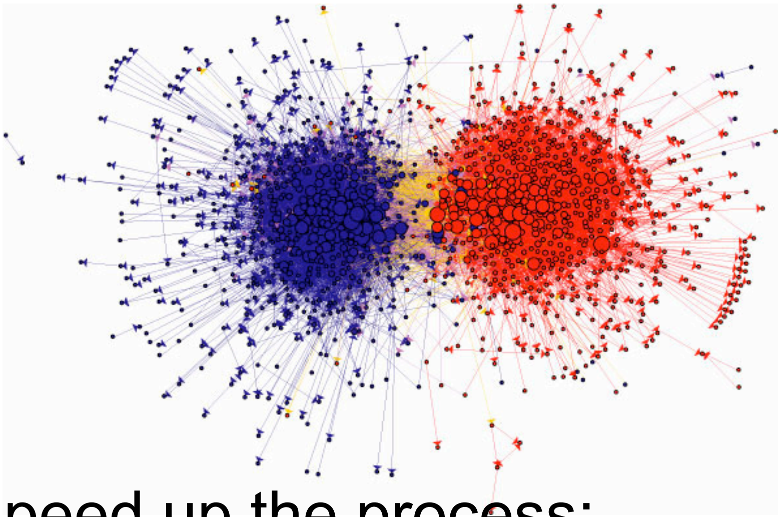
Size-4 Network Motifs in 'coli1_1Inter_st.txt' Page 1

Generated with FANMOD - Fast Network Motif Detection. Written by Sebastian Wernicke and Florian Rasche

ID	Adj	Frequency [Original]	Mean-Freq [Random]	Standard-Dev [Random]	Z-Score	p-Value
206		0.007152%	2.352e-006%	5.2569e-007	136	0
206		0.004768%	2.3727e-006%	5.3028e-007	89.87	0
2188		0.002384%	1.1822e-006%	3.7394e-007	63.739	0
206		0.003576%	3.5635e-006%	6.4999e-007	54.961	0

Lab task

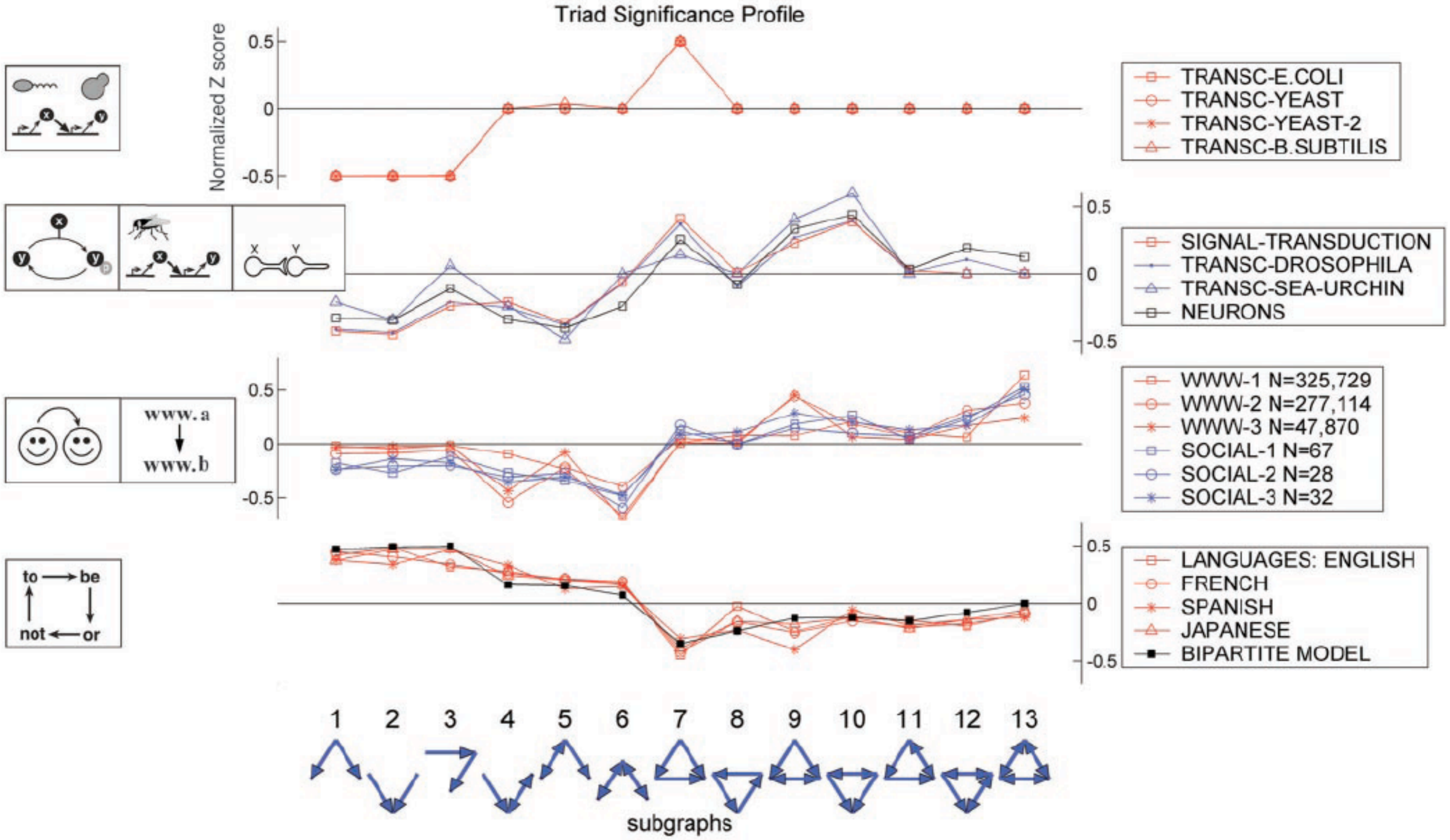
- Download the file poliblogmfinder.txt. It is this network:



- In order to speed up the process:
  - sample rather than doing a full enumeration (10,000 samples rather than 100,000)
  - select 100 rather than 1000 randomized graphs

source: Adamic & Glance, LinkKDD2005

Which of the following "superfamilies" does your network most look like?



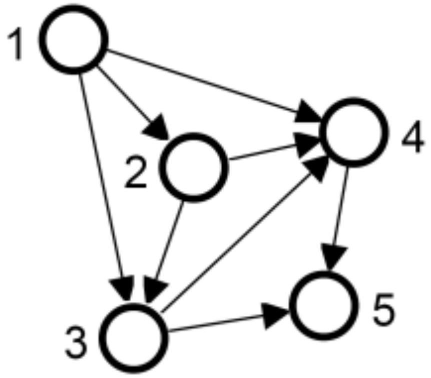
source: Milo et al., Superfamilies of Evolved and Designed Networks, Science 303:1538-1542, 2004

Hierarchical clustering

- Process:
  - after calculating the weights W for all pairs of vertices
  - start with all n vertices disconnected
  - add edges between pairs one by one in order of decreasing weight

Motifs: recap

- Given a particular structure, search for it in the network, e.g. complete triads
- advantage: motifs an correspond to particular functions, e.g. in biological networks
- disadvantage: don’t know if motif is part of a larger cohesive community



Hierarchical clustering in Pajek

http://mrvar.fdv.uni-lj.si/sola/info4/nusa/doc/block1.pdf

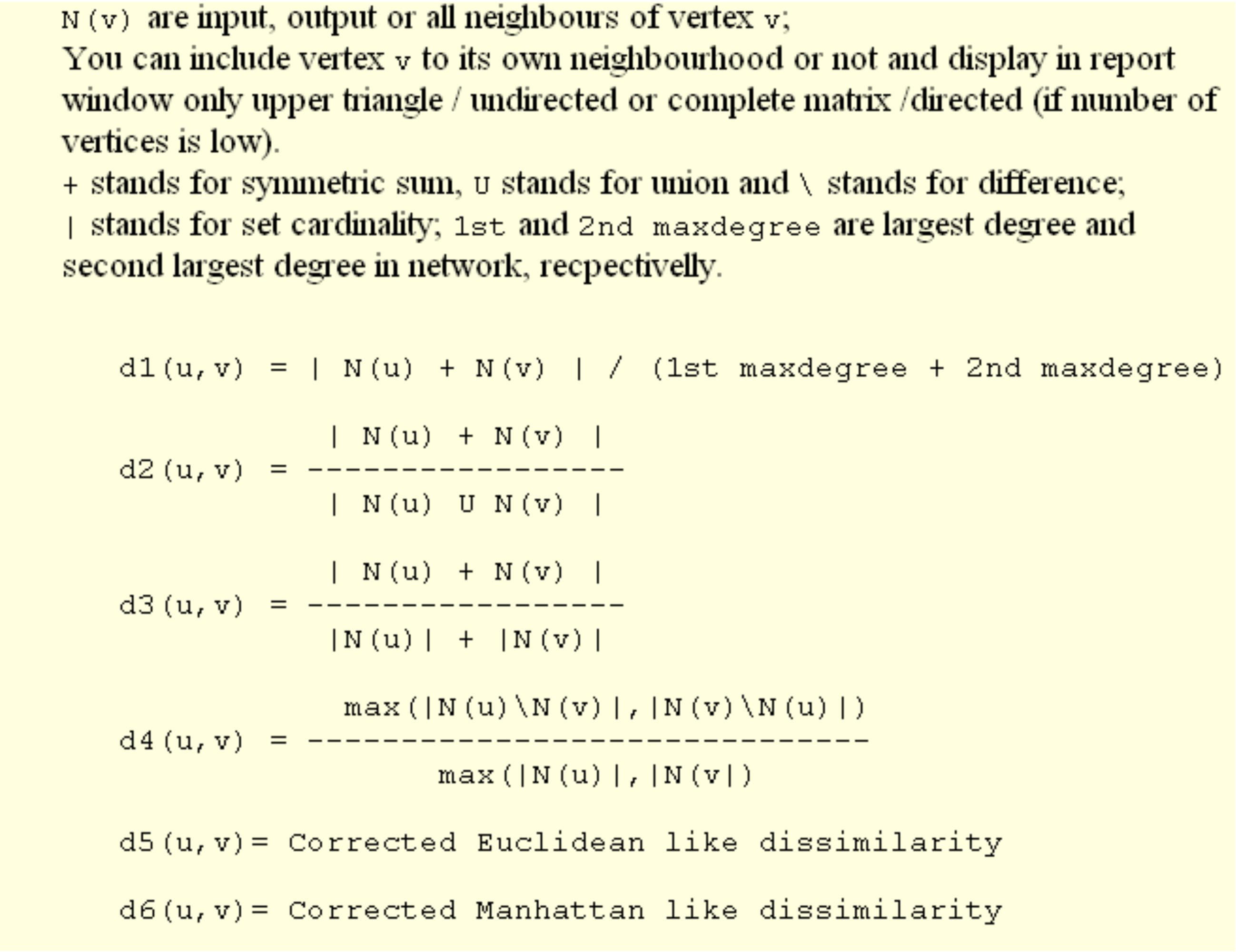
- Procedure
  - generate a complete cluster using Cluster->Create Complete Cluster
  - compute the dissimilarity matrix
    - run Operations->Dissimilarity
      - select “d1/All” to consider network as a binary matrix
      - select “Corrected Euclidean” or “Corrected Manhattan” distance for valued networks

Hierarchical clustering in Pajek

http://mrvar.fdv.uni-lj.si/sola/info4/nusa/doc/block1.pdf

- Procedure (continued)
  - the above will use the dissimilarity matrix to hierarchically cluster nodes and output
    - a dissimilarity matrix
    - EPS picture of the dendrogram
    - permutation of vertices according to the dendrogram
    - hierarchy representing hierarchical clustering
      - to visualize:
        - Edit->Show Subtree
        - Select nodes (Edit->Change Type or Ctrl+T)
        - transform the hierarchy into a partition (Hierarchy->Make Partition)

computing dissimilarities in Pajek



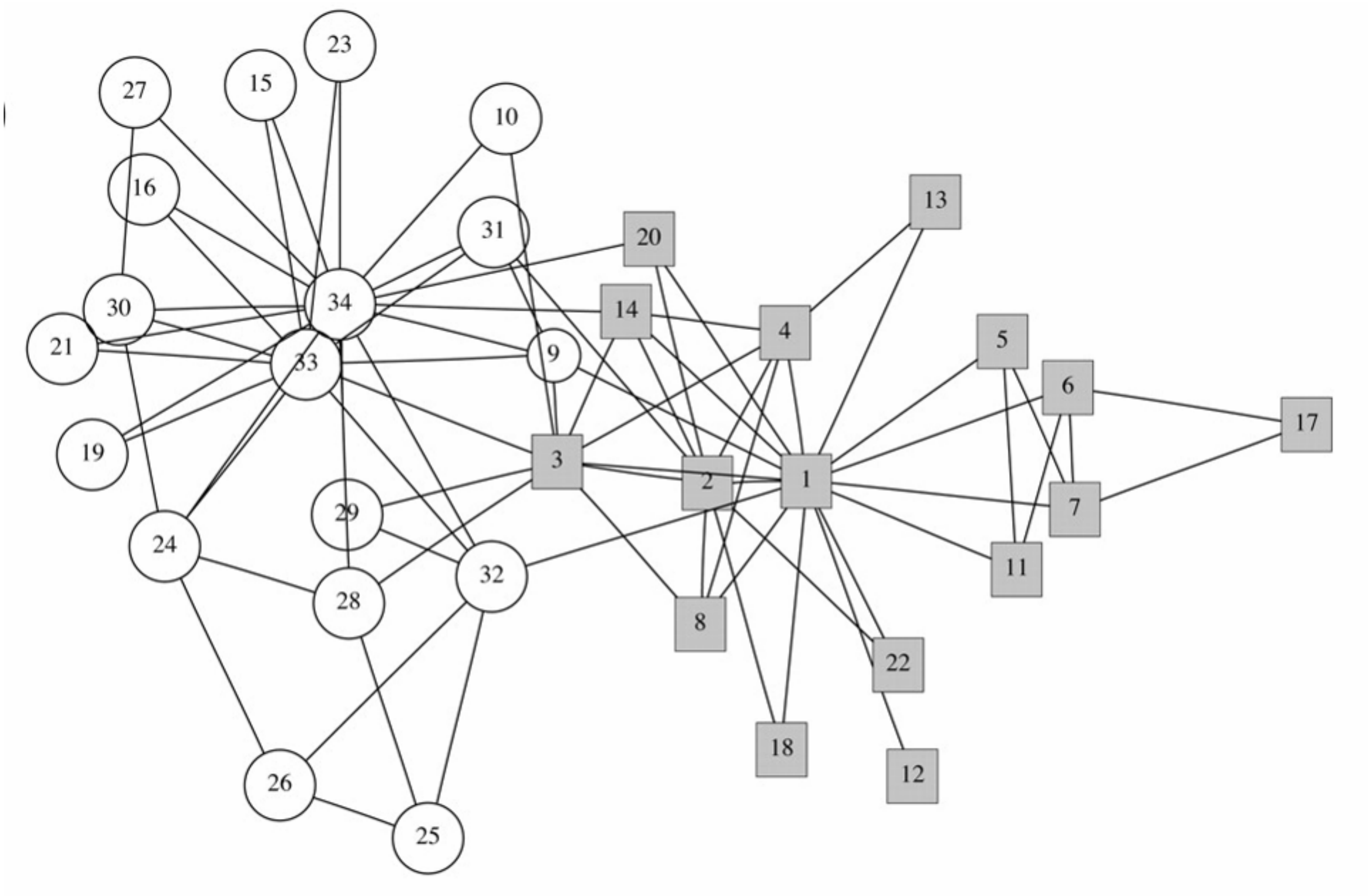
N(v) are input, output or all neighbours of vertex v;
You can include vertex v to its own neighbourhood or not and display in report window only upper triangle / undirected or complete matrix /directed (if number of vertices is low).
+ stands for symmetric sum, U stands for union and \ stands for difference;
| stands for set cardinality; 1st and 2nd maxdegree are largest degree and second largest degree in network, recpectivelly.

```
d1(u,v)  = | N(u)  + N(v)  | /  (1st maxdegree + 2nd maxdegree)

            | N(u)  + N(v)  |
d2(u,v)  = -----------------
            | N(u)  U N(v)  |

            | N(u)  + N(v)  |
d3(u,v)  = -----------------
            |N(u)|  + |N(v)|

             max(|N(u)\N(v)|,|N(v)\N(u)|)
d4(u,v)  = ------------------------------
                  max(|N(u)|,|N(v|)

d5(u,v)= Corrected Euclidean like dissimilarity

d6(u,v)= Corrected Manhattan like dissimilarity
```

the “+” denotes an XOR, the nodes that are either in N(u) or N(v) but not in both

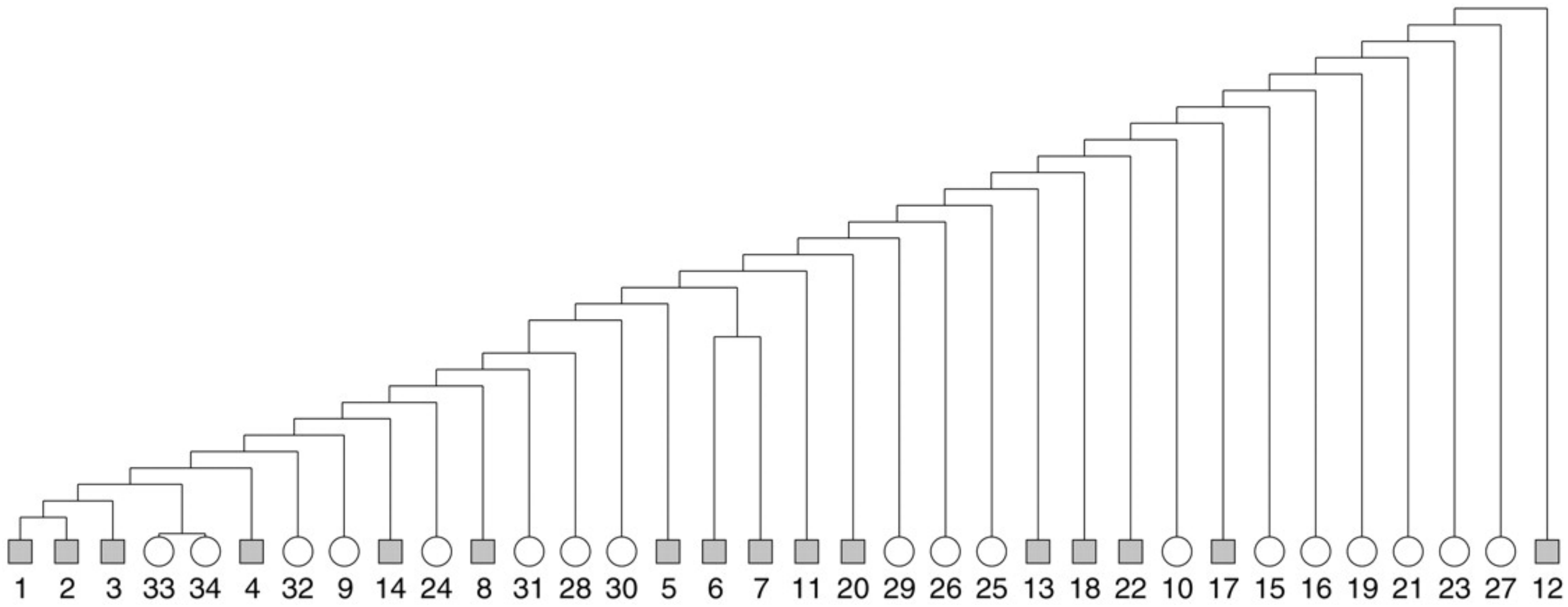
Source: Pajek Manual - http://vlado.fmf.uni-lj.si/pub/networks/pajek/doc/pajekman.pdf

Hierarchical clustering: Zachary Karate Club



source: Girvan and Newman, PNAS June 11, 2002 99(12):7821-7826

Is hierarchical clustering really this bad?



Zachary karate club data hierarchical clustering tree using edge-independent path counts

source: Girvan and Newman, PNAS June 11, 2002 99(12):7821-7826

step by step

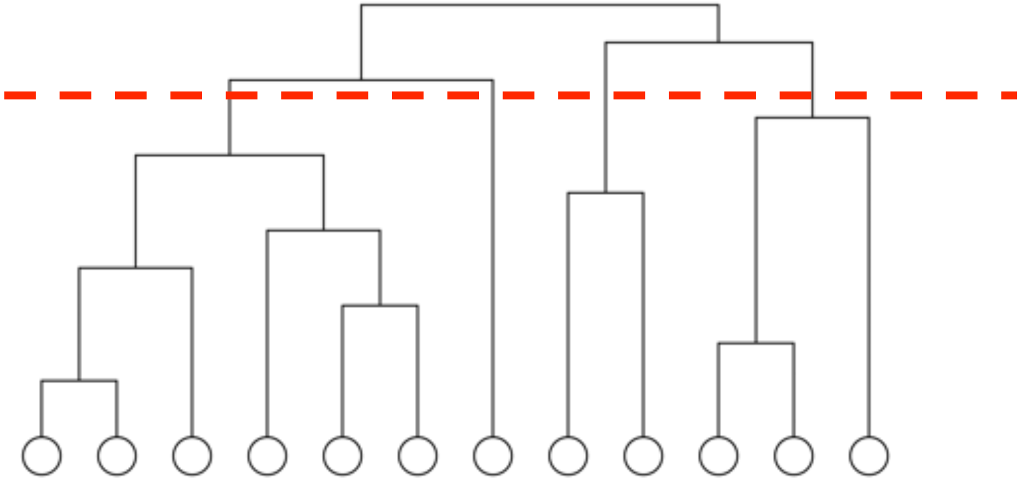
- load the file zachary.net
- create a complete cluster Operations-> Dissimilarity > d1/All
- save the dendrogram as an EPS (Pajek will prompt you after computing the dissimilarity matrix)

step by step (continued)

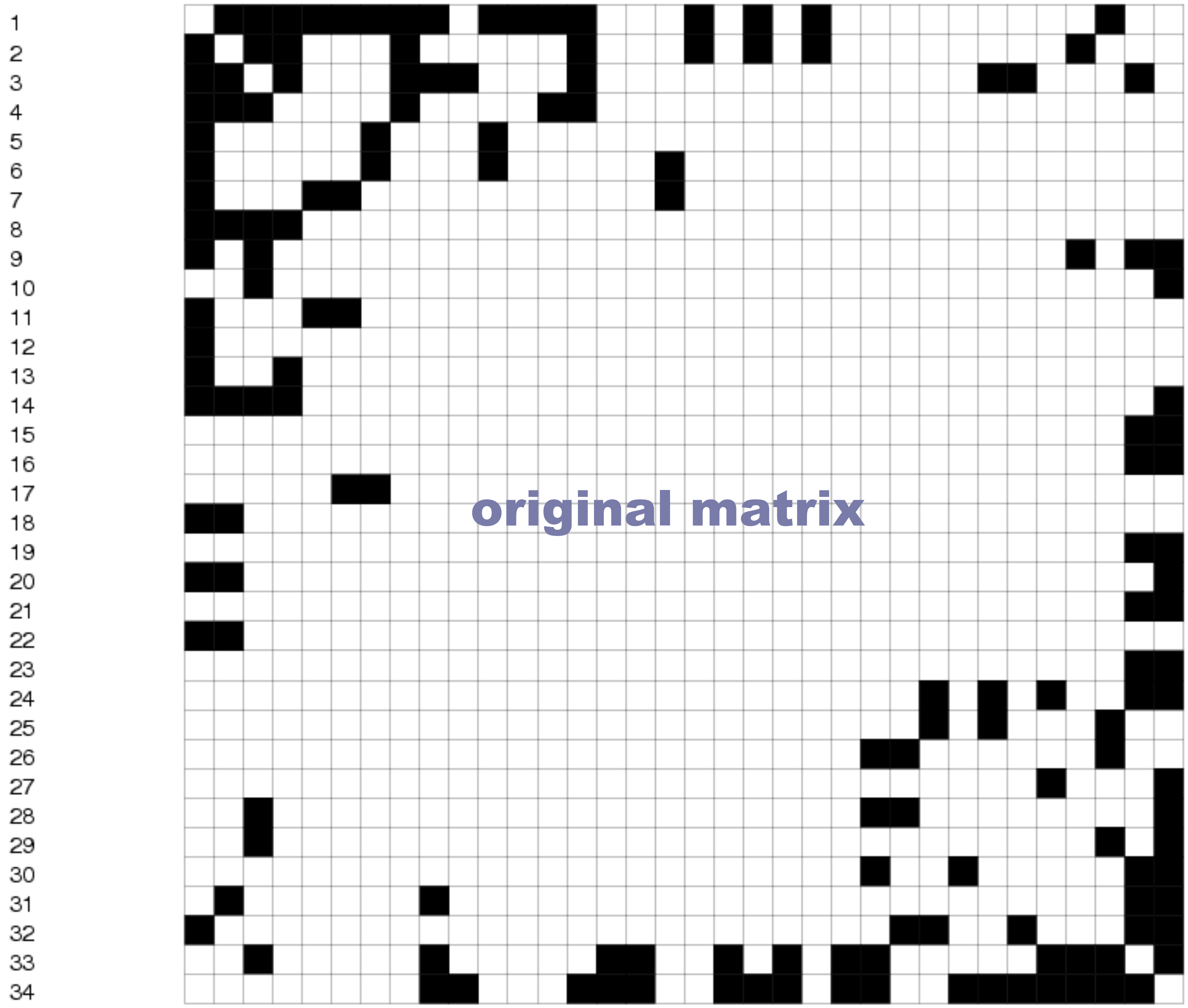
- save the matrix as an EPS (make sure you have the original, rather than the distance matrix selected)
- File > Network > Export matrix to EPS > Using permutation
- open the EPS files in ghostview, or illustrator, etc.
  - on the Mac EPS be converted to PDF by Adobe Distiller

Hierarchical clustering

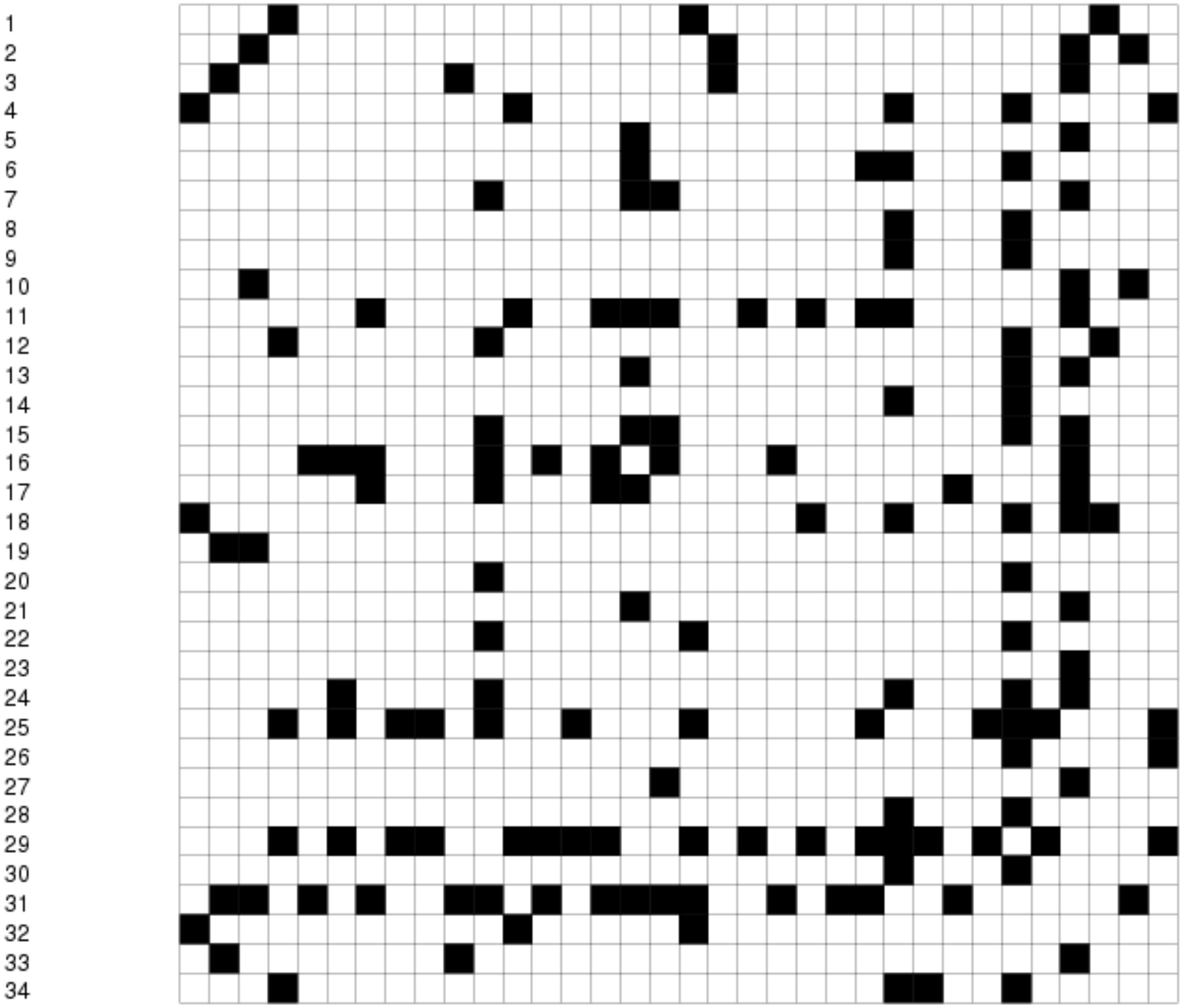
- result: nested components, where one can take a ‘slice’ at any level of the tree



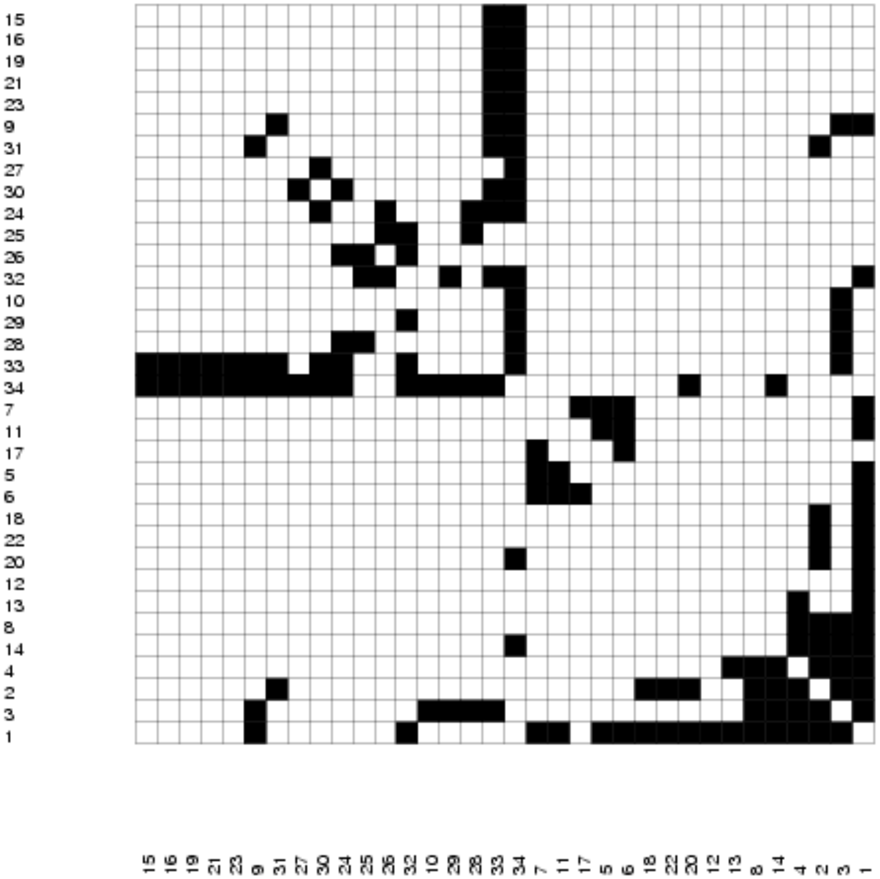
source: Girvan and Newman, PNAS June 11, 2002 99(12):7821-7826


original matrix
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34

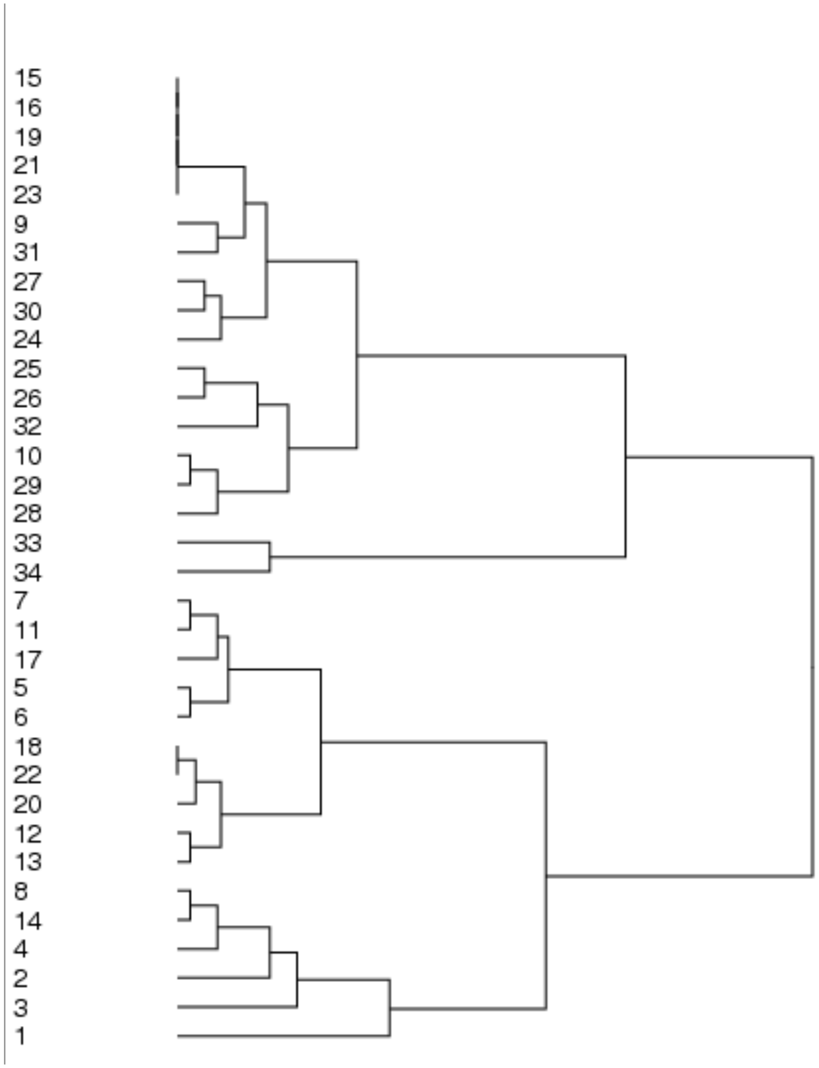
randomized karate club matrix



permuted matrix



dendrogram

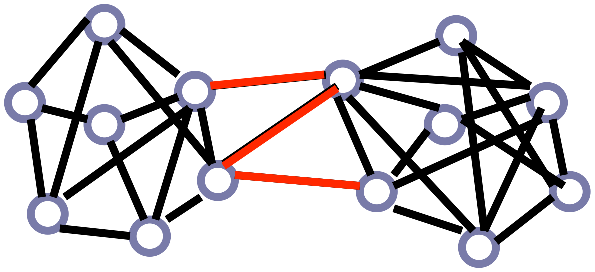

15
16
19
21
23
9
31
27
30
24
25
26
32
10
29
28
33
34
7
11
17
5
6
18
22
20
12
13
8
14
4
2
3
1

Girvan & Newman: betweenness clustering

- Algorithm
  - compute the betweenness of all edges
  - while (betweenness of any edge > threshold):
    - remove edge with highest betweenness
    - recalculate betweenness

- Betweenness needs to be recalculated at each step
  - removal of an edge can impact the betweenness of another edge
  - very expensive: all pairs shortest path – $O(N^3)$
  - may need to repeat up to N times
  - does not scale to more than a few hundred nodes, even with the fastest algorithms

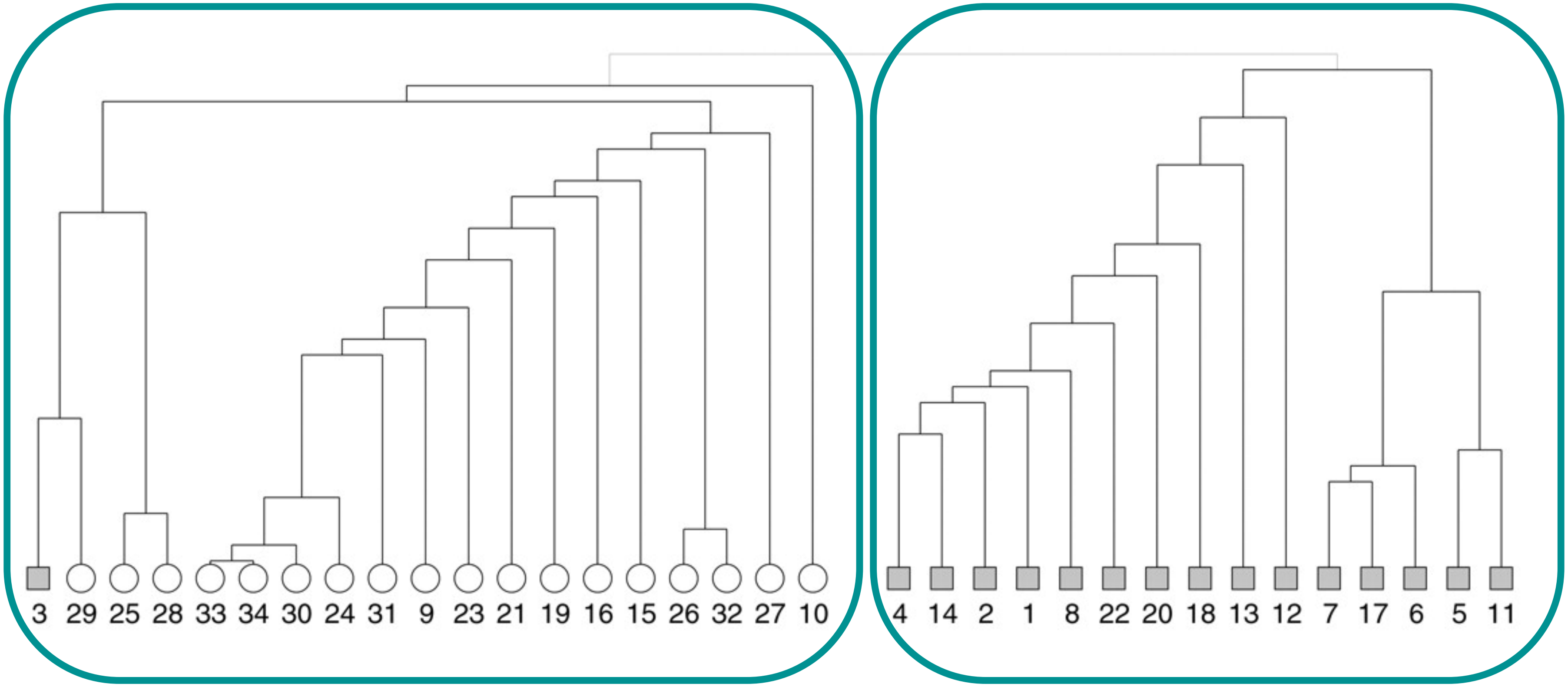
betweenness clustering algorithm



Step by step

- Run Guess
- Open the GDF zacharykarate.gdf
- Run the script betweennessclustering.py
  - File > Run Script ….
  - Click on “remove edge” to remove one edge at a time
  - Click on “next breakup” to remove edges until you separate a community

betweenness clustering algorithm & the karate club data set



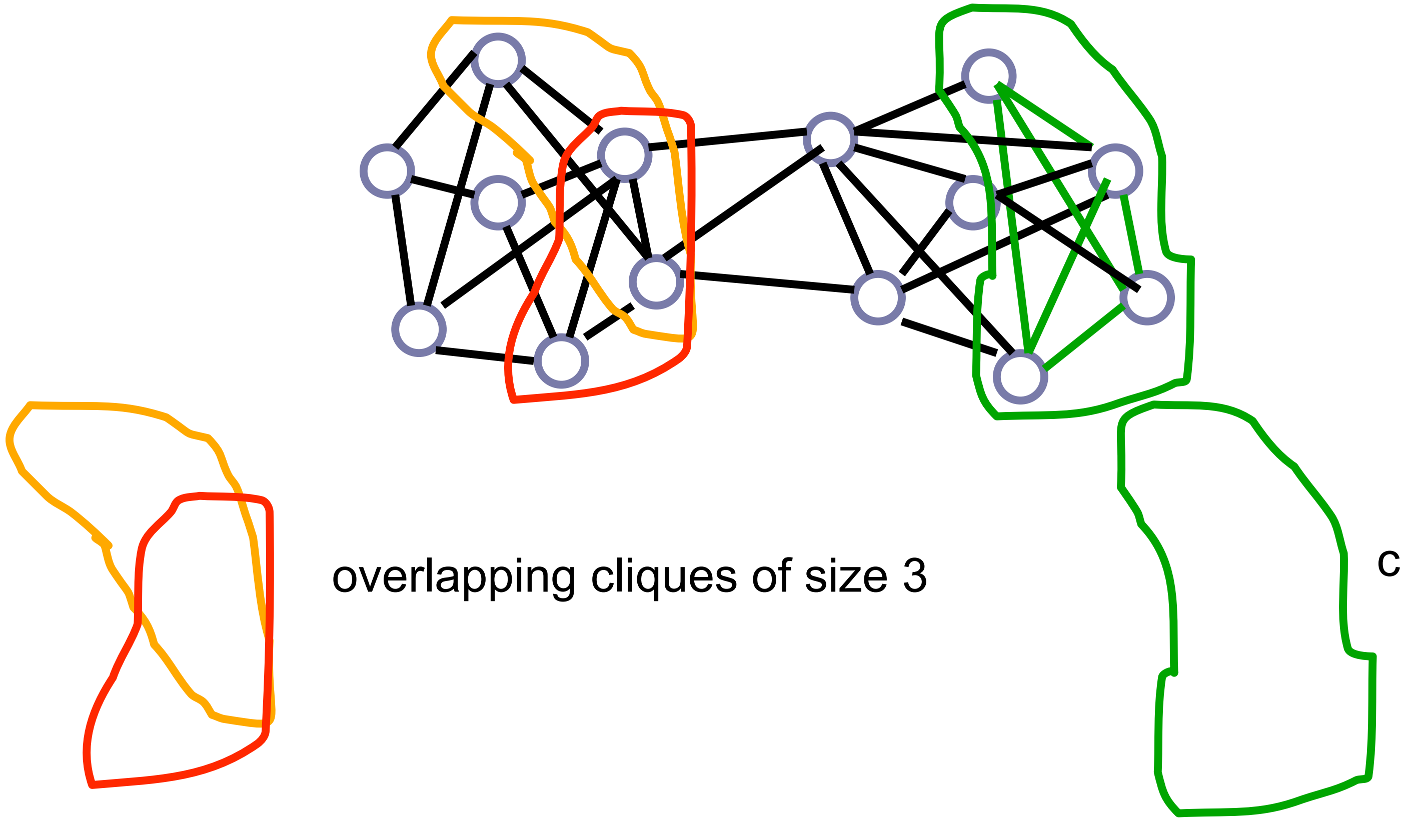
source: Girvan and Newman, PNAS June 11, 2002 99(12):7821-7826

What general properties indicate cohesion?

- mutuality of ties
  - everybody in the group knows everybody else
- closeness or reachability of subgroup members
  - individuals are separated by at most n hops
- frequency of ties among members
  - everybody in the group has links to at least k others in the group
- relative frequency of ties among subgroup members compared to nonmembers

Cliques

- Every member of the group has links to every other member
- Cliques can overlap



overlapping cliques of size 3

clique of size 4

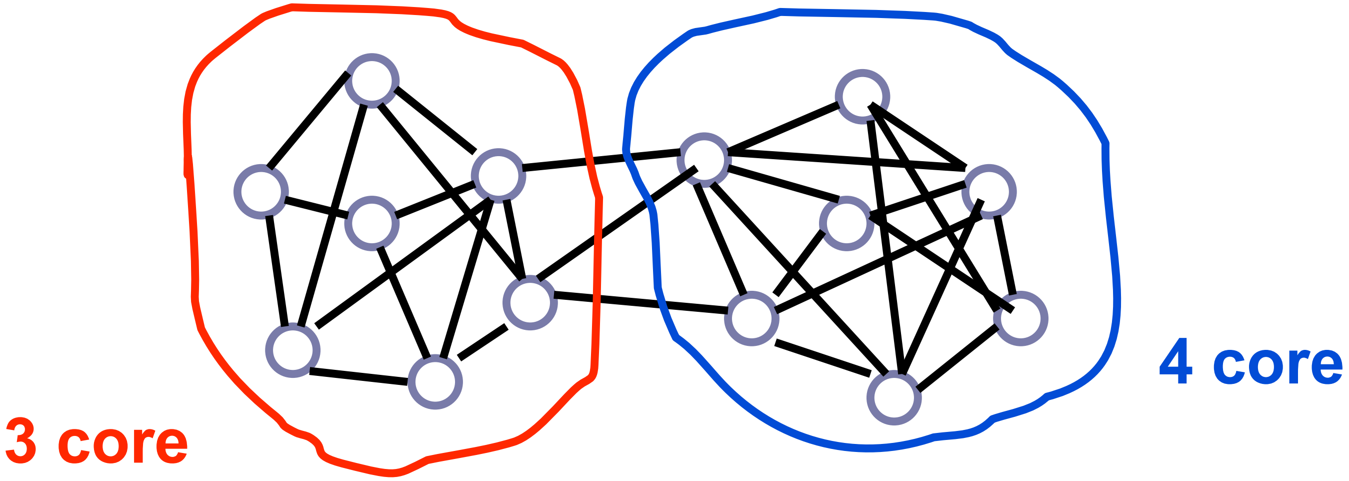
Considerations in using cliques as subgroups

- Not robust
  - one missing link can disqualify a clique
- Not interesting
  - everybody is connected to everybody else
  - no core-periphery structure
  - no centrality measures apply
- How cliques overlap can be more interesting than that they exist

- Pajek
  - just as for motifs:
    - construct a network that is a clique of the desired size
    - Nets>Fragment (1 in 2)>Find

a less stingy definition of cohesive subgroups: k cores

- Each node within a group is connected to k other nodes in the group

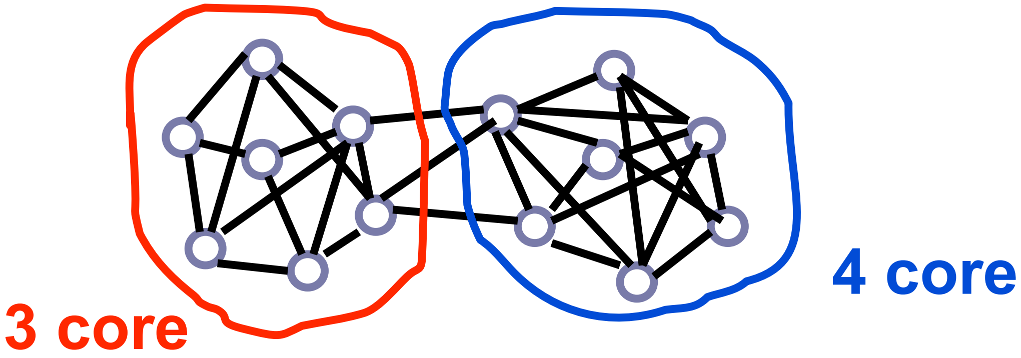


Pajek: Net>Partitions>Core>Input,Output,All

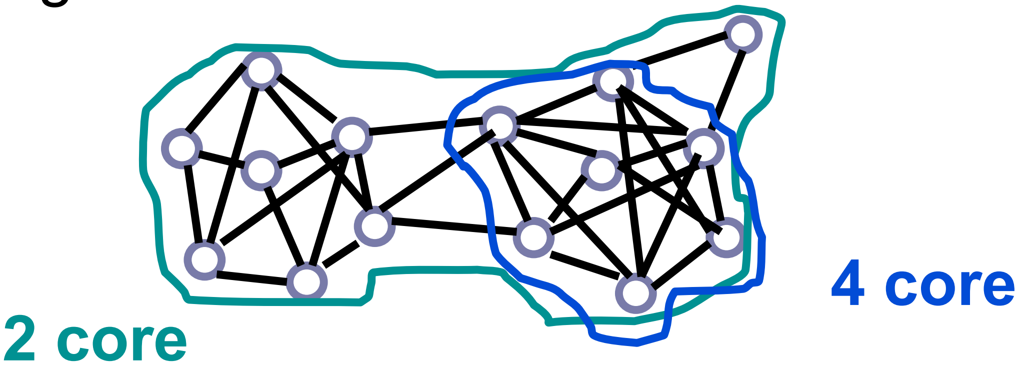
Assigns each vertex to the largest k-core it belongs to

k-cores

- Each node within a group is connected to k other nodes in the group

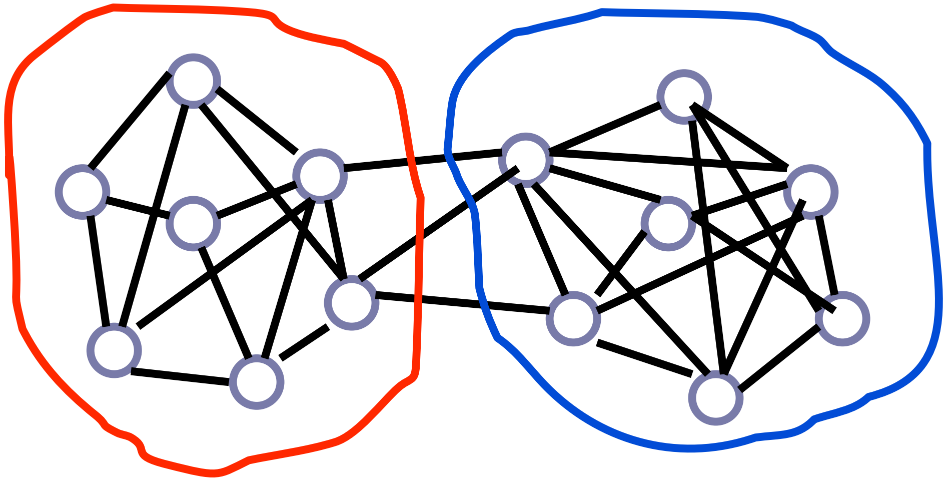


- but even this is too stringent of a requirement for identifying natural communities



subgroups based on reachability and diameter

- n – cliques
  - maximal distance between any two nodes in subgroup is n

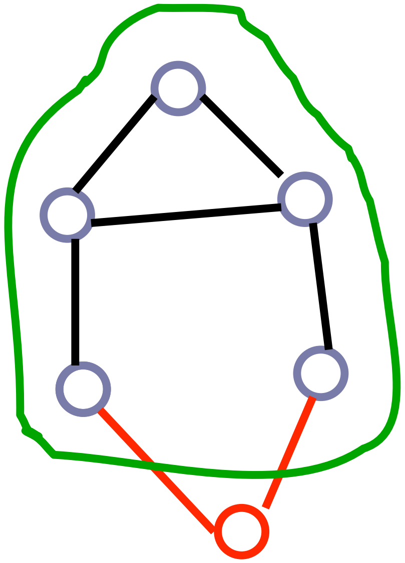


2-cliques

- theoretical justification
  - information flow through intermediaries

considerations with n-cliques

- problem
  - diameter may be greater than n
  - n-clique may be disconnected (paths go through nodes not in subgroup)



2 – clique

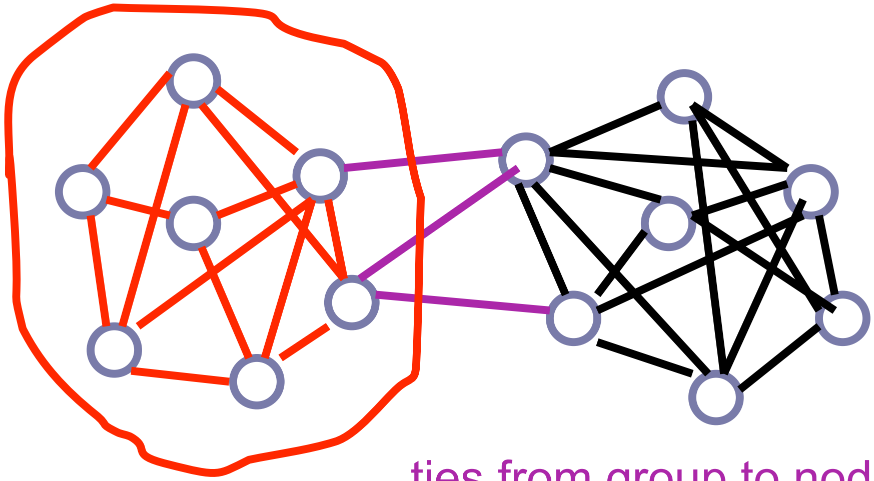
diameter = 3

path outside the 2-clique

- fix
  - n-club: maximal subgraph of diameter 2

p-cliques: frequency of in group ties

- partition the network into clusters where vertices have at least a proportion p (number between 0 and 1) of neighbors inside the cluster.



within-group ties

ties from group to nodes external to the group

Pajek:

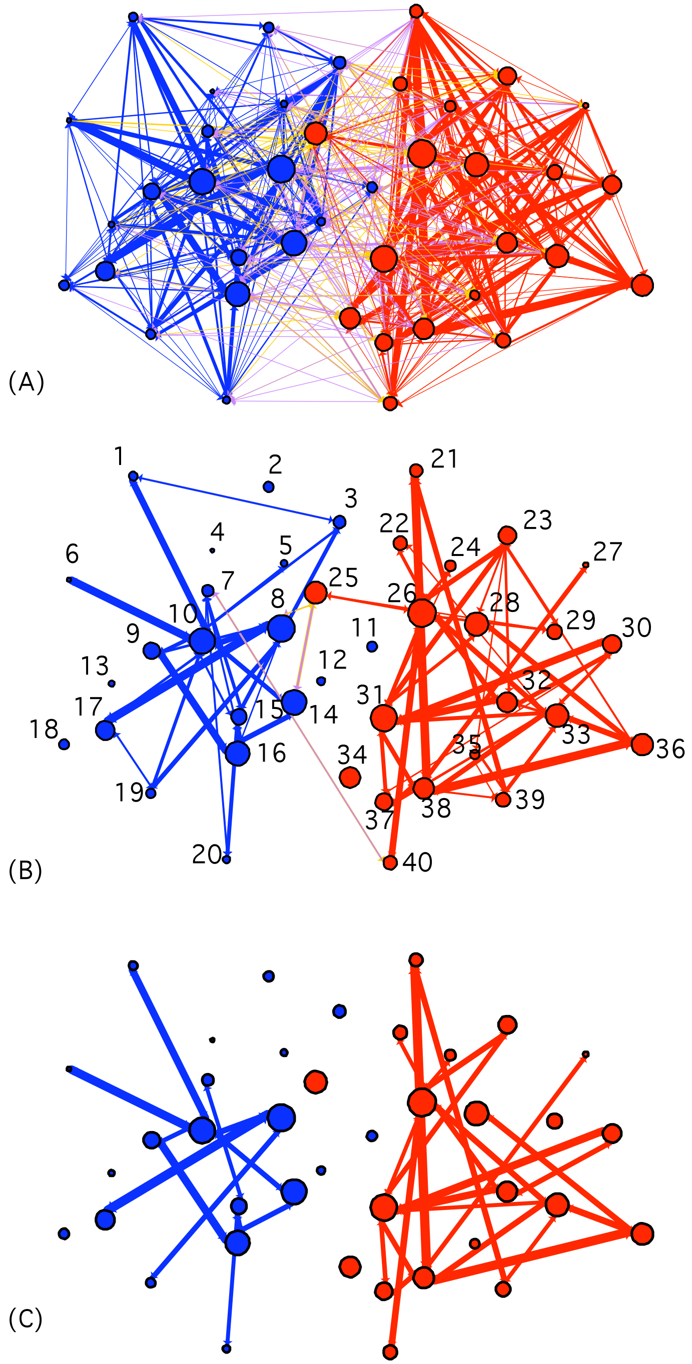
Net > Partition > p-Cliques…

Has the problem already discussed – can have high p if many or all vertices belong to one big cluster

cohesion in directed and weighted networks

- something we’ve already learned how to do:
  - find strongly connected components

- keep only a subset of ties before finding connected components
  - reciprocal ties
  - edge weight above a threshold



1 Digbys Blog
2 James Walcott
3 Pandagon
4 blog.johnkerry.com
5 Oliver Willis
6 America Blog
7 Crooked Timber
8 Daily Kos
9 American Prospect
10 Eschaton
11 Wonkette
12 Talk Left
13 Political Wire
14 Talking Points Memo
15 Matthew Yglesias
16 Washington Monthly
17 MyDD
18 Juan Cole
19 Left Coaster
20 Bradford DeLong

21 alwaReport
22 Voka Pundit
23 Roger L Simon
24 Tim Blair
25 Andrew Sullivan
26 Instapundit
27 Blogs for Bush
28 Little Green Footballs
29 Belmont Club
30 Captain's Quarters
31 Powerline
32 Hugh Hewitt
33 INDC Journal
34 Real Clear Politics
35 Winds of Change
36 Allahpundit
37 Michelle Malkin
38 WizBang
39 Dean's World
40 Volokh

Example: political blogs

(Aug 29th – Nov 15th, 2004)

A) all citations between A-list blogs in 2 months preceding the 2004 election

B) citations between A-list blogs with at least 5 citations in both directions

C) edges further limited to those exceeding 25 combined citations

only 15% of the citations bridge communities

source: Adamic & Glance, LinkKDD2005

Other reasons to care

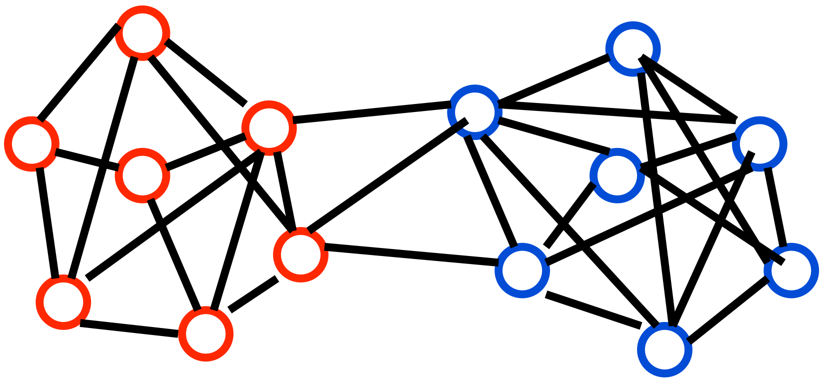
- Discover communities of practice

- Measure isolation of groups

- Threshold processes:
  - I will adopt an innovation if some number of my contacts do
  - I will vote for a measure if a fraction of my contacts do

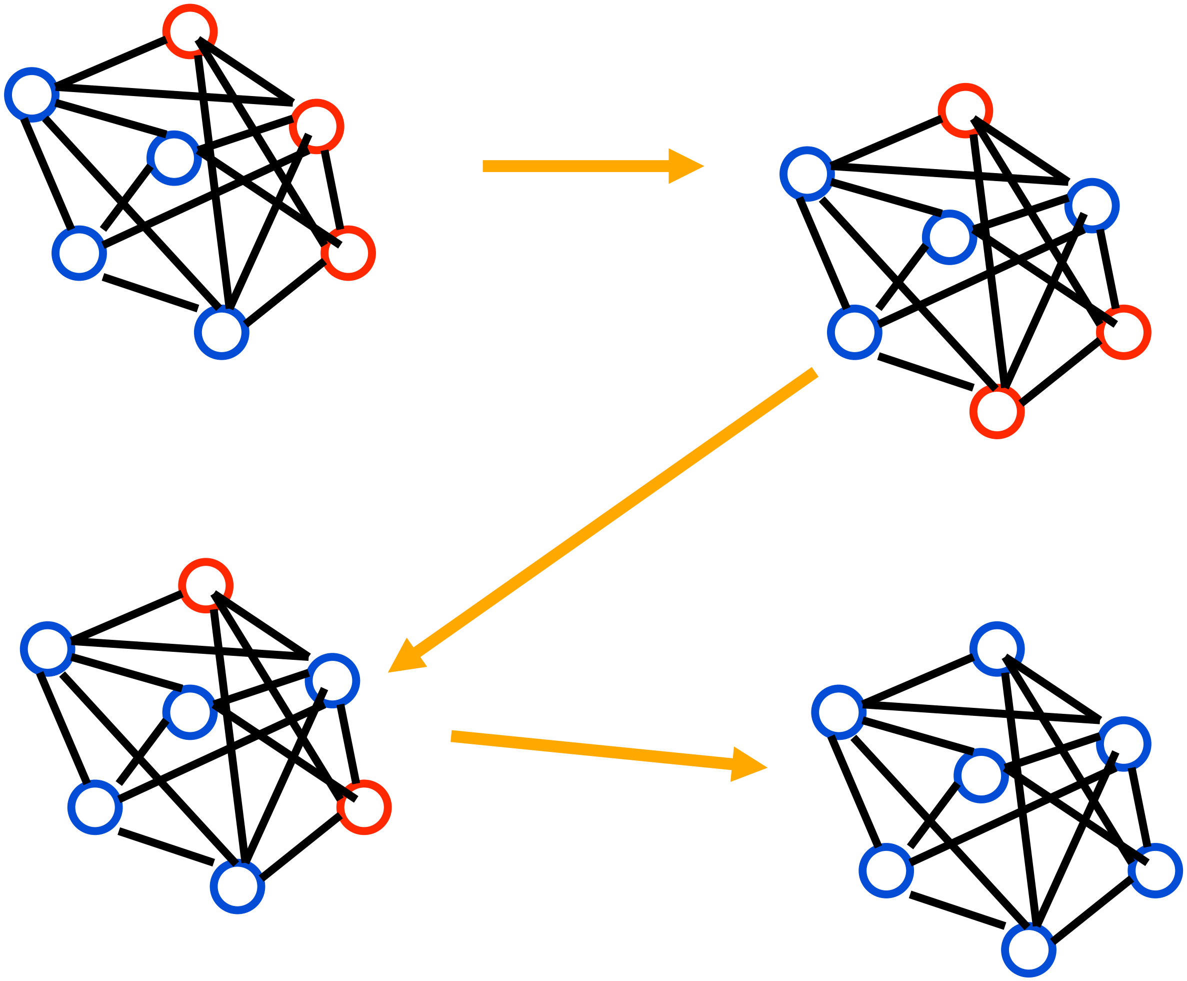
Why care about group cohesion?

- opinion formation and uniformity



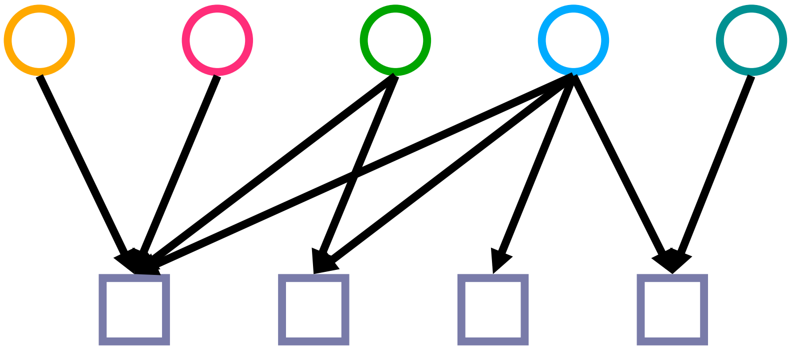
- if each node adopts the opinion of the majority of its neighbors, it is possible to have different opinions in different cohesive subgroups

within a cohesive subgroup – greater uniformity



Affiliation networks

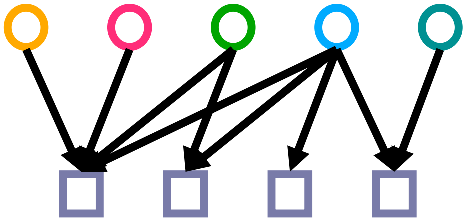
- otherwise known as
  - membership network
    - e.g. board of directors
  - hypernetwork or hypergraph
  - bipartite graphs
  - interlocks

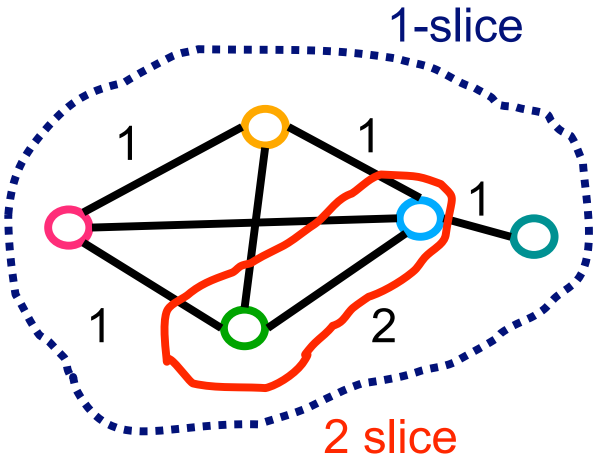


m-slices

- transform to a one-mode network
- weights of edges correspond to number of affiliations in common
- m-slice: maximal subnetwork containing the lines with a multiplicity equal to or greater than m

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 2 & 2 & 0 \\ 1 & 1 & 2 & 4 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$





Pajek:

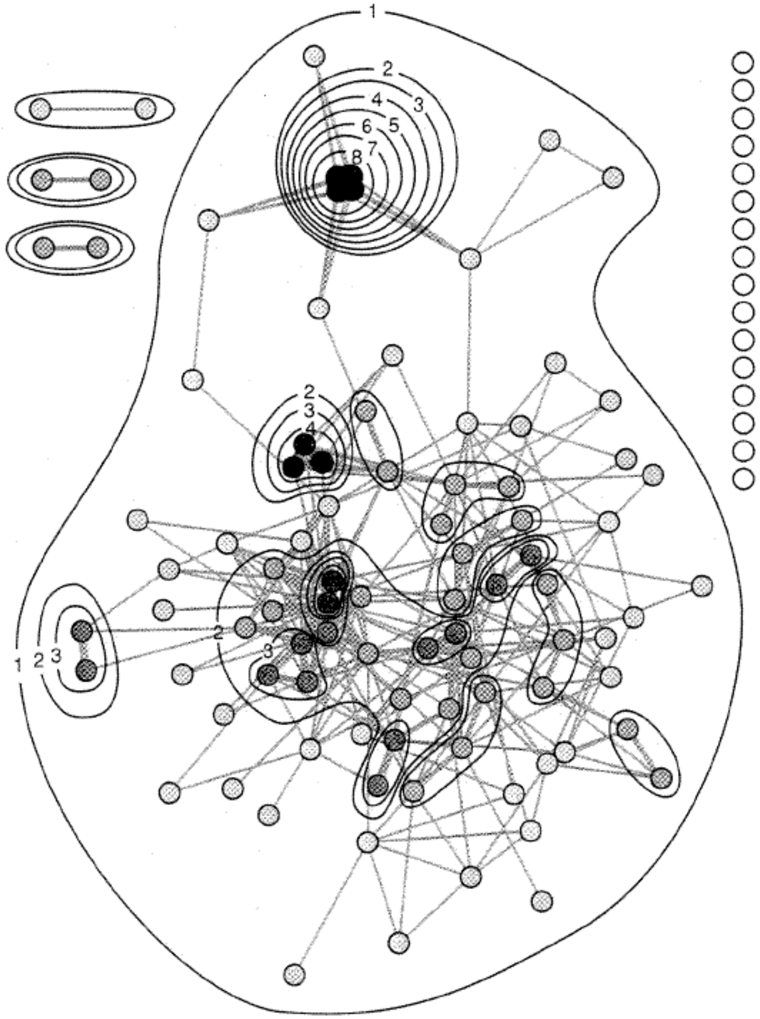


Figure 53. *m*-Slices in the network of Scottish firms, 1904–5 (contours added manually).

File > Pajek Project
File > Scotland.paj

Net>Transform>2-Mode to 1-Mode> Include Loops, Multiple Lines

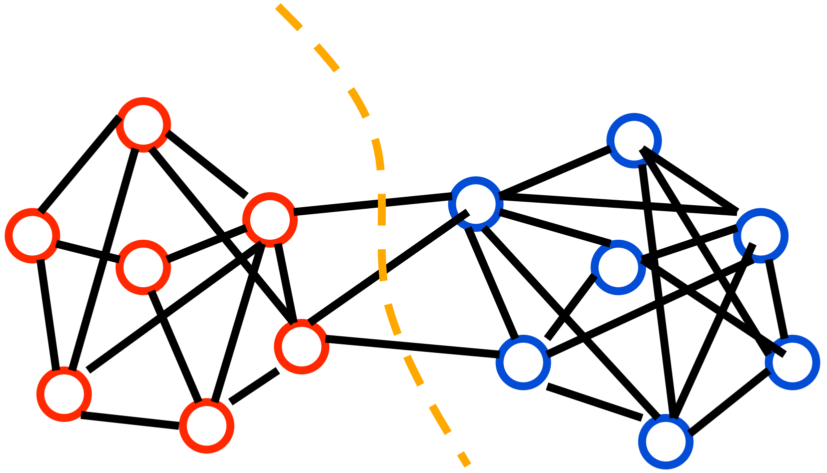
Info>Network>Line Values (to view)

Net>Partitions>Valued Core>First threshold and step

source: de Nooy et al., Exploratory Social Network Analysis with Pajek, Cambridge U. Press, 2005.

Community finding vs. other approaches

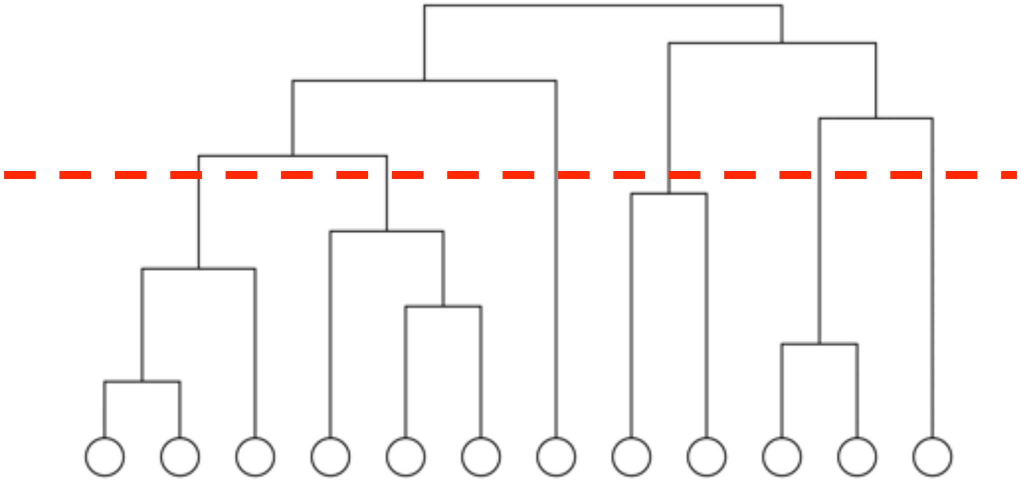
- Social and other networks have a natural community structure
- We want to discover this structure rather than impose a certain size of community or fix the number of communities



- Without “looking”, can we discover community structure in an automated way?

Hierarchical clustering

- Process:
  - after calculating the “distances” for all pairs of vertices
  - start with all n vertices disconnected
  - add edges between pairs one by one in order of decreasing weight
  - result: nested components, where one can take a ‘slice’ at any level of the tree



Hierarchical clustering in Pajek
http://mrvar.fdv.uni-lj.si/sola/info4/nusa/doc/block1.pdf

- Procedure
  - generate a complete cluster using Cluster->Create Complete Cluster
  - compute the dissimilarity matrix
    - run Operations->Dissimilarity
      - select “d1/All” to consider network as a binary matrix
      - select “Corrected Euclidean” or “Corrected Manhattan” distance for valued networks

Hierarchical clustering in Pajek
http://mrvar.fdv.uni-lj.si/sola/info4/nusa/doc/block1.pdf

- Procedure (continued)
  - the above will use the dissimilarity matrix to hierarchically cluster nodes and output
    - a dissimilarity matrix
    - EPS picture of the dendrogram
    - permutation of vertices according to the dendrogram
    - hierarchy representing hierarchical clustering
      - to visualize:
        - Edit->Show Subtree
        - Select nodes (Edit->Change Type or Ctrl+T)
        - transform the hierarchy into a partition (Hierarchy->Make Partition)

Finding community structure in very large networks

Authors: Aaron Clauset, M. E. J. Newman, Cristopher Moore

2004

- Consider edges that fall within a community or between a community and the rest of the network
- Define modularity:

if vertices are in the same community

$$Q = \frac{1}{2m} \sum_{vw} \left[A_{vw} - \frac{k_v k_w}{2m} \right] \delta(c_v, c_w)$$

adjacency matrix

probability of an edge between two vertices is proportional to their degrees

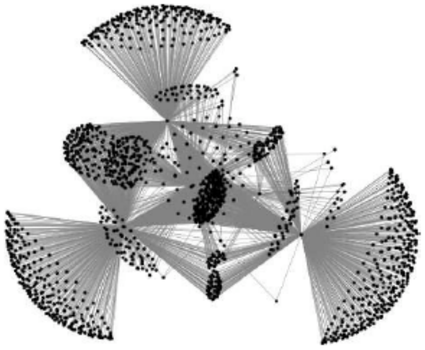
- For a random network, Q = 0
  - the number of edges within a community is no different from what you would expect

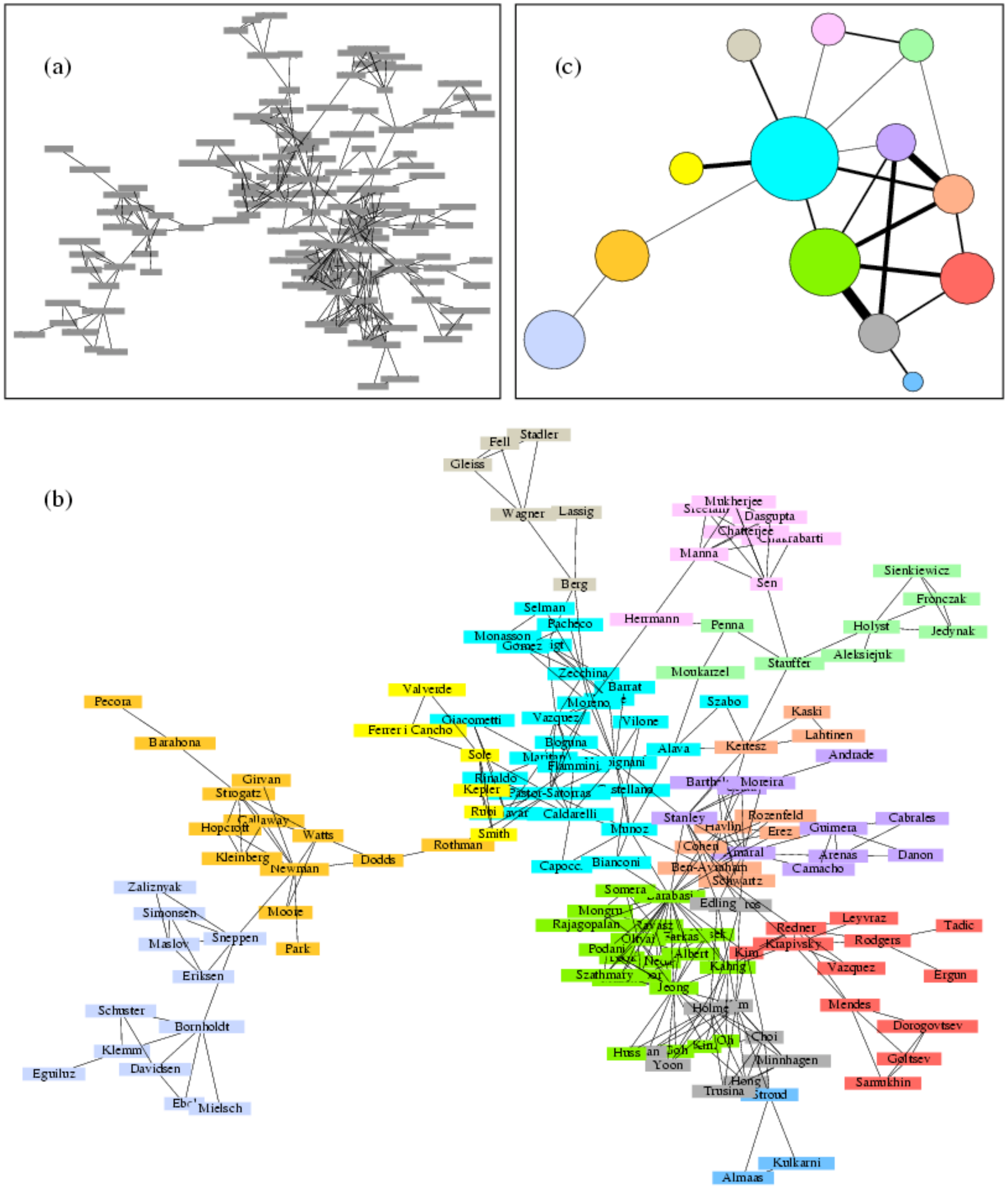
Finding community structure in very large networks

Authors: Aaron Clauset, M. E. J. Newman, Cristopher Moore

2004

- Algorithm
  - start with all vertices as isolates
  - follow a greedy strategy:
    - successively join clusters with the greatest increase ΔQ in modularity
    - stop when the maximum possible ΔQ <= 0 from joining any two
  - successfully used to find community structure in a graph with > 400,000 nodes with > 2 million edges
    - Amazon’s people who bought this also bought that…
  - alternatives to achieving optimum ΔQ:
    - simulated annealing rather than greedy search



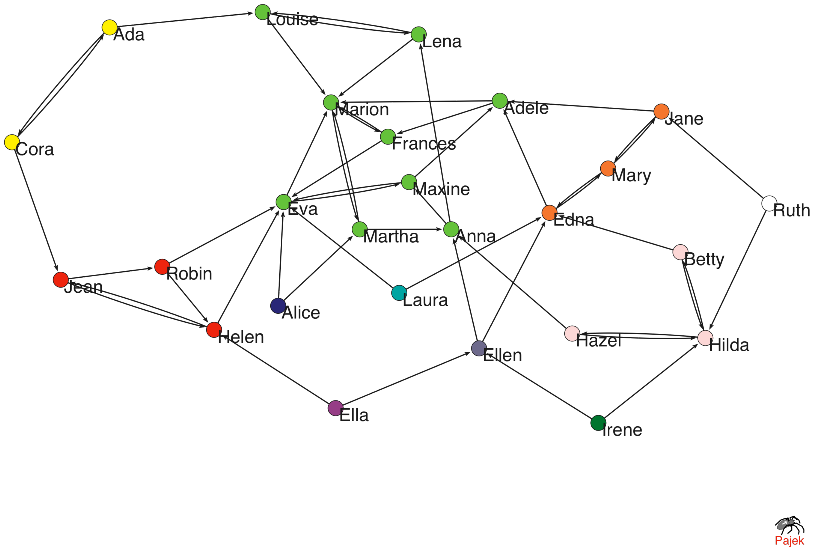


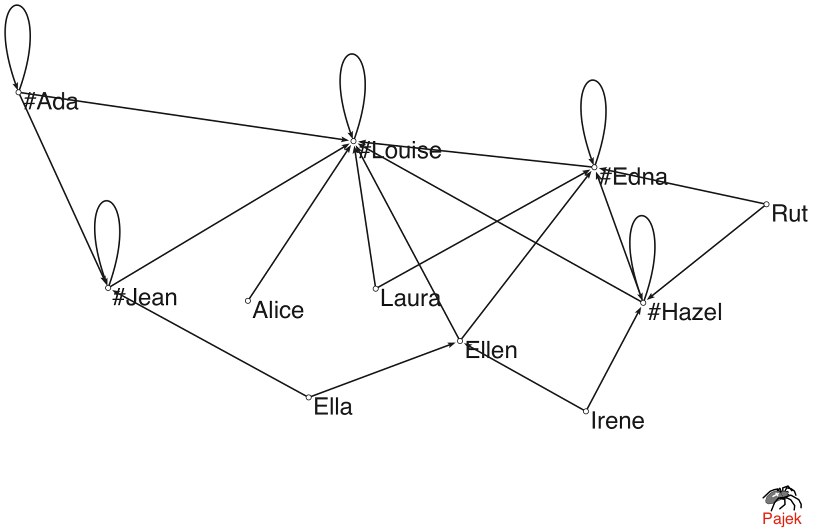
Reminder of how modularity can help us visualize large networks

source: M. E. J. Newman and M. Girvan, Finding and evaluating community structure in networks, Physical Review E 69, 026113 (2004).

network of components in pajek

- open dining.net (dining table partners data file)
- Net > Components > Strong
- Operations > Shrink network > Partition





lab wrap up

- What you’ve learned today
  - motif analysis – what is the micro structure of your network?
  - hierarchical clustering
    - what are the underlying communities in your network?
  - betweenness community finding
    - cohesive subcommunities
  - k-cores, k-cliques, m-cores
    - Pajek methods for discovering underlying cohesive subgroups
  - modularity-based clustering (download on your own or use igraph)