

A short course on

# Social Network Analysis

Lecture 1

Antonio Calcagni

`<antonio.calcagni@unipd.it>`

Master in Data Science per il Welfare  
University of Salento

AY 2023/24

# Outline

1 From data to networks

2 Descriptive analysis of network graphs

# Outline

1 From data to networks

2 Descriptive analysis of network graphs

# From data to networks

Data that are already represented as a network do exist. These datasets **naturally embody** the structure of nodes and edges without requiring transformation from another format usually not structured as a network.

Examples of these include: internet topology data, citation networks, social network data, transportation networks, neural and gene networks, financial networks.

# From data to networks

Quite generally, the process of constructing a **network graph representation** from a system of interest based on a set of measurements from that system is largely informal.

Indeed, users need to specify what should constitute a vertex and an edge to build up the associated graph. There could be many potential graph representations for the same set of measurements.

Two graphs  $\mathcal{G}$  and  $\mathcal{H}$  are said to be **isomorphic** if their structure remains unchanged after relabelling their vertices and edges.

# From data to networks

Although a graph  $\mathcal{G}$  is just a pair of particular sets, the geometry for representing  $\mathcal{G}$  can be quite informal unless one defines a problem of **network topology inference**.

# From data to networks

Although a graph  $\mathcal{G}$  is just a pair of particular sets, the geometry for representing  $\mathcal{G}$  can be quite informal unless one defines a problem of **network topology inference**.

Given a set of measurements and a set of candidates  $\{\mathcal{G}_1, \mathcal{G}_2, \dots\}$  the goal is that of finding  $\mathcal{G}_i$  that best captures the underlying state of the system. This can be performed by defining an apriori model (e.g., ERGM, SBM, LSM) and check whether it applied to the system being studied (**confirmatory approach**).

# From data to networks

Although a graph  $\mathcal{G}$  is just a pair of particular sets, the geometry for representing  $\mathcal{G}$  can be quite informal unless one defines a problem of **network topology inference**.

Given a set of measurements and a set of candidates  $\{\mathcal{G}_1, \mathcal{G}_2, \dots\}$  the goal is that of finding  $\mathcal{G}_i$  that best captures the underlying state of the system. This can be performed by defining an apriori model (e.g., ERGM, SBM, LSM) and check whether it applied to the system being studied (**confirmatory approach**).

Another way is to infer the network topology directly from the data, for instance by predicting links or by inferring the interior of the graph using only vertices that are at the perimeter of the structure (tomographic inference).

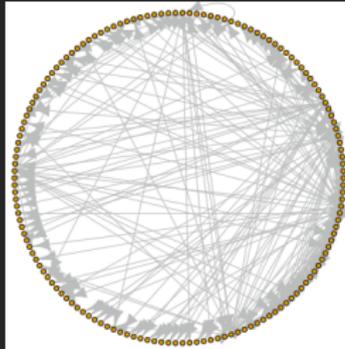
# From data to networks

When the research interest does not lie in inferring the network topology, **visualizing a graph**  $\mathcal{G}$  (i.e., assigning a topology) is not an easy task.

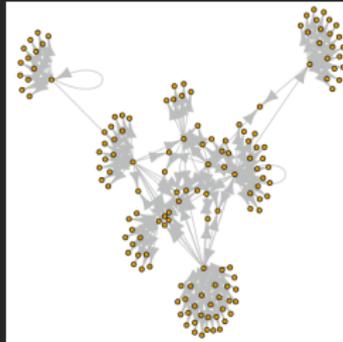
There are several algorithms that can be used to embed  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  into two- or three-dimensional Euclidean space:

- spring-embedder methods
- energy-placement methods
- multidimensional-scaling methods
- ...

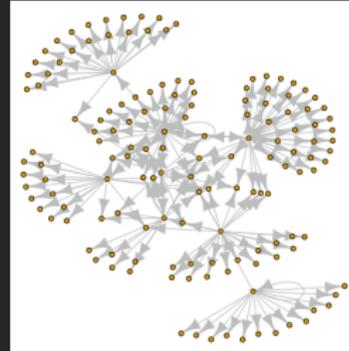
# From data to networks



(a) Circular layout



(b) Spring-embedder



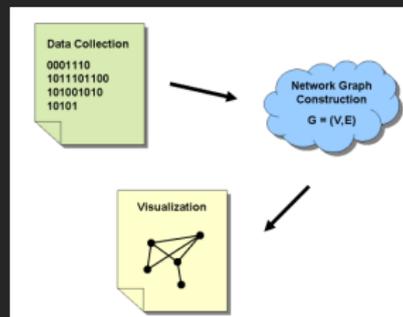
(c) MD scaling

Source: Kolaczyk E., Csardi G. (2014). *Statistical analysis of network data with R*. Springer

# From data to networks

## Mapping networks

- 1 Collecting measurements from a system
- 2 Creating the graph
  - (a) Identify nodes (e.g., in a social network, nodes could be individuals)
  - (b) Define edges (e.g., in a social network, edges could represent friendships or connections)
  - (c) Assign attributes (e.g., edges might have weights representing the strength of the connection)
- 3 Visualize and analyze the graph



# From data to networks

## Mapping networks

Quite often network graphs are constructed from basic measurements for a collection of **units** and information about them (e.g., **interactions**, associations, co-occurrences). The choice of what is meant by *vertices* and *edges* is important since it influences the network structure as well as the analyses we can run on it.

# From data to networks

## Mapping networks

Quite often network graphs are constructed from basic measurements for a collection of **units** and information about them (e.g., **interactions**, associations, co-occurrences). The choice of what is meant by *vertices* and *edges* is important since it influences the network structure as well as the analyses we can run on it.

Units can be of any type, including individuals (e.g., in social networks), server machines (e.g., in internet network), airplanes (e.g., in air traffic networks), questionnaire items or traders (e.g., in socio-economic networks).

# From data to networks

## Mapping networks

Quite often network graphs are constructed from basic measurements for a collection of **units** and information about them (e.g., **interactions**, associations, co-occurrences). The choice of what is meant by *vertices* and *edges* is important since it influences the network structure as well as the analyses we can run on it.

Units can be of any type, including individuals (e.g., in social networks), server machines (e.g., in internet network), airplanes (e.g., in air traffic networks), questionnaire items or traders (e.g., in socio-economic networks).

Interactions can be regarded as friendship (e.g., in social networks), package exchanges (e.g., in internet network), routes (e.g., in air traffic networks), item correlations or transactions (e.g., in socio-economic networks).

# From data to networks

## Mapping networks

Measurements among units can be of any type, including counts (i.e., natural number), presence/absent (i.e., integer), densities, fluxes, or other continuous measurement (i.e., reals).

The starting point of any analysis is to construct the lists of vertices, nodes, and adjacencies (weighted or not).

# From data to networks

## Mapping networks: *Zachary's karate club*

The well-known **Zachary's dataset** contains thirty-four members of a karate club alongside their social interactions. During the Zachary's study from 1970 to 1972, a conflict between the administrator Mr. Hi and the instructor John A. led to the club splitting into two. Half joined the admin's new club, while others either found a new instructor or left karate altogether.

# From data to networks

## Mapping networks: *Zachary's karate club*

The dataset is a dictionary containing 78 elements of members connections (the element 1 indicates Mr. Hi whereas the element 34 John A.) and their frequency (third list).

1	2	4
1	3	5
1	4	3
1	5	3
1	6	3
⋮	⋮	⋮
31	33	3
31	34	3
32	34	4
33	34	5

# From data to networks

## Mapping networks: *Zachary's karate club*

A graph  $\mathcal{G}$  for the karate club can be built by letting the thirty-four members be the vertices and the social interactions be the edges:

$$\mathcal{V} = \{1, 2, \dots, 34\}$$

$$\mathcal{E} = \{\{1, 2\}, \{1, 3\}, \dots, \{32, 34\}, \{33, 34\}\}$$

The corresponding adjacency matrix is as follows:

$$\mathbf{A}_{34 \times 34} = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ \vdots & & & & \vdots & & & & \vdots \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

# From data to networks

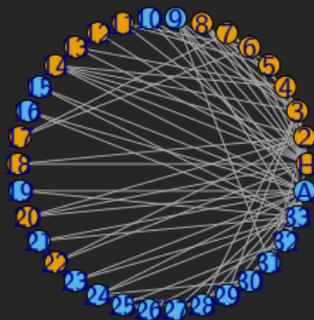
Mapping networks: *Zachary's karate club*

Unless one proceeds by inferring a proper network topology from the data, the last step of the mapping procedure consists of rendering the network graph using one of the available algorithms or techniques. Here we propose the results of four well-known algorithms.

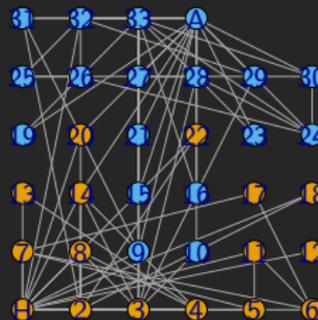
# From data to networks

Mapping networks: *Zachary's karate club*

(A) Circular



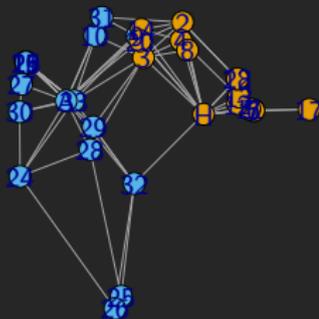
(B) Grid



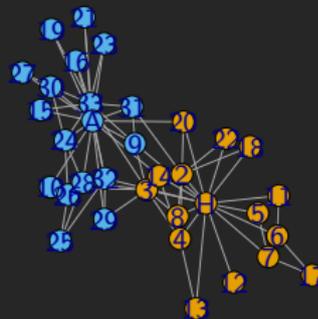
# From data to networks

Mapping networks: *Zachary's karate club*

(C) MD scaling



(D) Fruchterman-Reingold



# Outline

■ From data to networks

2 Descriptive analysis of network graphs

# Descriptive analysis of network graphs

Once a graph network graph representation has been obtained from the data, it is common to explore the characteristics and structural properties of the network.

The goal is that of summarizing the topological structure using simple metrics or more complex relational patterns.

The tools used in this type of structural analysis can be grouped into **three levels**:

- Node or edges-level analysis (A)
- Network-level analysis (B)
- Temporal-level analysis (for dynamic network graphs)

# Descriptive analysis of network graphs

## (A) Degree distribution

Let  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$  be an undirected graph with adjacency matrix  $\mathbf{A}_{\mathcal{G}}$ . For each node/vertex  $v \in \mathcal{V}$  the number of edges  $d$  incident upon  $v$  can be computed. The collection  $\{f_d\}_{d>0}$  is the *degree distribution* of  $\mathcal{G}$ , which is simply the histogram of the degree sequence.

Studying the distribution of  $\mathbf{d}_{\mathcal{G}}$  provides insights into the type of network  $\mathcal{G}$  embeds:

Scale-free	$\mathbf{d}_{\mathcal{G}} \sim \mathcal{PL}(d; \alpha)$	Power-Law distribution
Random network (Erdős–Rényi)	$\mathbf{d}_{\mathcal{G}} \sim \mathcal{Poi}(d; \lambda)$	Poisson
Regular network	$\mathbf{d}_{\mathcal{G}} \sim \delta(d - d_0)$	Degenerated distribution on $d_0$

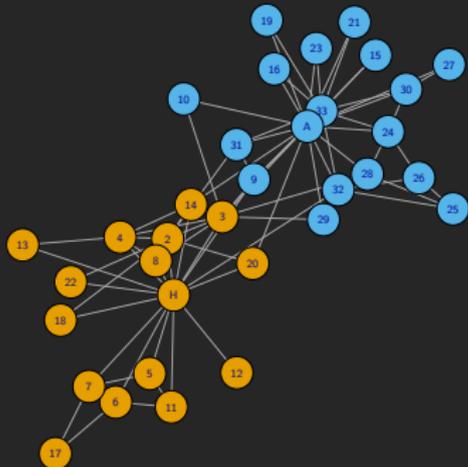
# Descriptive analysis of network graphs

## (A) Degree distribution

When  $\mathcal{G}$  is **directed**, the degree distribution is computed by summing the weights of adjacent vertices. In this case, one get the so-called **strength** measure.

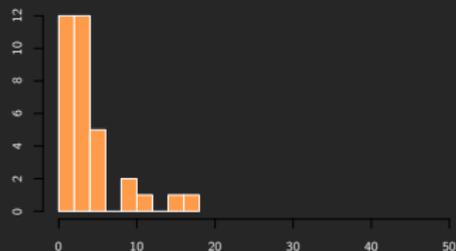
# Descriptive analysis of network graphs

## (A) Degree distribution

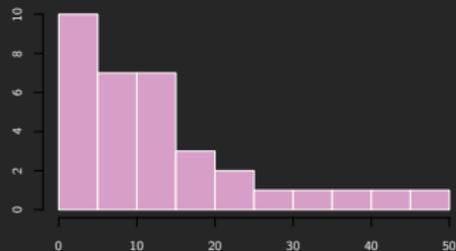


Zachary's karate club

Degree distribution



Strength distribution



# Descriptive analysis of network graphs

## (A) Centrality

Measures of centrality seek to quantify to what extent a vertex/node  $v \in \mathcal{V}$  is important in the graph  $\mathcal{G}$ .

# Descriptive analysis of network graphs

## (A) Centrality

Closeness: A vertex  $v$  is central if it is *close* to many other vertices. It is computed as

$$c_{Clo}(v) = \left( \sum_{u \in \mathcal{V}} \text{dist}(v, u) \right)^{-1}$$

where  $\text{dist}(v, u)$  is the geodesic distance (shortest path length) between  $v$  and  $u$ . The Dijkstra's algorithm can be used in this case.

# Descriptive analysis of network graphs

## (A) Centrality

Betweenness: A vertex  $v$  is central if it *lies* on paths between other vertices. It is computed as

$$c_{Btw}(v) = \sum_{u \neq v \neq z \in \mathcal{V}} \frac{\sigma(u, v|z)}{\sigma(u, v)}$$

where

$\sigma(u, v|z)$  is the number of shortest paths between  $u$  and  $v$  passing through  $z$

$\sigma(u, v) = \sum_{z \in \mathcal{V}} \sigma(u, v|z)$  is the number of paths between  $u$  and  $v$  regardless of  $z$

# Descriptive analysis of network graphs

## (A) Centrality

Eigen-centrality: A vertex  $v$  is central if it is *connected* to other vertices that are themselves central. It is computed by solving the Eigenvector problem of the adjacency matrix  $\mathbf{A}$ :

$$c_{Eigen}(v) = \frac{1}{\lambda} \sum_{u \in \mathcal{V}} a(u, v) c_u$$

where

$a(u, v)$  is the adjacency value for the pair  $\{u, v\}$

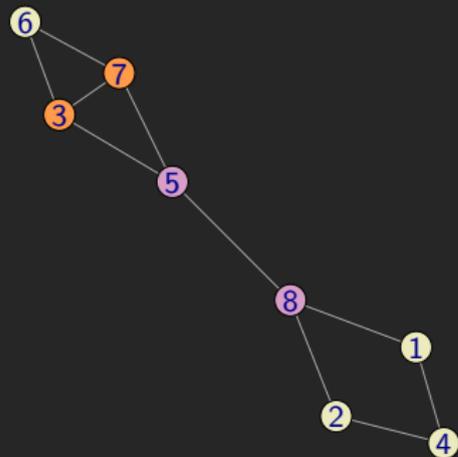
$\lambda$  is the largest eigenvalue of  $\mathbf{A}$

$c_u$  centrality of node  $u$

(initially unknown, it needs to be iteratively computed)

# Descriptive analysis of network graphs

## (A) Centrality



node	$C_{Clo}$	$C_{Btw}$	$C_{Eigen}$
1	0.44	2.50	0.29
2	0.44	2.50	0.29
3	0.47	2.50	1.00
4	0.35	0.50	0.22
5	0.58	12.00	0.95
6	0.35	0.00	0.74
7	0.47	2.50	1.00
8	0.58	12.50	0.57
centrality	0.53	0.49	3.32

# Descriptive analysis of network graphs

## (A) Centrality

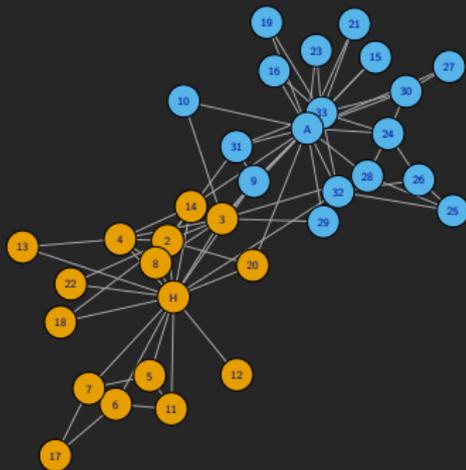
**Closeness:** Measures *how quickly* a node can interact with all other nodes in the network. Nodes with high closeness centrality are efficient in spreading information or influence across the network.

**Betweenness:** Measures the extent to which a node lies on the shortest paths between other nodes in the network. Nodes with high betweenness centrality *act as bridges* between different parts of the network.

**Eigen-centrality:** Measures the influence of a node based on the *centrality of its neighbors*. Nodes with high eigenvector centrality are influential due to their connections to other important nodes in the network.

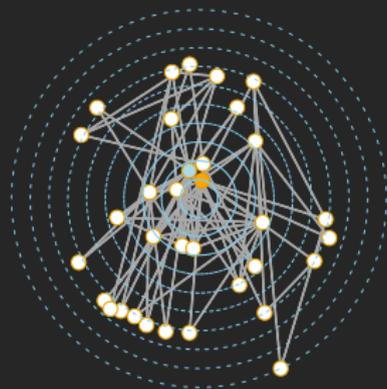
# Descriptive analysis of network graphs

## (A) Centrality



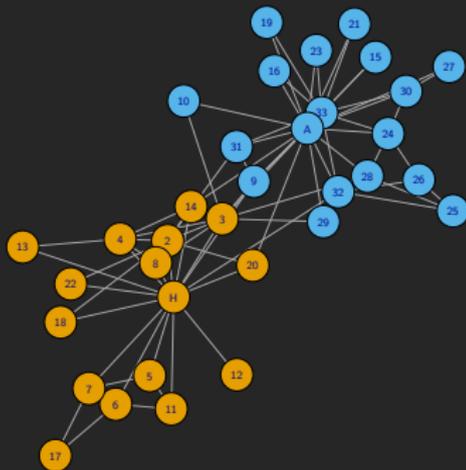
Zachary's karate club

## Closeness



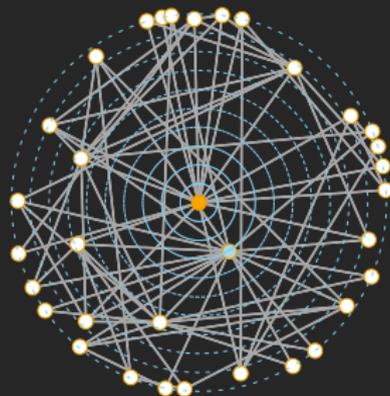
# Descriptive analysis of network graphs

## (A) Centrality



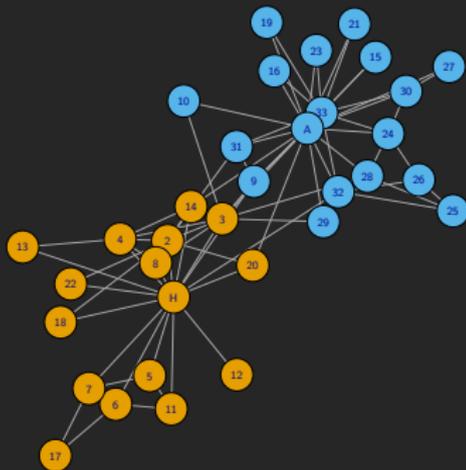
Zachary's karate club

## Betweenness



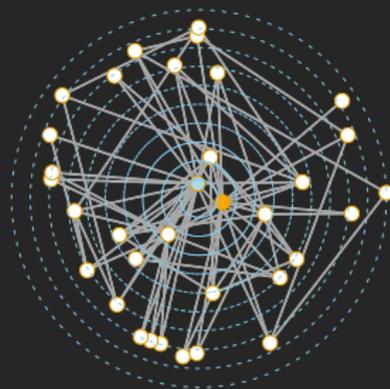
# Descriptive analysis of network graphs

## (A) Centrality



Zachary's karate club

## Eigen-centrality



# Descriptive analysis of network graphs

## (B) Subgraphs and cliques

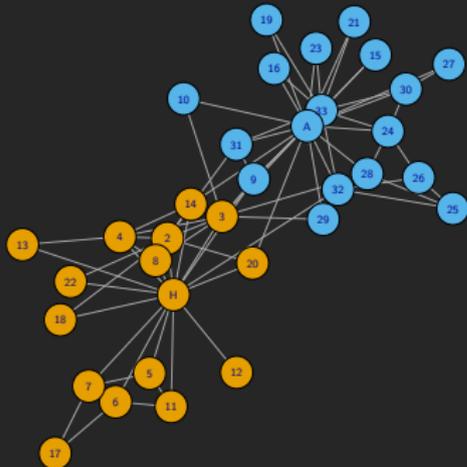
**Cliques** are complete subgraphs  $\mathcal{H}_m$  of size  $m$  with a subset of vertices  $\mathcal{V}_m \subseteq \mathcal{V}$  fully connected among them. They are computed using iterative algorithms (e.g., the Bron-Kerbosch algorithm).

Usually, cliques of larger sizes necessarily include cliques of smaller sizes. Instead, a **maximal clique** is a clique that is not a subset of a larger clique.

Note: this analysis is run by first defining a notion of substructure (e.g., clique of size 2), then looking to see if and how often it occurs in the graph.

# Descriptive analysis of network graphs

## (B) Subgraphs and cliques



clique	1	2	3	4	5
count	34	78	45	11	2

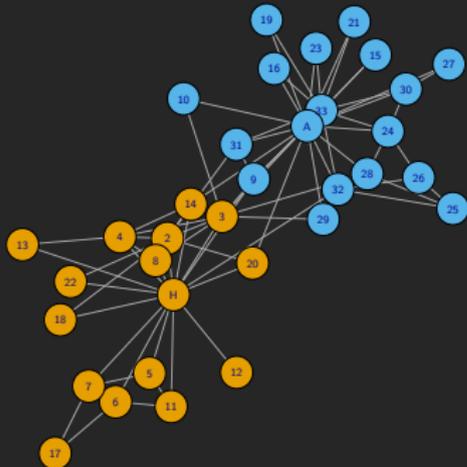
In the Zachary's karate network we have

- 34 cliques of size 1 (nodes)
- 78 cliques of size 2 (pairs)
- 45 cliques of size 3 (triads)
- 11 cliques of size 4 (quadriads)
- 2 cliques of size 5

Zachary's karate club

# Descriptive analysis of network graphs

## (B) Subgraphs and cliques



clique	2	3	4	5
count	11	21	2	2

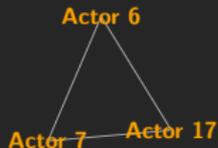
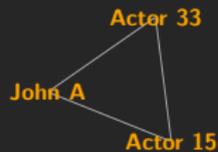
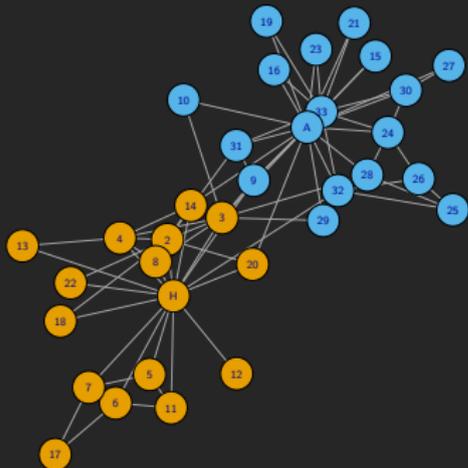
In the Zachary's karate network we have

- 11 maximal cliques of size 2
- 21 maximal cliques of size 3
- 2 maximal cliques of size 4
- 2 maximal cliques of size 5

Zachary's karate club

# Descriptive analysis of network graphs

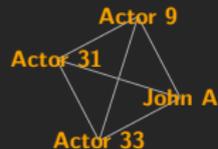
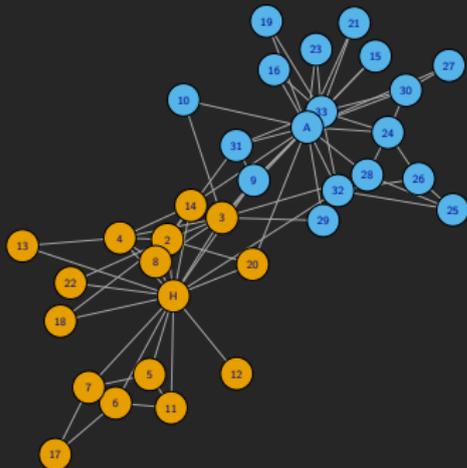
## (B) Subgraphs and cliques



Zachary's karate club

# Descriptive analysis of network graphs

## (B) Subgraphs and cliques



Zachary's karate club

# Descriptive analysis of network graphs

## (B) Transitivity

The transitivity of a graph  $\mathcal{G}$  reflects the degree to which vertices tend to cluster together. It indicates how tightly knit the groups within the network are. High transitivity suggests a network where nodes tend to create tightly connected groups, while low transitivity indicates a more loosely connected structure.

The transitivity index can be computed locally and globally.

# Descriptive analysis of network graphs

## (B) Transitivity

Global: It measures the overall level of clustering in the entire network as follows

$$C_T = \frac{3\tau_{\Delta}(\mathcal{G})}{\tau_3(\mathcal{G})}$$

where

$\tau_{\Delta}(\mathcal{G})$  is the number of triangles in the graph

(A triangle is a set of three nodes where each node is directly connected to the other two nodes, forming a closed loop)

$\tau_3(\mathcal{G})$  is the number of triples in the graph

(A triple is a set of three nodes that can be either open or closed. If it is closed, then the triple forms a triangle)

# Descriptive analysis of network graphs

## (B) Transitivity

Local: It measures the tendency of a single node's neighbors to be connected. For a vertex  $v \in \mathcal{V}$ , it is computed as

$$C_T(v) = \frac{2\tau_\Delta(\mathcal{G}|v)}{\tau_3(\mathcal{G}|v)}$$

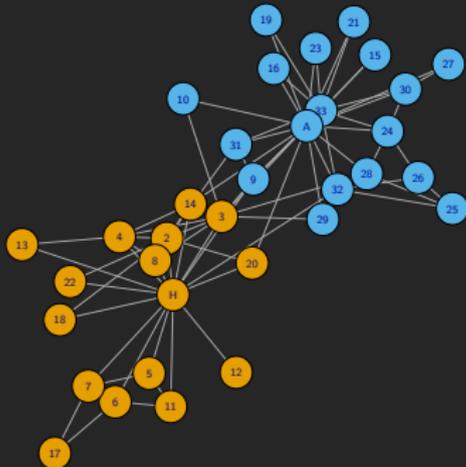
where

$\tau_\Delta(\mathcal{G}|v)$  is the number of triangles in the graph containing  $v$

$\tau_3(\mathcal{G}|v)$  is the number of triples in the graph for which two edges are both incident to  $v$

# Descriptive analysis of network graphs

## (B) Transitivity



Zachary's karate club

In the Zachary's karate network we have

$$C_T = 0.25$$

which indicates a *moderate* level of clustering within the network.

As Networks with higher transitivity tend to have more pronounced community structures where nodes are more likely to be connected within groups,  $C_T = 0.25$  suggests that while there are some community structures or clusters within the network, they are not as strongly defined as they would be with higher transitivity values.

# Descriptive analysis of network graphs

## (B) Graph partitioning

Graph partitioning refers to the task of finding subgraphs which are well-connected (dense) and at the same time are well-separated (sparse) from the others. The task is also known as **community detection** problem.

# Descriptive analysis of network graphs

## (B) Graph partitioning

Graph partitioning refers to the task of finding subgraphs which are well-connected (dense) and at the same time are well-separated (sparse) from the others. The task is also known as **community detection** problem.

There are several techniques used for this task, most of them resembling an agglomerative clustering on graphs:

- Greedy-based methods
- Louvain's method
- Spectral clustering
- Block modeling partitioning
- ...

# Descriptive analysis of network graphs

## (B) Graph partitioning

Both the *Greedy* and *Louvain* methods are popular iterative algorithms for detecting communities in large networks by optimizing the **modularity** measure. Particularly, the Louvain's method is scalable, produces high-quality communities, and it is easy to implement.

Other algorithms are based on different rationales. For instance, the *Spectral* clustering method seeks for agglomerates of subgraphs by inspecting the Laplacian matrix  $\mathbf{L}$  associated to the graph  $\mathcal{G}$ .

# Descriptive analysis of network graphs

## (B) Graph partitioning

The **modularity**  $Q(\mathcal{G})$  quantifies the density of edges *within communities* compared to edges *between communities*:

$$Q(\mathcal{G}) = \frac{1}{2N_e} \sum_{u,v \in \mathcal{V}} \left( a(u,v) - \frac{d_u d_v}{2N_e} \right) \chi(c_u, c_v)$$

where

$a(u, v)$  is the adjacency value for the pair  $\{u, v\}$

$d_x$  is the degree for the node  $x$

$\chi(c_u, c_v)$  is 1 if nodes  $u$  and  $v$  belong to the same community  $c$ , and 0 otherwise

# Descriptive analysis of network graphs

## (B) Graph partitioning

The modularity  $Q(\mathcal{G})$  quantifies the density of edges *within communities* compared to edges *between communities*:

$$Q(\mathcal{G}) = \frac{1}{2N_e} \sum_{u,v \in \mathcal{V}} \left( a(u,v) - \frac{d_u d_v}{2N_e} \right) \chi(c_u, c_v)$$

As  $Q(\mathcal{G}) \in [-1, 1]$ ,

$Q(\mathcal{G}) > 0$  indicates a higher density of edges within communities than expected by chance, suggesting a good community structure

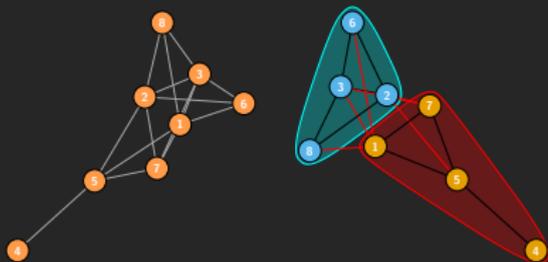
$Q(\mathcal{G}) < 0$  suggests that the network is less modular than expected (rare in practice)

$Q(\mathcal{G}) = 0$  indicates that the division into communities is no better than chance

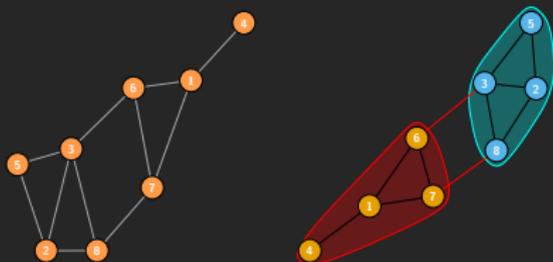
# Descriptive analysis of network graphs

## (B) Graph partitioning

Case1:  $Q(G) = 0.098$

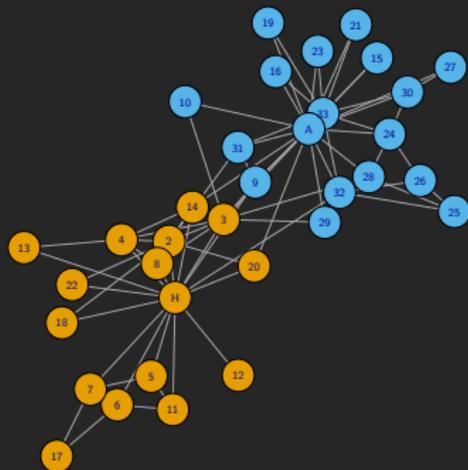


Case 2:  $Q(G) = 0.314$



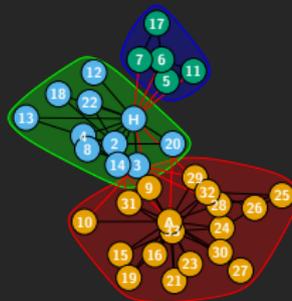
# Descriptive analysis of network graphs

## (B) Graph partitioning

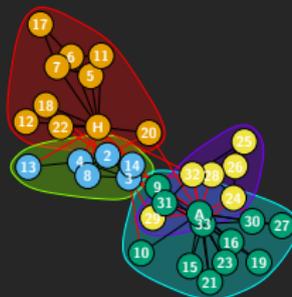


Zachary's karate club

### (A) Greedy algorithm



### (B) Louvain's algorithm



# Descriptive analysis of network graphs

## (B) Assortativity mixing

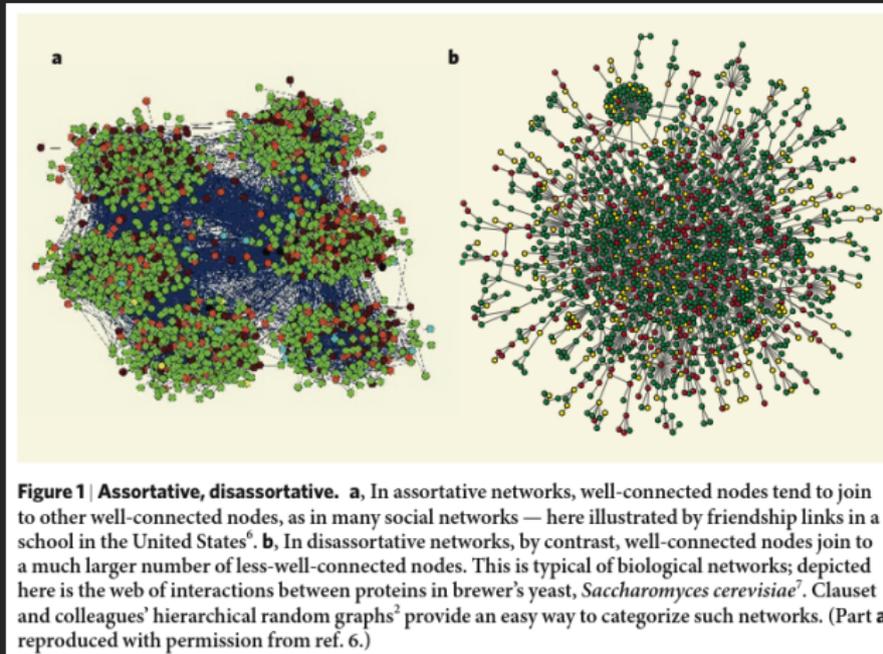
**Assortativity** is a measure of similarity that quantifies the tendency of nodes in a network to connect to similar nodes. It helps understand whether nodes with similar characteristics (e.g., degree, attributes) tend to connect more often than expected by chance.

Assortativity coefficient for a network is calculated using the Pearson correlation coefficient between the degrees of connected nodes. In this way, it quantifies the level of *homophily* of the graph, based on some vertex labeling or values assigned to vertices.

Positive assortativity indicates that similar vertices tend to connect to each other. Instead, negative assortativity (i.e., **disassortativity**) indicates that nodes with dissimilar attributes tend to connect more often.

# Descriptive analysis of network graphs

## (B) Assortativity mixing



Source: Chen, C., Liao, C., & Liu, Y. Y. (2023). Teasing out missing reactions in genome-scale metabolic networks through hypergraph learning. *Nature Communications*, 14(1), 2375

# Descriptive analysis of network graphs

## Summing up with a social network example

<b>Degree of a node</b>	Number of friends or connections persons/nodes do have
<b>Diameter</b>	Maximum number of acquaintanceships one person would need to traverse to reach another person
<b>Order of a network</b>	Number of individuals forming the social space or actually interacting
<b>Avg path length</b>	Between any two people, indicates how closely connected the network is overall
<b>Closeness</b>	A node with high closeness might be someone who can reach most other individuals quickly through direct or short paths
<b>Betweenness</b>	A node with high betweenness might be someone who connects different groups or communities of individuals
<b>Clique</b>	It might represent a group of friends where each person is friends with every other person in the group

# Descriptive analysis of network graphs

## Summing up with a social network example

- Global transitivity** A high global transitivity indicates that friends of a person are likely to be friends with each other as well
- Local transitivity** It helps understand how likely it is for mutual friends of a person to be friends with each other
- Modularity** It helps identify distinct groups of individuals who have more connections within their group than with individuals outside their group
- Assortativity** It might indicate whether individuals with many friends tend to be friends with other popular individuals (positive assortativity) or with less connected individuals (negative assortativity)