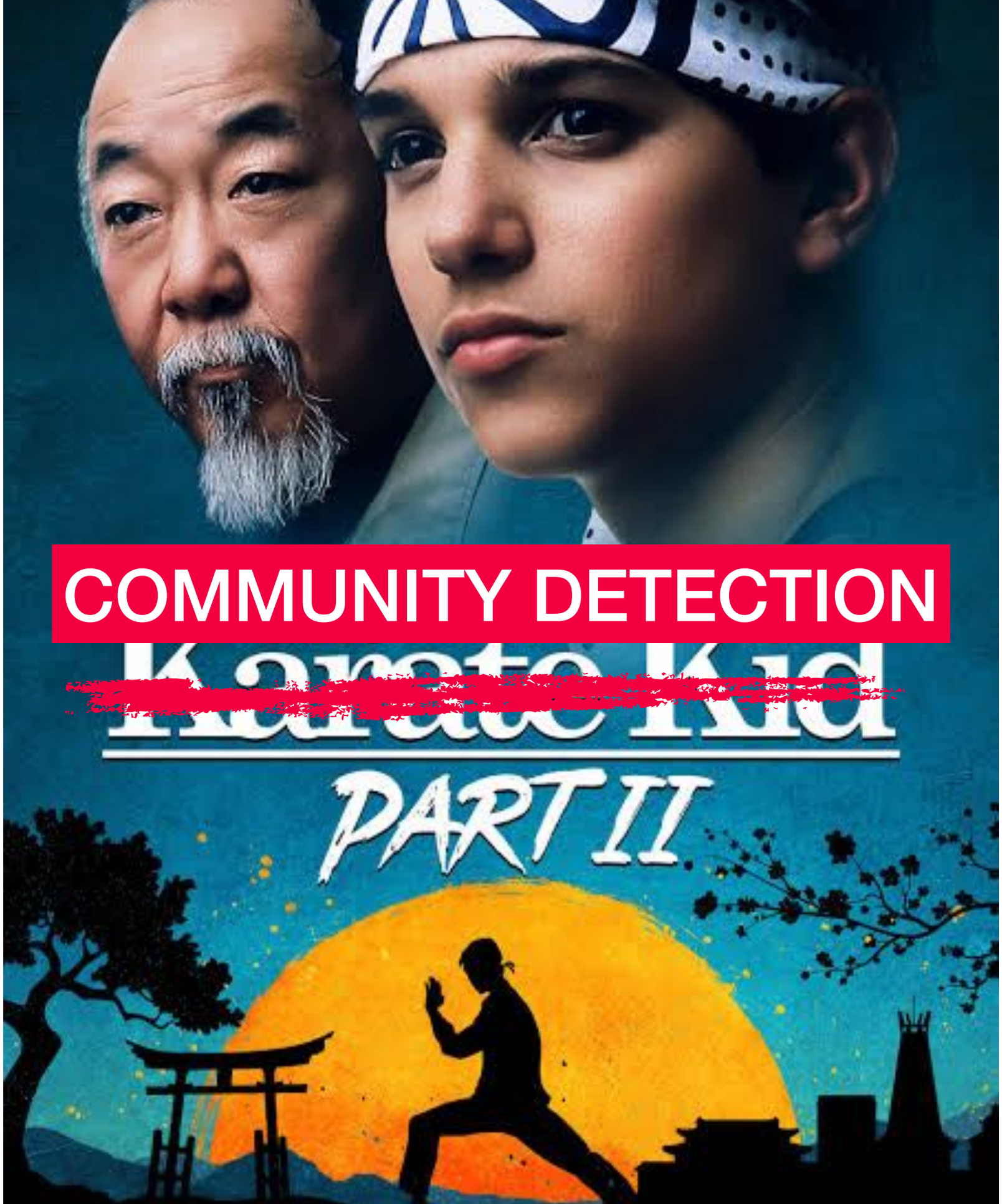

COMMUNITY DETECTION
Karate Kid
PART II

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

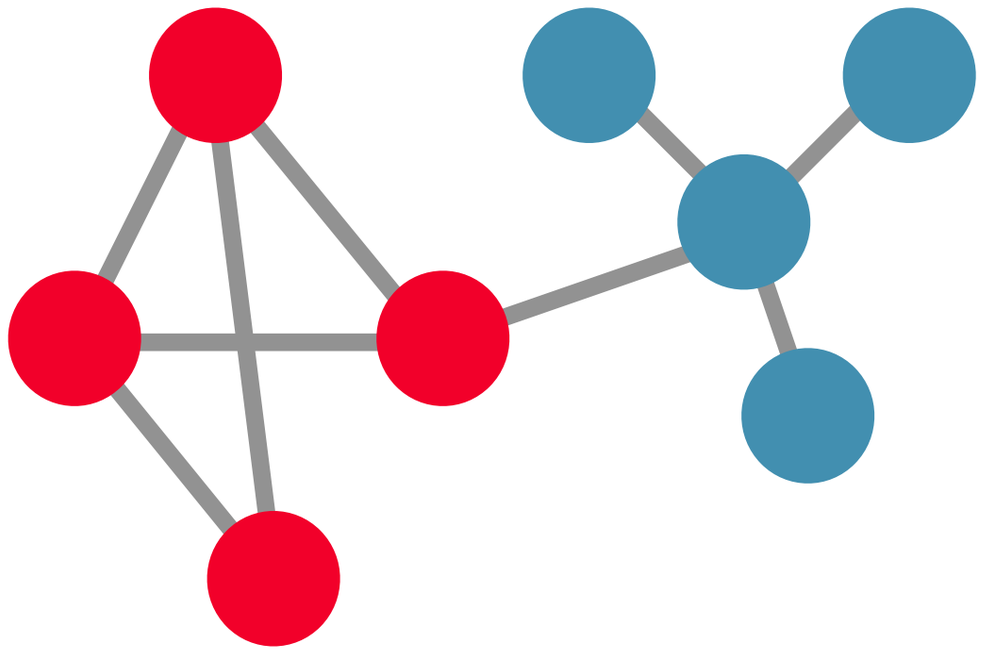
Bridge removal

Modularty maximisation

Label propagation

Stochastic block modelling

BRIDGE REMOVAL



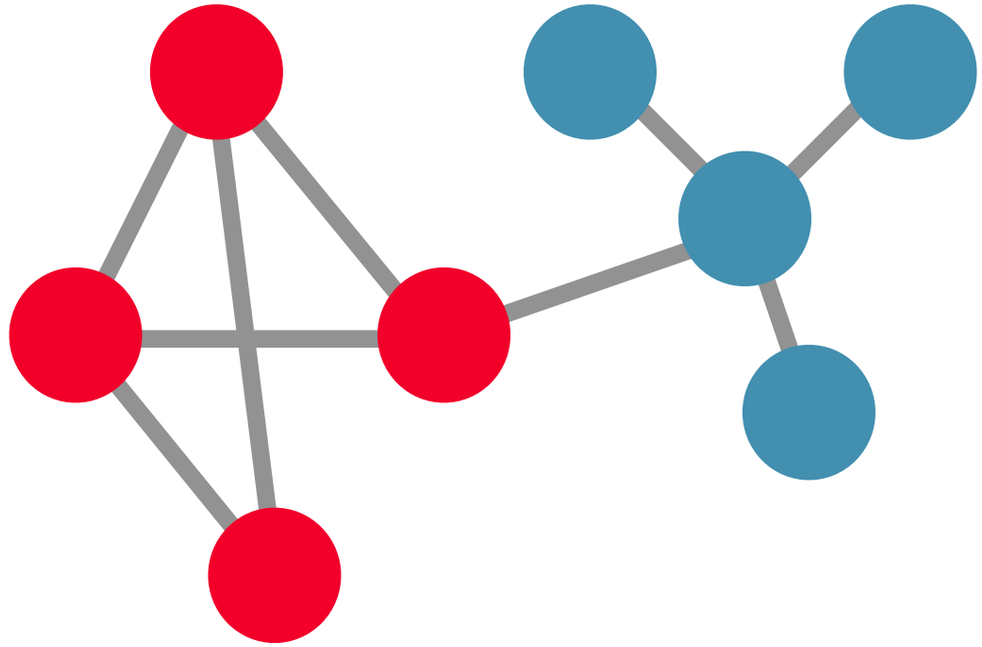
A bridge is a link whose removal breaks the network into two parts

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

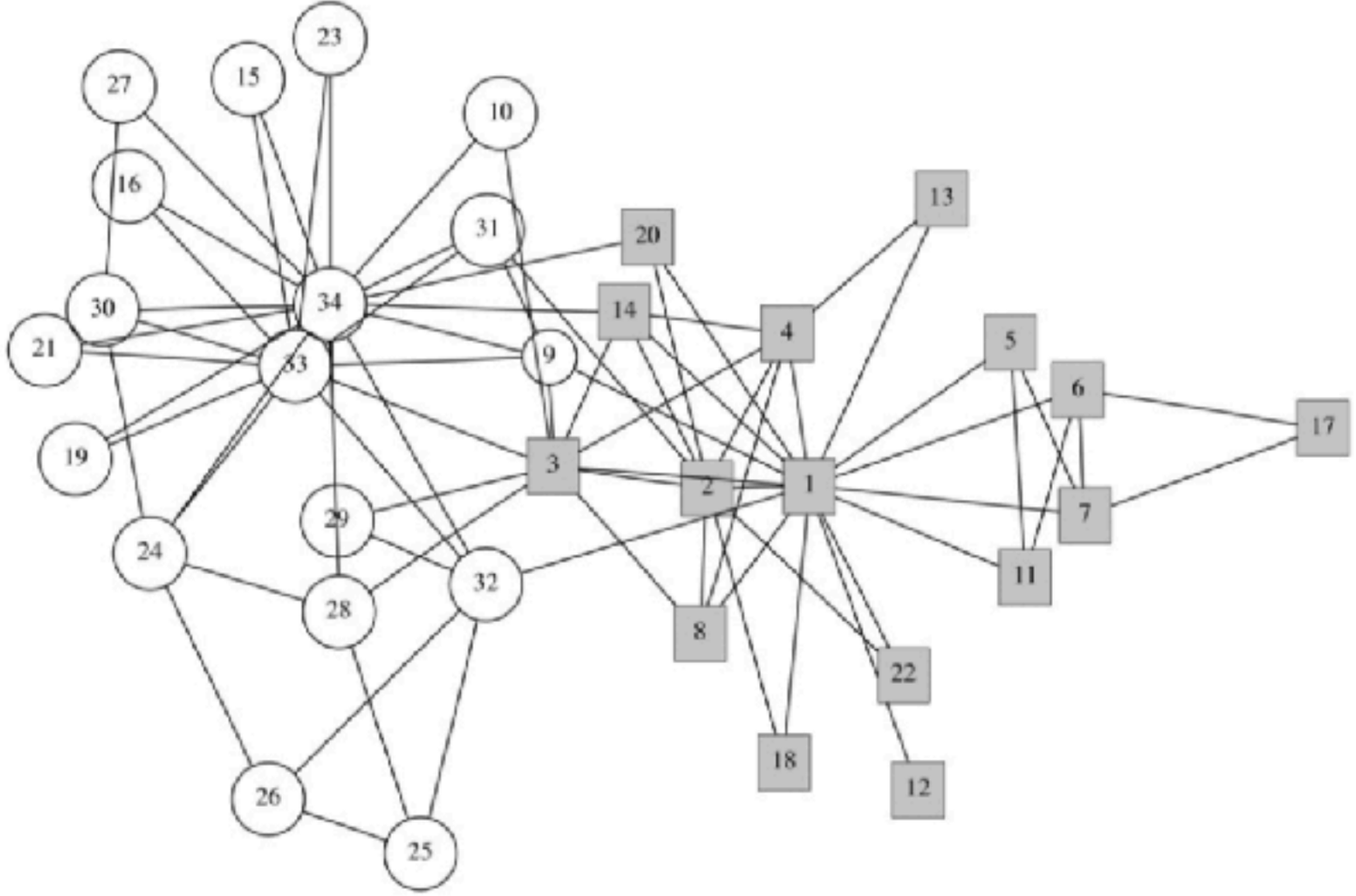
The most famous algorithm based on this approach is the **Girvan-Newman algorithm**

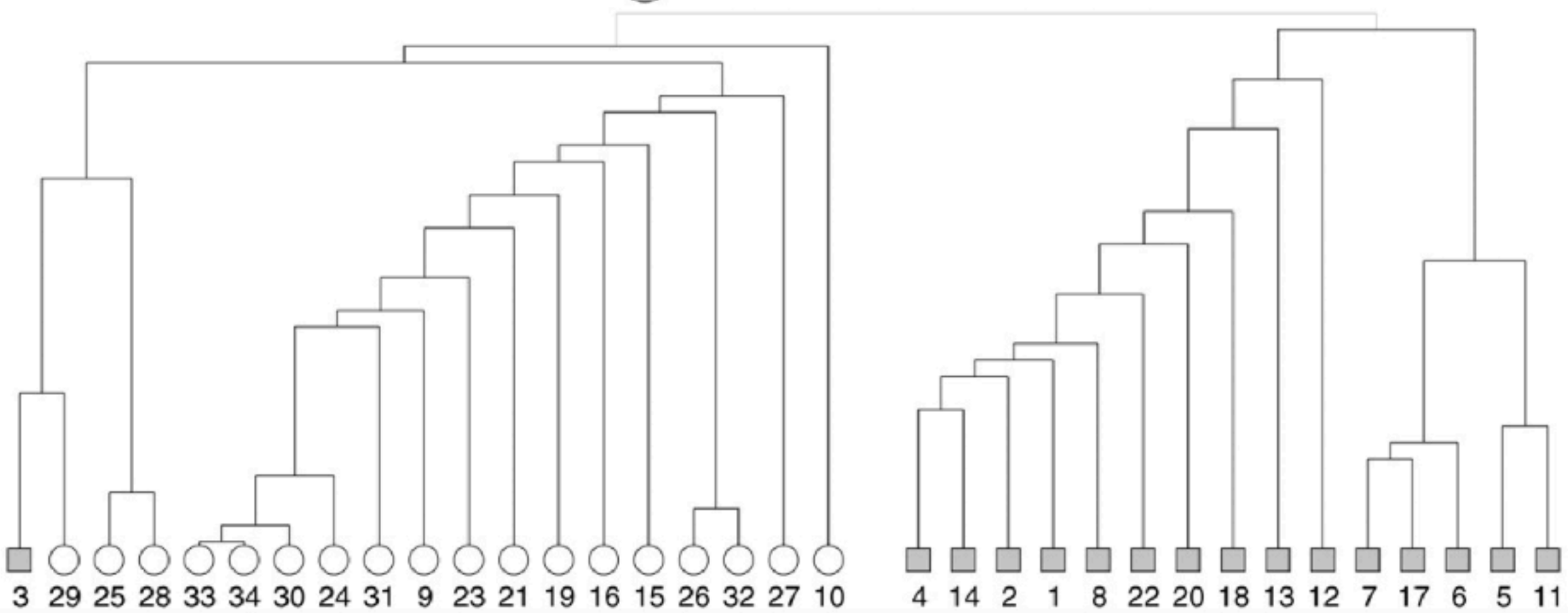
1 - compute link **betweenness** for all the links

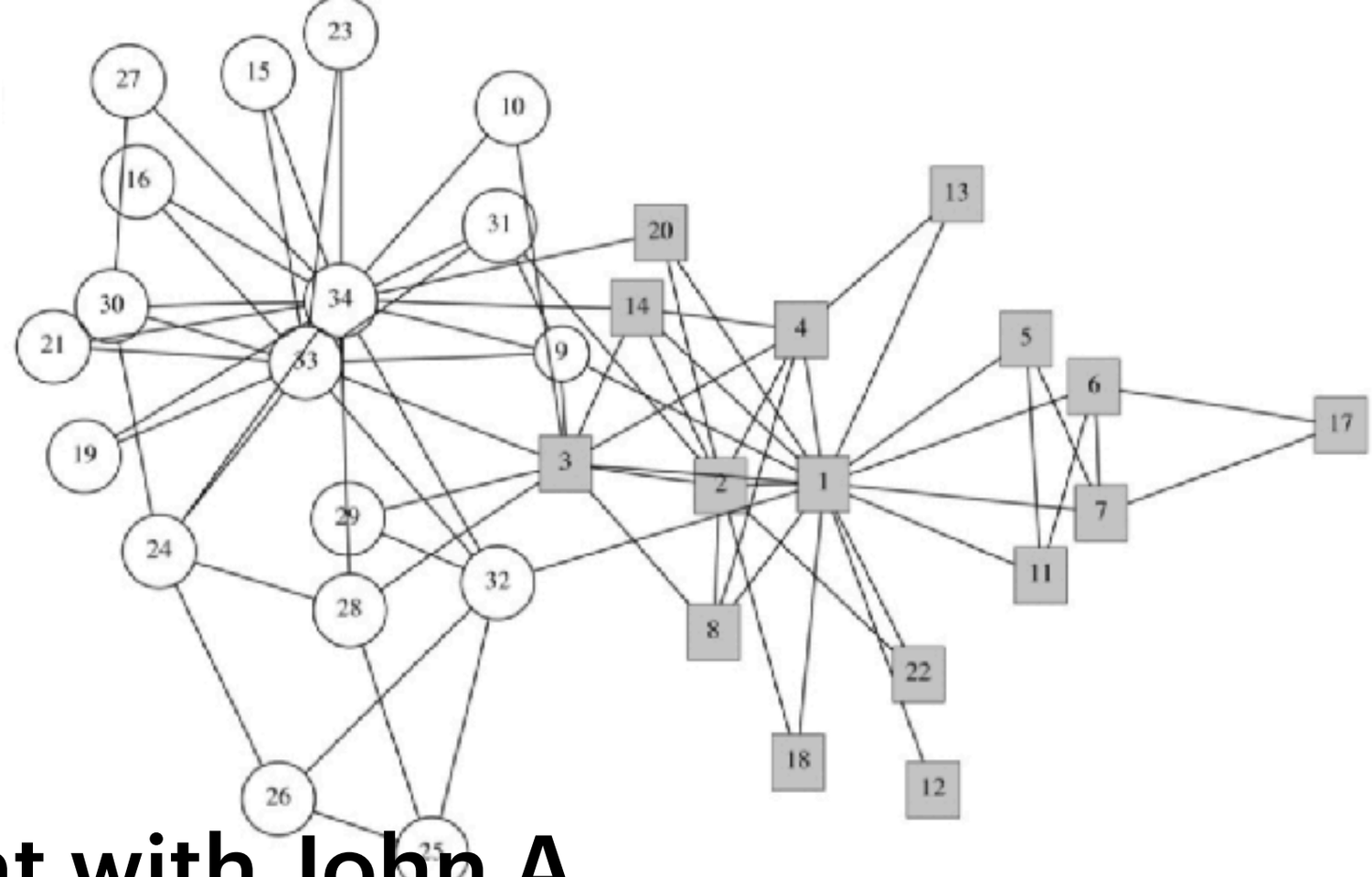
2 - **remove** the link with highest betweenness*

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

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

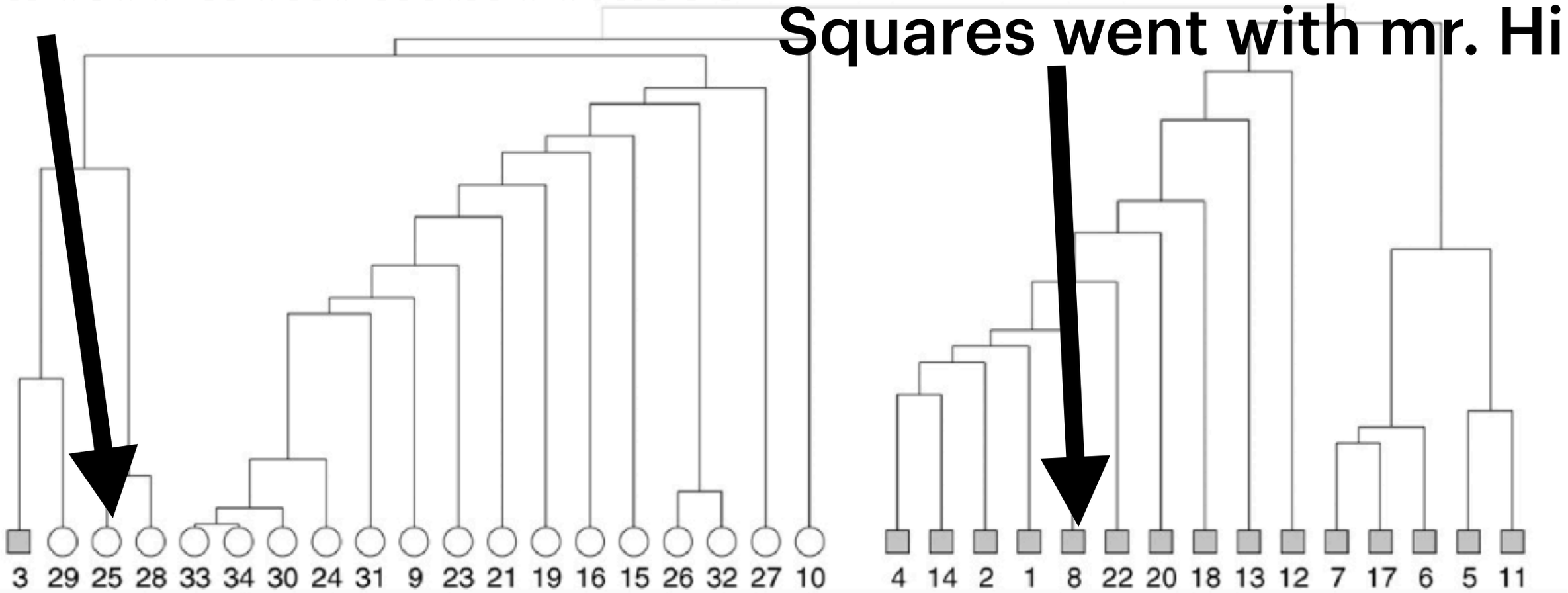

23
27
15
10
16
31
20
13
30
34
14
4
5
21
33
9
6
17
19
3
2
1
7
29
24
32
11
28
8
22
18
12
26
25


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



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

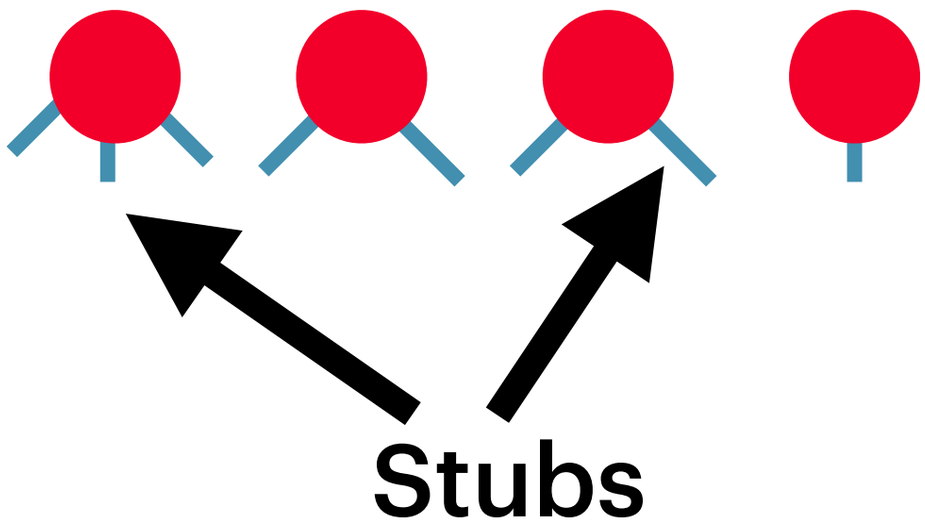
$$Q = \frac{1}{L} \sum_{C} \left(L_C - \frac{k_C^2}{4L} \right)$$

MODULARITY MAXIMISATION

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

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

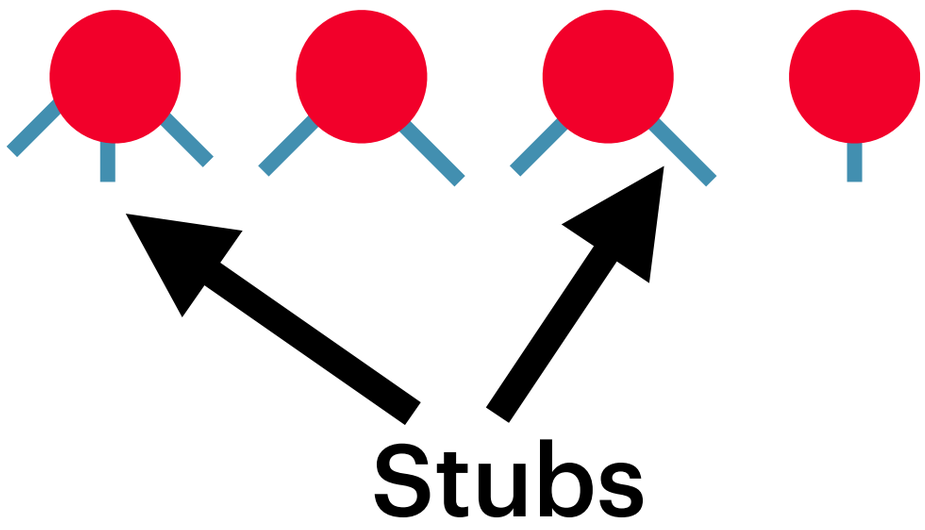
MODULARITY MAXIMISATION



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

$\frac{k_C}{2L}$ **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**

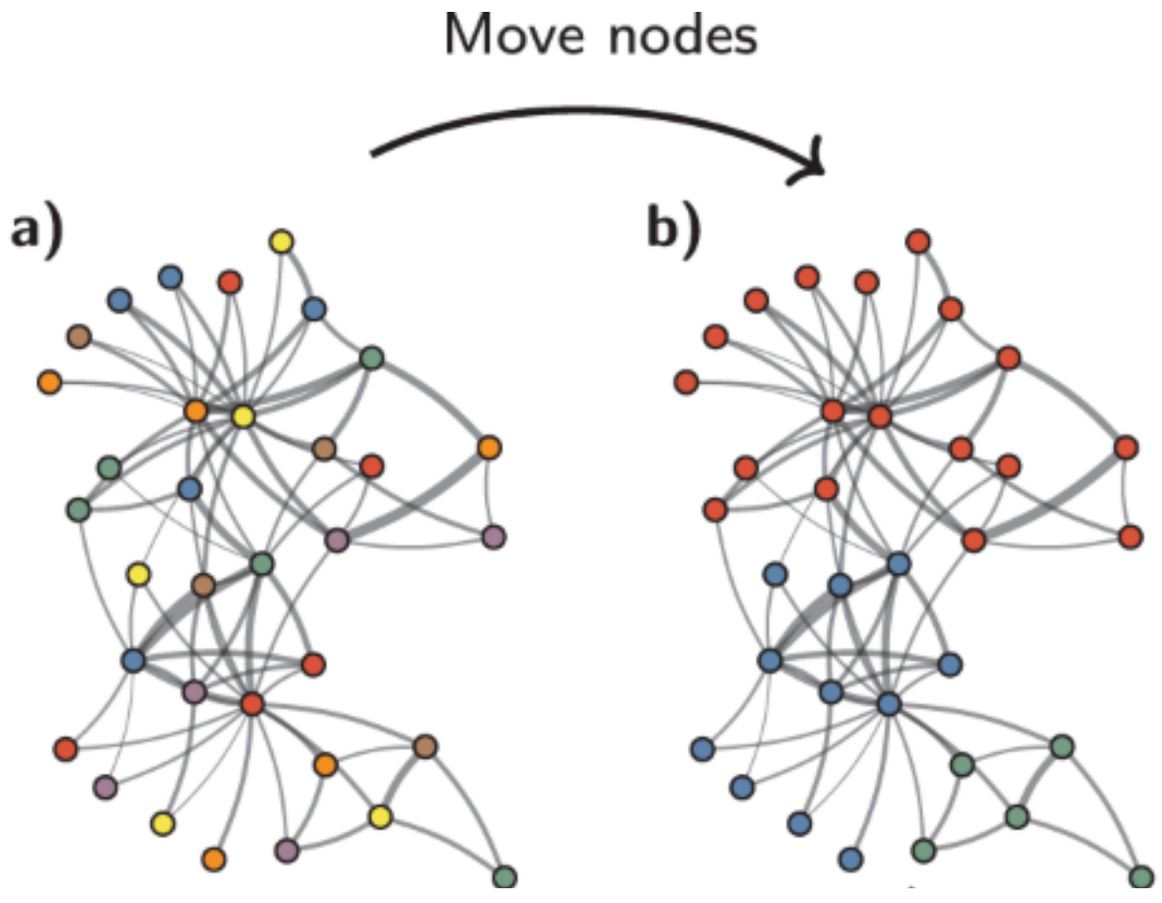
1) start with no communities. Every nodes is moved to a community so that **Q** Is maximised. Repeat until no modularity gain is possible

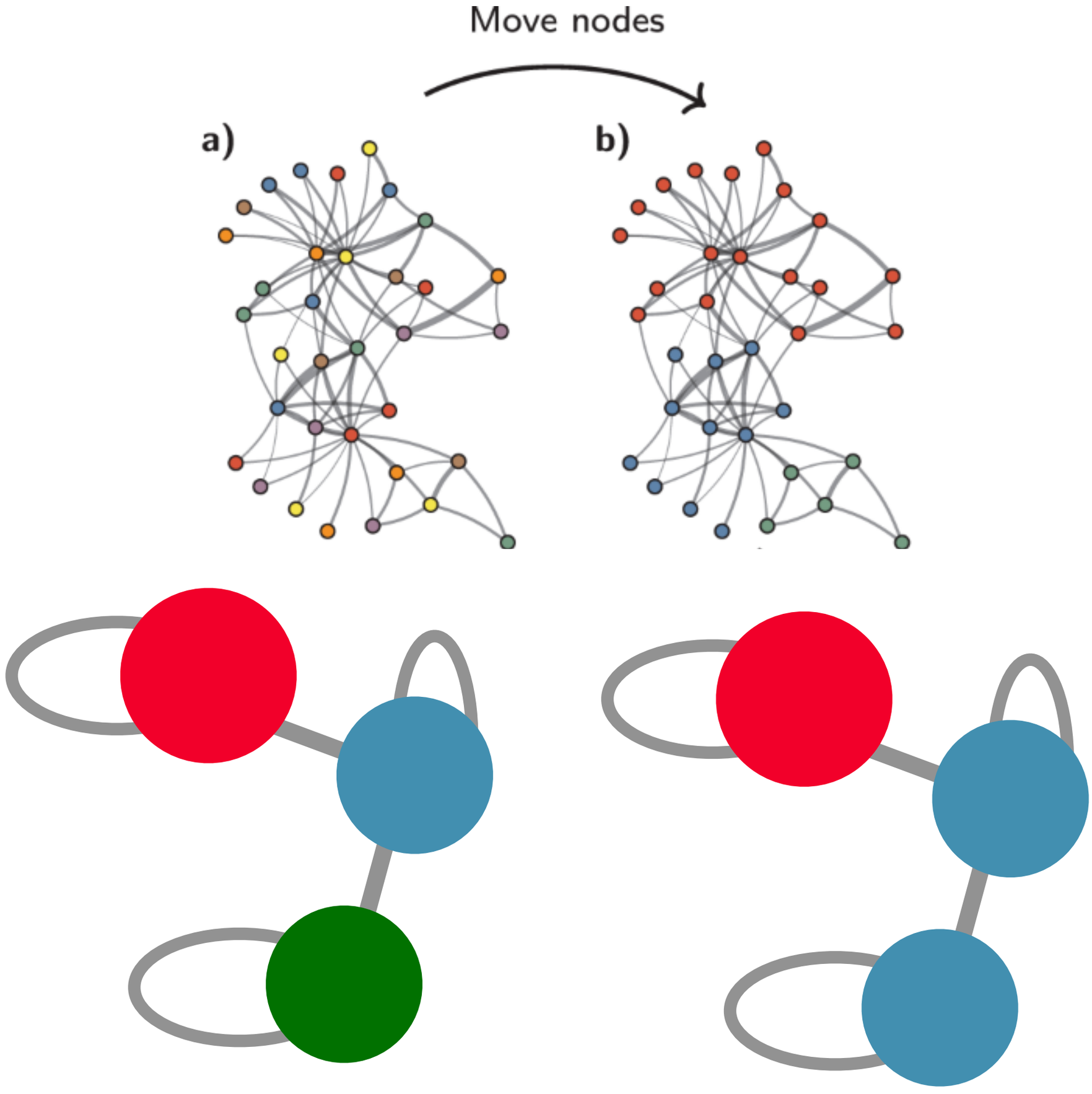
MODULARITY MAXIMISATION

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

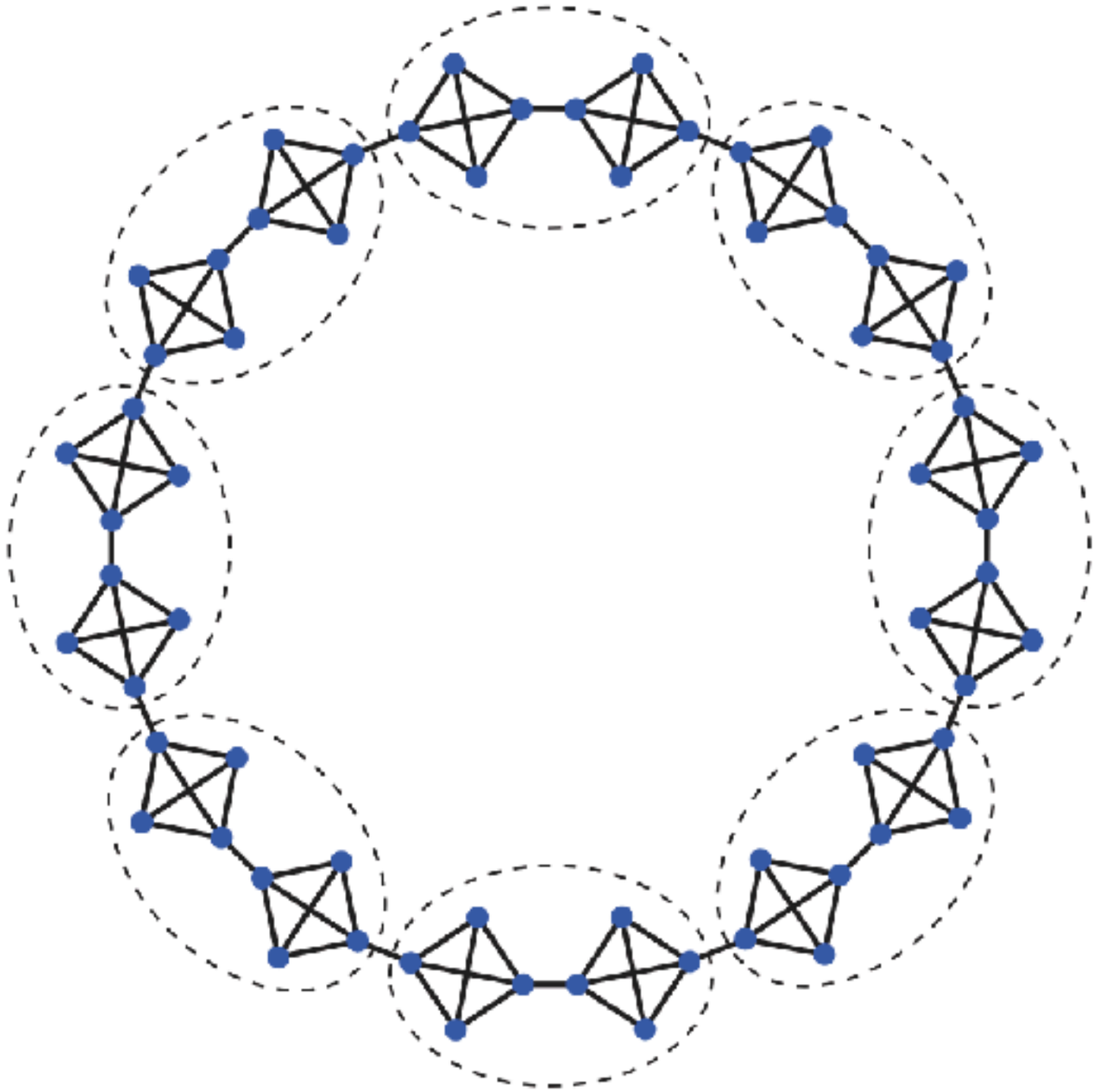
Comparison: On average Larger networks have larger modularity

Uncertainty: this approach can find positive modularity for random networks

Resolution: cannot find communities whose degree is smaller than

$$\sqrt{2L}$$

MODULARITY MAXIMISATION



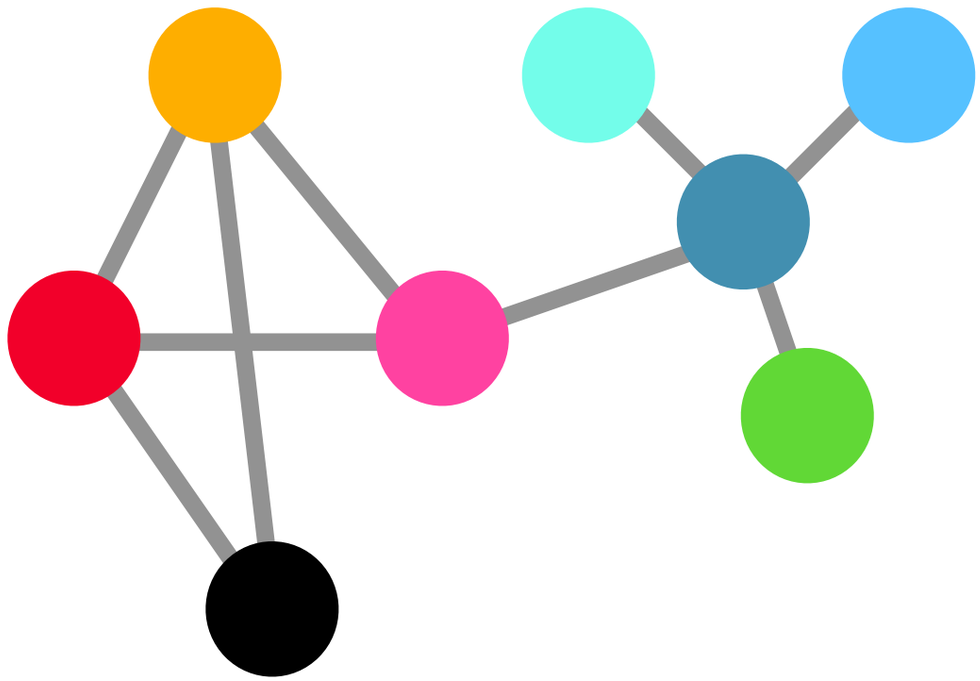
LABEL PROPAGATION

1) WE START WITH **SINGLETONS**

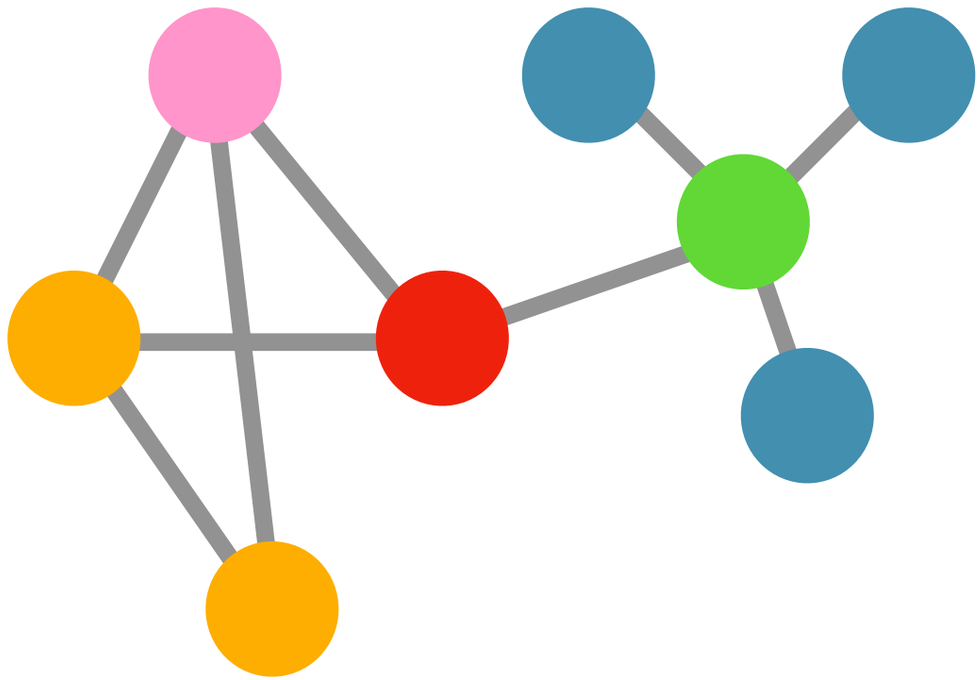
2) ONE BY ONE, WITH RANDOM ORDER, NODES TAKE THE "LABEL" (IE COMMUNITY MEMBERSHIP) OF THE **MAJORITY OF THEIR NEIGHBOURS**

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

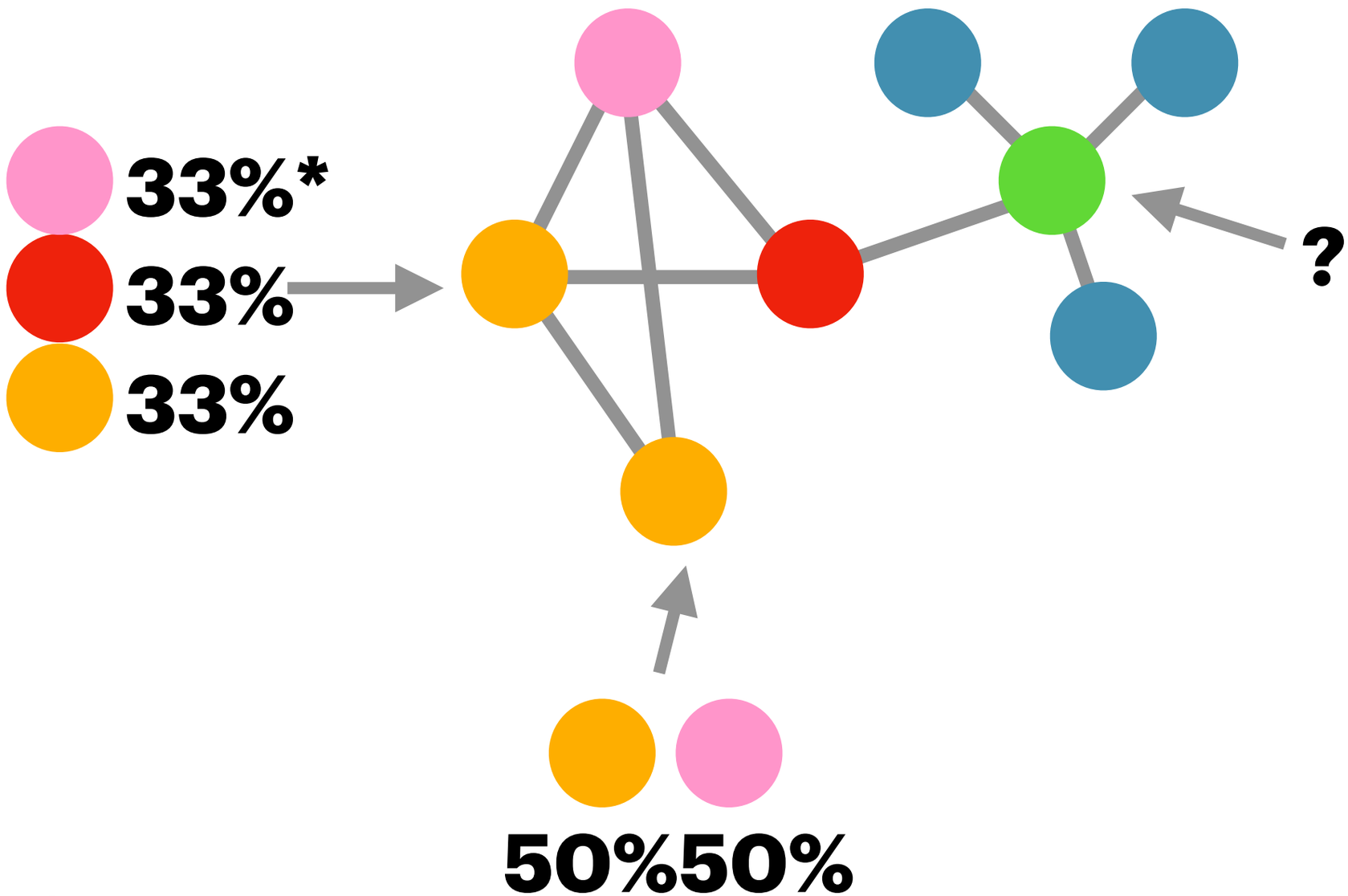
LABEL PROPAGATION



LABEL PROPAGATION

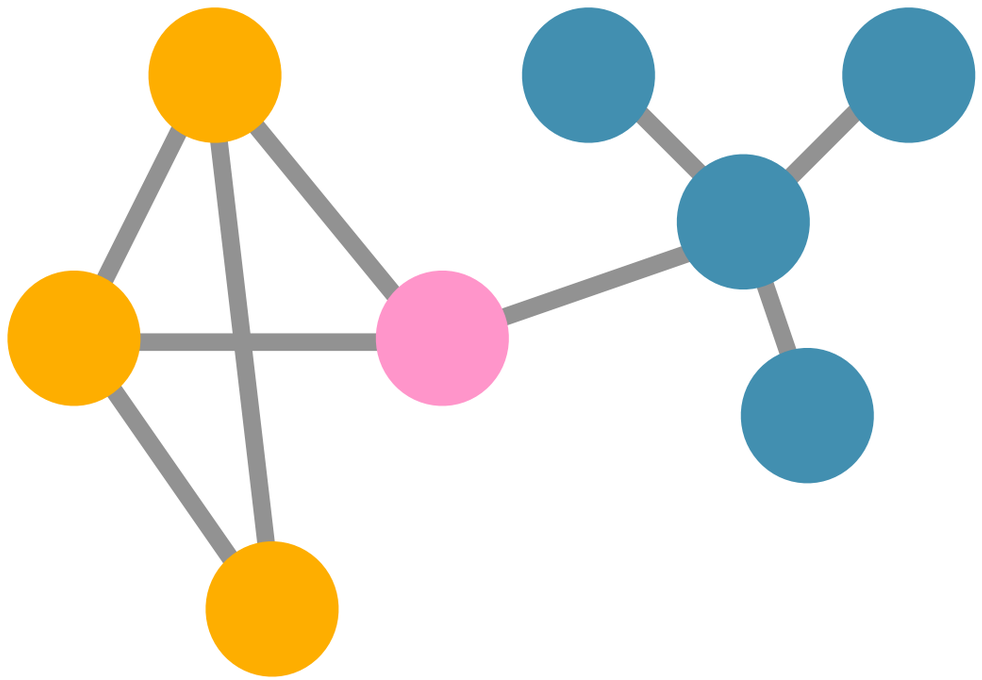


LABEL PROPAGATION

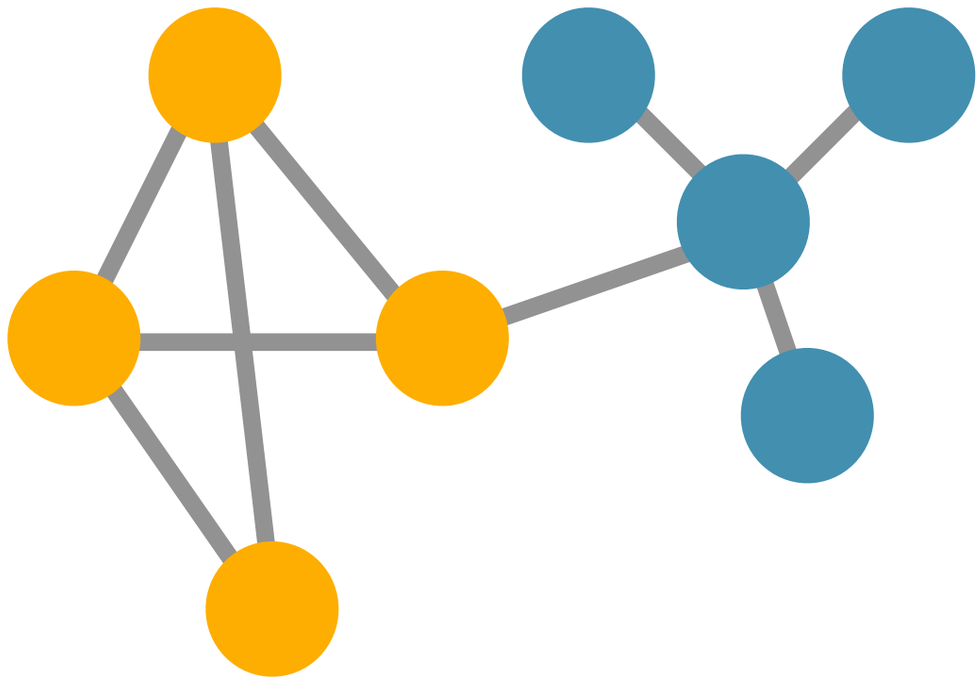


*Actually 1/3!!!

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

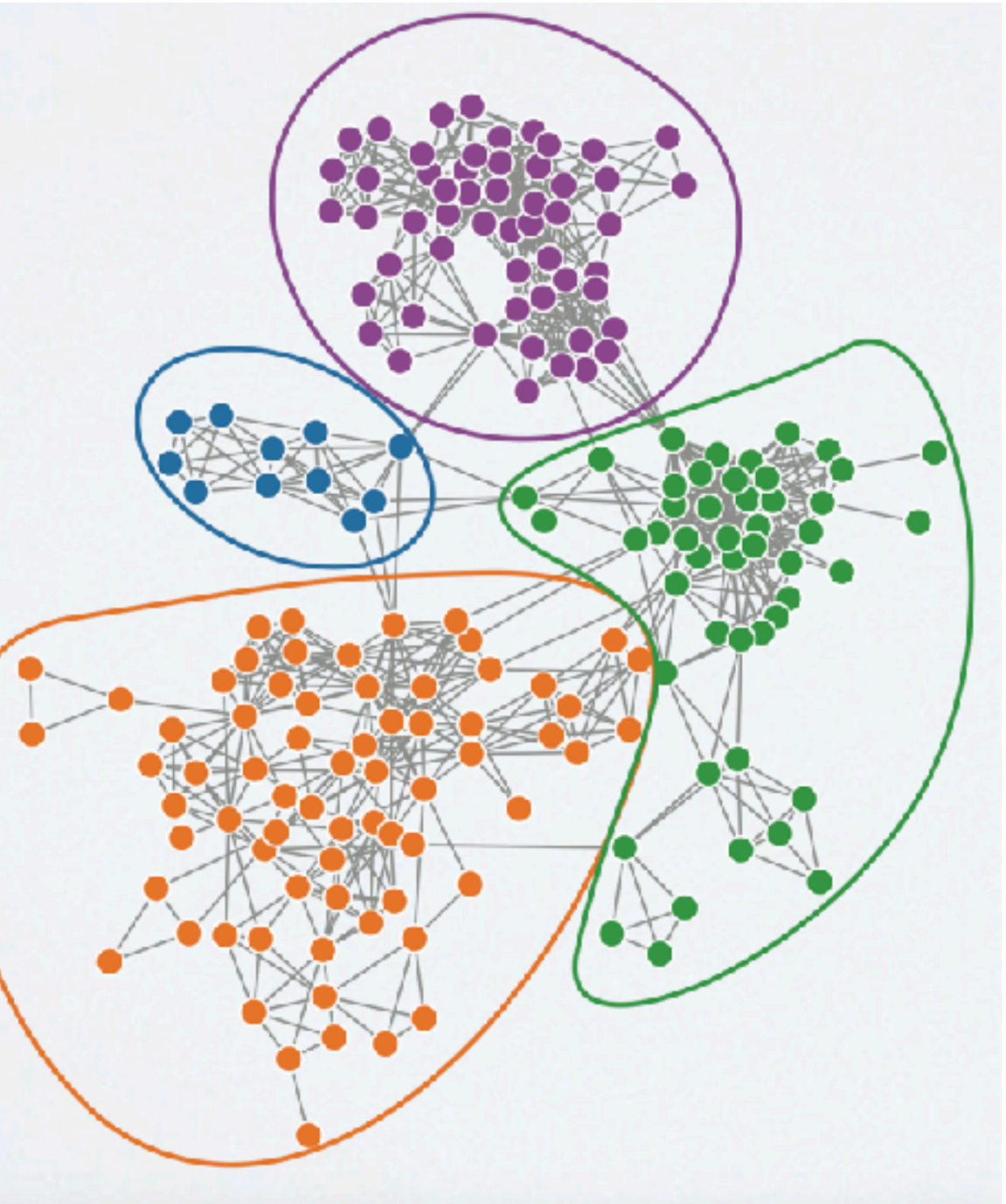
STOCHASTIC BLOCK MODEL

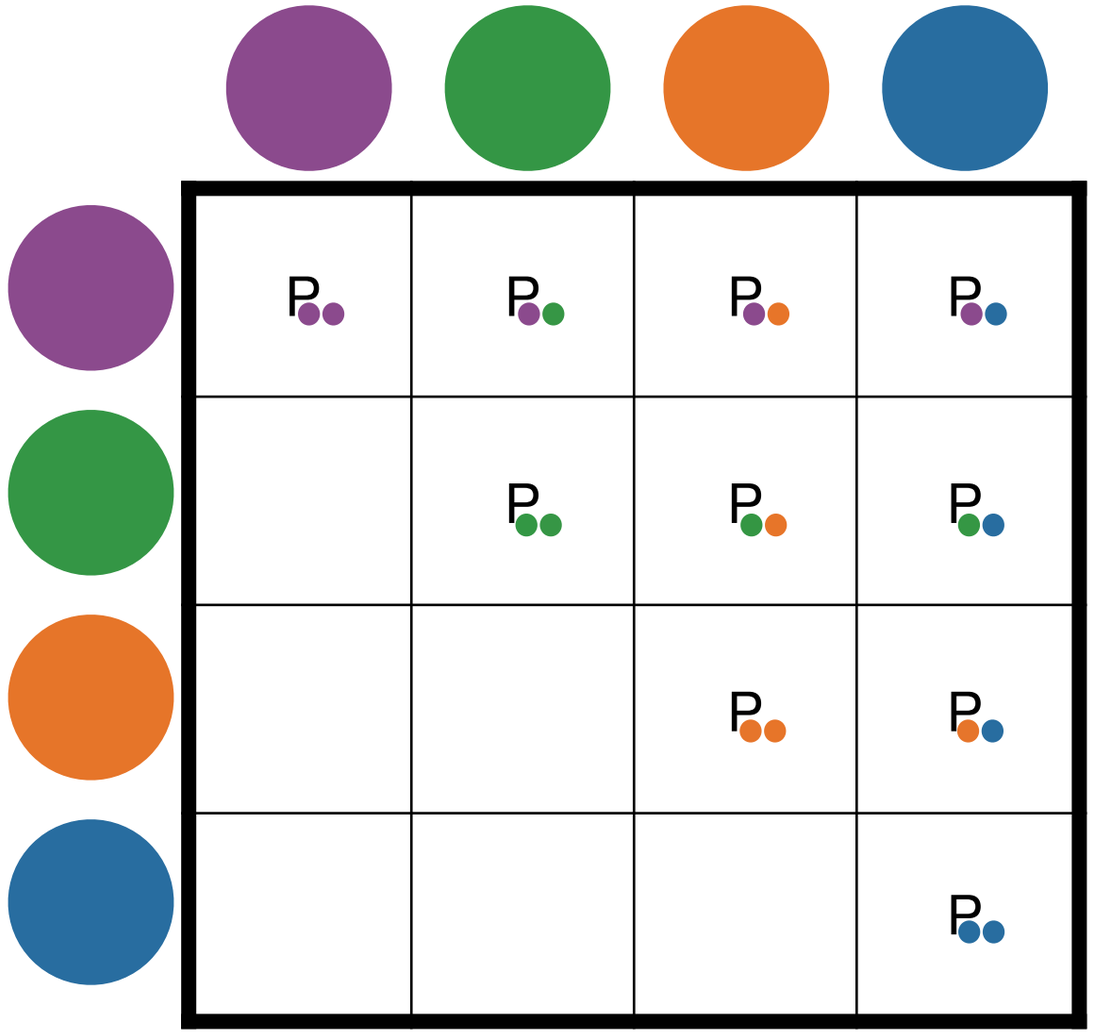
Generative algorithm

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

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

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

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

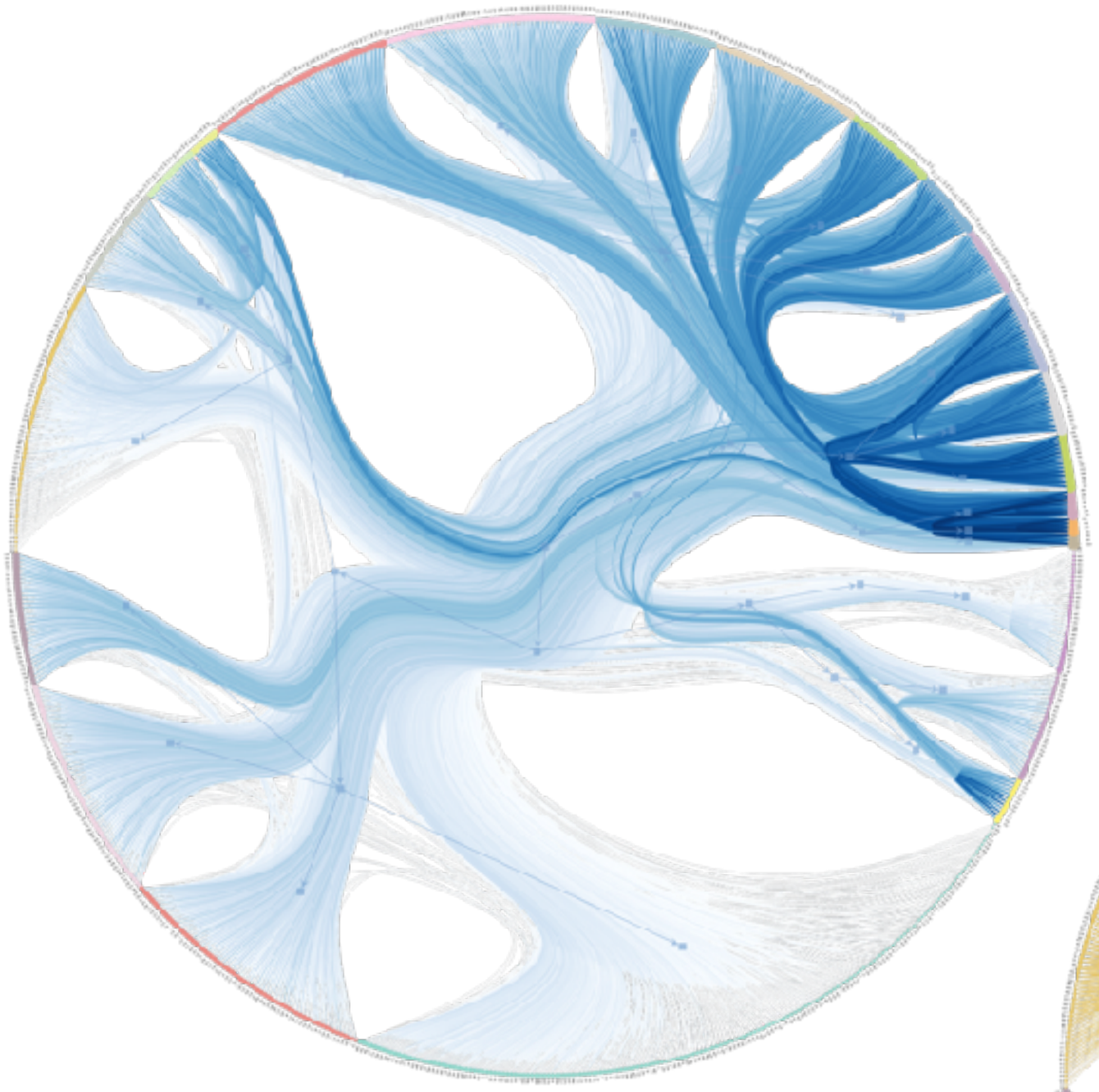
$$p_{rr} < p_{rs}$$ **Disassortative structure**

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



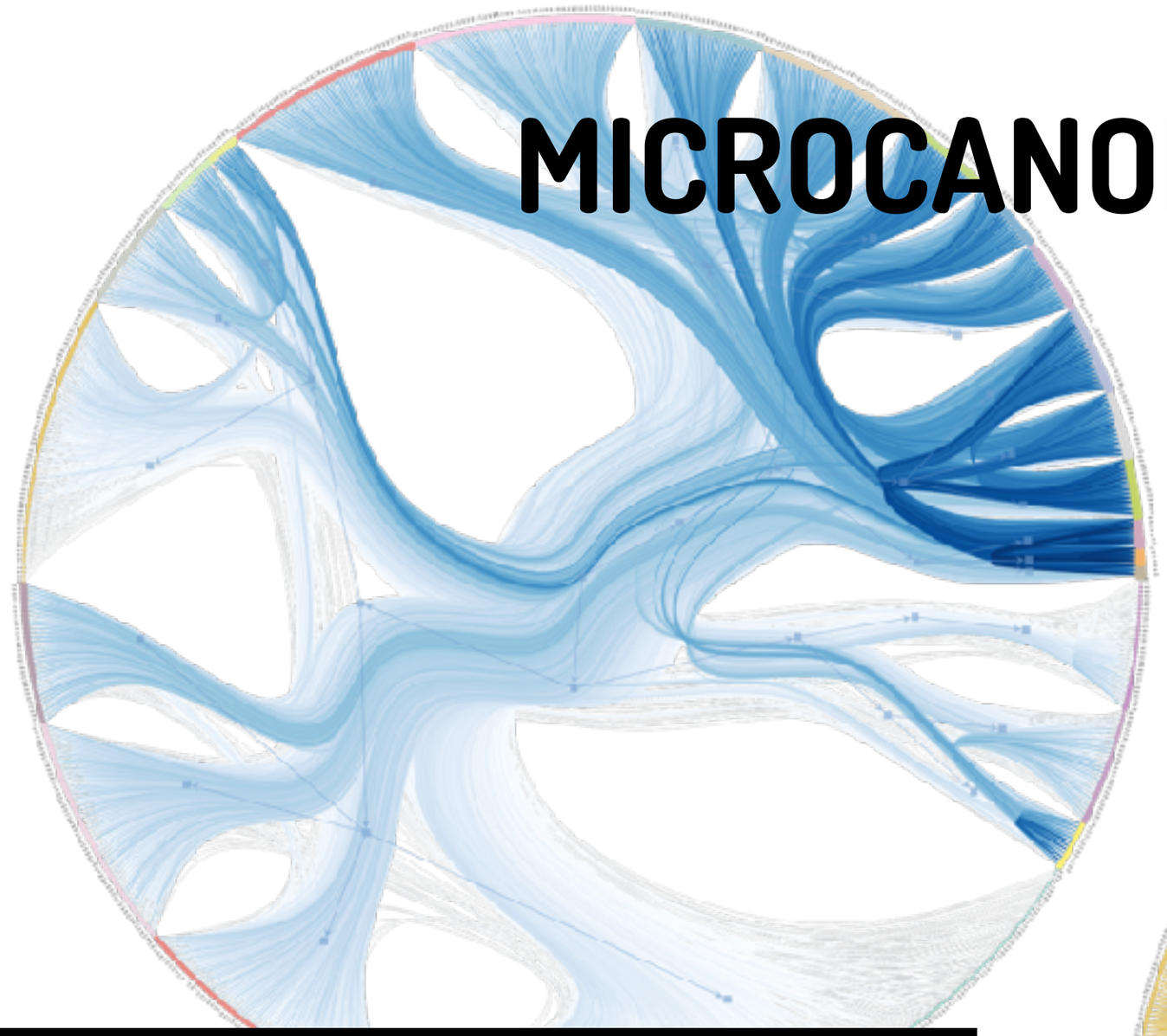
Uses Bayesian inference

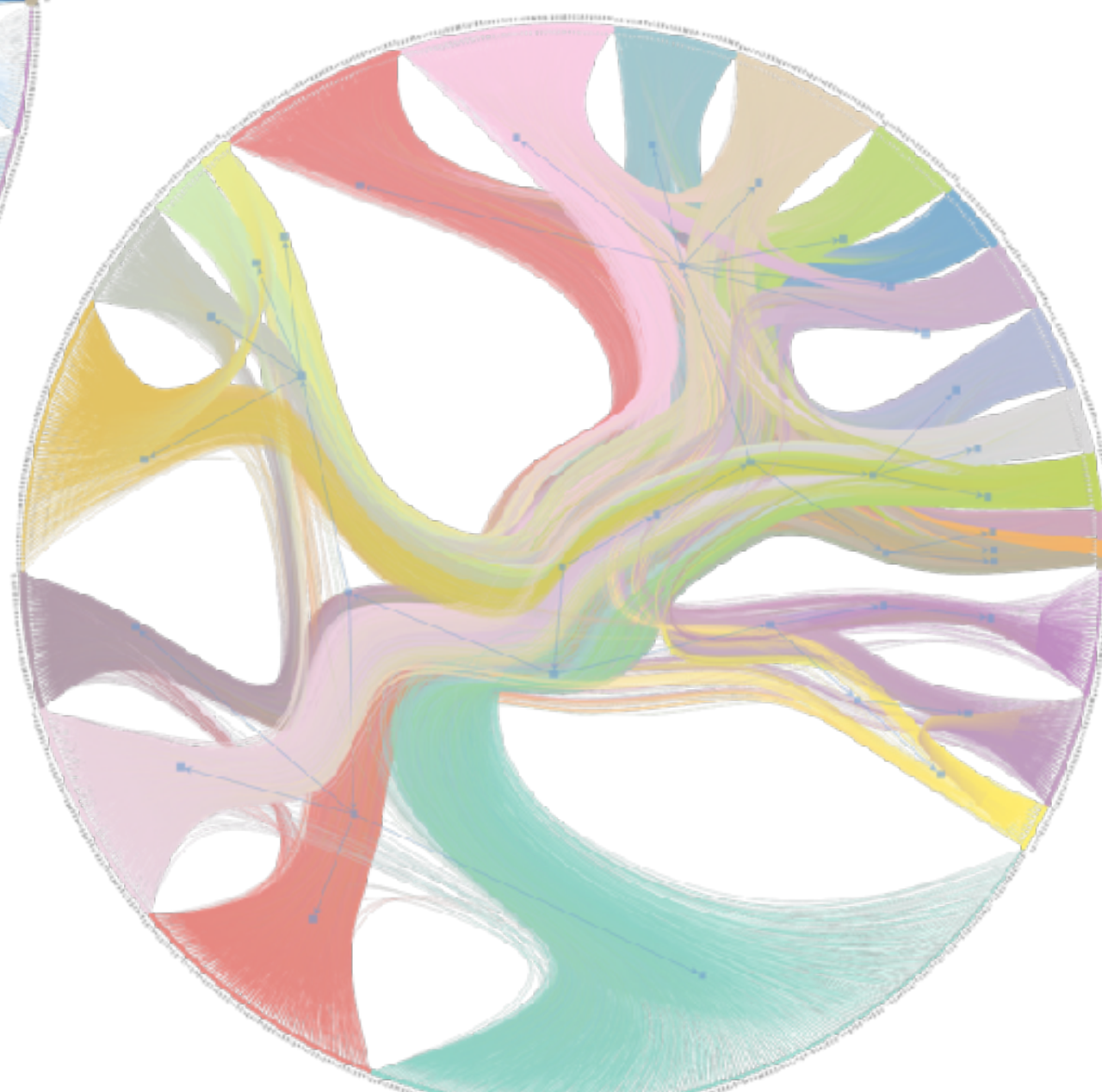
Does not require prior knowledge

Extremely versatile

MICROCANONICAL SBM

MICROCANONICAL SBM





Fast and scalable

Explainable

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