Community detection

LEARNING OUTCOMES

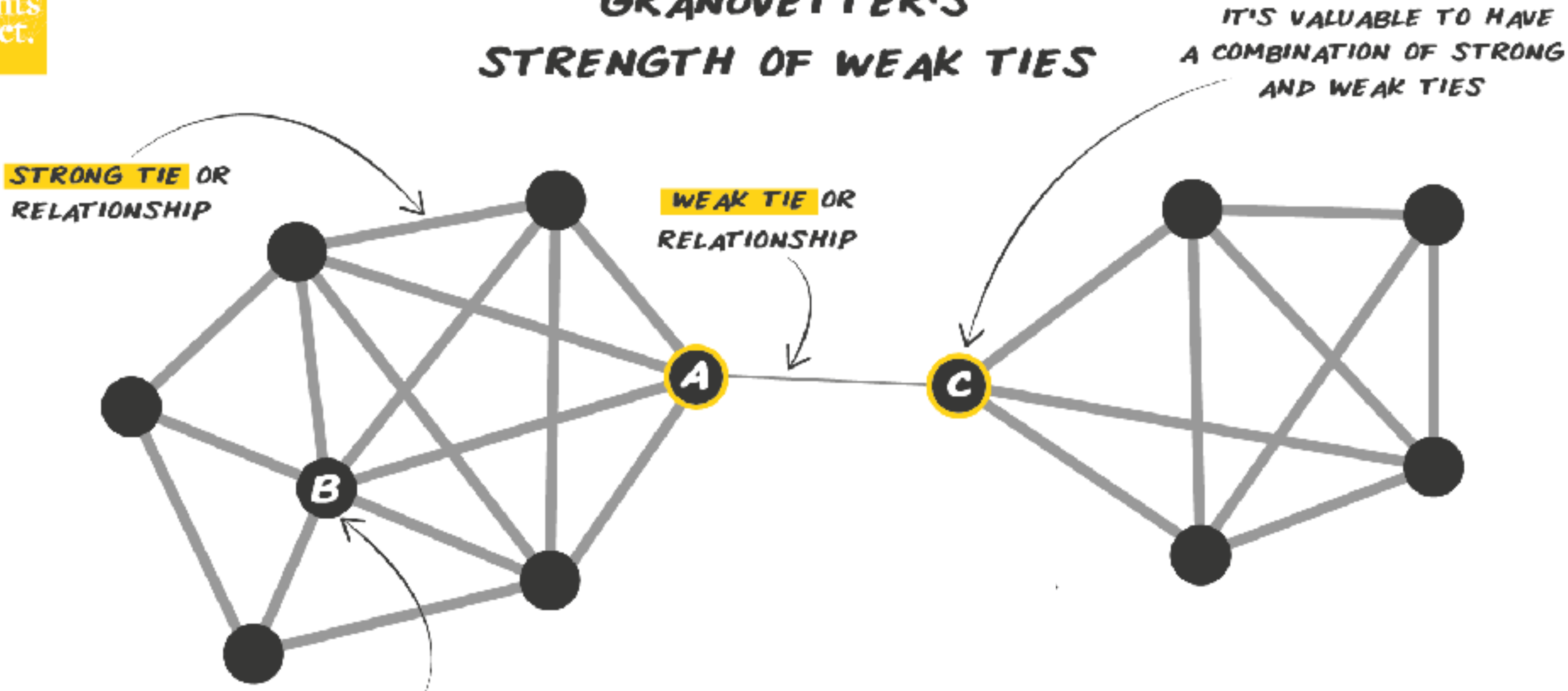
UNDERSTAND WHAT **COMMUNITIES** ARE

BE ABLE TO **DESCRIBE A NETWORK** IN TERMS OF COMMUNITIES

LEARN DIFFERENT TYPES OF COMMUNITY **CLASSIFICATIONS**

The Reliants Project.

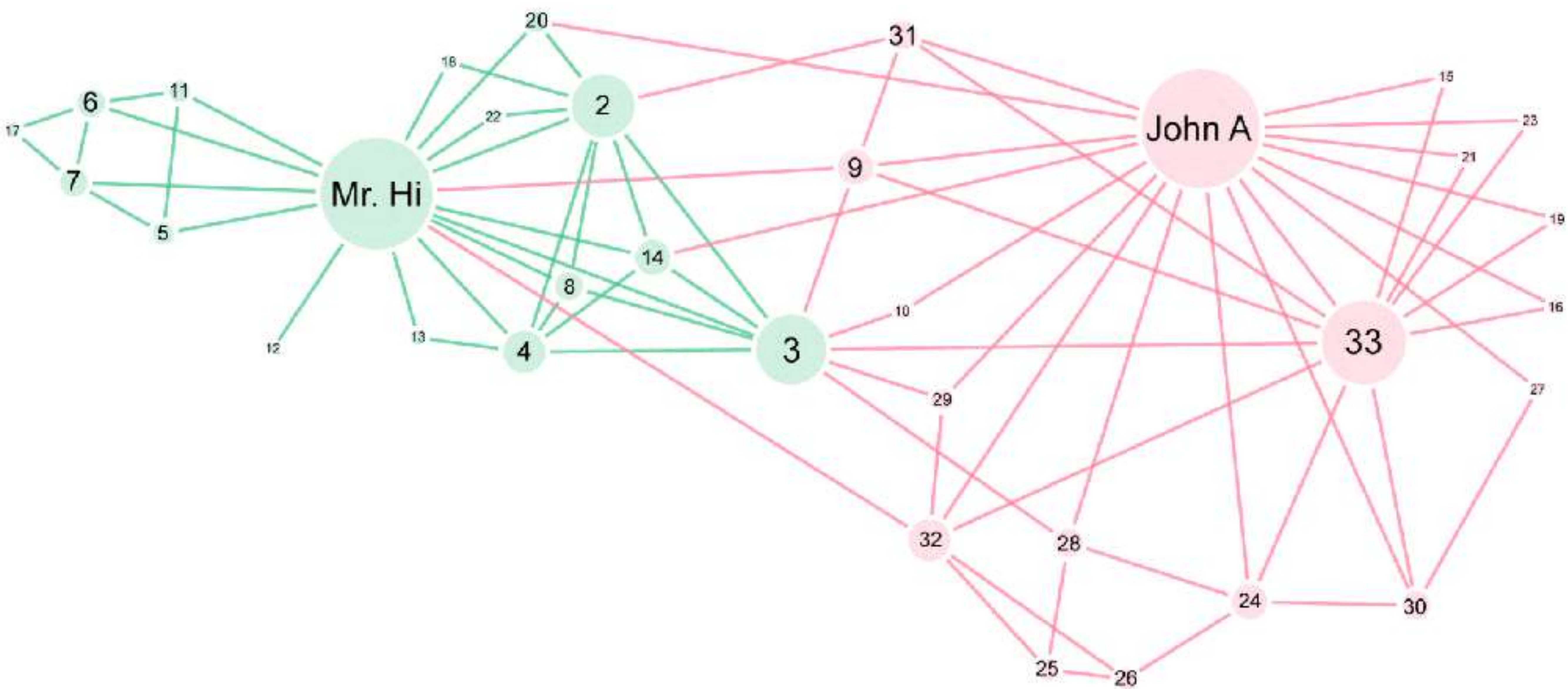
GRANOVETTER'S STRENGTH OF WEAK TIES

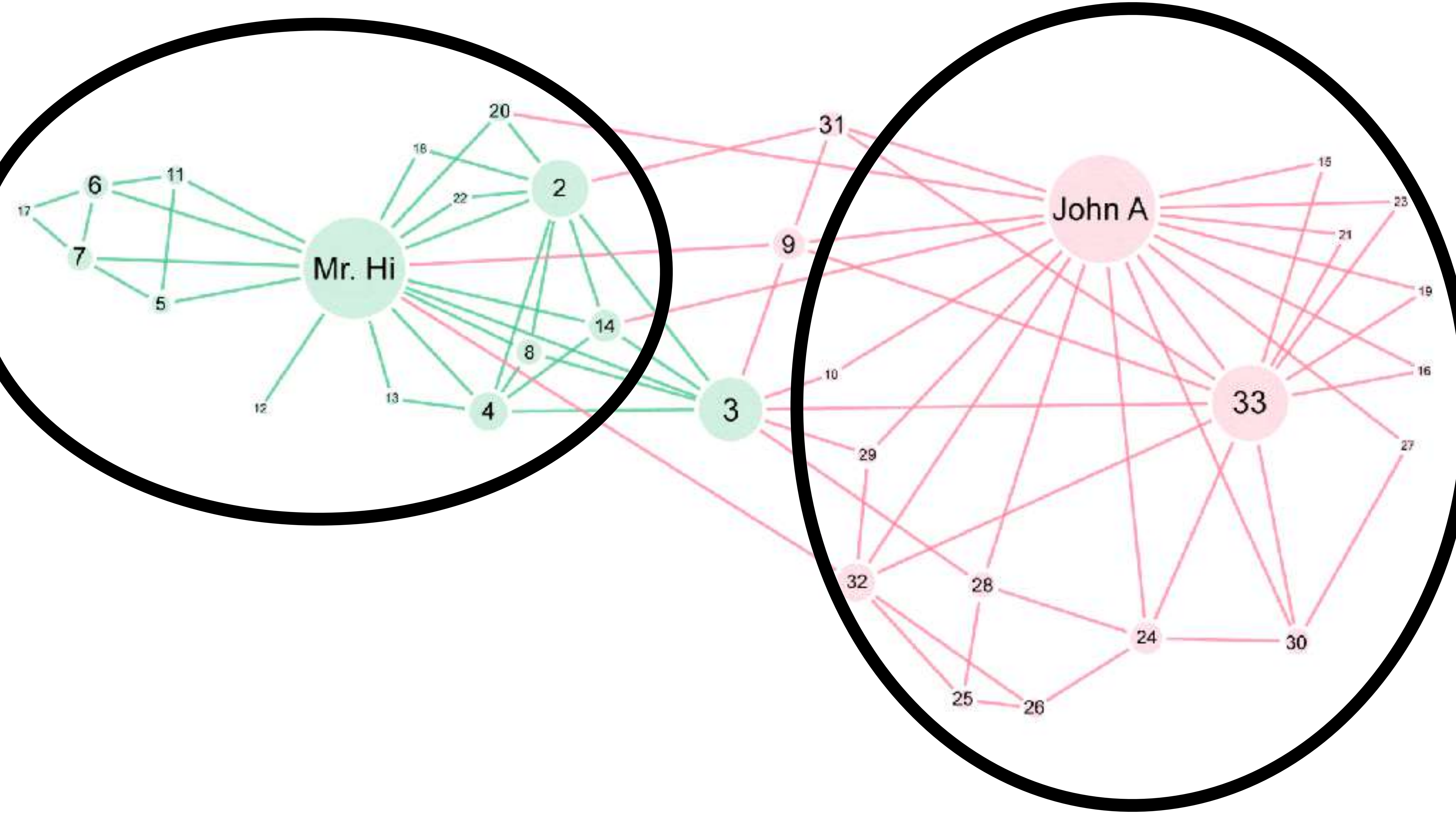


EVEN THOUGH B HAS MORE TIES THAN A, ALL THOSE TIES LIKELY HAVE THE SAME INFORMATION BECAUSE THEY ALL KNOW EACH OTHER WELL

FOR EXAMPLE, A CAN SHARE INFORMATION WITH C THAT C WOULDN'T GET FROM ANYONE ELSE IN THEIR GROUP, AND VICE VERSA.


ALL VALLEY KARATE CHAMPIONSHIP
SEMI-FINALS
FINALS


Mr. Hi
John A
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


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

DEFINITIONS

INTERNAL AND EXTERNAL DEGREE:
THE NUMBER OF NEIGHBOURS INSIDE AND OUTSIDE THE COMMUNITY

DEFINITIONS

INTERNAL AND EXTERNAL DEGREE:
THE NUMBER OF NEIGHBOURS INSIDE AND OUTSIDE THE COMMUNITY

$$k_i = k_i^{int} + k_i^{ext}$$

i is called internal node of community **c** if

$$k_i^{ext} = 0 \ \textbf{And} \ \ k_i^{int} > 0$$

i is called boundary node of community **c** if

$$k_i^{ext} > 0 \ \textbf{And} \ \ k_i^{int} > 0$$

DEFINITIONS

INTERNAL AND EXTERNAL DEGREE:

THE NUMBER OF NEIGHBOURS INSIDE AND OUTSIDE THE COMMUNITY

DEFINITIONS

INTERNAL AND EXTERNAL DEGREE:
THE NUMBER OF NEIGHBOURS INSIDE AND OUTSIDE THE COMMUNITY

NUMBER OF INTERNAL LINKS:
THE NUMBER OF LINKS BETWEEN NODES WITHIN THE COMMUNITY

DEFINITIONS

INTERNAL AND EXTERNAL DEGREE:
THE NUMBER OF NEIGHBOURS INSIDE AND OUTSIDE THE COMMUNITY

NUMBER OF INTERNAL LINKS:
THE NUMBER OF LINKS BETWEEN NODES WITHIN THE COMMUNITY

COMMUNITY DEGREE:
THE SUM OF DEGREE OF ALL THE NODES IN THE COMMUNITY

DEFINITIONS

COMMUNITY DEGREE:

THE SUM OF DEGREE OF ALL THE NODES IN THE COMMUNITY

$$k_C = \sum_{i \in C} k_i$$

DEFINITIONS

INTERNAL AND EXTERNAL DEGREE:
THE NUMBER OF NEIGHBOURS INSIDE AND OUTSIDE THE COMMUNITY

NUMBER OF INTERNAL LINKS:
THE NUMBER OF LINKS BETWEEN NODES WITHIN THE COMMUNITY

COMMUNITY DEGREE:
THE SUM OF DEGREE OF ALL THE NODES IN THE COMMUNITY

INTERNAL LINK DENSITY:
DENSITY THAT CONSIDERS ONLY LINKS BETWEEN MEMBERS OF THE COMMUNITY

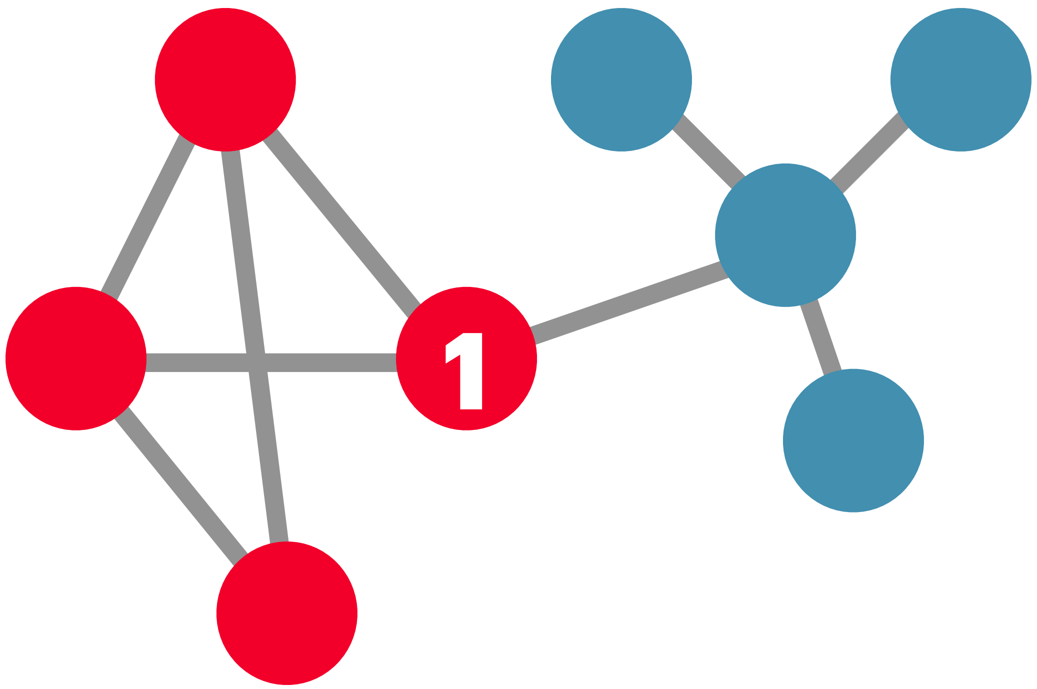
DEFINITIONS

INTERNAL LINK DENSITY:

DENSITY THAT CONSIDERS ONLY LINKS BETWEEN MEMBERS OF THE COMMUNITY

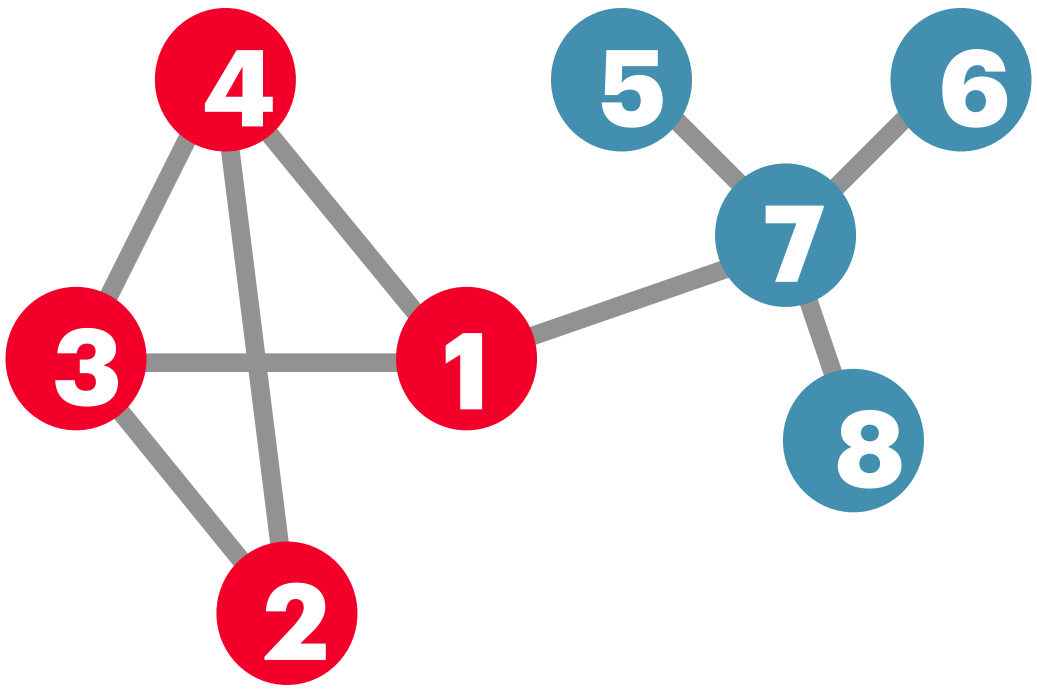
$$\delta_C^{int} = \frac{L_C}{L_C^{max}} = \frac{2L_C}{N_C(N_C - 1)}$$

DEFINITIONS



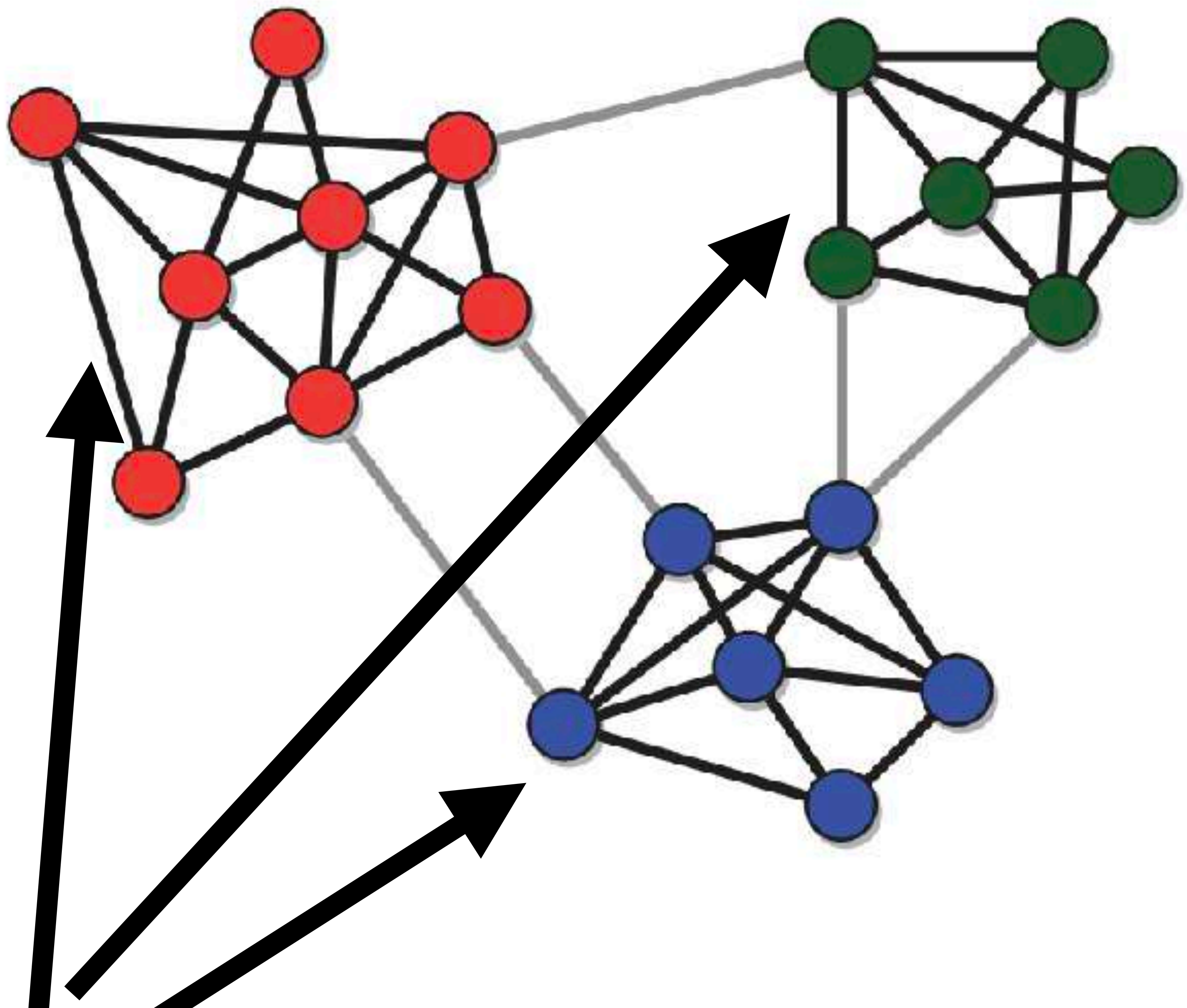
? $k_1^{ext}, k_1^{int}, \delta_{red}^{int}, k_{blue}$

DEFINITIONS



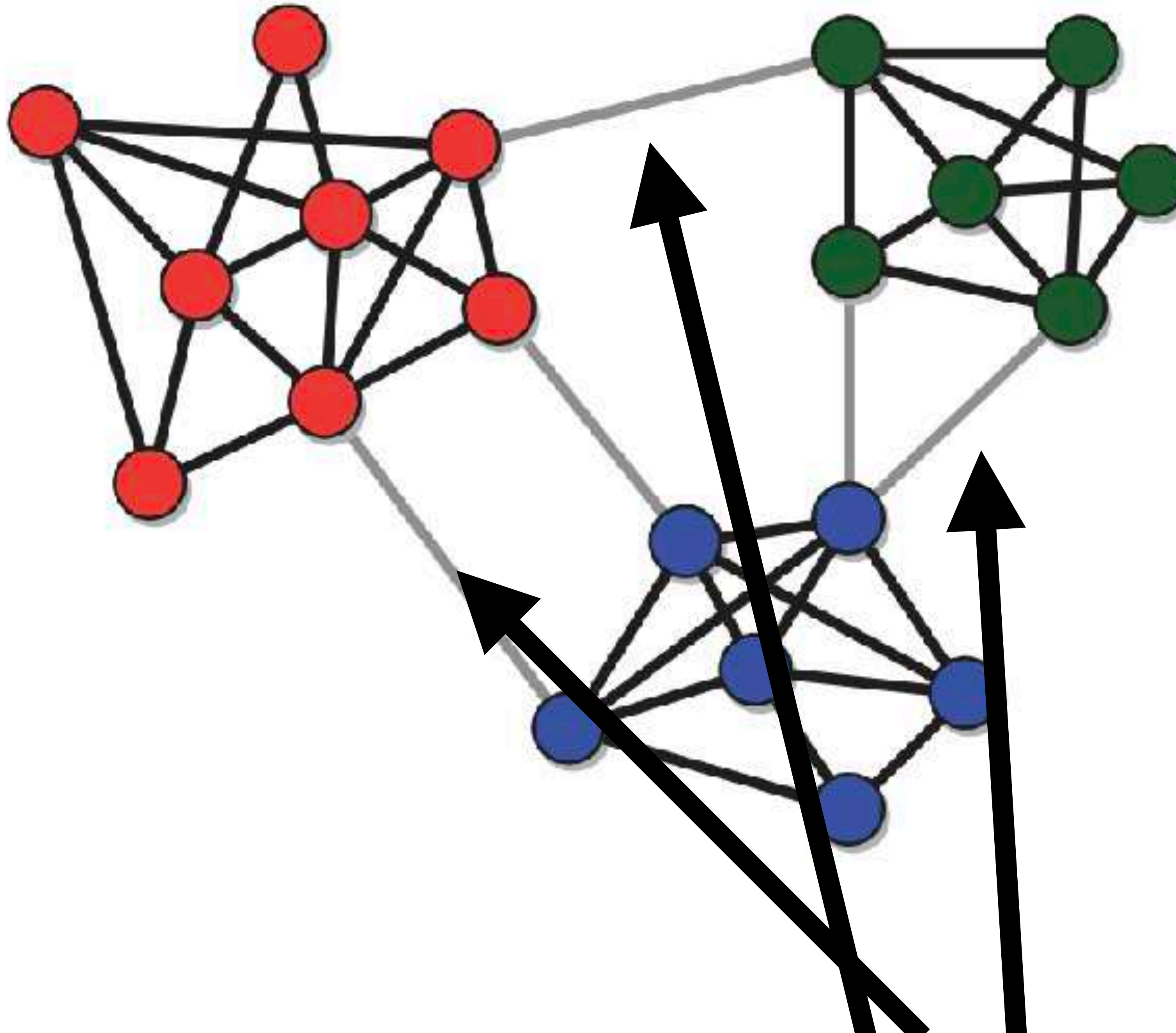
Which are the boundary nodes?

DEFINITIONS



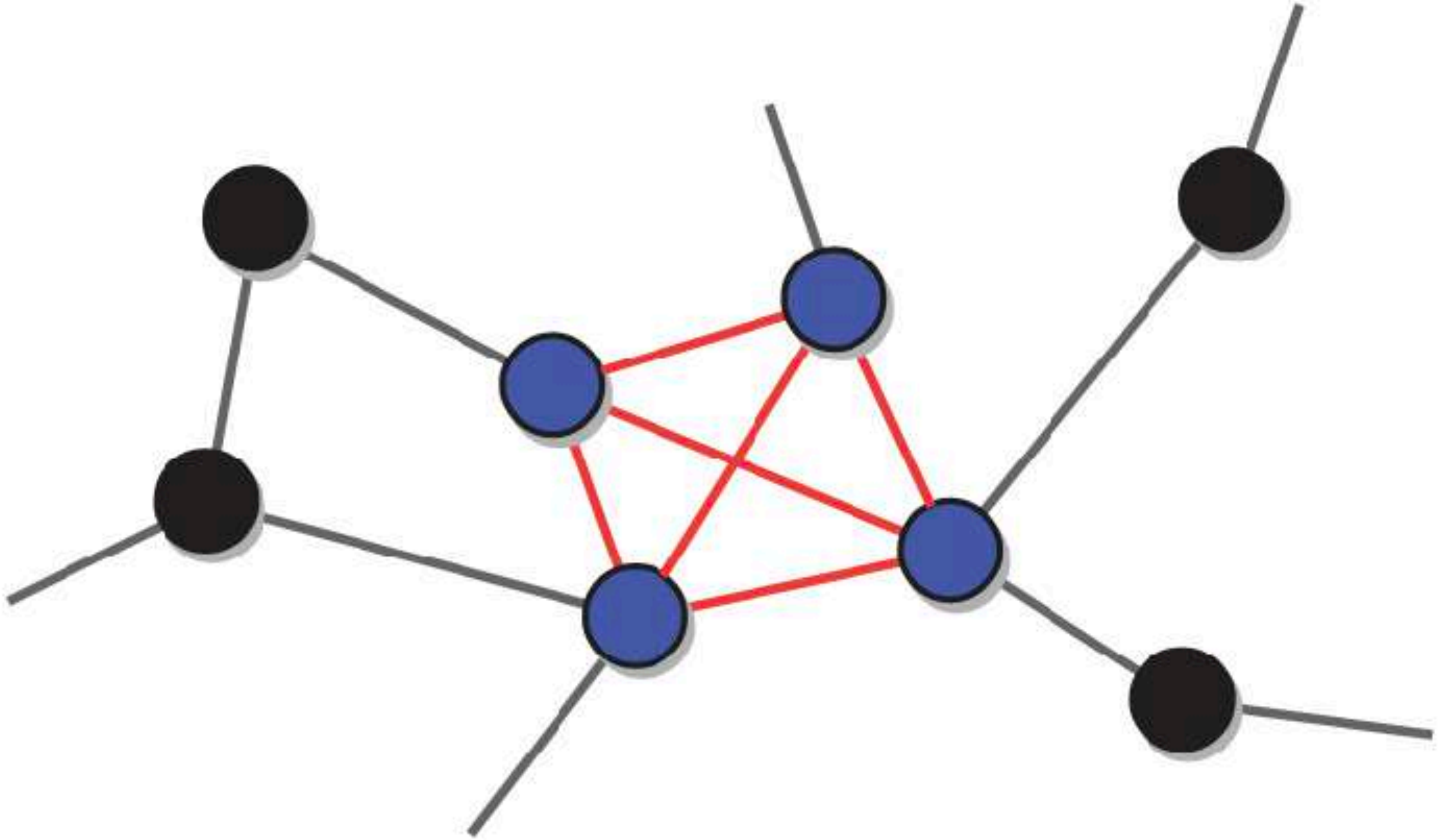
high cohesion, high separation

DEFINITIONS



high cohesion, high separation

DEFINITIONS



clique (a fully connected subgraph)

DEFINITIONS

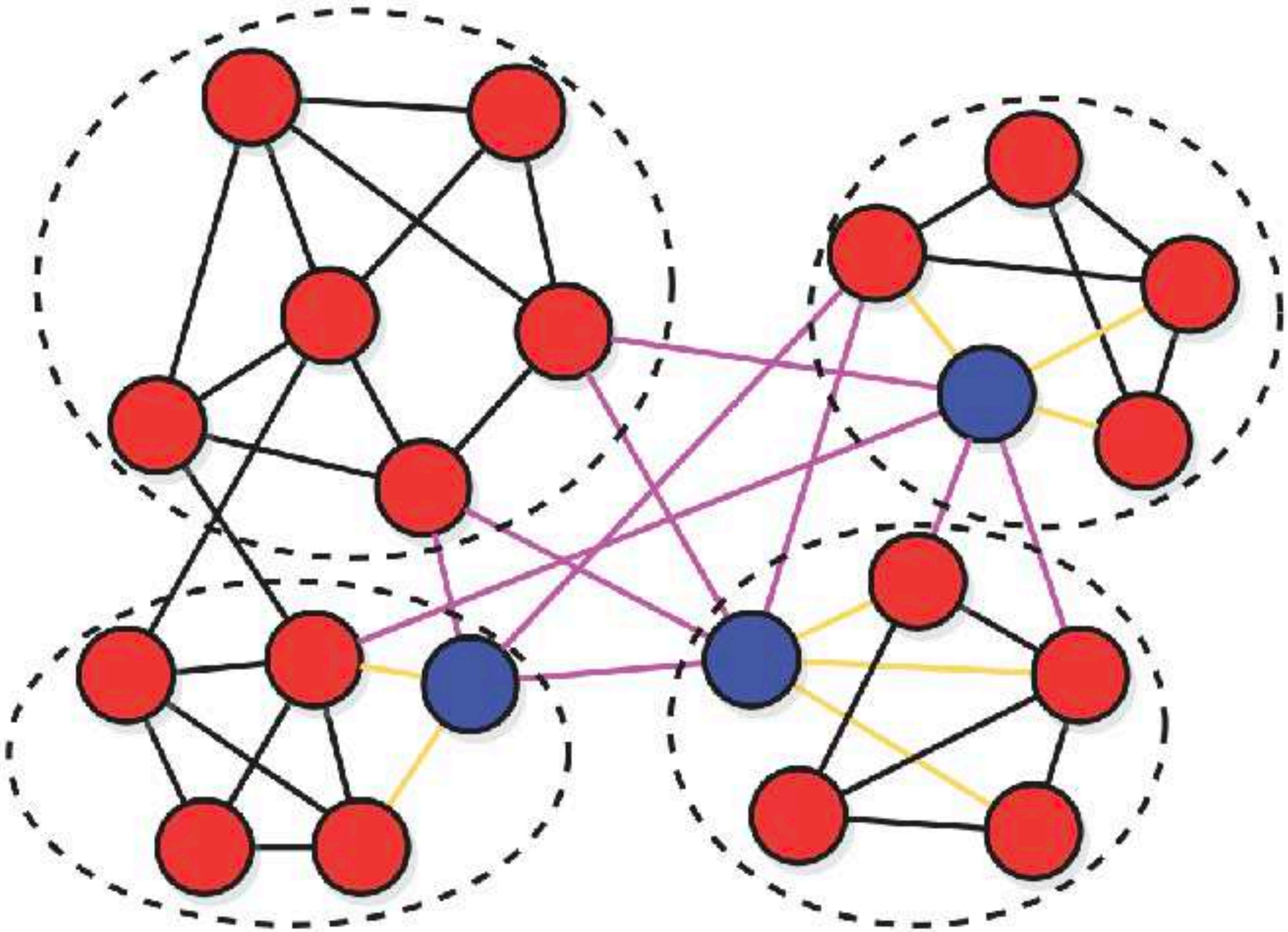
Strong community: $\forall i \in C : k_i^{int} > k_i^{ext}$

Weak community: $\sum_{i \in C} k_i^{int} > \sum_{i \in C} k_i^{ext}$

DEFINITIONS

Strong community: $\forall i \in C : k_i^{int} > k_i^{\cancel{ext}}$

Weak community: $$\sum_{i \in C} k_i^{int} > \sum_{i \in C} k_i^{\cancel{ext}}$$



Strong and weak communities. The four subnetworks enclosed in the dashed contours are weak communities according to both definitions we have given. They are also strong communities according to the less stringent definition, as the internal degree of each node exceeds the number of links joining the node with those of every other community. However, three of the subnetworks are not strong communities in the more stringent sense, because some nodes (in blue) have external degree larger than their internal degree (the internal and external links of these nodes are colored in yellow and magenta, respectively). Adapted from Fortunato and Hric (2016).

PARTITIONS

A PARTITION IS A DIVISION OF THE NETWORK IN COMMUNITIES

PARTITIONS

SUPPOSE YOU HAVE A NETWORK **G** WITH **10 NODES** 1,2,...,10

{1,2,...,10}

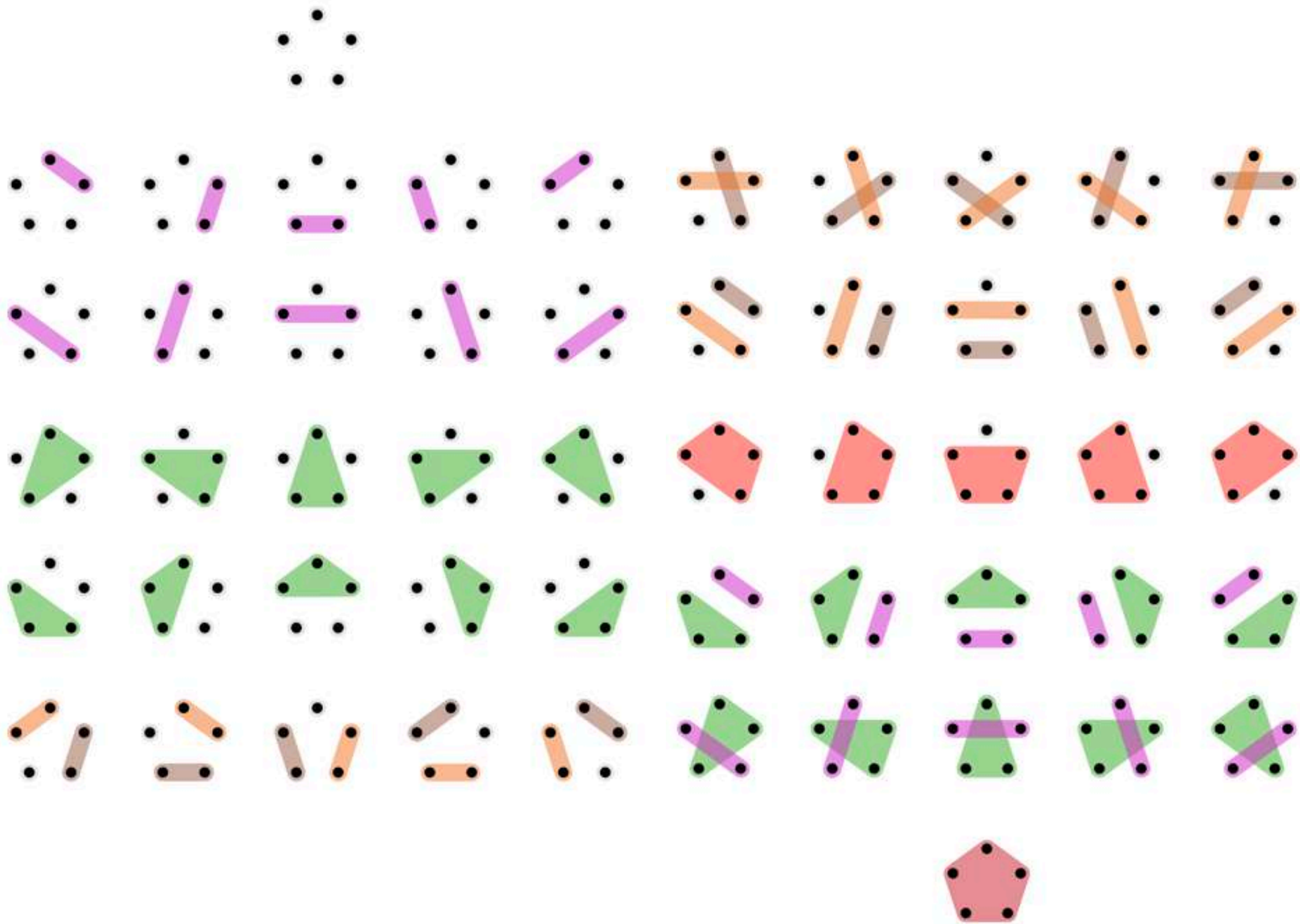
{1} {2} {3} ... {10}

{1,2} {3,6,9} {5,8,10} {7,4}

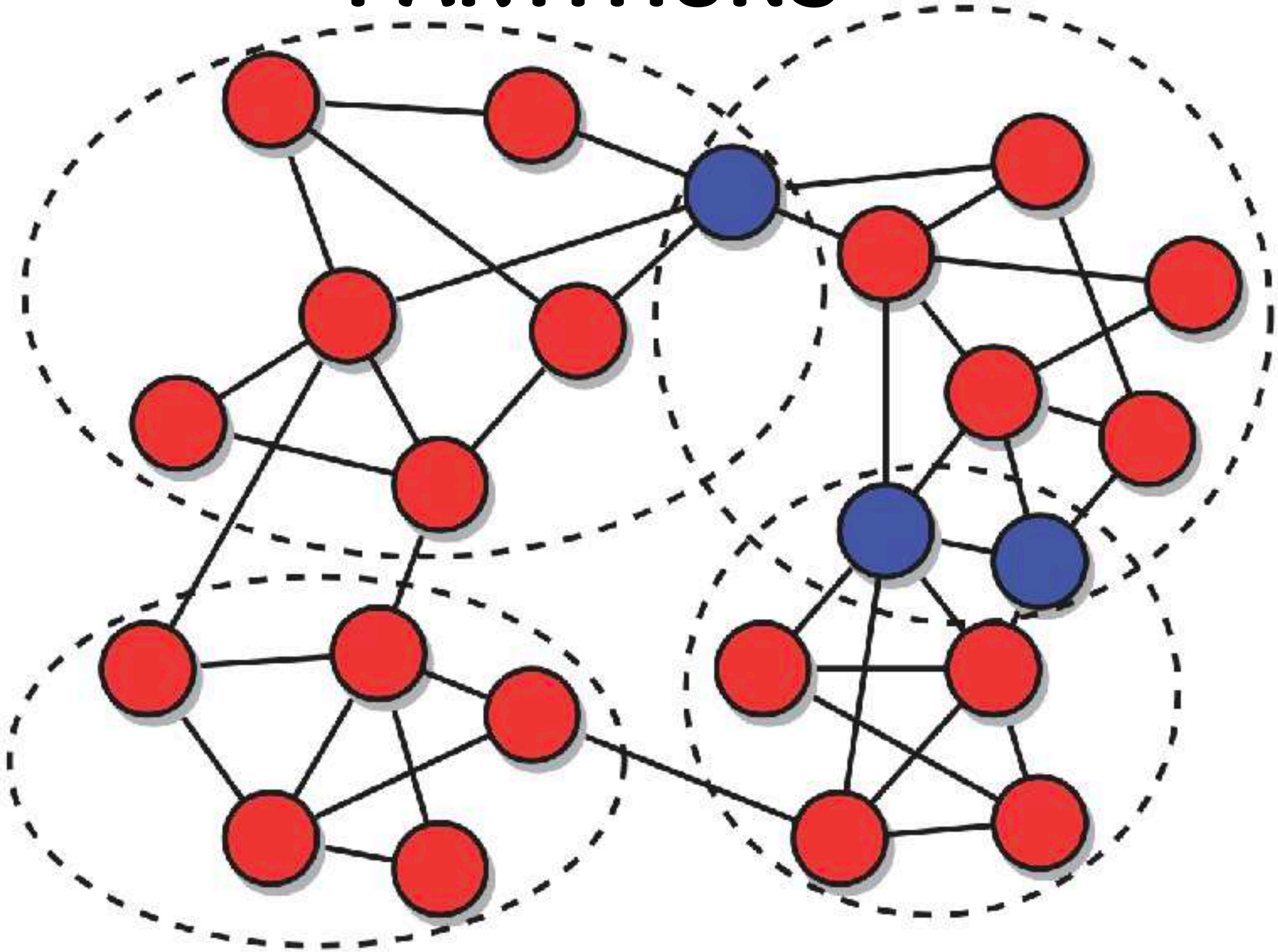
THESE ARE ALL VALID PARTITIONS

The number of possible partitions grows super-exponentially

PARTITIONS



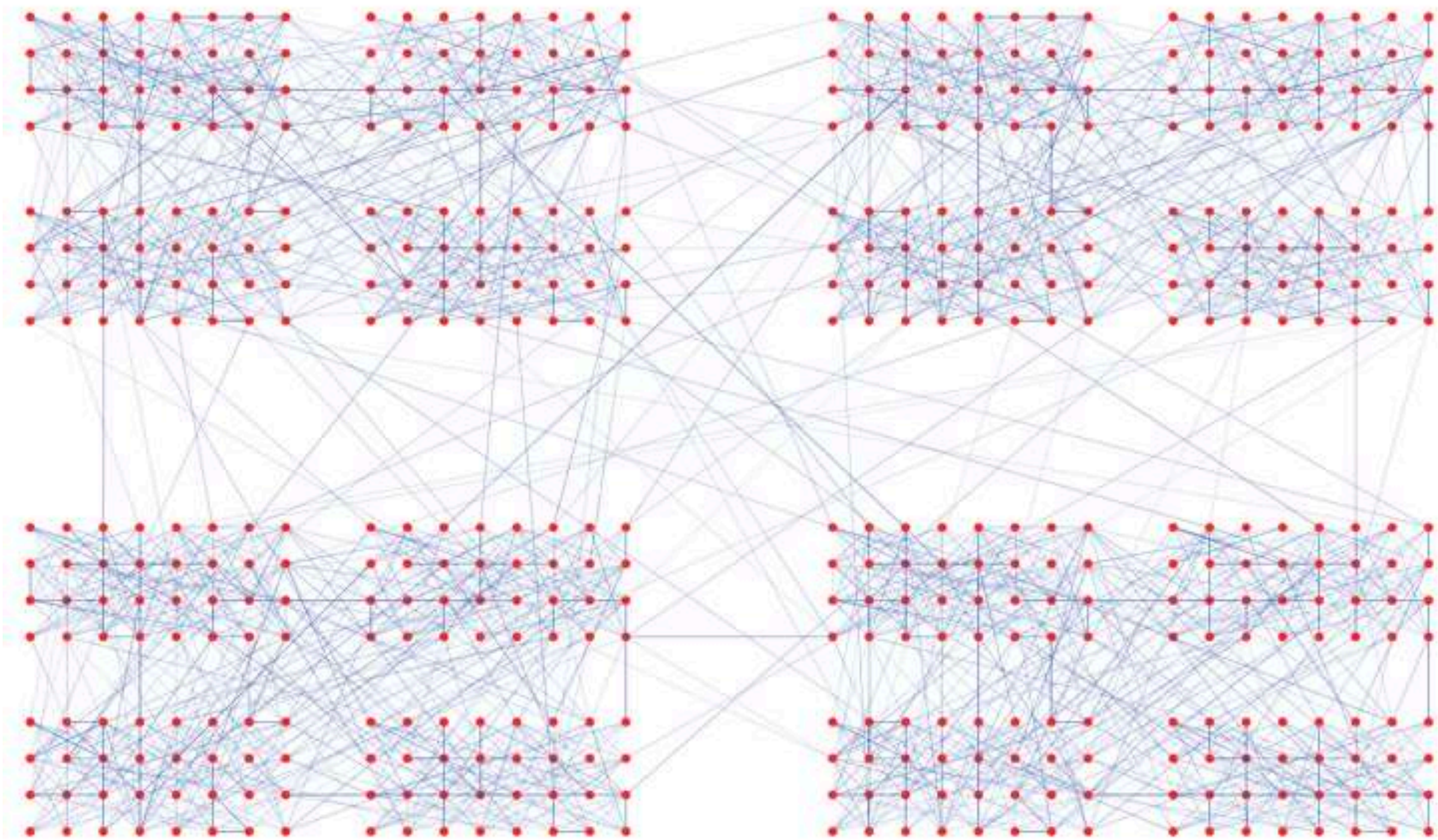
PARTITIONS



COMMUNITIES CAN OVERLAP

(You are part of different communities, think about it)

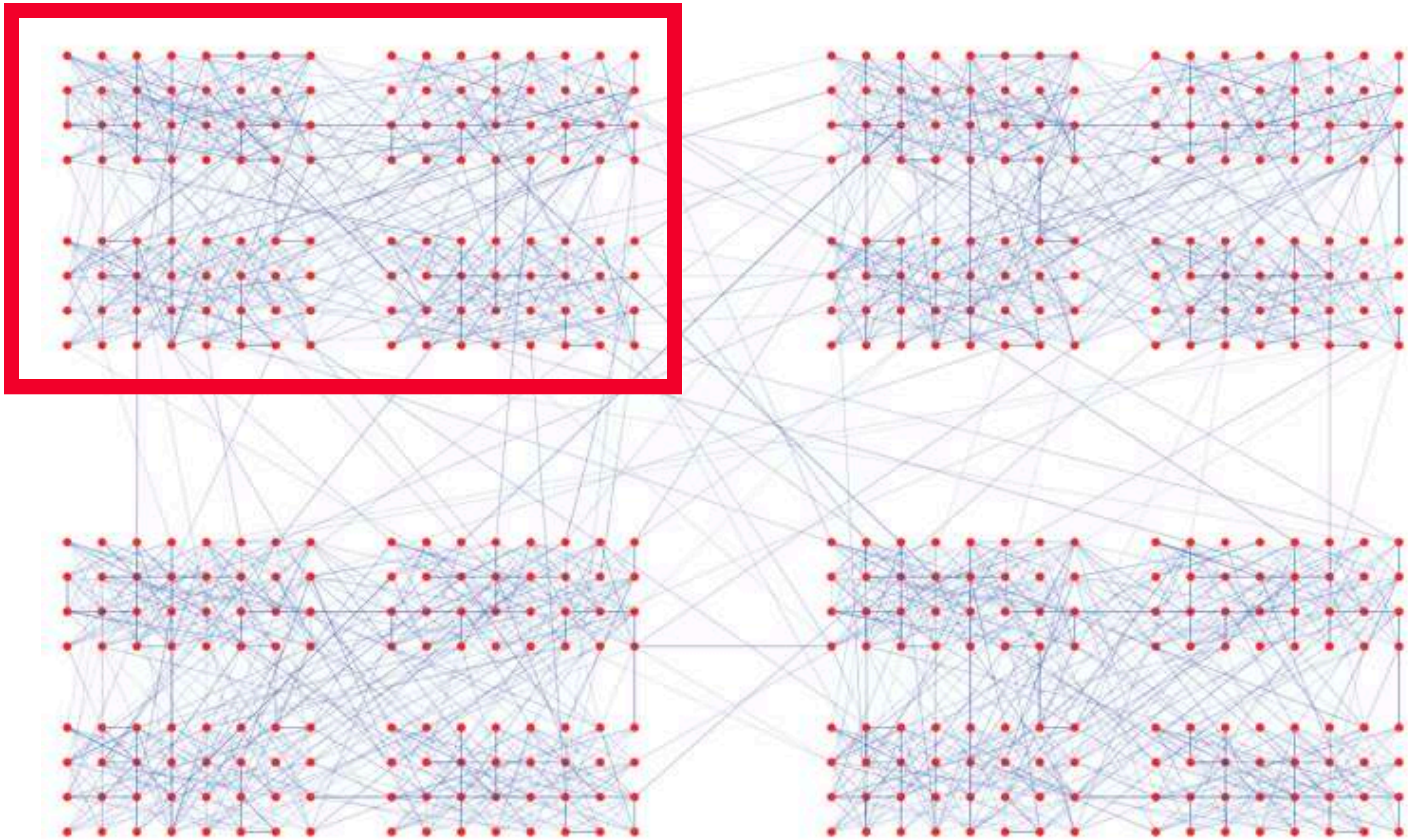
PARTITIONS



COMMUNITIES CAN BE HIERARCHICAL

(There might be communities within communities)

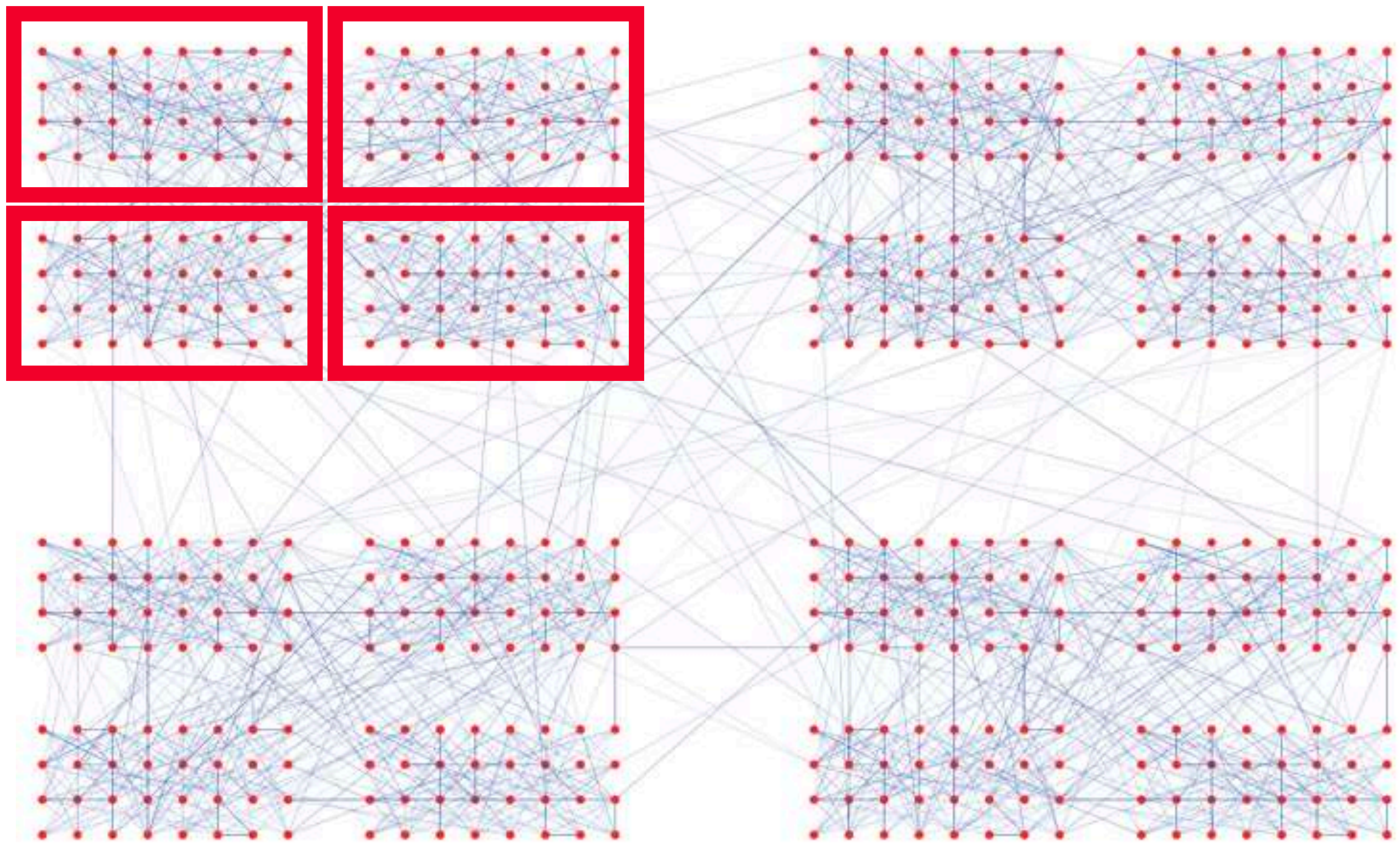
PARTITIONS



COMMUNITIES CAN BE HIERARCHICAL

(There might be communities within communities)

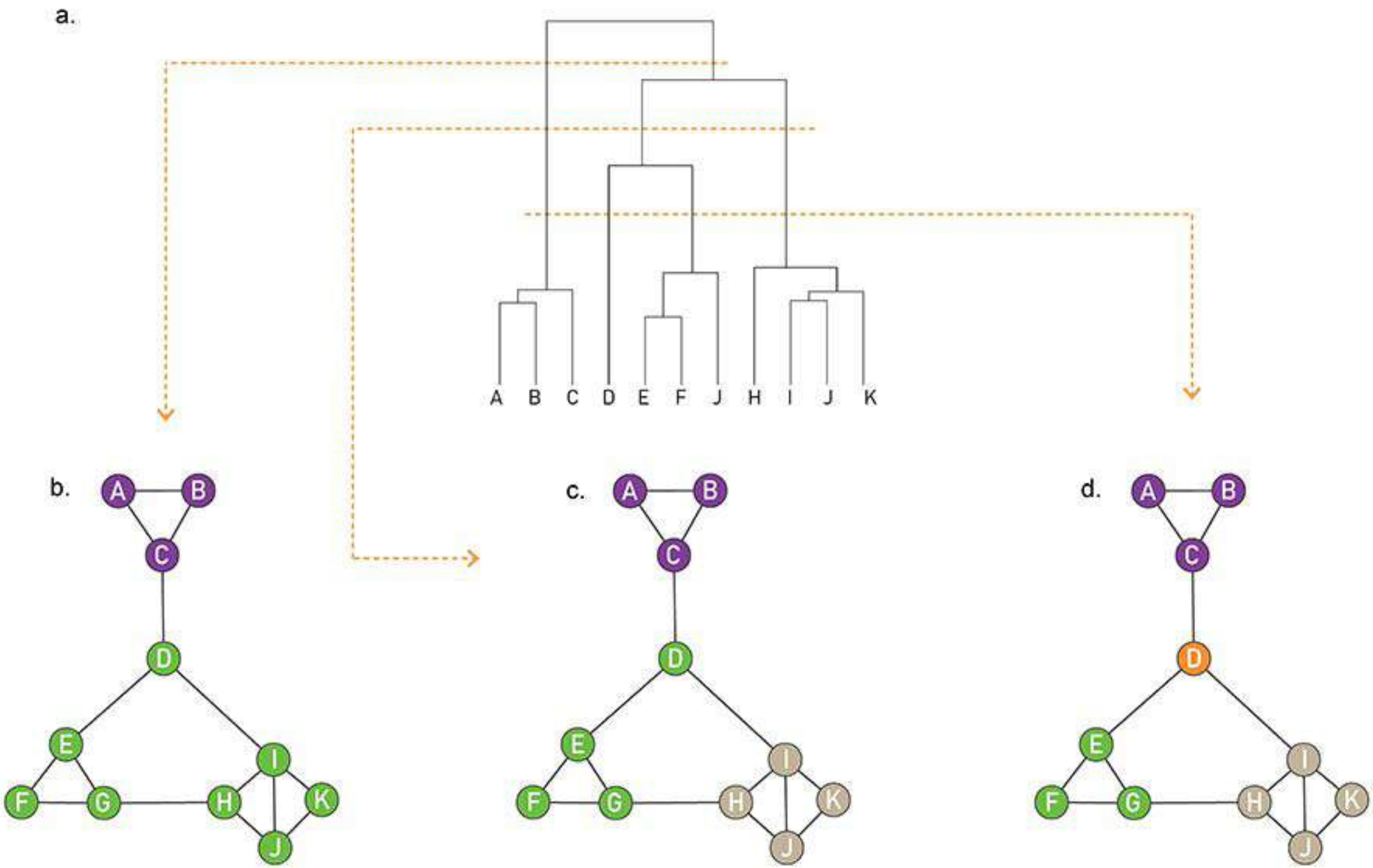
PARTITIONS



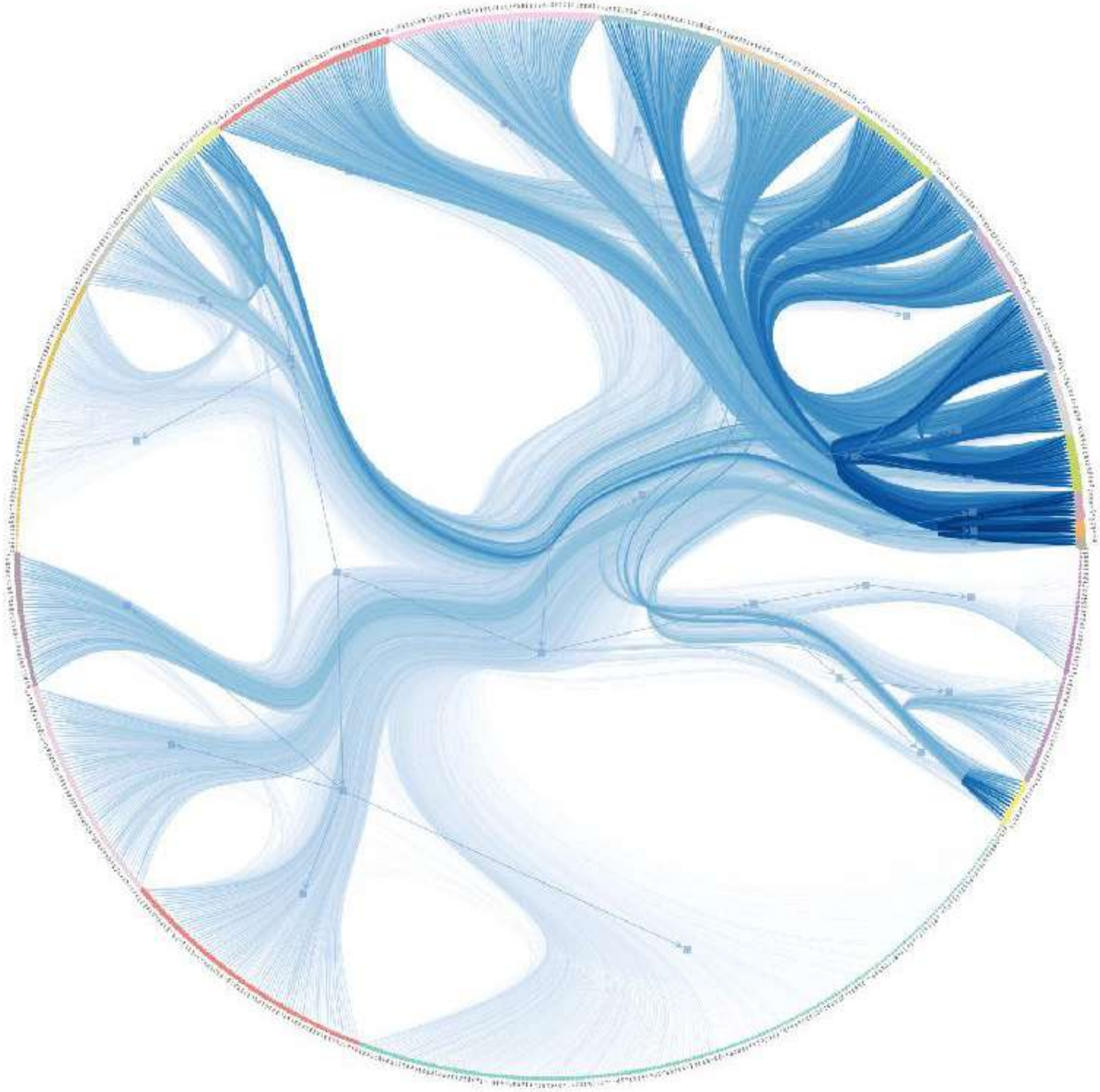
COMMUNITIES CAN BE HIERARCHICAL

(There might be communities within communities)

PARTITIONS



Dendrogram



EXERCISE

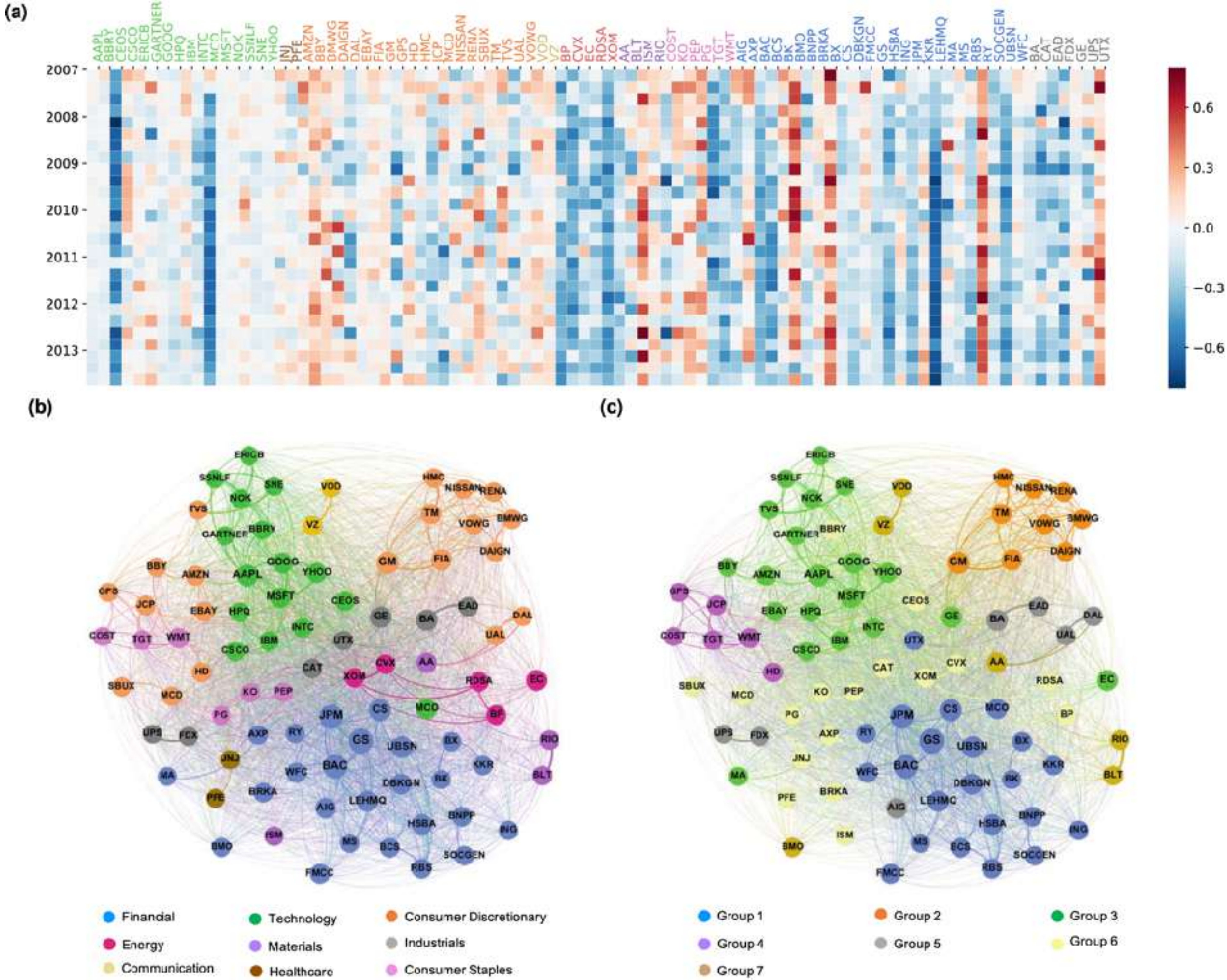
MAKE SOME EXAMPLES
OF SOCIAL AND FINANCIAL
NETWORKS WITH COMMUNITIES

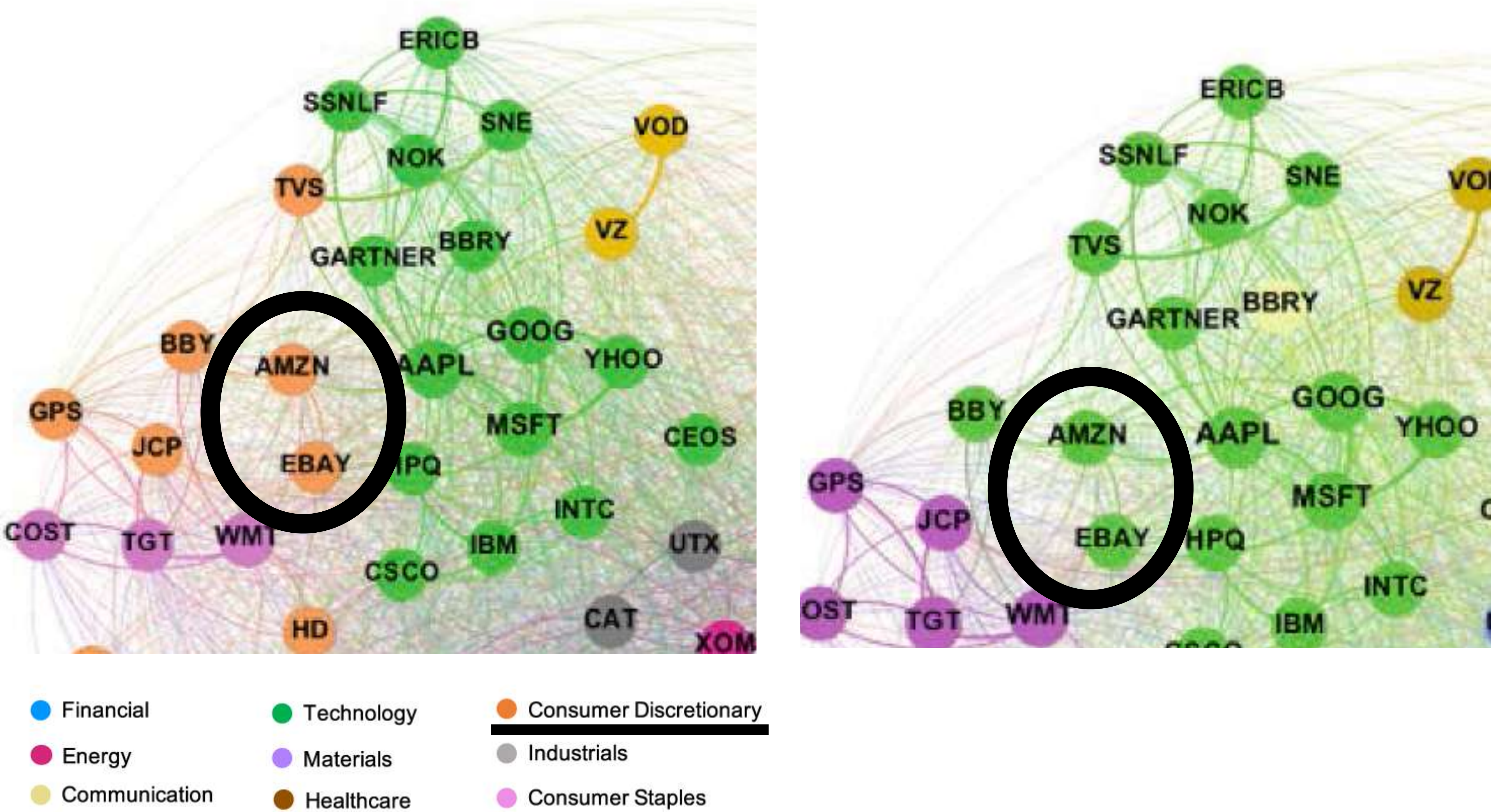
PART I RECAP

We saw what communities are and how they are **defined**

We explored some **examples**

We now have all the tools to learn about **community detection**


(a)
2007
2008
2009
2010
2011
2012
2013
0.6
0.3
0.0
−0.3
−0.6
(b)
(c)
Financial
Technology
Consumer Discretionary
Energy
Materials
Industrials
Communication
Healthcare
Consumer Staples
Group 1
Group 2
Group 3
Group 4
Group 5
Group 6
Group 7


ERICB
SSNLF
SNE
VOD
NOK
TVS
VZ
GARTNER
BBRY
BBY
AMZN
AAPL
GOOG
YHOO
GPS
JCP
MSFT
CEOS
EBAY
HPQ
COST
TGT
WMT
INTC
UTX
IBM
CSCO
HD
CAT
XOM
ERICB
SSNLF
SNE
NOK
TVS
VZ
GARTNER
BBRY
BBY
GOOG
AMZN
AAPL
YHOO
GPS
JCP
MSFT
EBAY
HPQ
INTC
TGT
WMT
IBM
Financial
Technology
Consumer Discretionary
Energy
Materials
Industrials
Communication
Healthcare
Consumer Staples

COMMUNITY DETECTION

The task of **finding communities** in a network

We now have all the tools to learn about **community detection**

COMMUNITY DETECTION

FOUR APPROACHES

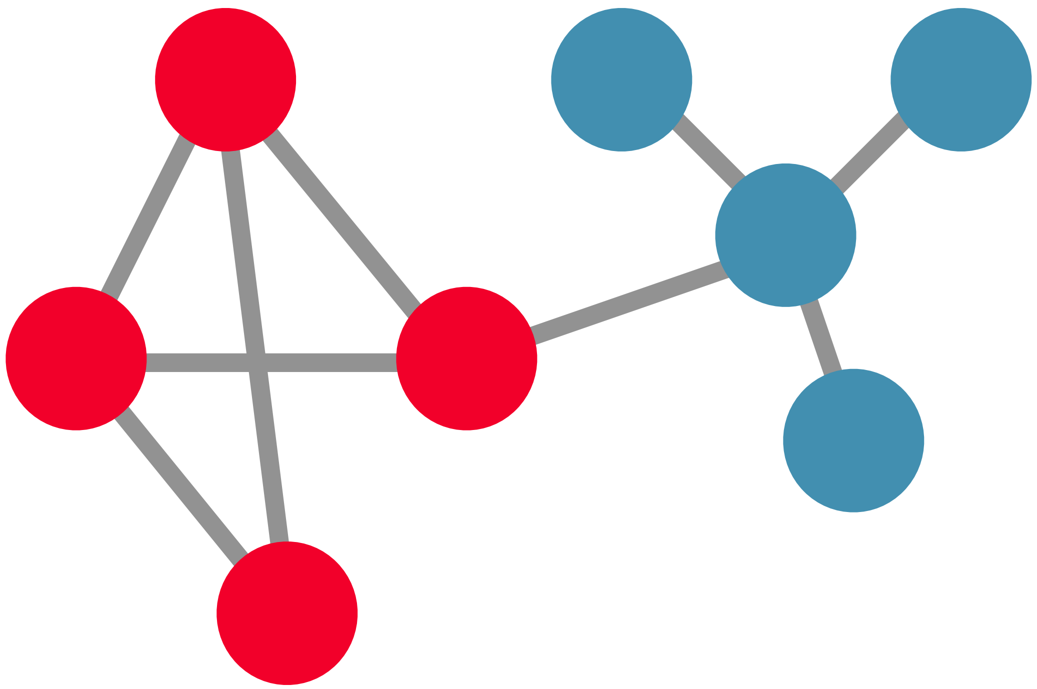
Bridge removal

Modularty maximisation

Label propagation

Stochastic block modelling

BRIDGE REMOVAL



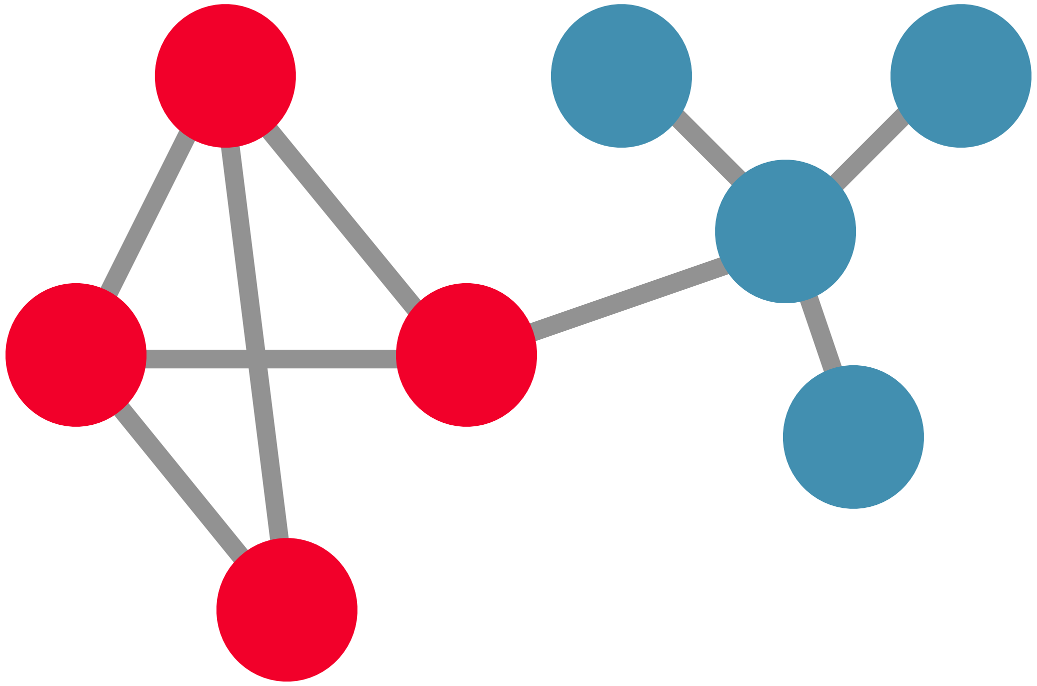
A bridge is a link whose removal breaks the network into two parts

BRIDGE REMOVAL

The most famous algorithm based on this approach is the **Girvan-Newman algorithm**

BRIDGE REMOVAL

The most famous algorithm based on this approach is the **Girvan-Newman algorithm**



How do we find a bridge?

BRIDGE REMOVAL

The most famous algorithm based on this approach is the **Girvan-Newman algorithm**

1 - compute link **betweenness** for all the links

BRIDGE REMOVAL

The most famous algorithm based on this approach is the **Girvan-Newman algorithm**

1 - compute link **betweenness** for all the links

2 - **remove** the link with highest betweenness*

***in case of a tie, pick a random one among those with highest betweenness**

BRIDGE REMOVAL

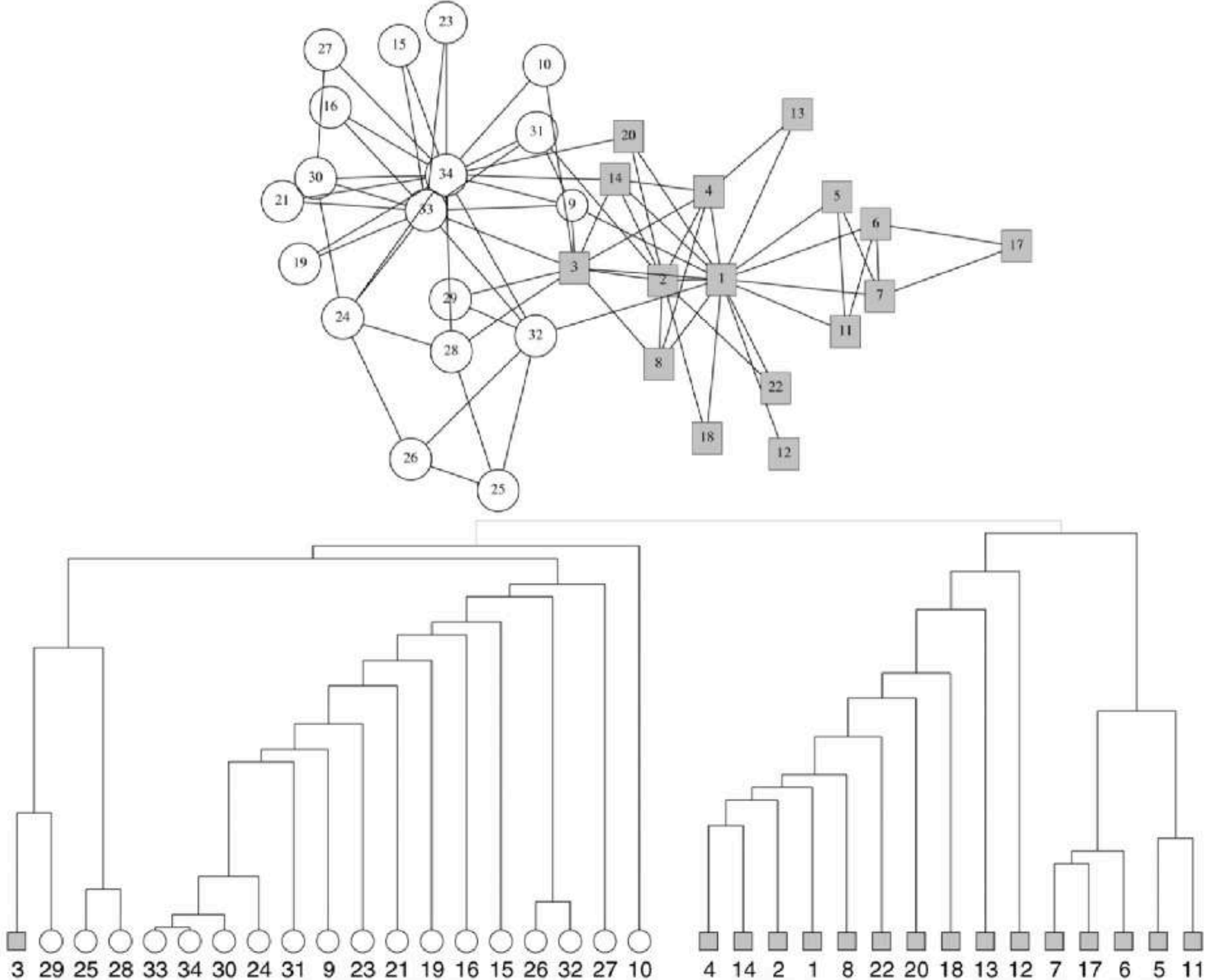
The most famous algorithm based on this approach is the **Girvan-Newman algorithm**

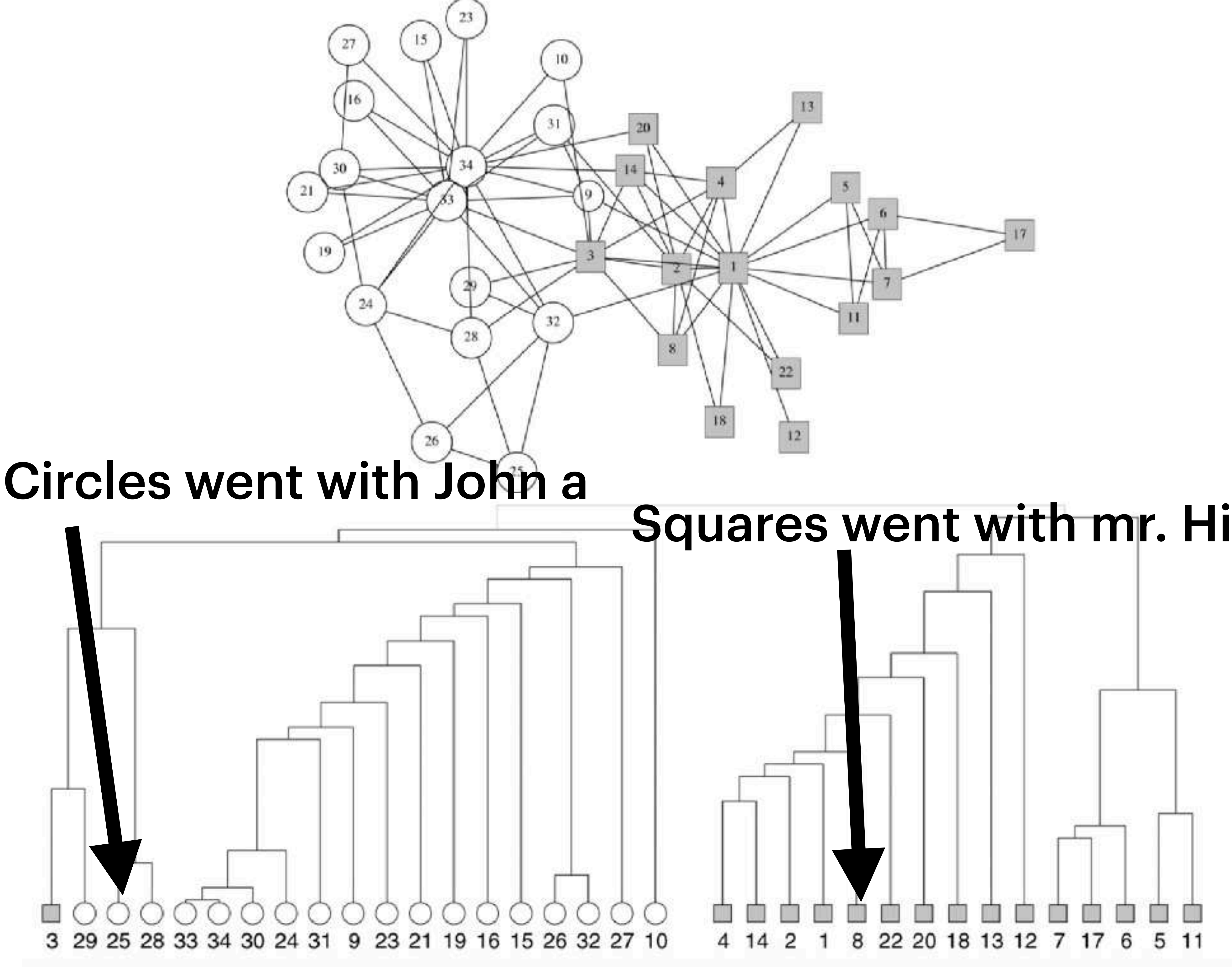
1 - compute link **betweenness** for all the links

2 - **remove** the link with highest betweenness*

3 - **repeat** 1 and 1 until you have no links left

***in case of a tie, pick a random one among those wit**


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


Circles went with John a
Squares went with mr. Hi
3 29 25 28 33 34 30 24 31 9 23 21 19 16 15 26 32 27 10
4 14 2 1 8 22 20 18 13 12 7 17 6 5 11

FINAL VERDICT

GREAT FIRST ATTEMPT, BUT COMPUTING LINK BETWEENNESS FOR LARGE NETWORKS THAT MANY TIMES IS IMPOSSIBLE

MODULARITY MAXIMISATION

MAIN IDEA: WE CALCULATE HOW GOOD A COMMUNITY IS VS RANDOM BASELINE

MODULARITY MAXIMISATION

MAIN IDEA: WE CALCULATE HOW GOOD A COMMUNITY IS VS RANDOM BASELINE

Originally introduced to know **where to cut** the dendrogram in Girvan-Newman

MODULARITY MAXIMISATION

MAIN IDEA: WE COUNT HOW MANY LINKS INSIDE COMMUNITY VS RANDOM NETWORK

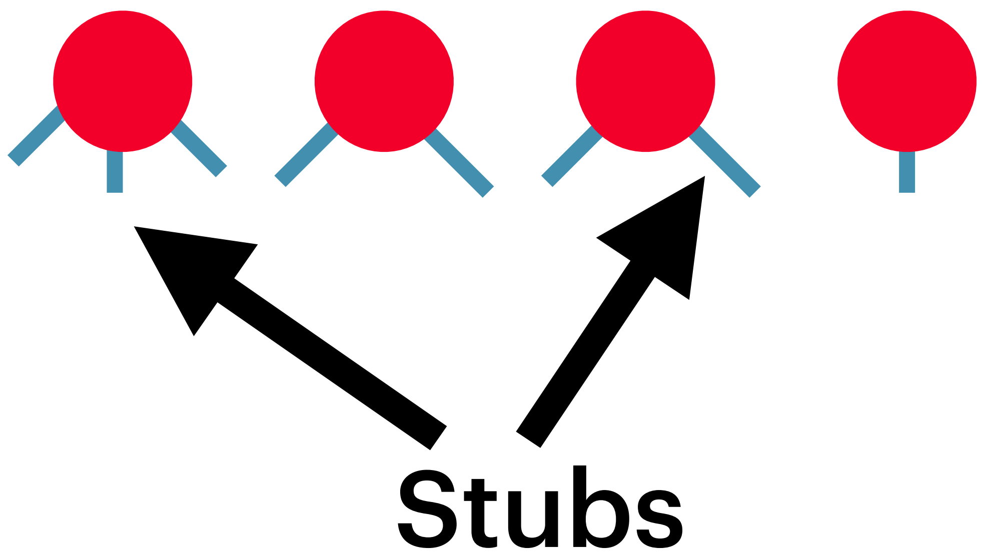
$$Q = \frac{1}{L} \sum_{C} \left(L_C - \frac{k_C^2}{4L} \right)$$

MODULARITY MAXIMISATION

Difference between links in c
and expected links in c with
configuration model

$$Q = \frac{1}{L} \sum_C \left(L_C - \frac{k_C^2}{4L} \right)$$

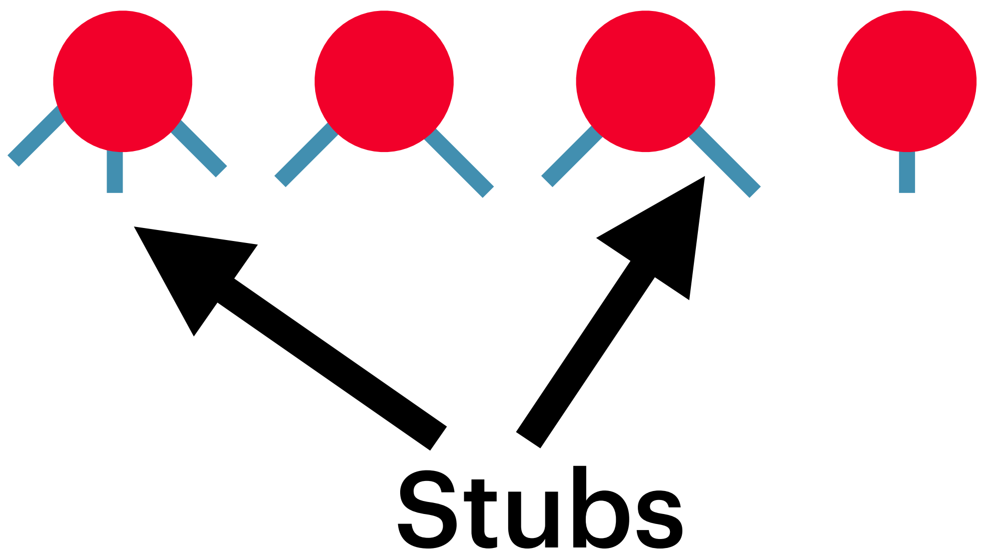
MODULARITY MAXIMISATION



$$Q = \frac{1}{L} \sum_{C} \left(L_C - \frac{k_C^2}{4L} \right)$$

$\frac{k_C}{2L}$ Is the probability of randomly choosing **one stub** in the community

MODULARITY MAXIMISATION



$$Q = \frac{1}{L}\sum_C \left(L_C - \frac{k_C^2}{4L} \right)$$

$\left(\frac{k_C}{2L}\right)^2$ Is the probability of randomly choosing **two stubs** in the community

MODULARITY MAXIMISATION

There are **L** links in the network

MODULARITY MAXIMISATION

There are **L** links in the network

Each link joins two stubs from community c with probability

$$\left(\frac{k_C}{2L}\right)^2$$

MODULARITY MAXIMISATION

There are **L** links in the network

Each link joins two stubs from community c with probability

$$\left(\frac{k_C}{2L}\right)^2$$

Then, the expected number of links in the community is

$$L\left(\frac{k_C}{2L}\right)^2 = \frac{k_C^2}{4L}$$

MODULARITY MAXIMISATION

Average

$$Q = \frac{1}{L}\sum_{C}\left(L_C - \frac{k_C^2}{4L}\right)$$

Difference between actual links in c and expected links in c

MODULARITY MAXIMISATION

Directed $$Q_d = \frac{1}{L}\sum_C \left(L_C - \frac{k_C^{in} k_C^{out}}{L} \right)$$

Weighted $$Q_w = \frac{1}{W}\sum_C \left(W_C - \frac{s_C^2}{4W} \right)$$

Weighted and directed $$Q_{dw} = \frac{1}{W}\sum_C \left(W_C - \frac{s_C^{in} s_C^{out}}{W} \right)$$

MODULARITY MAXIMISATION

Most famous algorithms: **Louvain, Leiden**

MODULARITY MAXIMISATION

Most famous algorithms: **Louvain, Leiden**

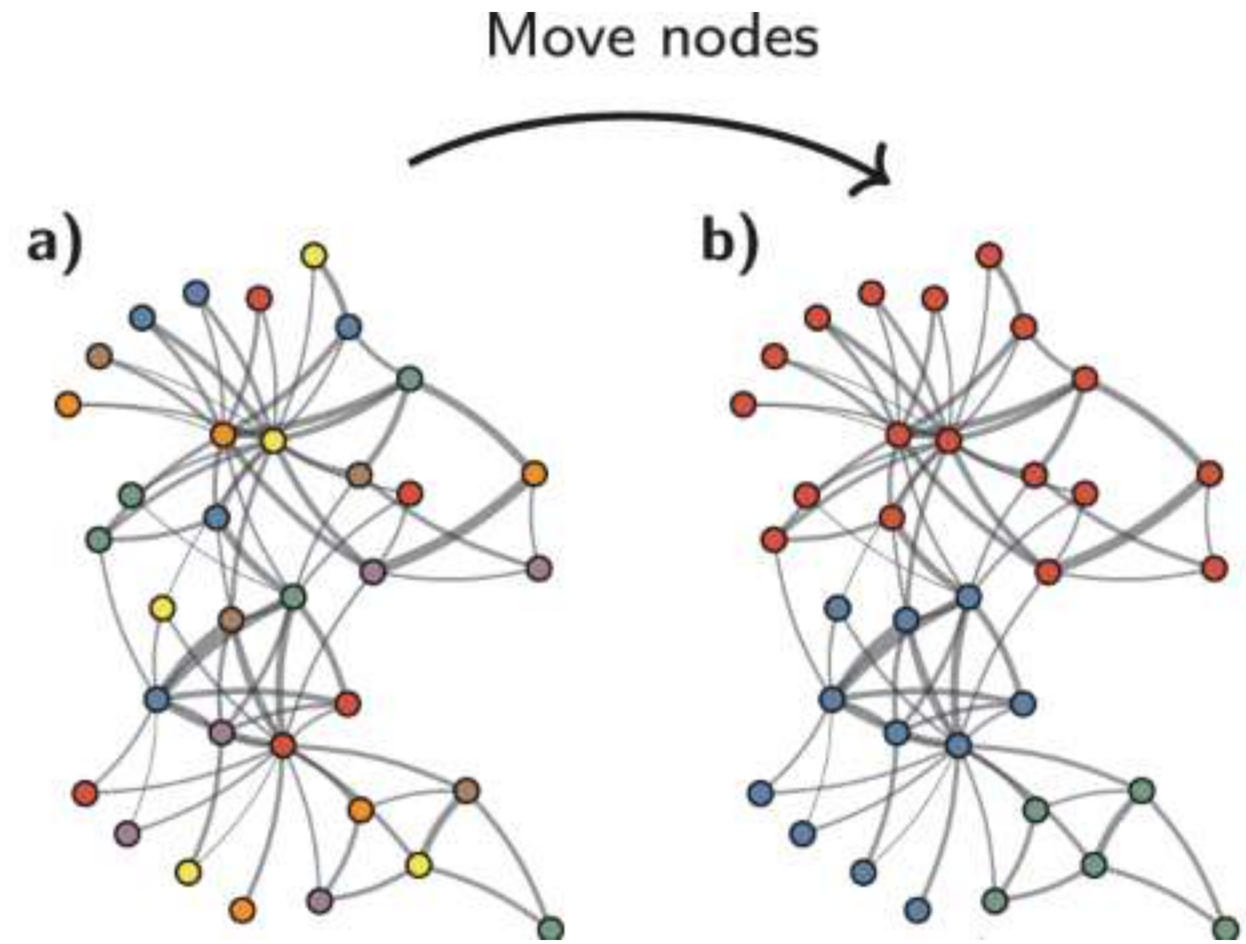
1) start with no communities. Every nodes is moved to a community so that **Q** Is maximised. Repeat until no modularity gain is possible

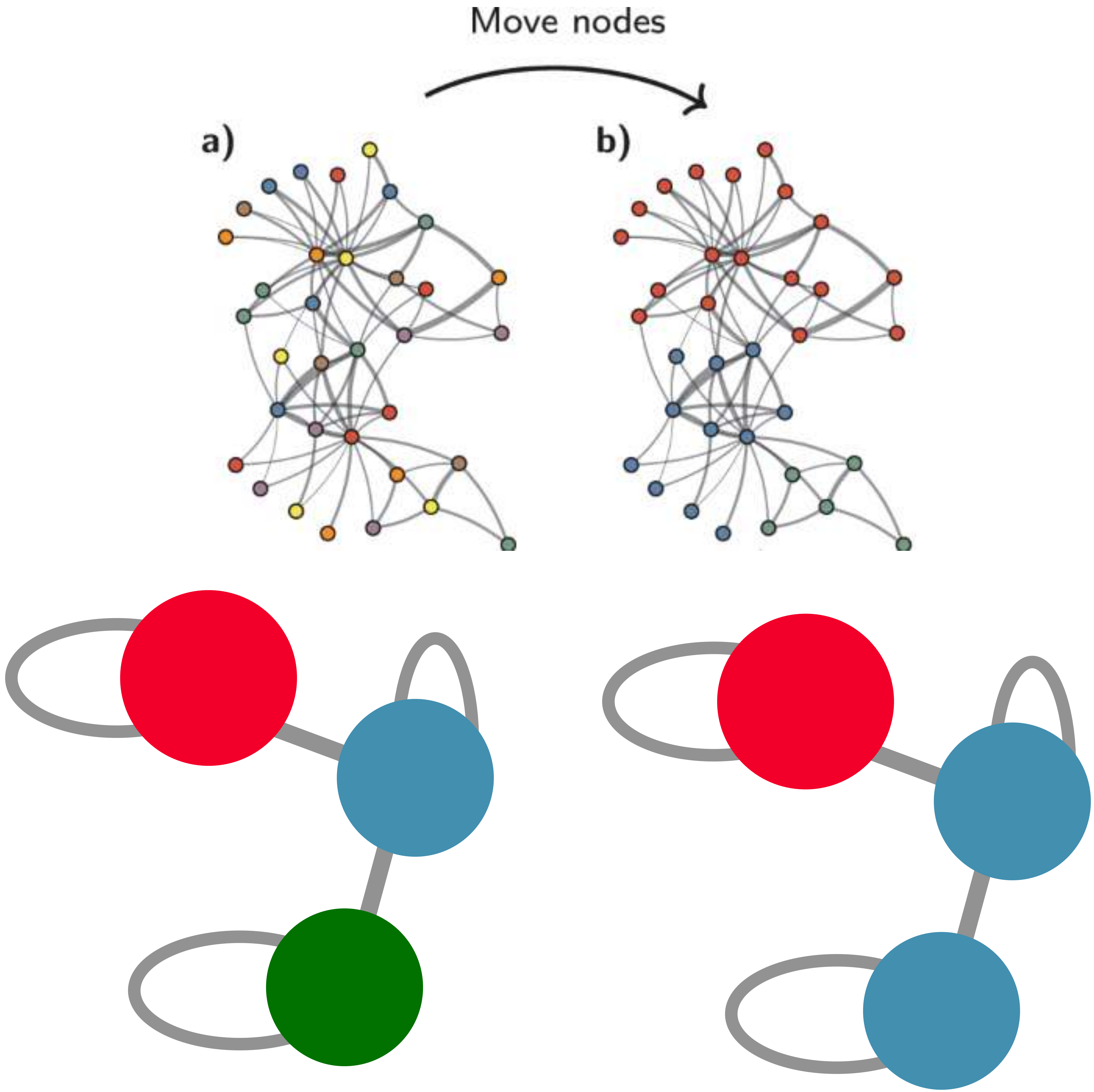
MODULARITY MAXIMISATION

Most famous algorithms: **Louvain, Leiden**

1) start with no communities. Every nodes is moved to a community so that **Q** Is maximised. Repeat until no modularity gain is possible

2) the network becomes a weighted super-network, in which nodes are the communities of the original network, and weights are the number of links between communities (this includes self-loops)


Move nodes
a)
b)


Move nodes
a)
b)

MODULARITY MAXIMISATION PROBLEMS

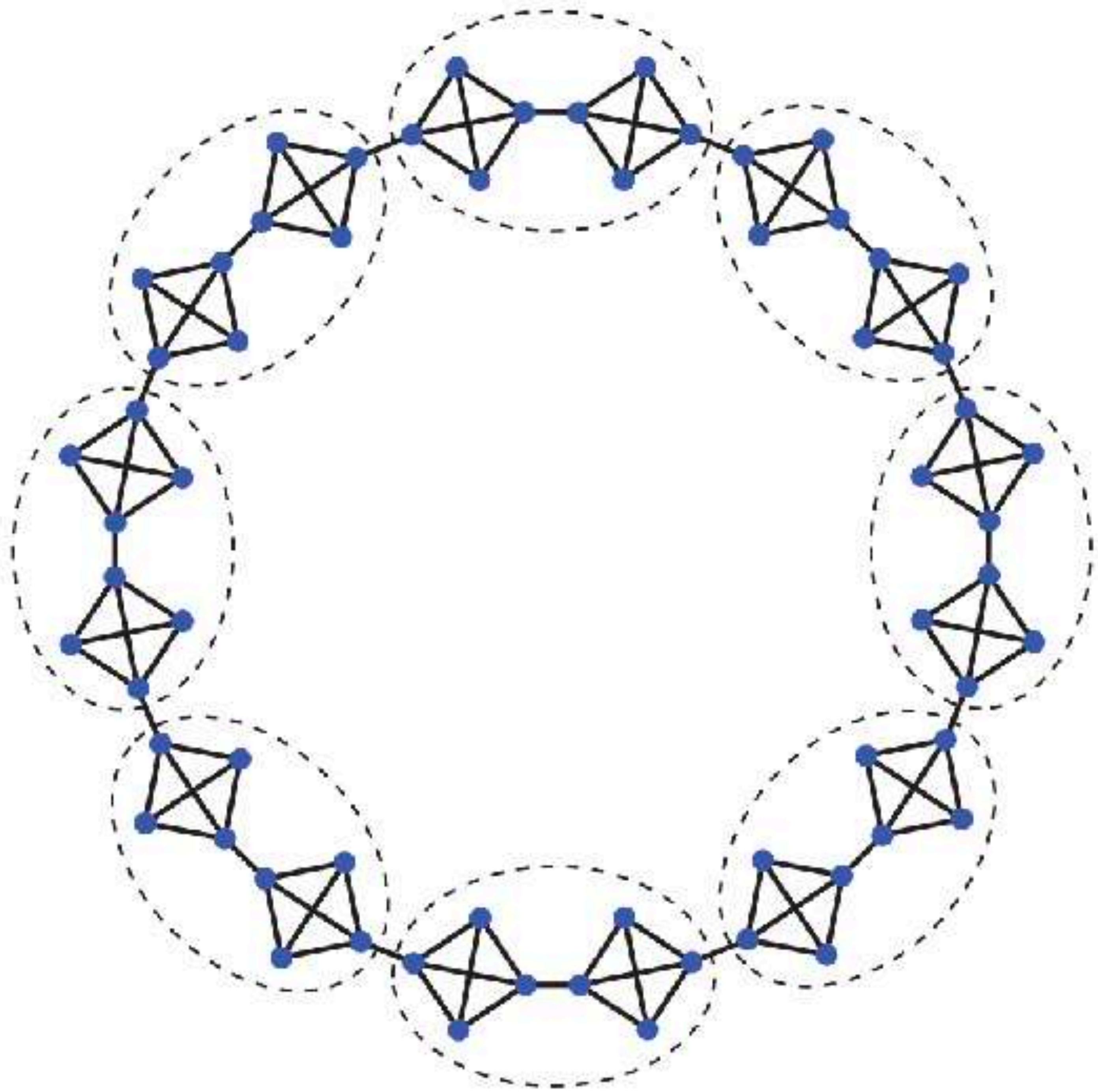
Comparison: On average Larger networks have larger modularity

Uncertainty: this approach can find positive modularity for random networks

Resolution: cannot find communities whose degree is smaller than

$$\sqrt{2L}$$

MODULARITY MAXIMISATION



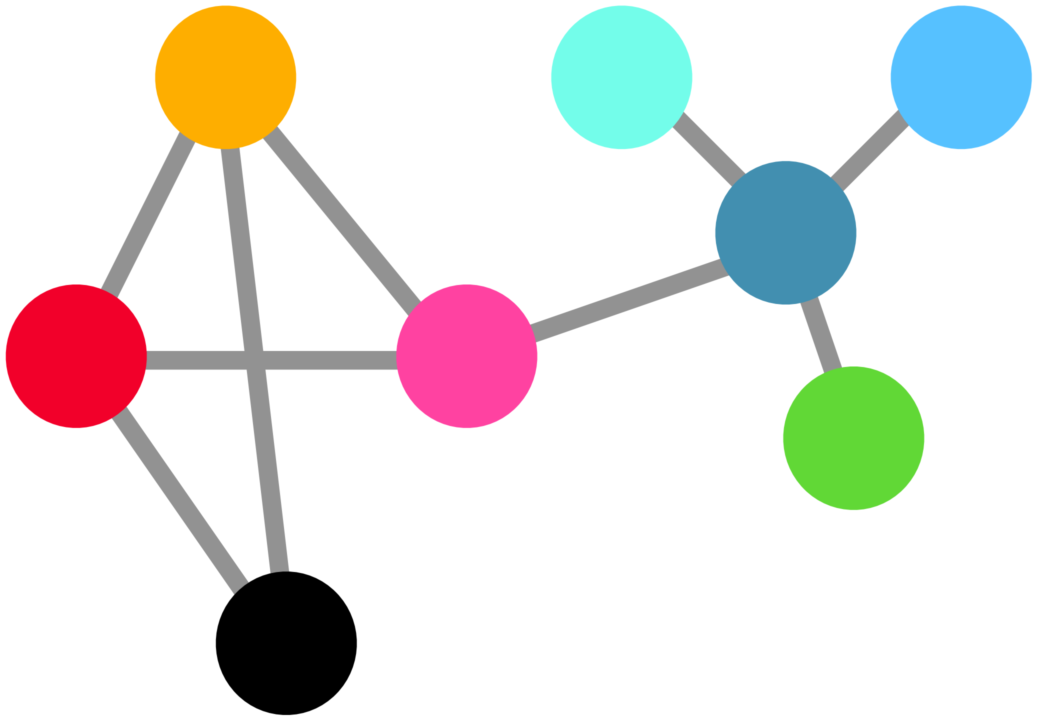
LABEL PROPAGATION

1) WE START WITH **SINGLETONS**

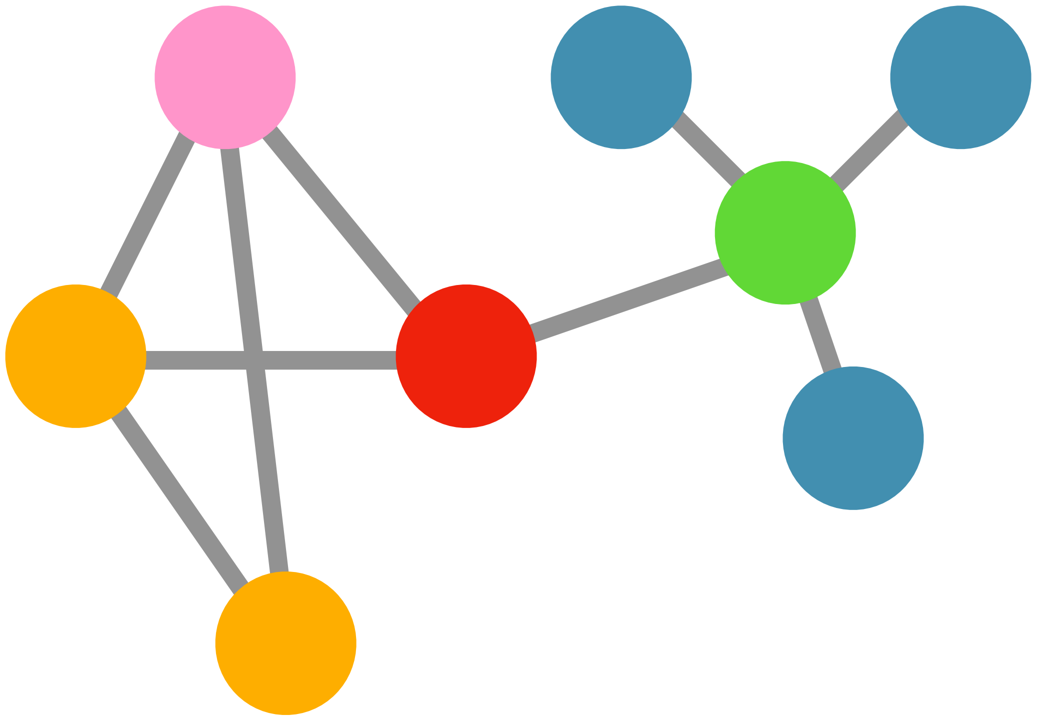
2) ONE BY ONE, WITH RANDOM ORDER, NODES TAKE THE "LABEL" (IE COMMUNITY MEMBERSHIP) OF THE **MAJORITY OF THEIR NEIGHBOURS**

3) WE REPEAT THIS UNTIL THE **PARTITION IS STABLE** (IE THERE ARE NO POSSIBLE CHANGES)

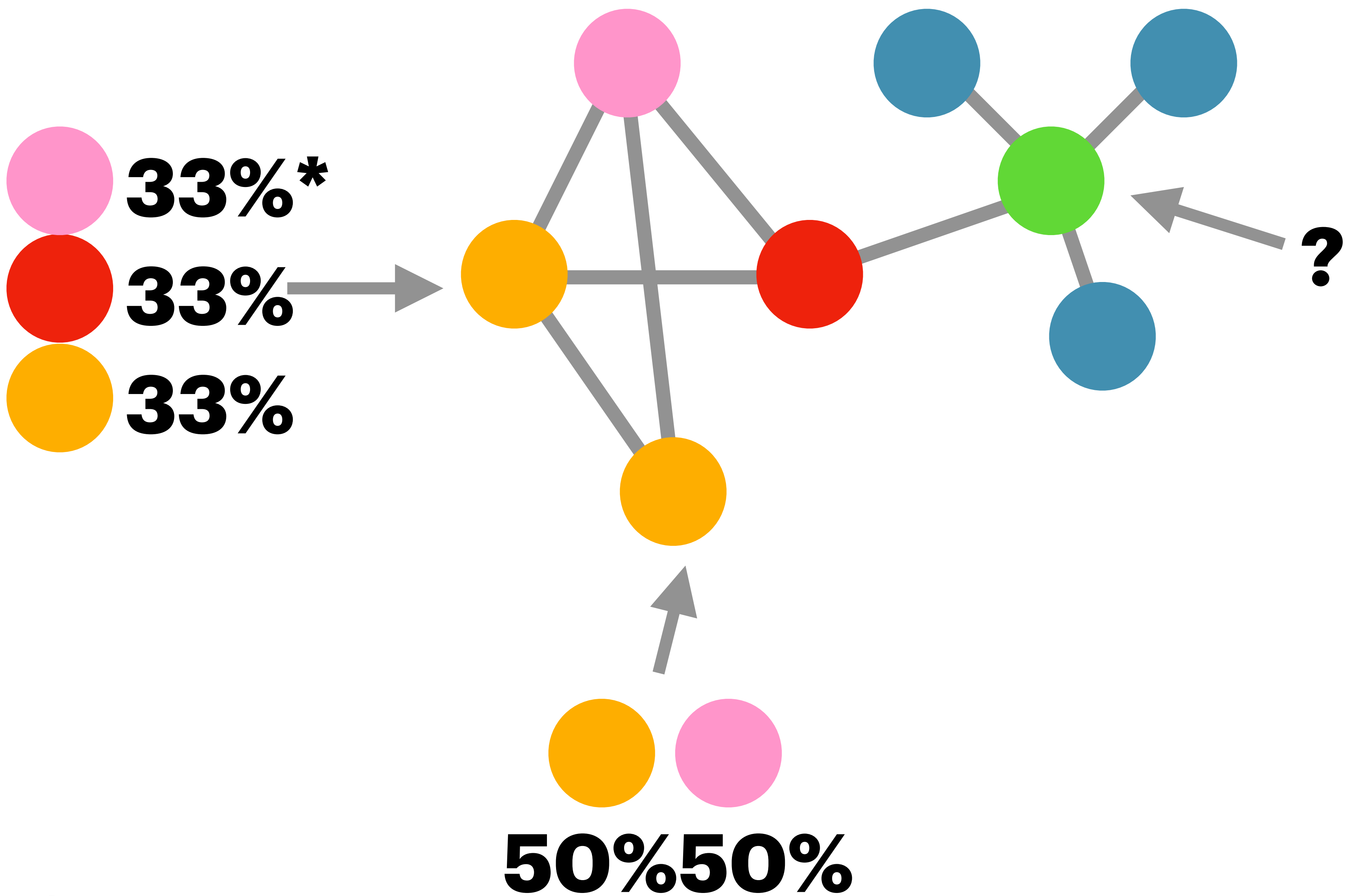
LABEL PROPAGATION



LABEL PROPAGATION

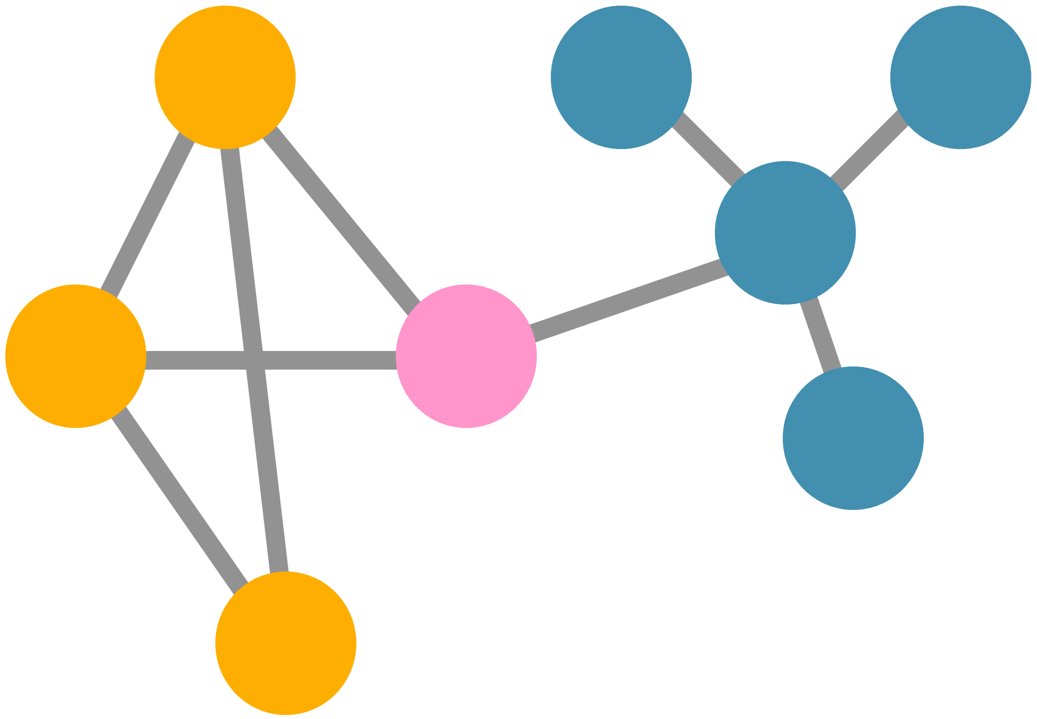


LABEL PROPAGATION

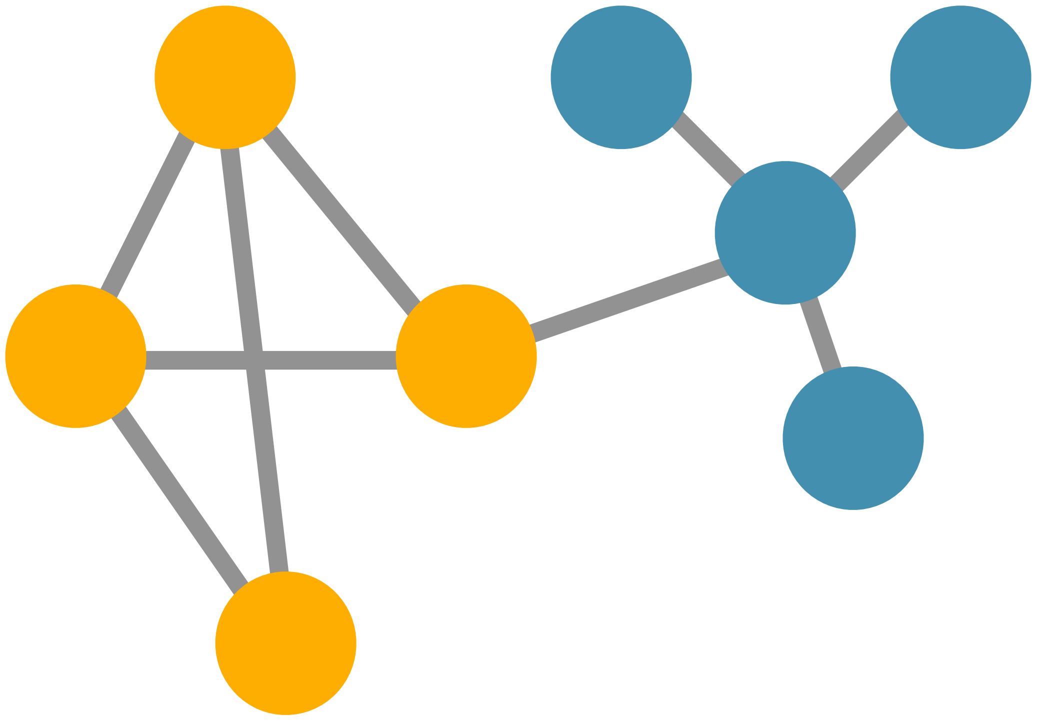


*Actually 1/3!!!

LABEL PROPAGATION



LABEL PROPAGATION



LABEL PROPAGATION

ISSUES

DIFFERENT RUNS FIND DIFFERENT COMMUNITIES

NEEDS TO BE RUN MULTIPLE TIMES

STRENGTHS

VERY FAST

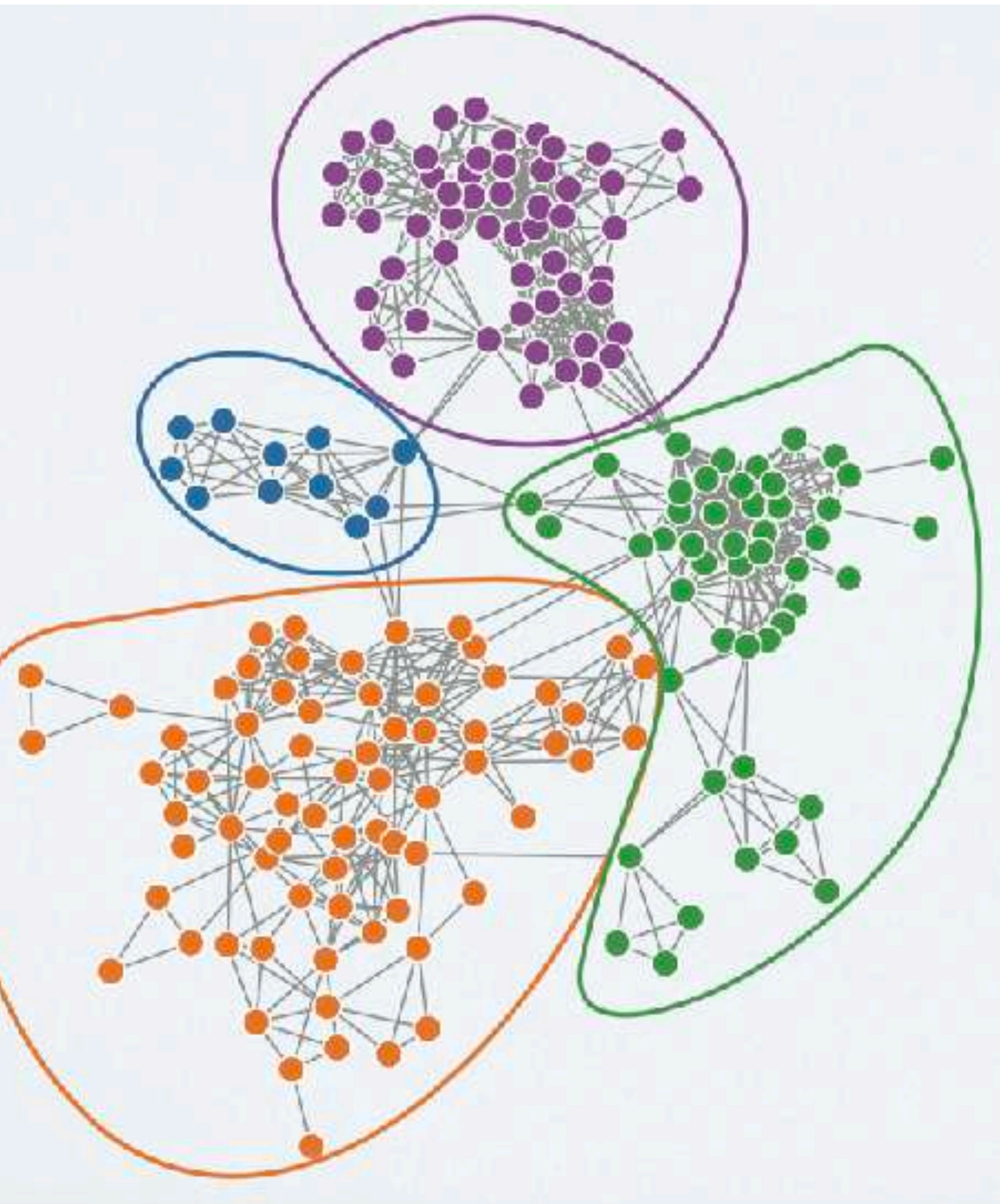
IF SOME MEMBERSHIPS ARE KNOWN, THEY CAN BE USED TO INITIALISE THE NETWORK

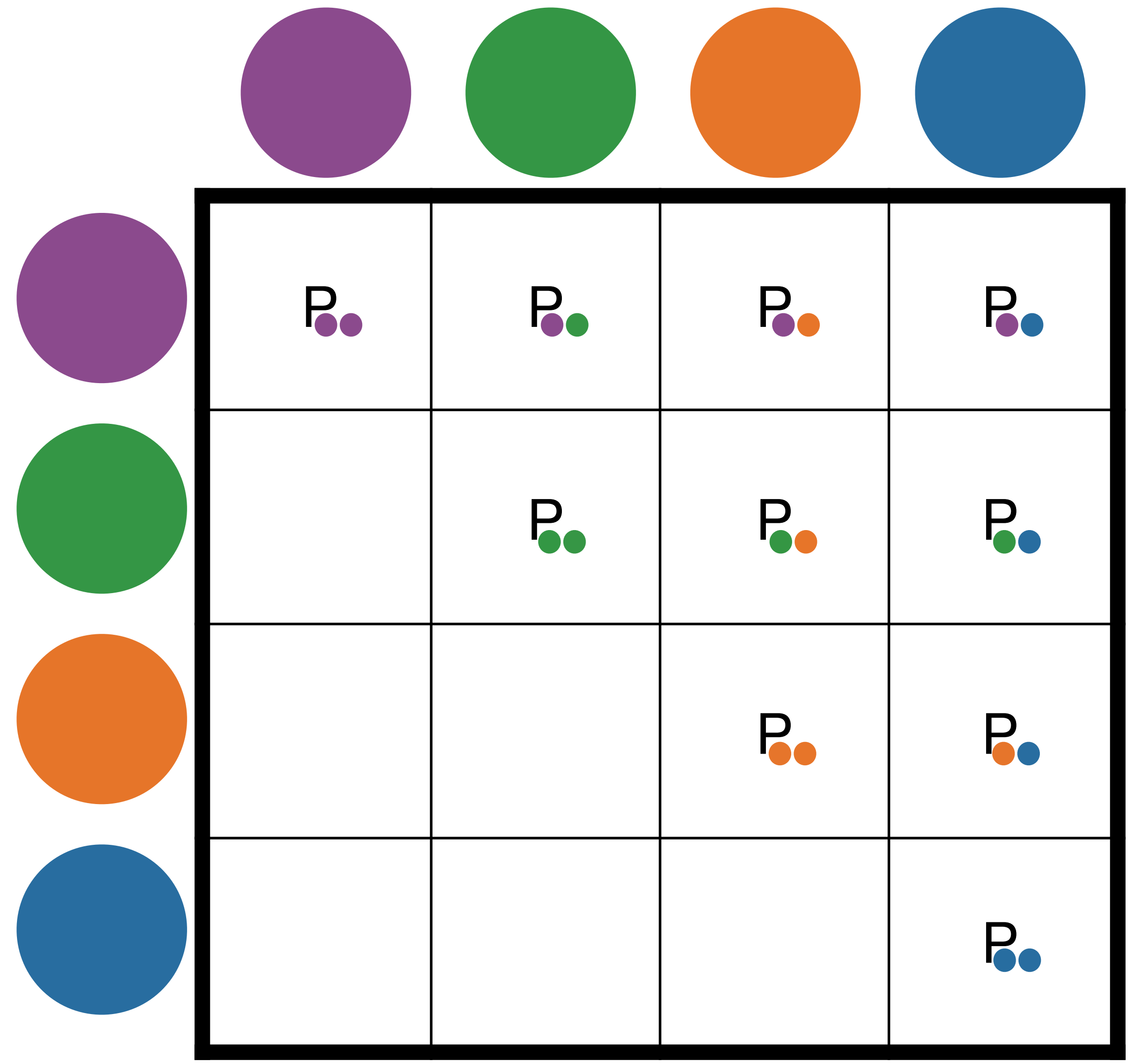
STOCHASTIC BLOCK MODEL

Generative algorithm

generates communities with given probabilities, chooses the most likely

STOCHASTIC BLOCK MODEL





STOCHASTIC BLOCK MODEL

CAN PERFORM COMMUNITY DETECTION ON A LOT OF DIFFERENT NETWORK TYPES

FOR EXAMPLE: IF $\forall r, p_{rr} = 0$ **THIS REPRESENTS MULTIPARTITE NETWORKS**

STOCHASTIC BLOCK MODEL

CAN PERFORM COMMUNITY DETECTION ON A LOT OF DIFFERENT NETWORK TYPES

And can discover more than just communities

$\forall r, s$	$p_{rr} > p_{rs}$	**Classic communities**
	$p_{rr} < p_{rs}$	**Disassortative structure**
$\forall r$	$p_{rr} = 0$	**Multipartite network**

STOCHASTIC BLOCK MODEL

CAN PERFORM COMMUNITY DETECTION ON A LOT OF DIFFERENT NETWORK TYPES

And can discover more than just communities

$\forall r, s \quad p_{rr} > p_{rs}$ **Classic communities**

$p_{rr} < p_{rs}$ **Disassortative structure**

$\forall r \quad p_{rr} = 0$ **Multipartite network**

$\forall r, s \quad p_{rr} = p_{rs} = p$ **Random network**

STOCHASTIC BLOCK MODEL

LIMITS:

NEEDS PRIOR KNOWLEDGE ON NUMBER OF COMMUNITIES

STRENGHTS:

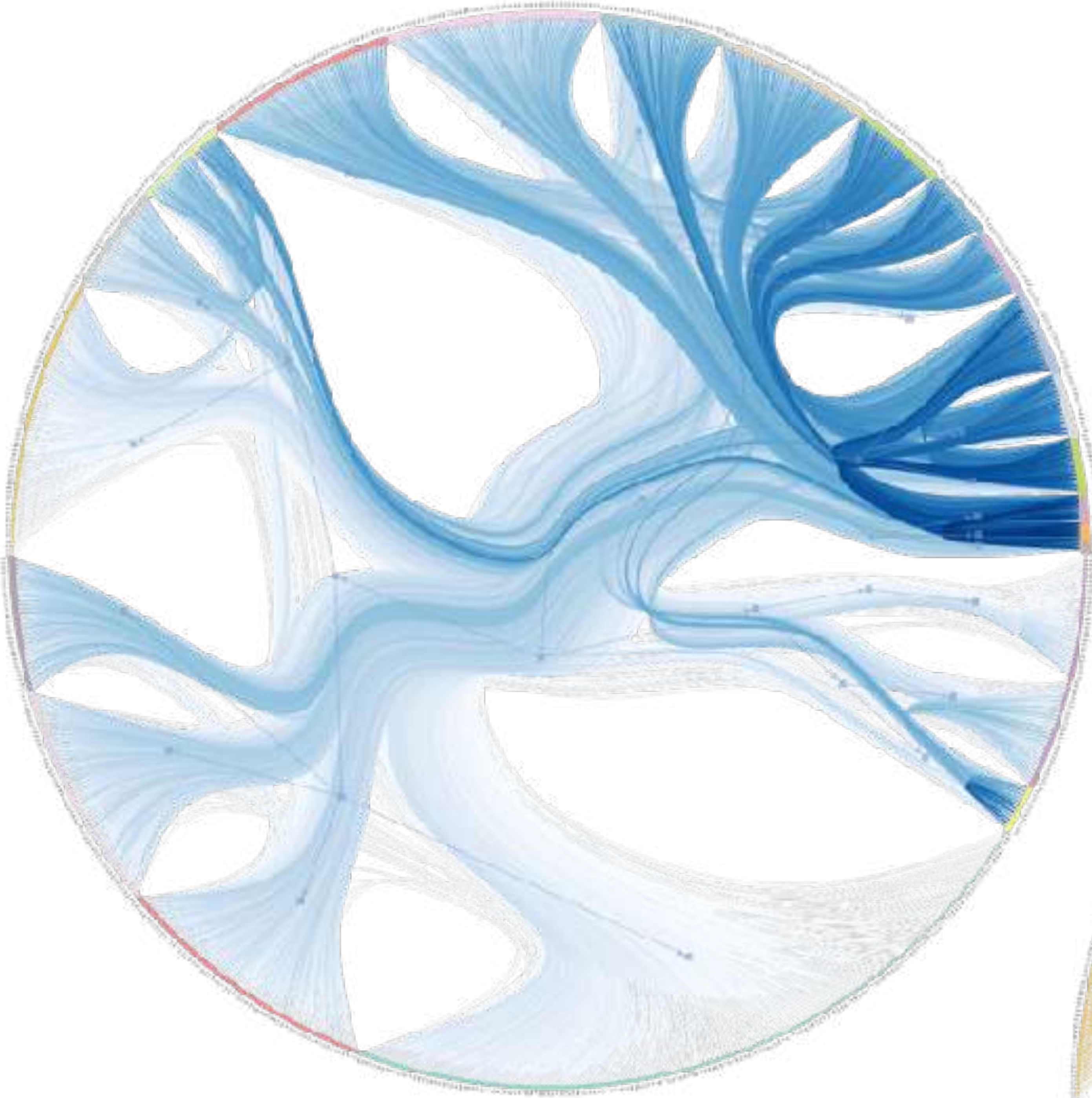
EVERYTHING ELSE

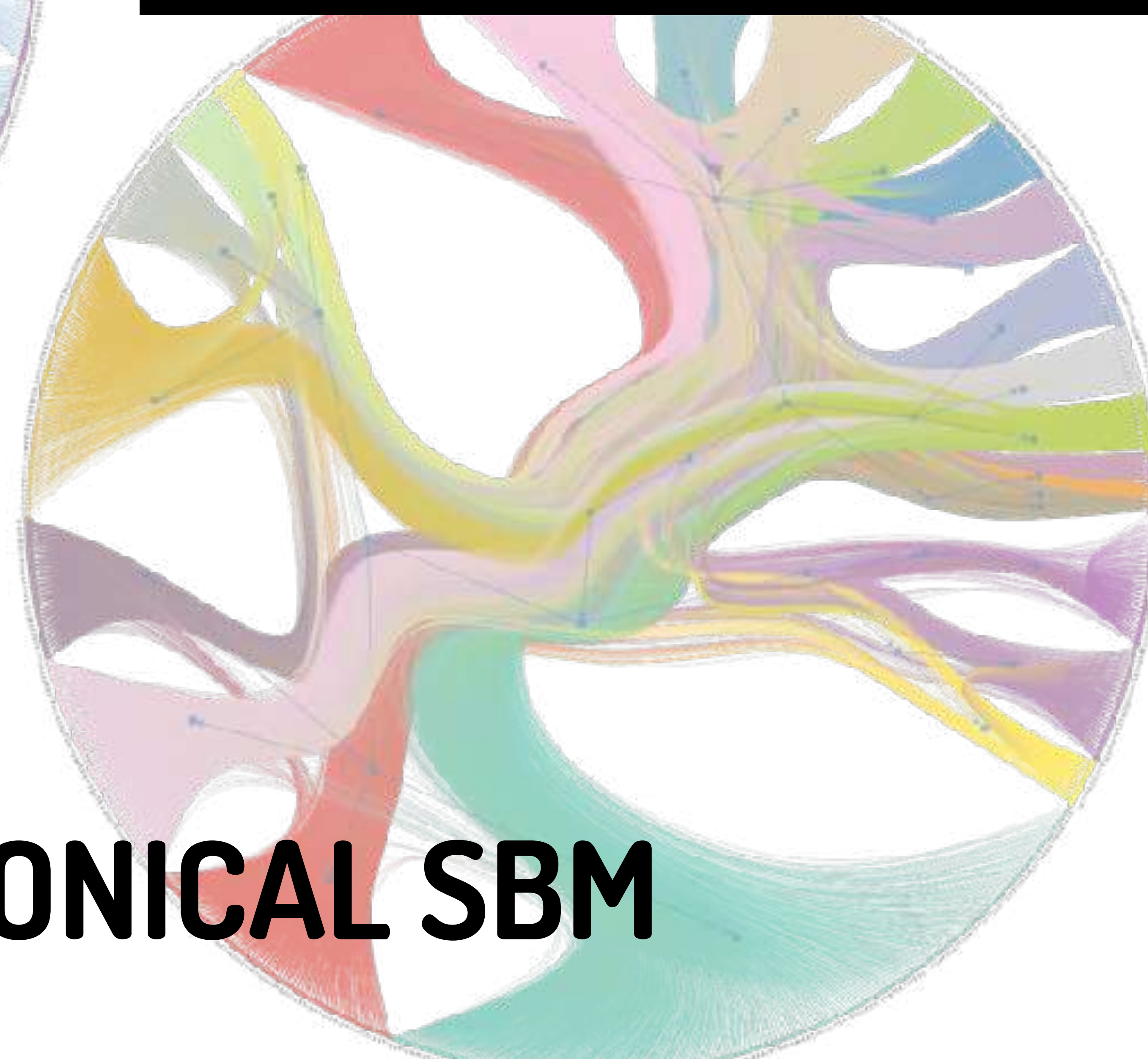
MICROCANONICAL SBM

Uses Bayesian inference

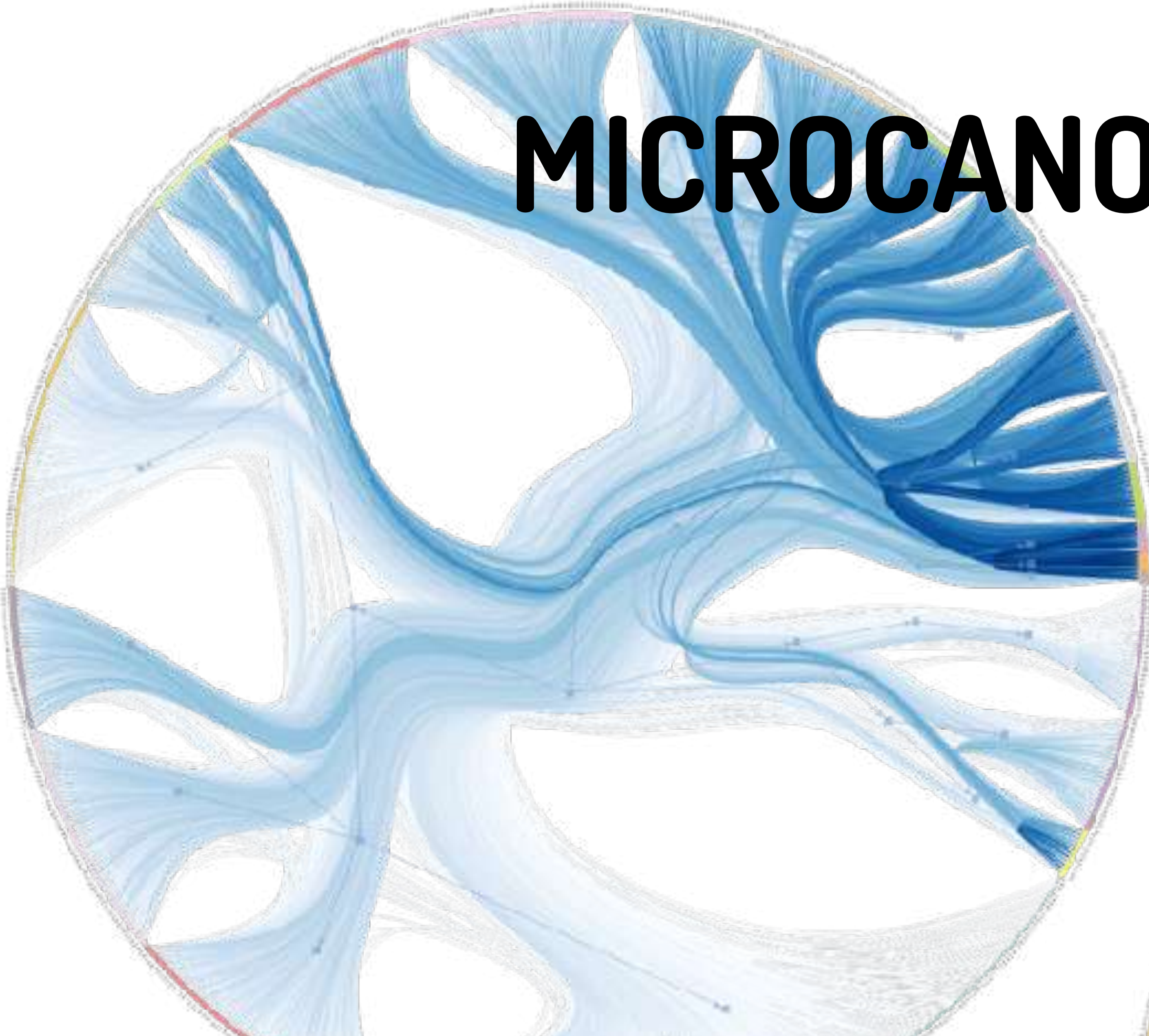
Does not require prior knowledge

Extremely versatile





MICROCANONICAL SBM



Fast and scalable

Explainable

There's a library that does it all and produces beautiful figures