“Persistent Homology” Summer School - Rabat

Persistent Homology in Complex Network Analysis

Ulderico Fugacci

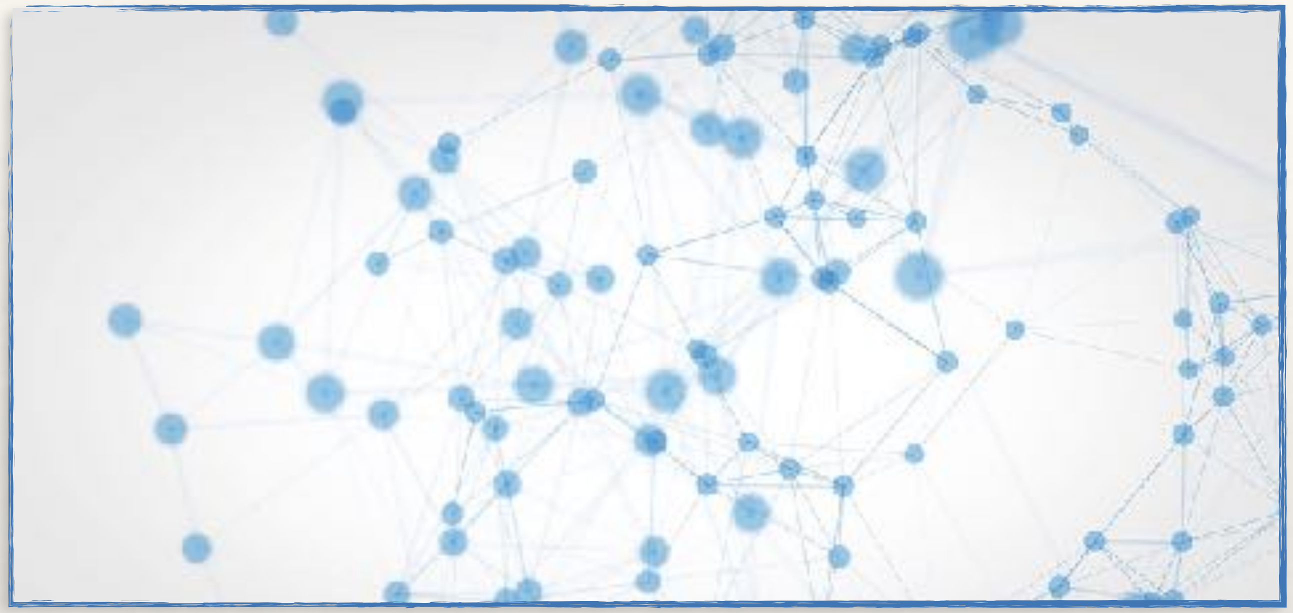
Kaiserslautern University of Technology

Department of Computer Science

Technische Universität Kaiserslautern

July 7, 2017

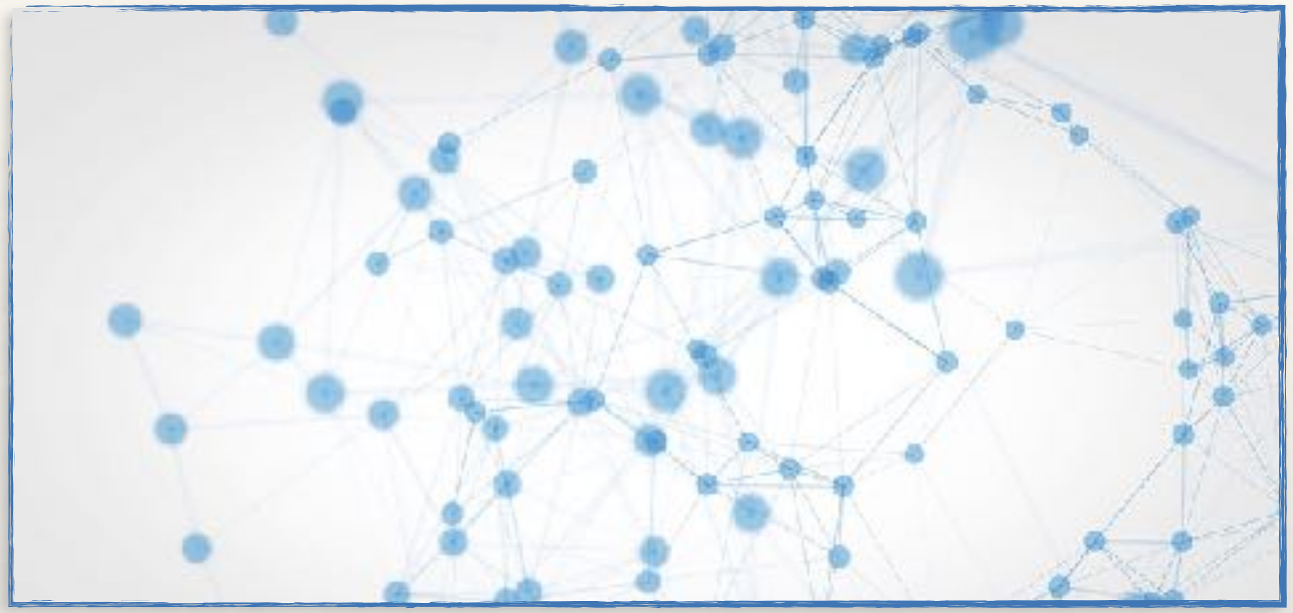
Anything has Shape



"Data has shape and
shape has meaning"

Gunnar Carlsson

Anything has Shape



"Data has shape and
shape has meaning"

Gunnar Carlsson

Persistent Homology:

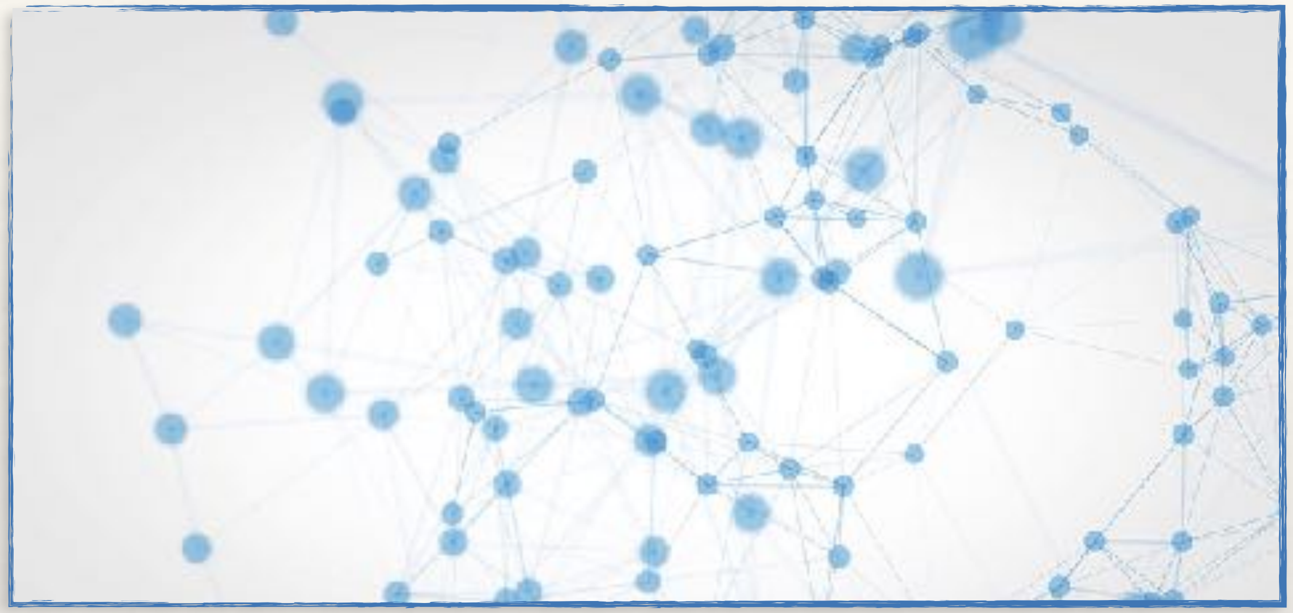
From:

Shape Analysis

To:

Neuroscience
Chemistry
Geography
Shape Analysis
Biophysics
Biology
Network Analysis
Oncology
…

Anything has Shape



"Data has shape and shape has meaning"

Gunnar Carlsson

Persistent Homology:

From:

Shape Analysis

To:

Neuroscience

Chemistry

Geography

Shape Analysis

Biophysics

Biology

Network Analysis

Oncology

...

Complex Networks

Definition:

A ***network*** is a **complex system** consisting of **individuals** or **entities** connected by specific **ties** such as

- ***Personal Relationship***
- ***Shared Knowledge***
- …

References:
M. Newman, *Networks: An Introduction*, 2010
J. Scott, *Social Network Analysis*, 2017

Complex Networks

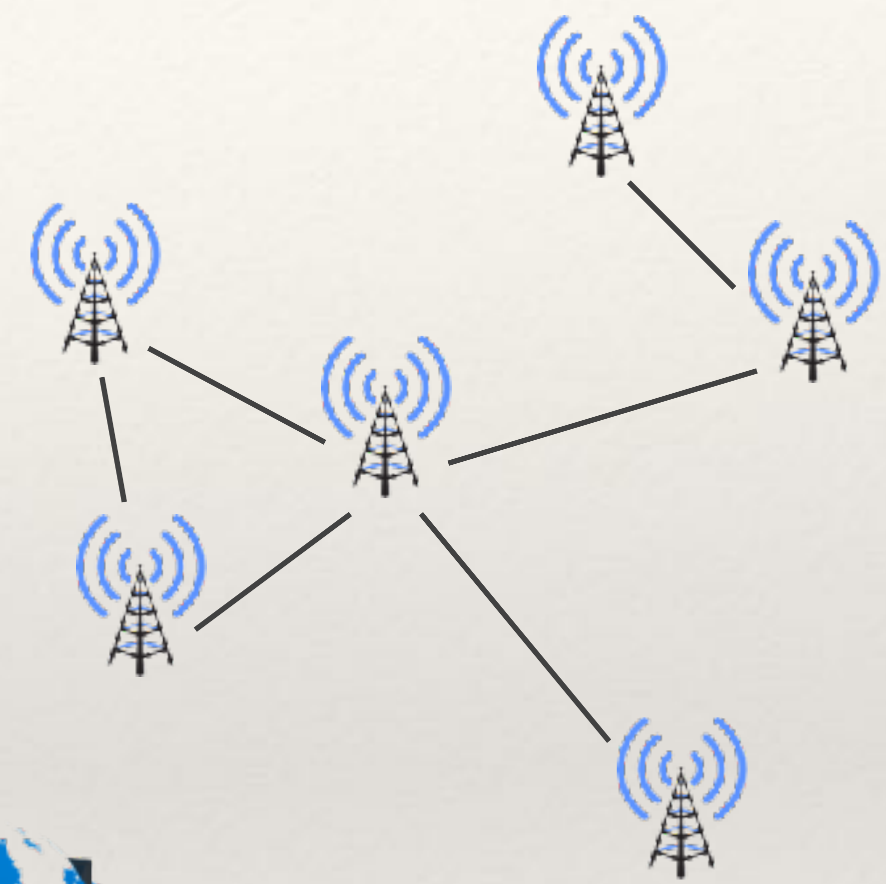
A Bunch of Examples:

- ***Social** Networks*

Complex Networks

A Bunch of Examples:

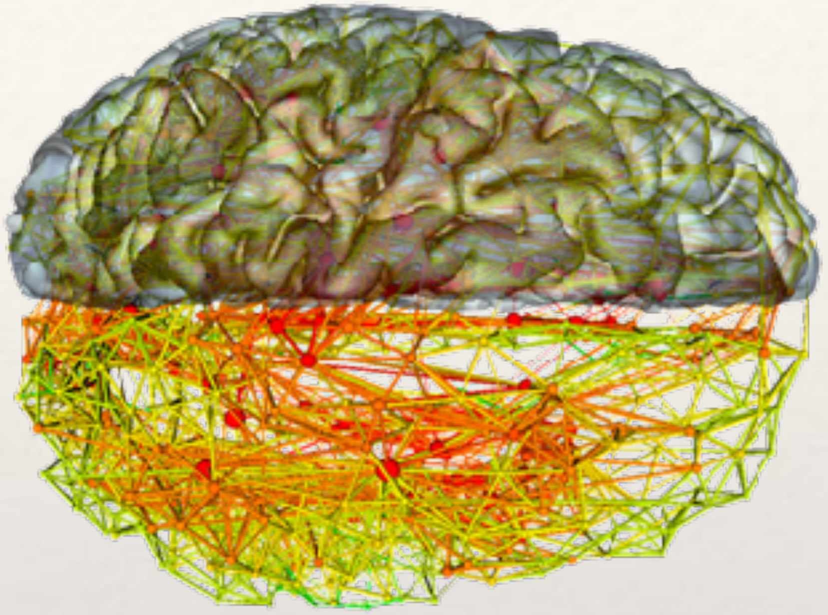
- ***Social*** *Networks*
- ***Sensor*** *Networks*

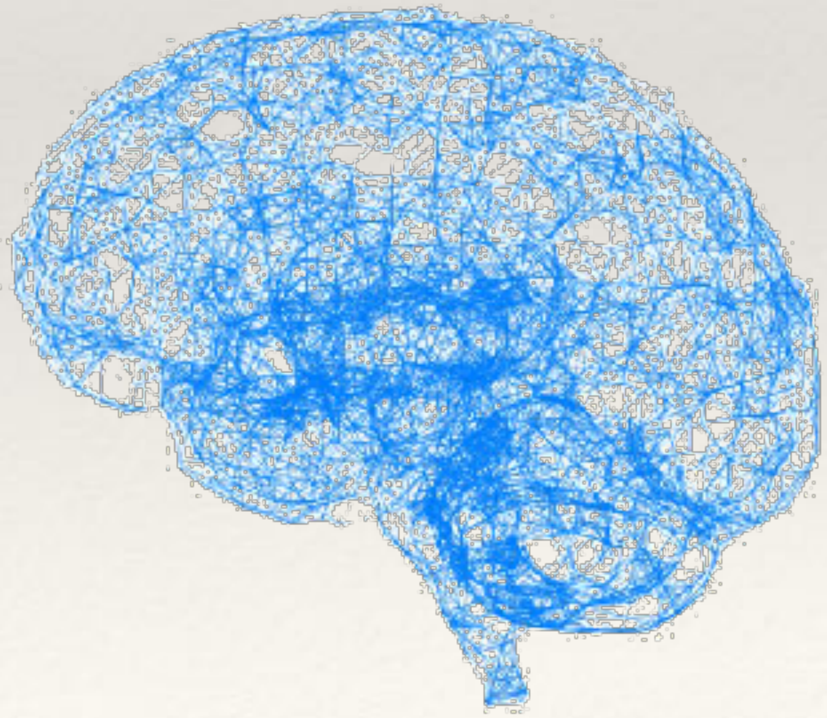


Complex Networks

A Bunch of Examples:

- ***Social*** *Networks*
- ***Sensor*** *Networks*
- ***Biological*** *Networks*





Complex Networks

A Bunch of Examples:

- ***Social*** *Networks*
- ***Sensor*** *Networks*
- ***Biological*** *Networks*
- ***Collaborative*** *Networks*

Complex Networks

A Bunch of Examples:

- ***Social*** *Networks*
- ***Sensor*** *Networks*
- ***Biological*** *Networks*
- ***Collaborative*** *Networks*
- …

Outline

Brief Introduction to Complex Network Analysis

Outline

Brief Introduction to
Complex Network
Analysis

Persistence-based
Network Analysis

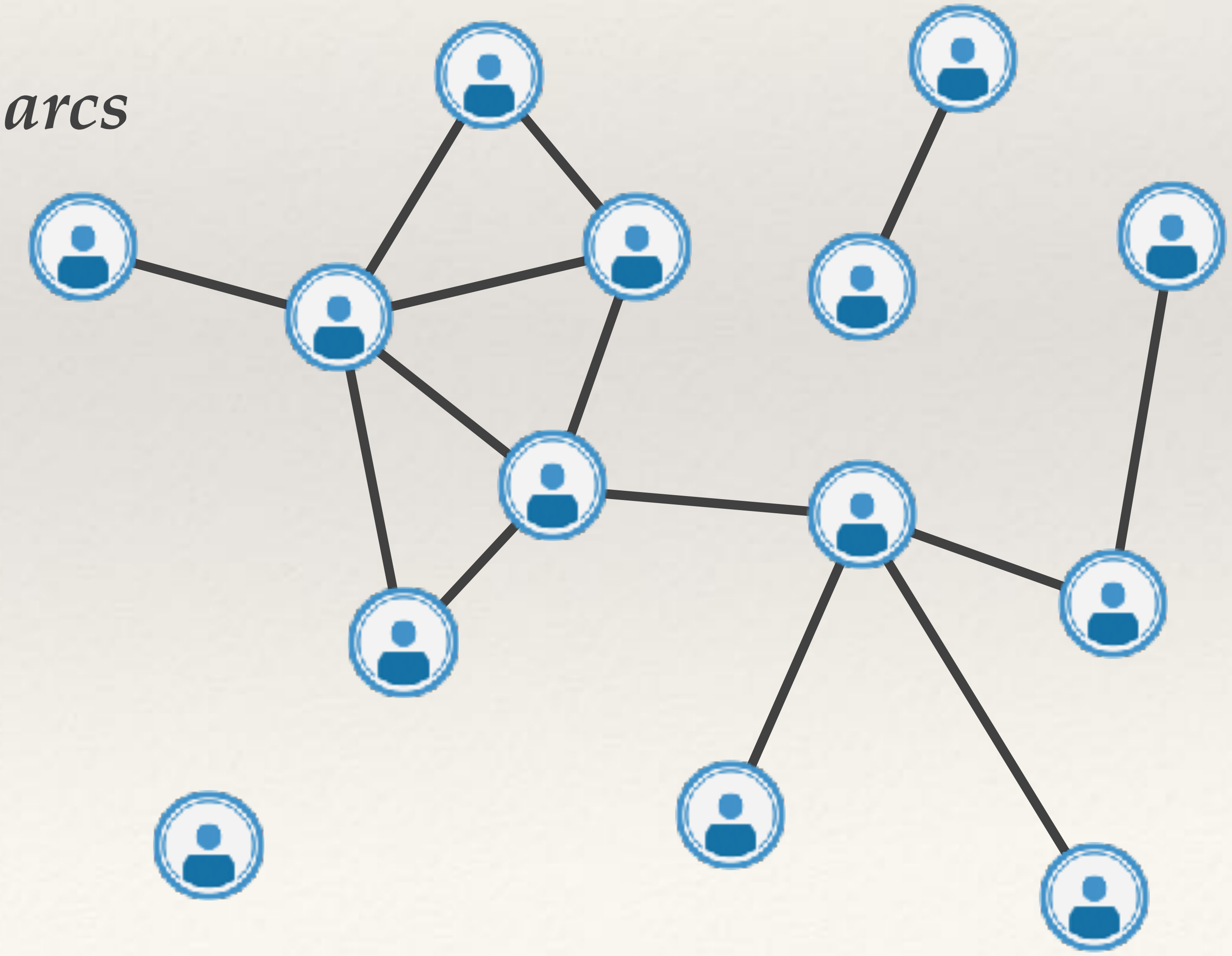
Network Analysis

Representation:

A network can be represented by a **graph** ***G=(V, E)*** such that:

individuals ⟺ *nodes*

ties ⟺ *arcs*

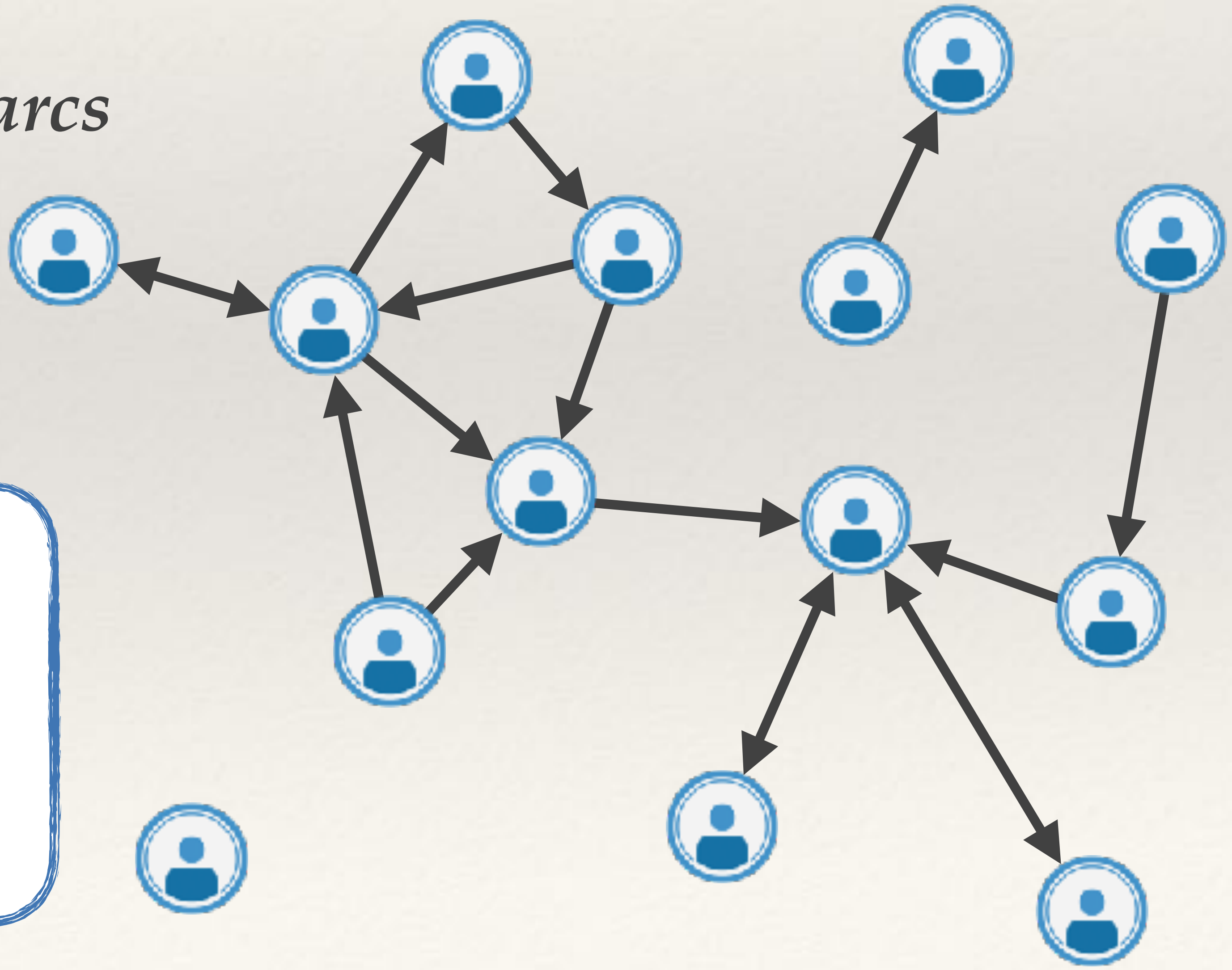


Network Analysis

Representation:

A network can be represented by a **graph** ***G=(V, E)*** such that:

individuals	⟺	*nodes*
ties	⟺	*arcs*



Arcs can be:

- ***directed***
- ***weighted***

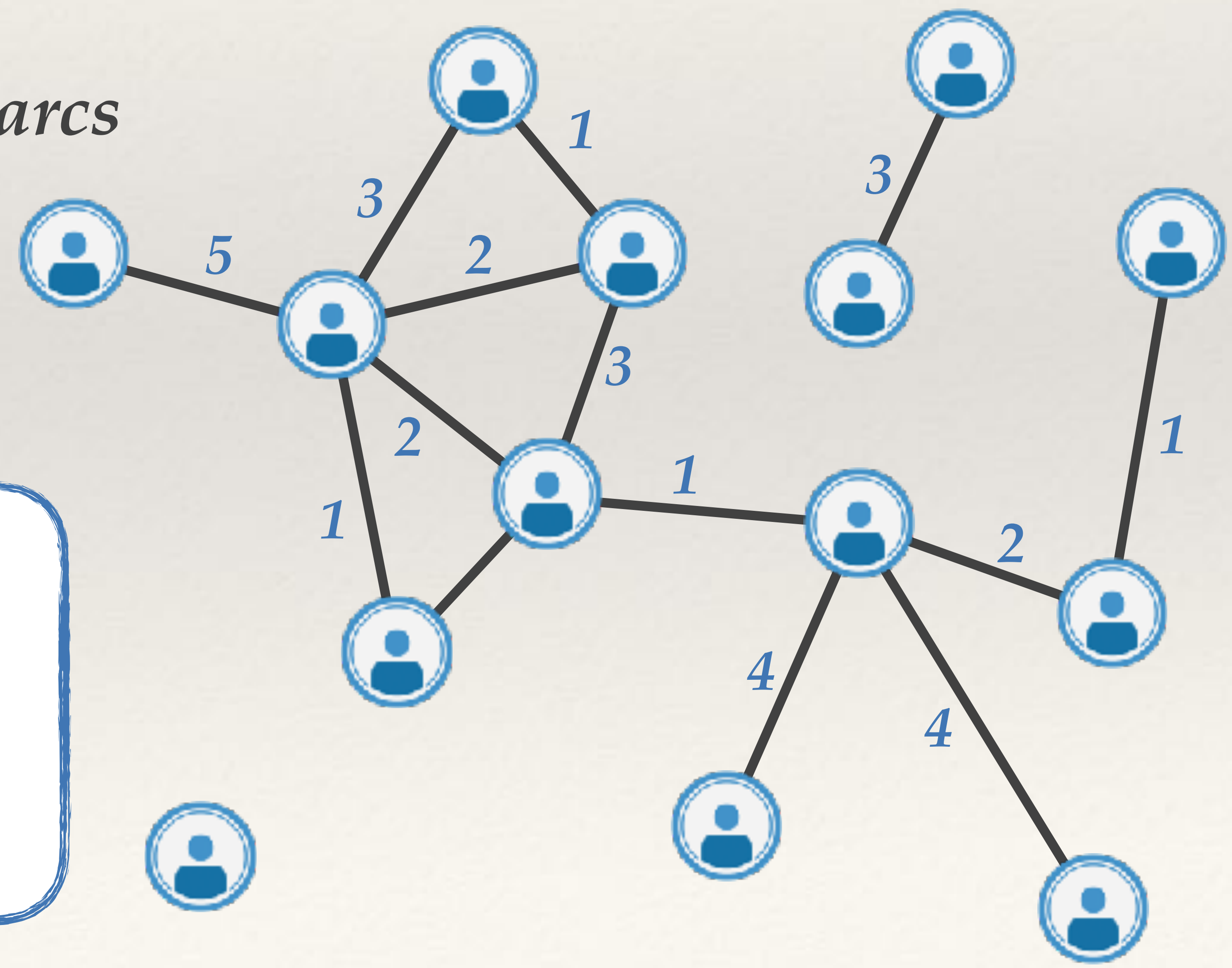
Network Analysis

Representation:

A network can be represented by a **graph** ***G=(V, E)*** such that:

individuals ⟺ *nodes*

ties ⟺ *arcs*



Arcs can be:

- *directed*
- *weighted*

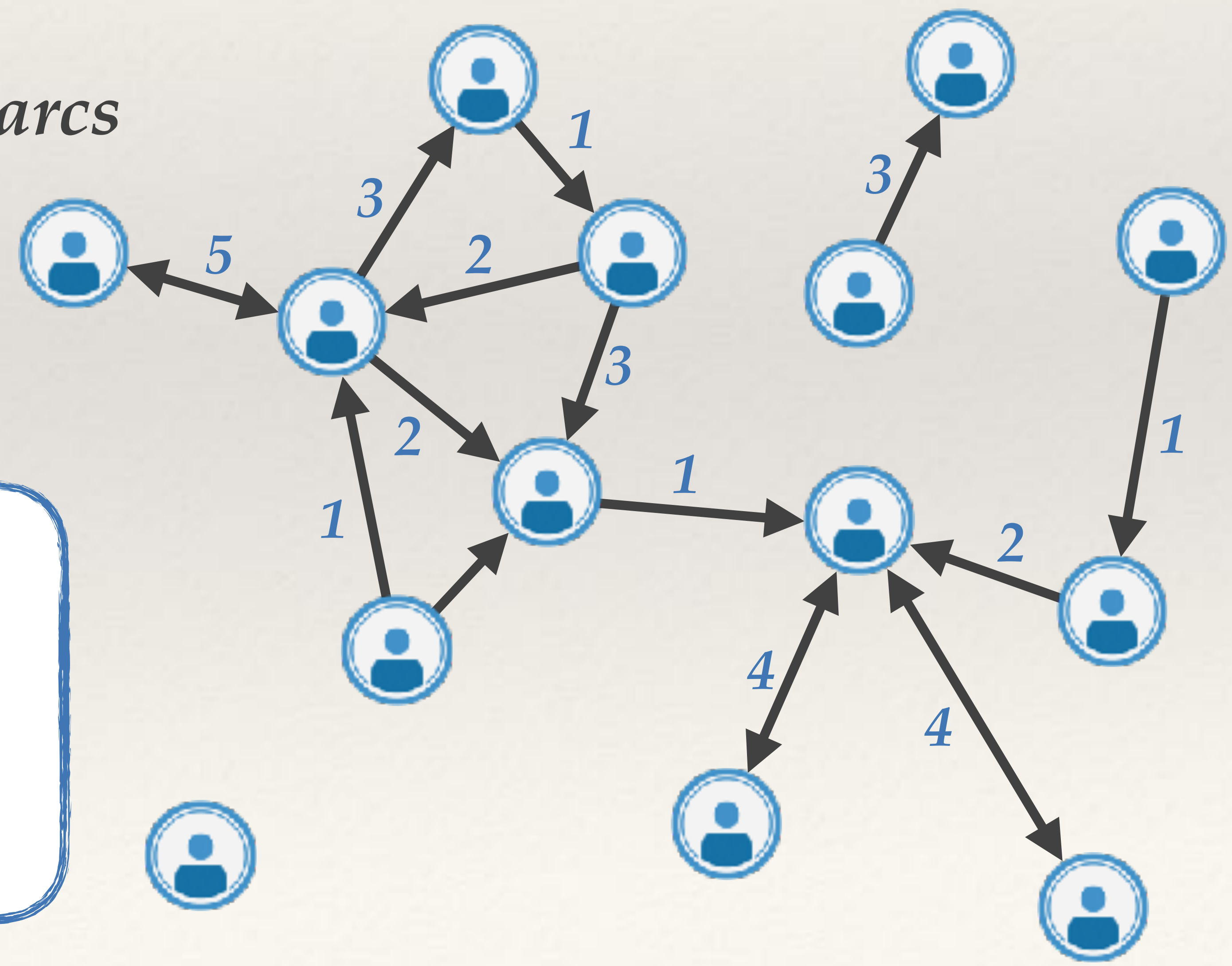
Network Analysis

Representation:

A network can be represented by a **graph** ***G=(V, E)*** such that:

individuals ⟺ *nodes*

ties ⟺ *arcs*

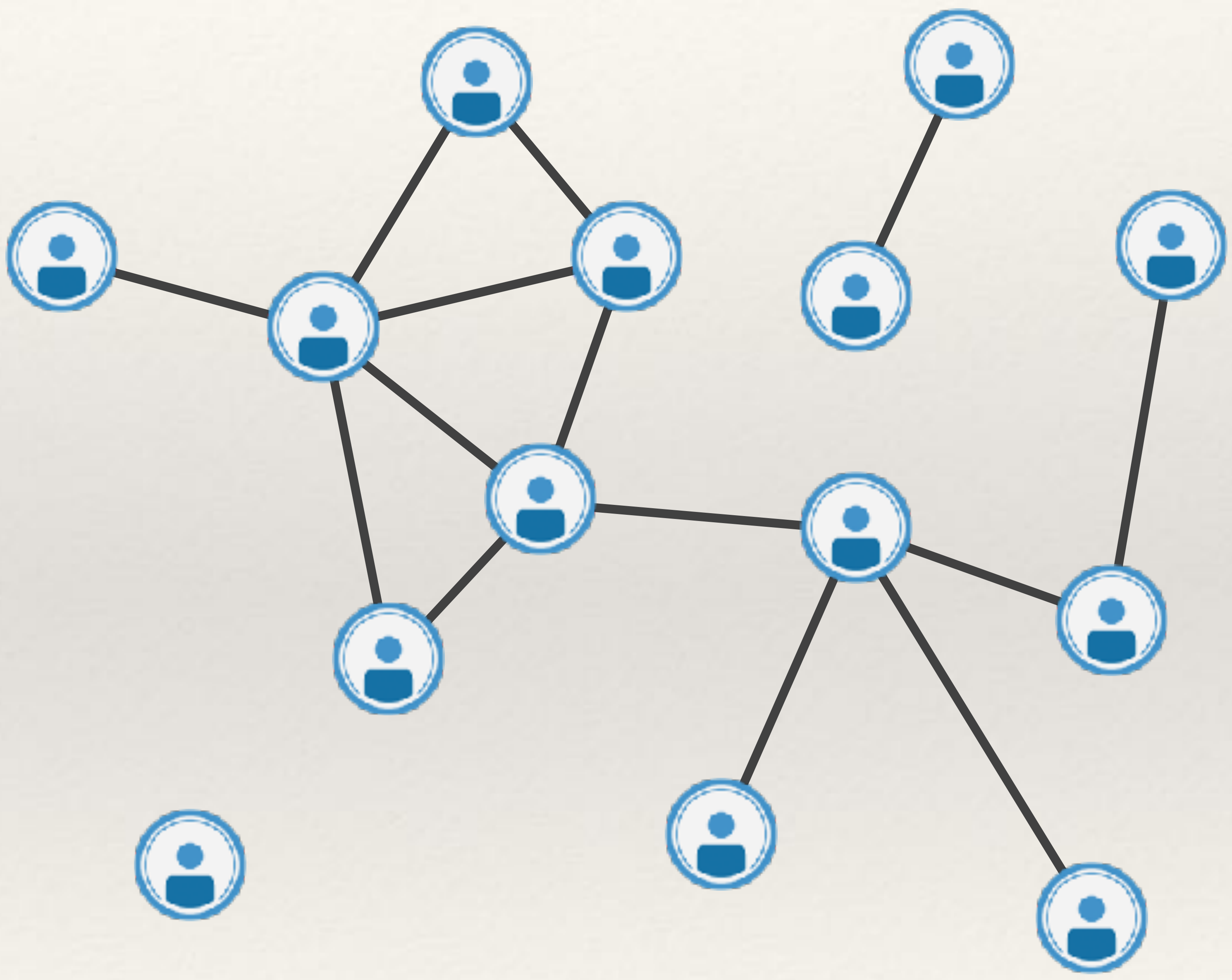


Arcs can be:

- *directed*
- *weighted*

Network Analysis

A Two-level Analysis:



Network Analysis

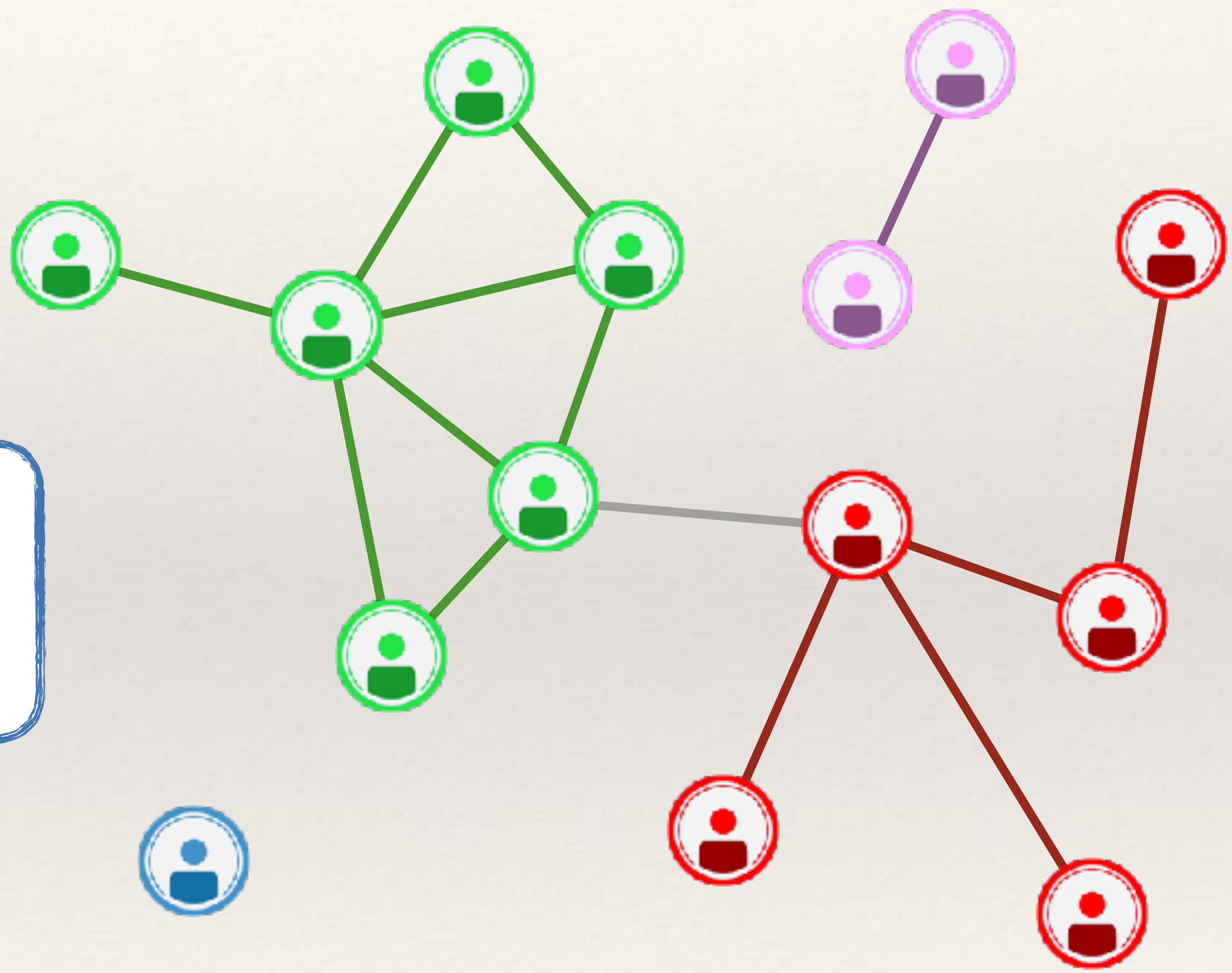
A Two-level Analysis:

- *Egocentric*
- *Sociocentric*

Network Analysis

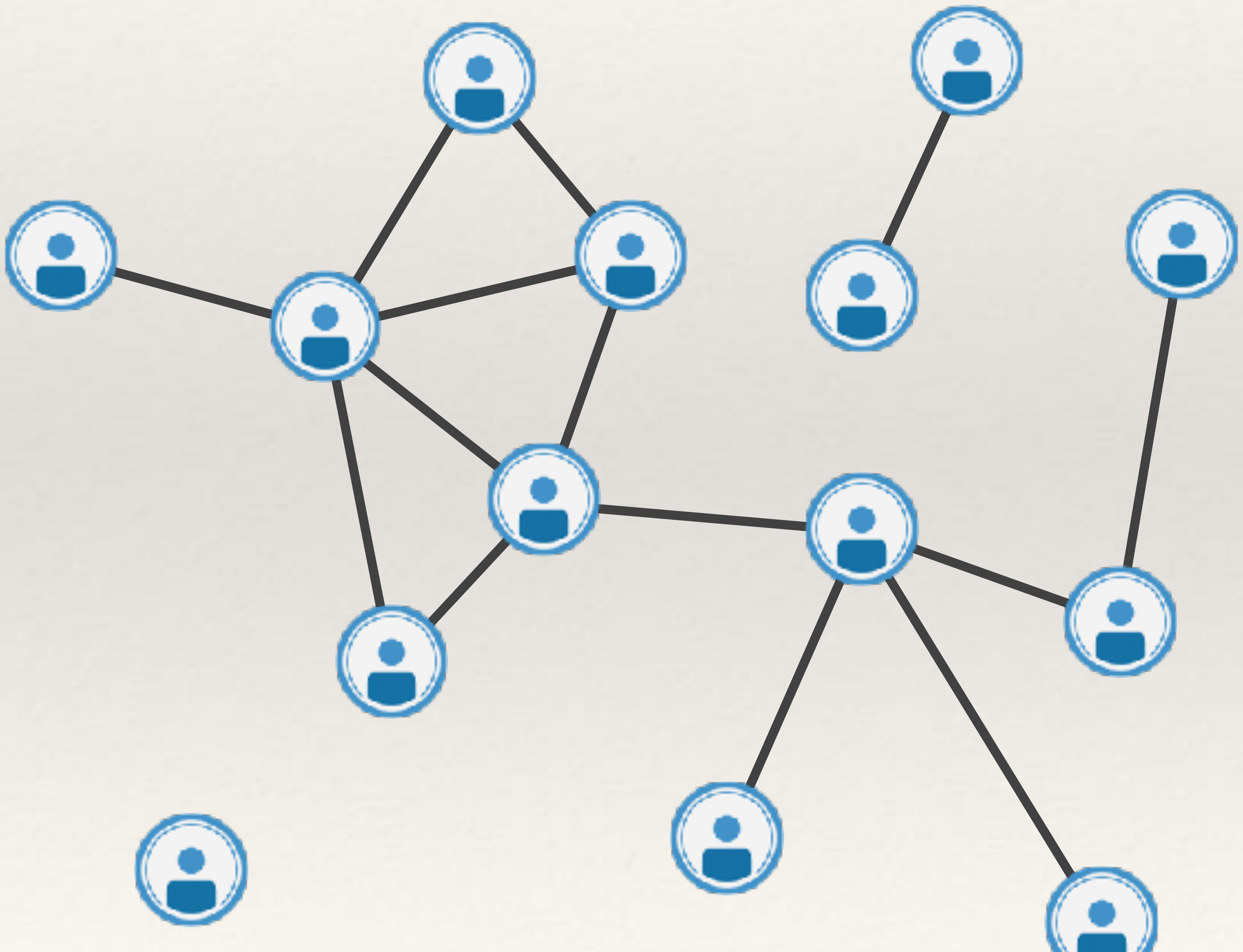
A Two-level Analysis:

- *Egocentric*
- ***Sociocentric***



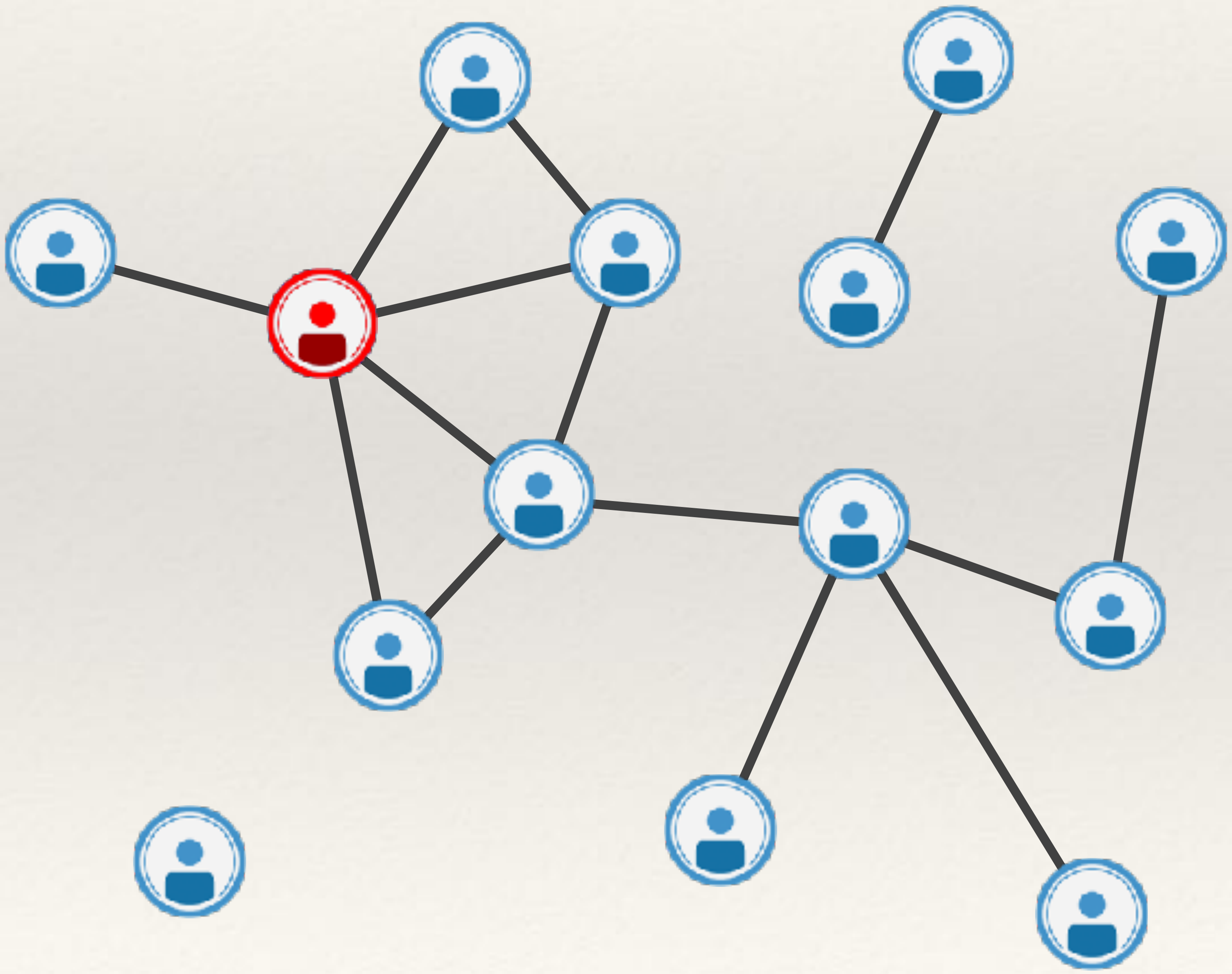
Identifying Key Players

What is the most important individual?



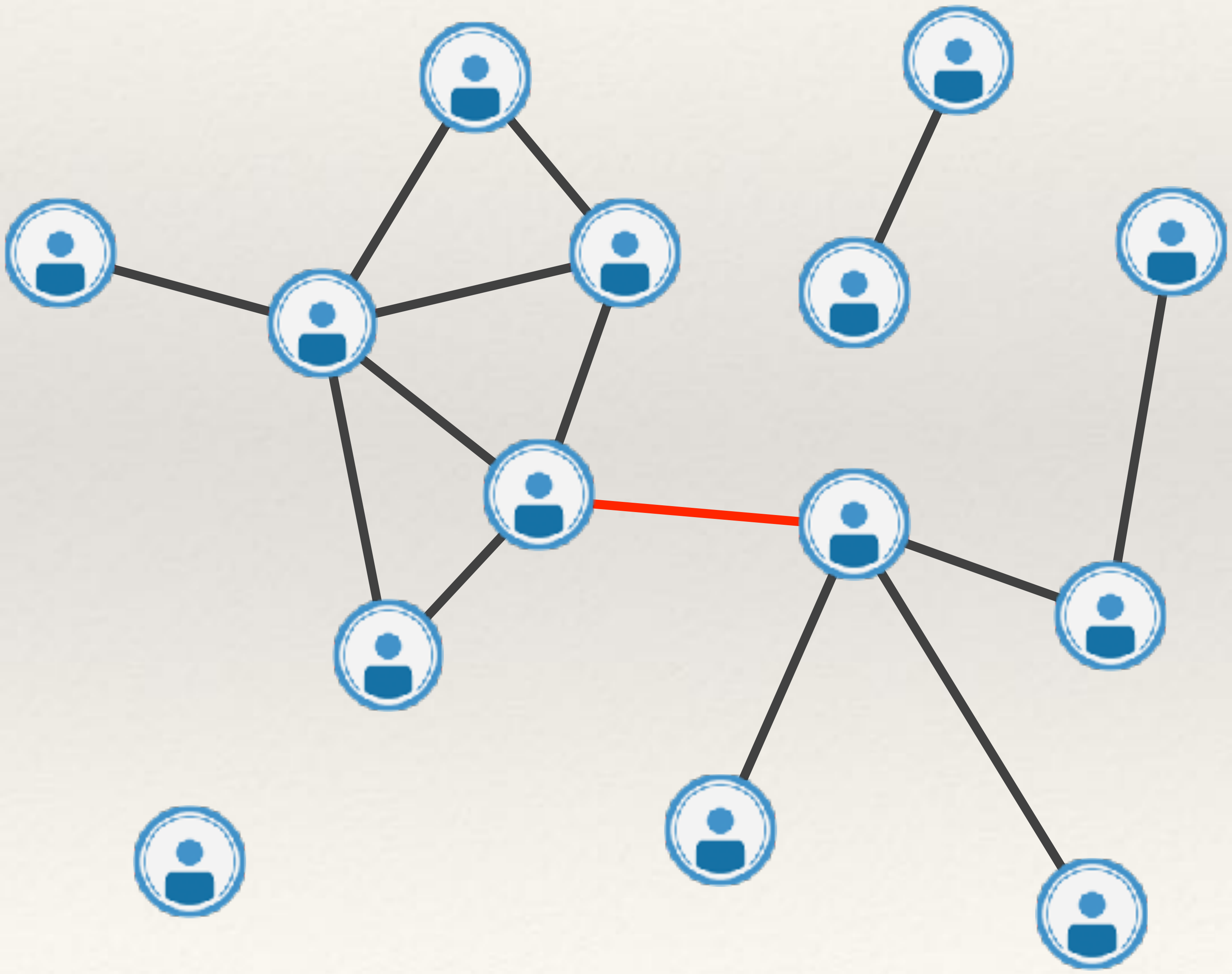
Identifying Key Players

What is the most important individual?



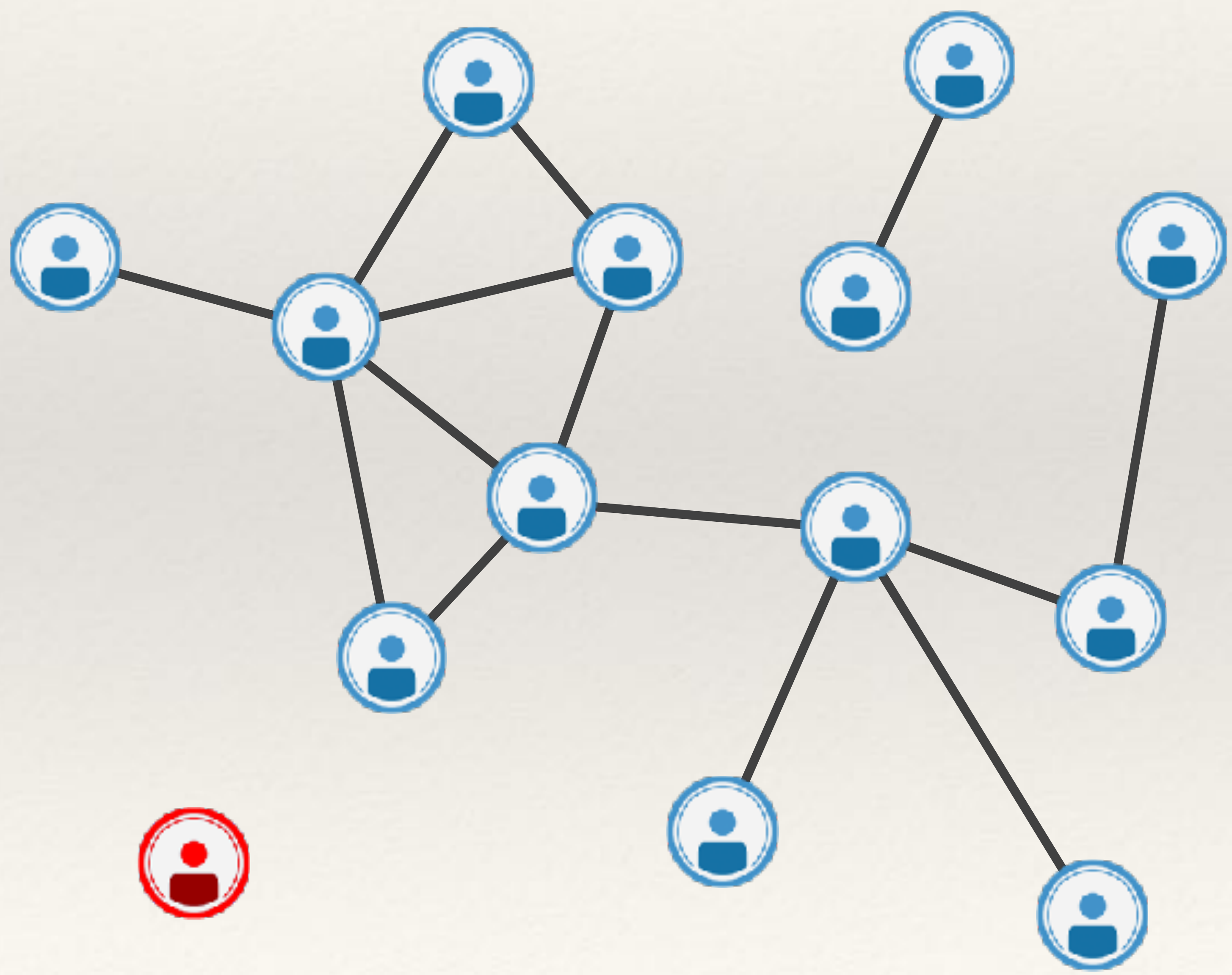
Identifying Key Players

What is the most important individual?



Identifying Key Players

What is the most important individual?



Identifying Key Players

Centrality Measures:

Different criteria to underline **different roles**:

- *Key players*
- *Brokers*
- *Bridges*
- *Isolated*
- …

A function $F : V \longrightarrow R$ assigning to each node a ***"centrality"*** value:

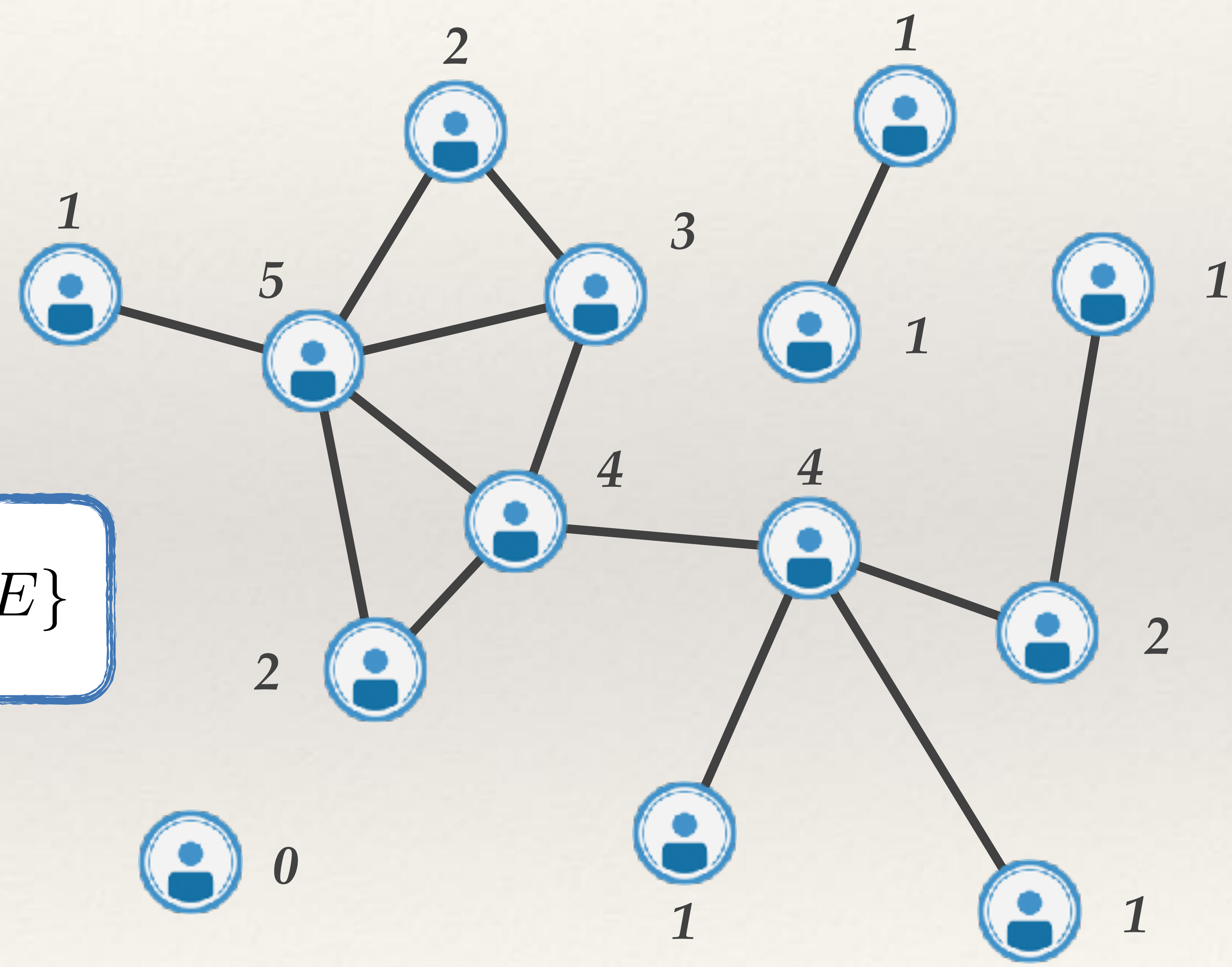
- ***Degree*** *centrality*
- ***Betweenness*** *centrality*
- ***Closeness*** *centrality*
- ***Eigenvector*** *centrality*
- ***Erdös*** *distance*

Identifying Key Players

Degree Centrality:

Given a node v of G=(V, E),

$$D(v) := \#\{u \in V \mid (u,v) \in E\}$$

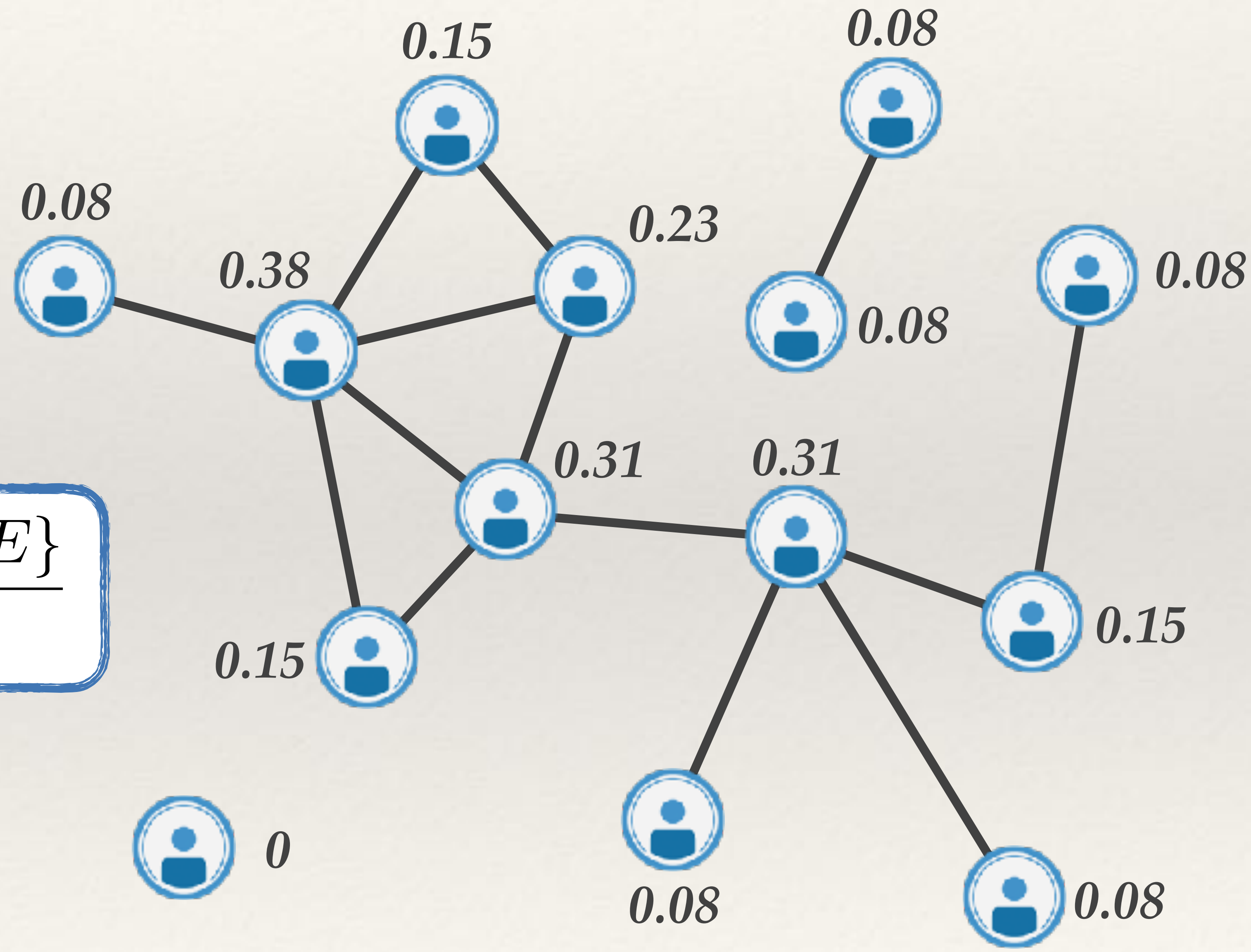


Identifying Key Players

Degree Centrality:

Given a node v of $G=(V, E)$,

$$D(v) := \frac{\#\{u \in V \mid (u,v) \in E\}}{\#V - 1}$$



Identifying Key Players

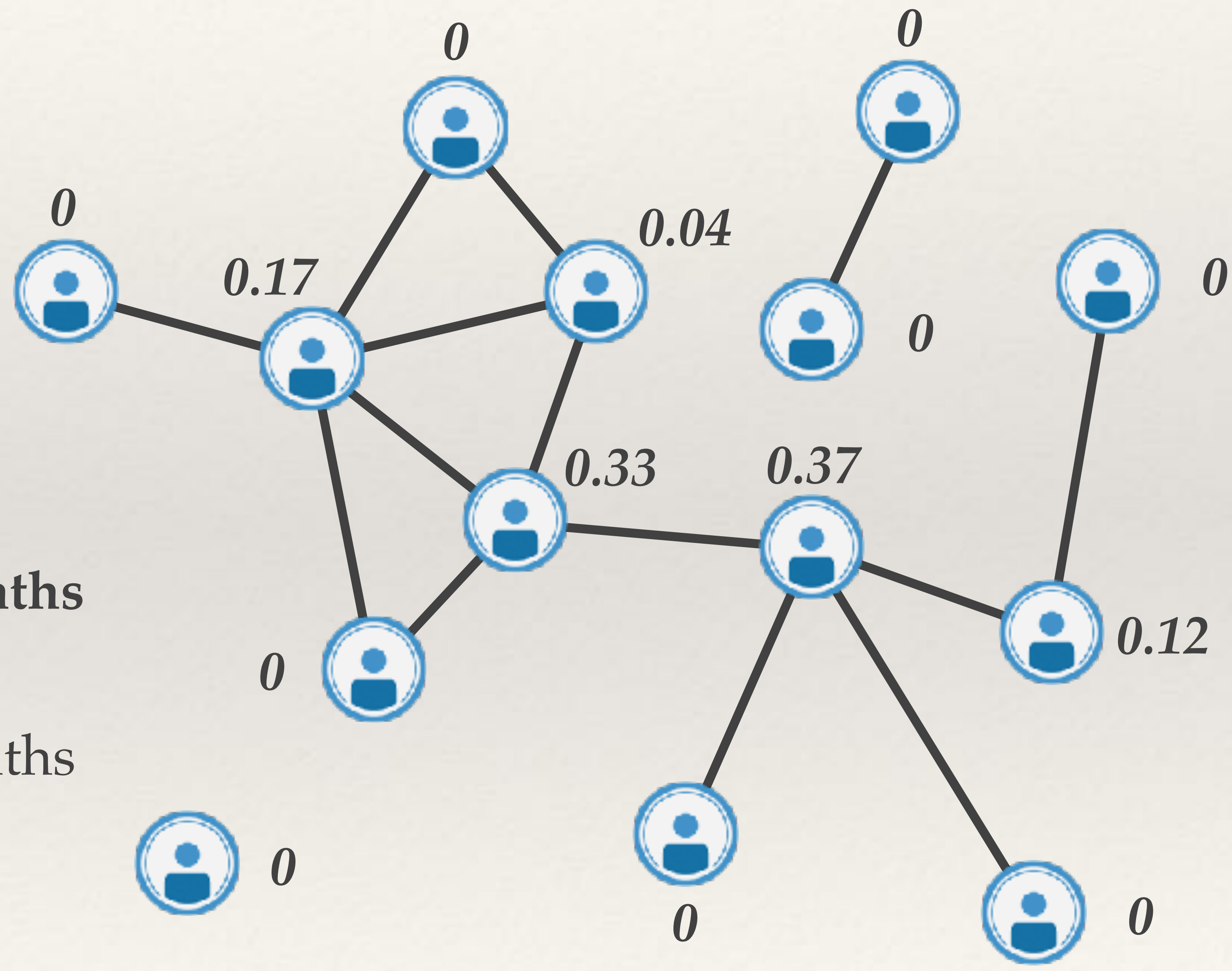
Betweenness Centrality:

Given a node v of $G=(V, E)$,

$$B(v) := \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

where:

- σ_{st} is the number of **shortest paths from s to t**
- $\sigma_{st}(v)$ is the number of those paths **passing through v**

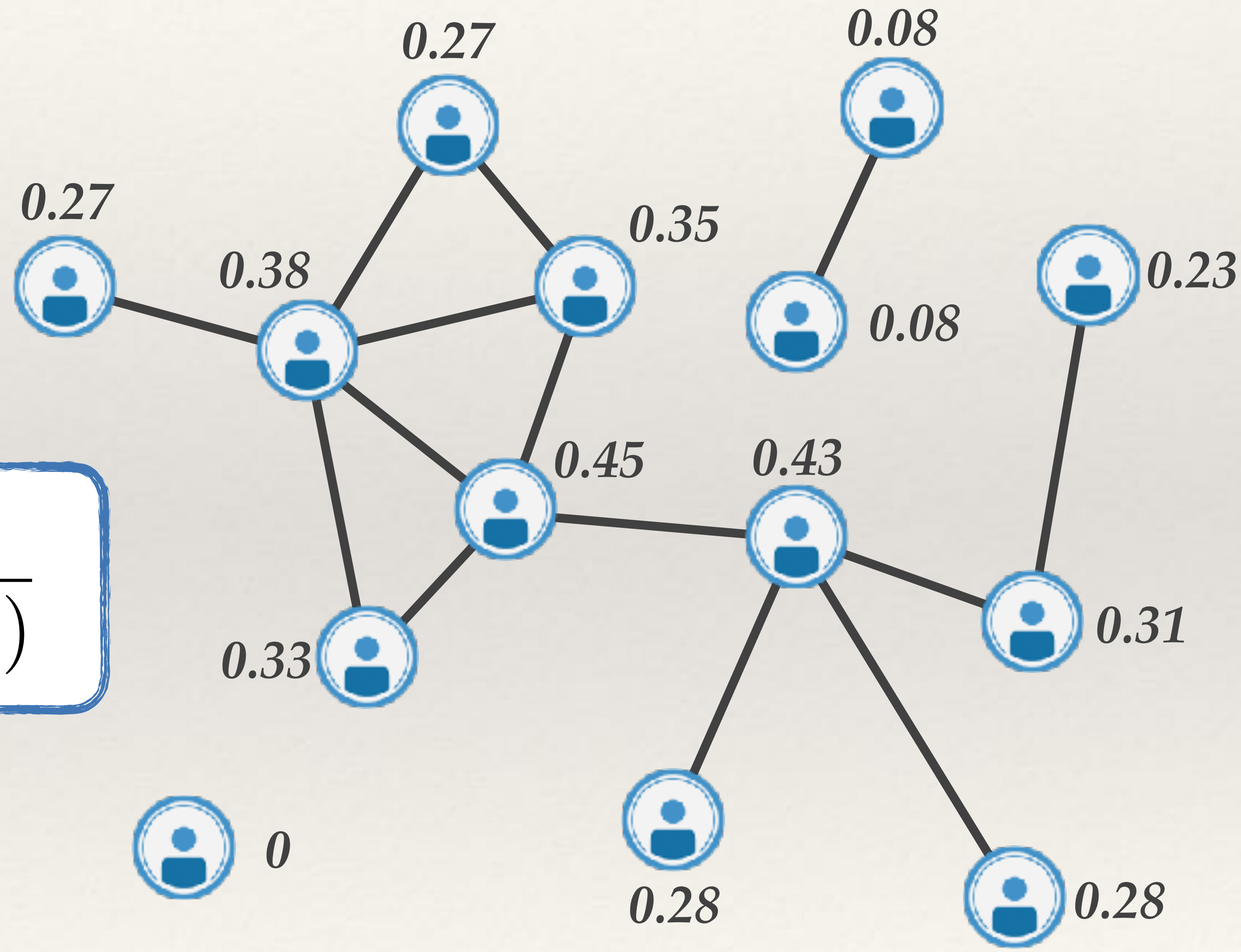


Identifying Key Players

Closeness Centrality:

Given a node v of $G=(V, E)$,

$$C(v) := \frac{\#V - 1}{\sum_{u \in V} d(u,v)}$$



Identifying Key Players

Eigenvector Centrality:

Given a node v of $G=(V, E)$,

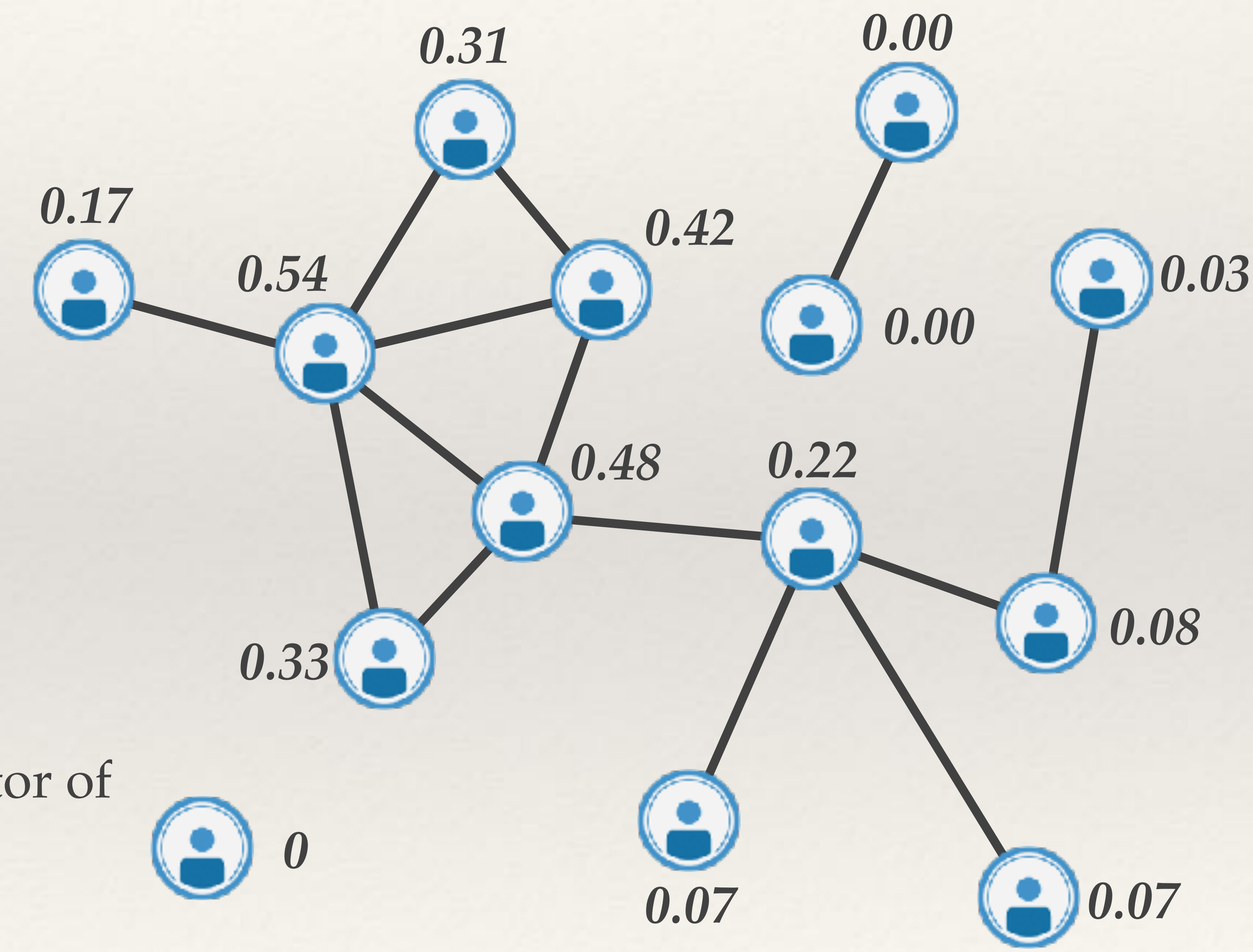
$$x_v := \frac{1}{\lambda} \sum_{u \in V} A_{uv} \, x_u$$

where λ is constant and

$$A_{uv} := \begin{cases} 1 & if(u,v) \in E \\ 0 & otherwise \end{cases}$$

i.e., the v^{th} entry of the eigenvector of

$$\boldsymbol{Ax} = \boldsymbol{\lambda x}$$



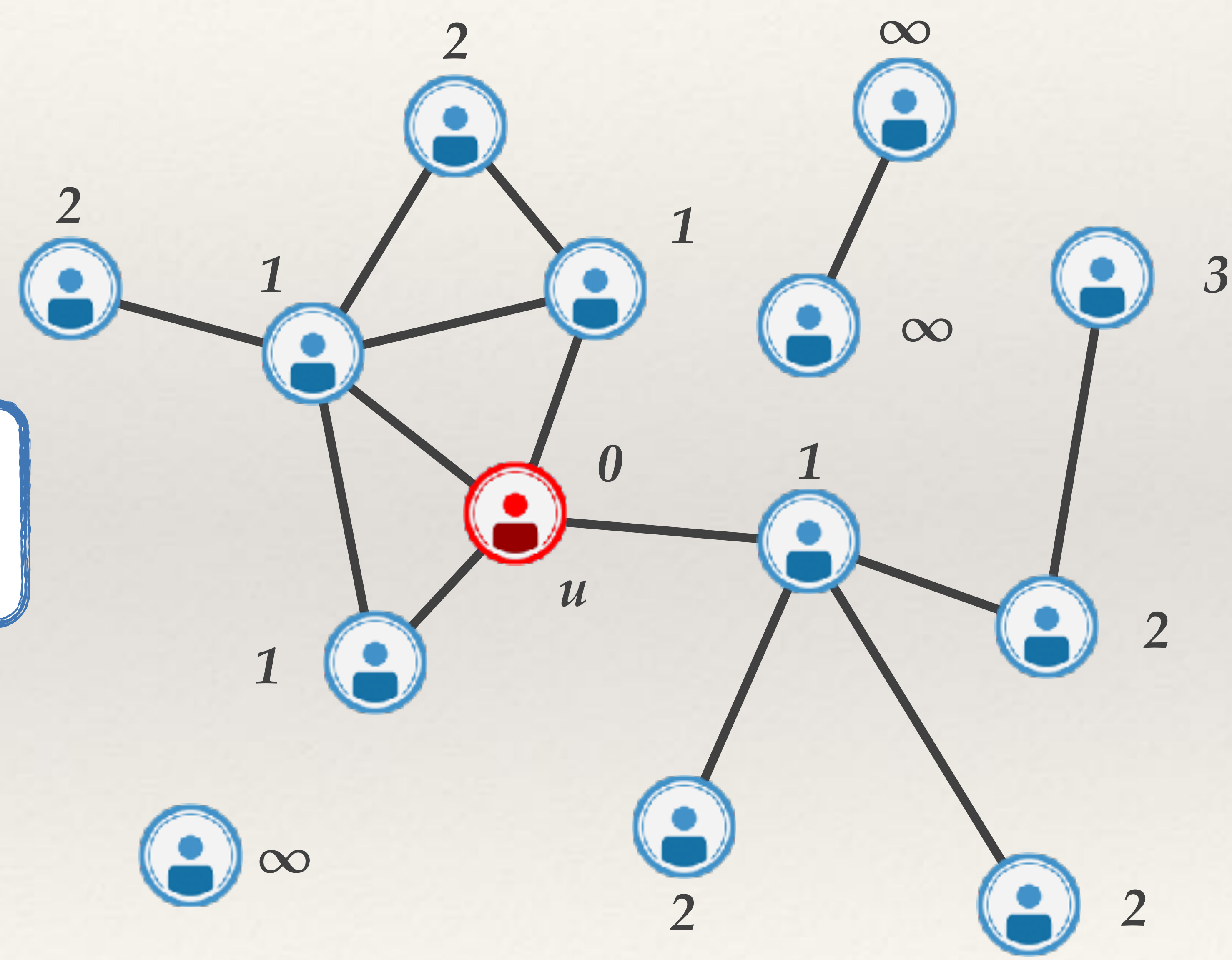
x*>*0 implies λ must be the largest eigenvalue of A and $\boldsymbol{x}$ the corresponding eigenvector

Identifying Key Players

Erdös Distance:

Given two nodes u, v of $G=(V, E)$,

$$E_u(v) := d(u, v)$$



Named after ***Paul Erdös***,

- one of the most prolific mathematicians of the 20th century

Identifying Key Players

Centrality Measures:

A centrality measure for ***any query***

Degree	***How many individuals can v reach directly?***
Betweenness	***How likely is v to be the most direct route between two individuals?***
Closeness	***How fast can v reach everyone in the network?***
Eigenvector	***How well is v connected to other well-connected individuals?***
Erdös	***How far is v from a specific individual?***

Structural Analysis

Sociocentric Networks:

- **Structural Metrics:**
  - *Average of a Centrality Measure*
  - *Diameter*
  - *Density*
  - *Transitivity*
  - …
- **Community Decomposition:**
  - *Atomic Communities*
  - *Clustering Techniques*

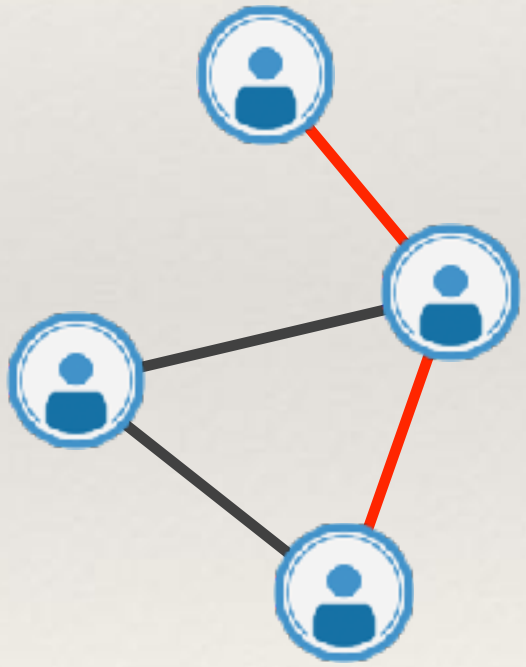
Structural Analysis

Structural Metrics:

- *How far are two individuals at most?*

Diameter:

The longest shortest path between any two nodes



$Diameter(G) = 2$

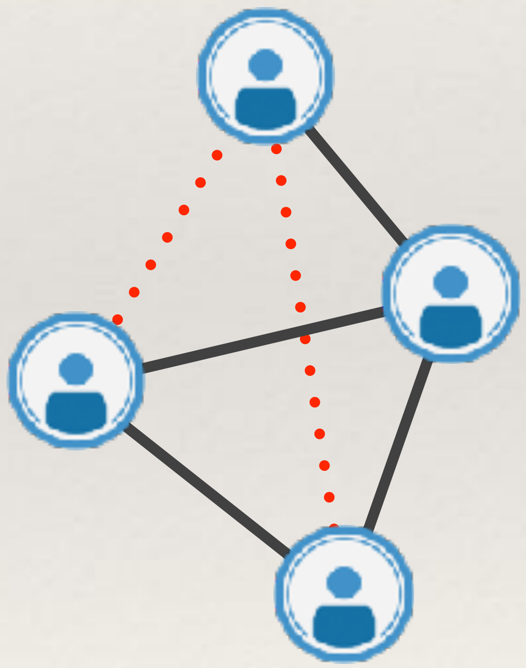
Structural Analysis

Structural Metrics:

- *How close is G to being an "everyone knows everyone" network?*

Density:

$$\frac{\textit{Number of edges of G}}{\textit{Number of all possible edges}}$$



Density(G) = 4/6= 0.67

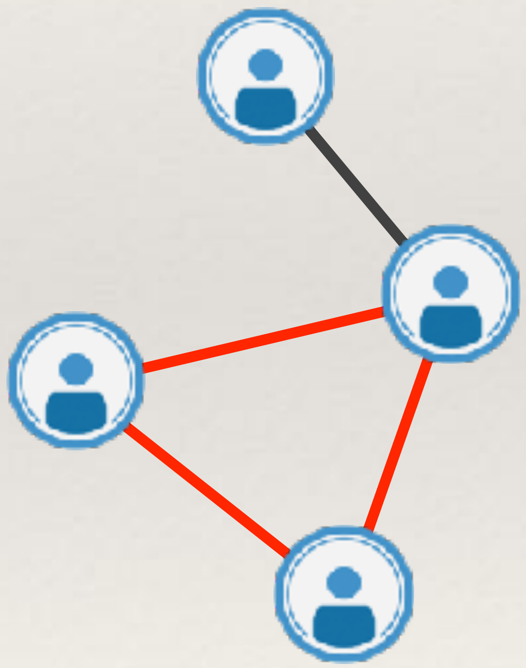
Structural Analysis

Structural Metrics:

- *How likely are two individuals connected to an individual v to be connected to each other?*

Transitivity:

$$\frac{\textit{Number of closed triplets of nodes}}{\textit{Number of connected triplets}}$$



Transitivity(G) = 1/3 = 0.33

Structural Analysis

Community Decomposition:

- *Atomic Communities:*
  - *Clique*
  - *n-Clique*
  - *n-Clan*
  - *n-Club*
  - *k-Plex*
  - *k-Core*
  - …

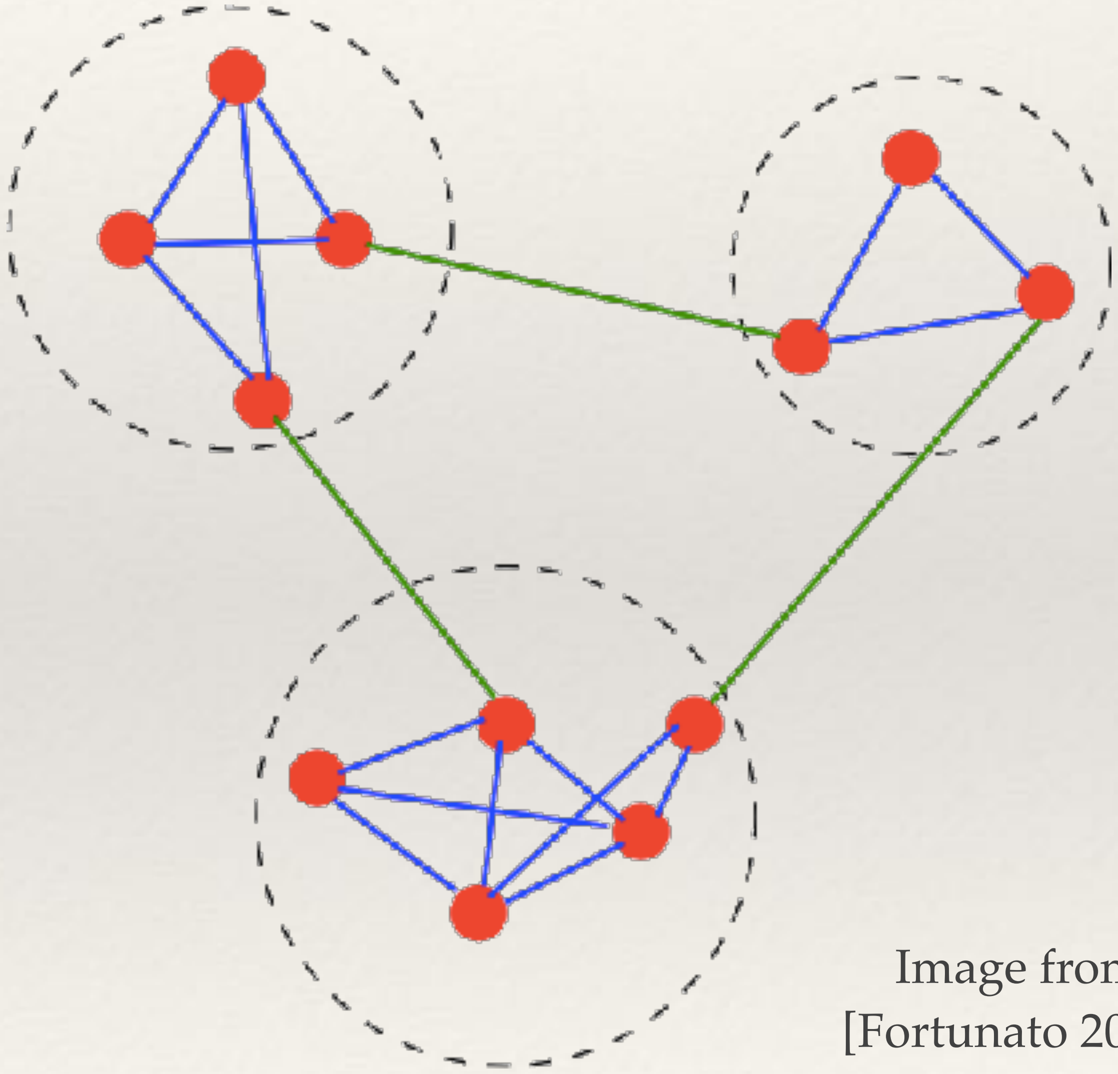


Image from [Fortunato 2009]

Structural Analysis

Community Decomposition:

- *Atomic Communities:*
  - *Clique*
  - *n-Clique*
  - *n-Clan*
  - *n-Club*
  - *k-Plex*
  - *k-Core*
  - …

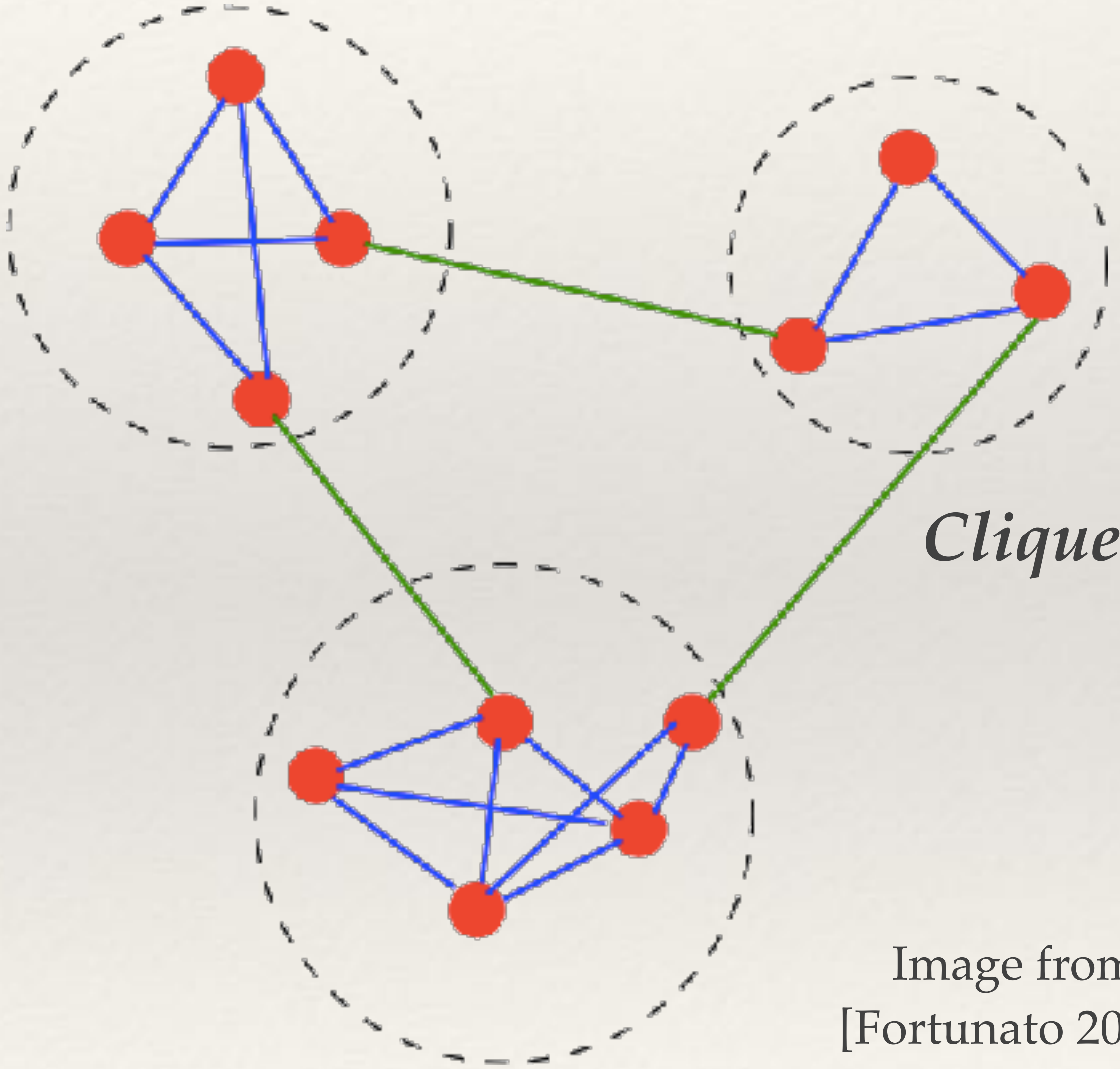


Image from [Fortunato 2009]

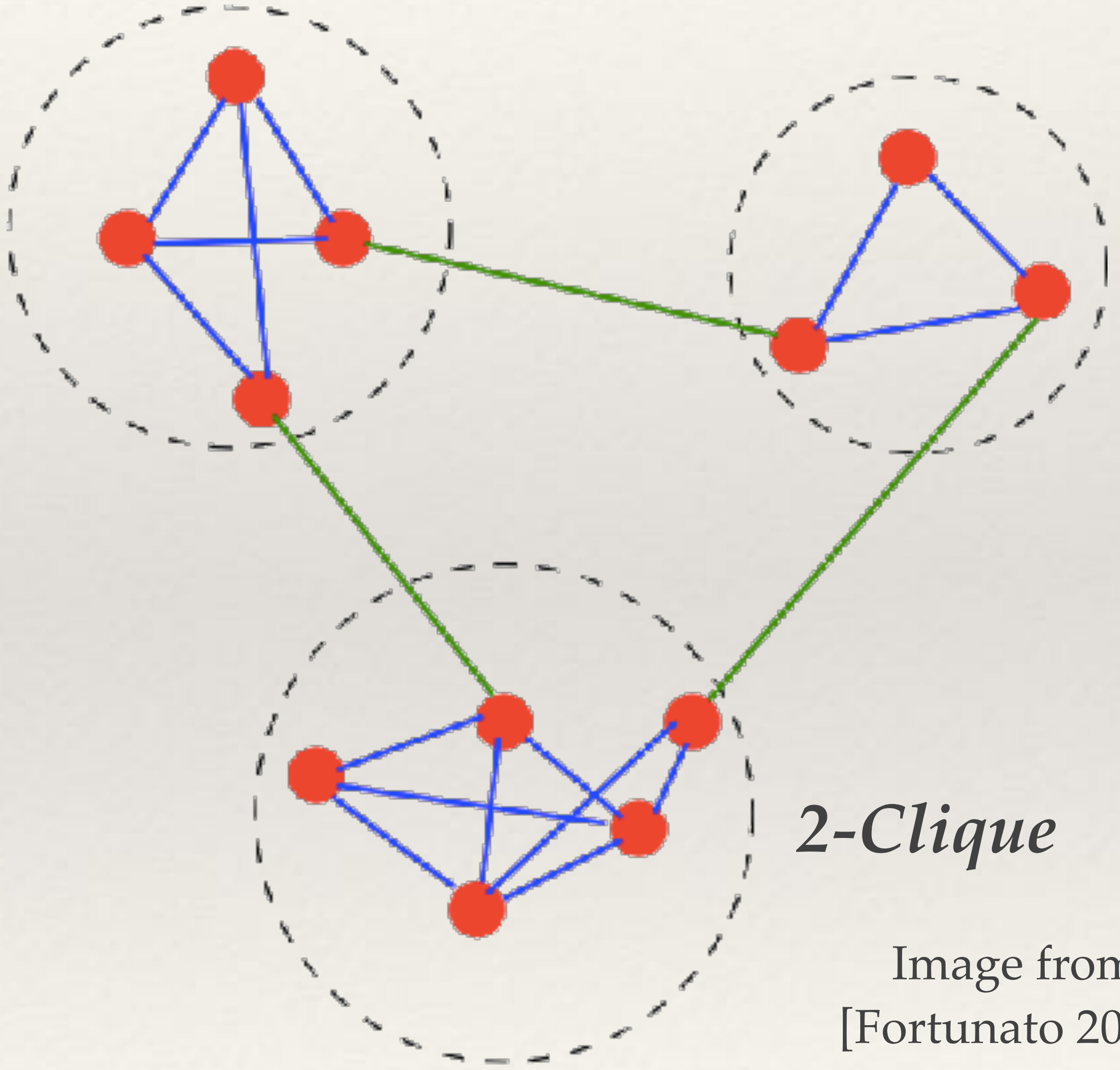
Clique:

maximal subgraph whose nodes are all adjacent to each other

Structural Analysis

Community Decomposition:

- *Atomic Communities:*
  - *Clique*
  - *n-Clique*
  - *n-Clan*
  - *n-Club*
  - *k-Plex*
  - *k-Core*
  - …



2-Clique

Image from
[Fortunato 2009]

***n*-Clique:**

maximal subgraph such that the distance of each pair of its nodes is not greater than n

Structural Analysis

Community Decomposition:

- *Atomic Communities:*
  - *Clique*
  - *n-Clique*
  - *n-Clan*
  - *n-Club*
  - *k-Plex*
  - *k-Core*
  - …

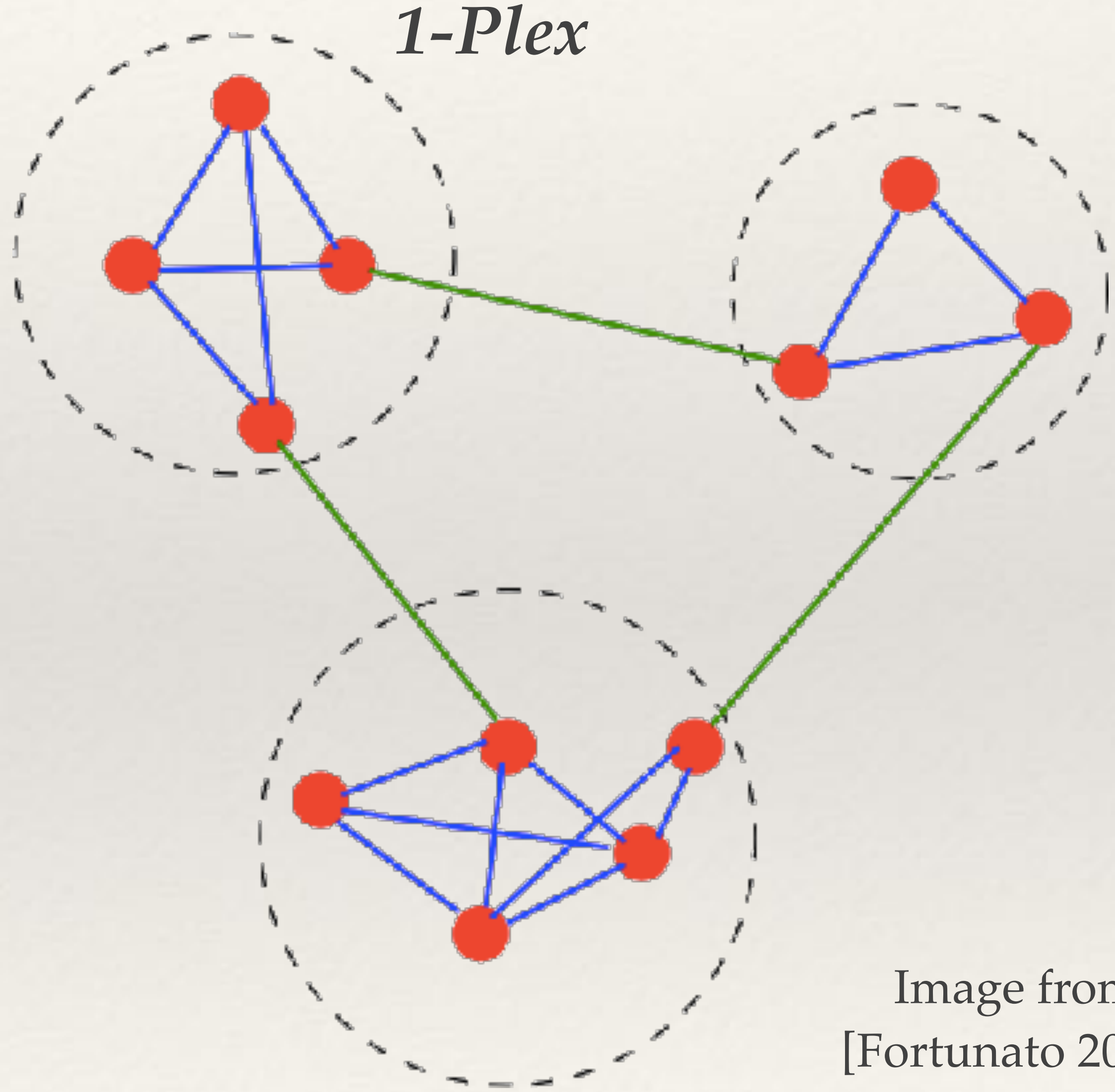


Image from [Fortunato 2009]

***k*-Plex:**

maximal subgraph in which each node is adjacent to all other nodes of the subgraph except at most *k* of them

Structural Analysis

Clustering Techniques:

Agglomerative (bottom-up)
Divisive (top-dow)

approach based on

- Centrality Measures
- Atomic Communities
- Quality Functions

Structural Analysis

Clustering Techniques:

Agglomerative (bottom-up)

Divisive (top-dow)

approach based on

Centrality Measures

Atomic Communities

Quality Functions

Girvan-Newman Algorithm:

Iterated removal of the edge with largest ***betweenness centrality***

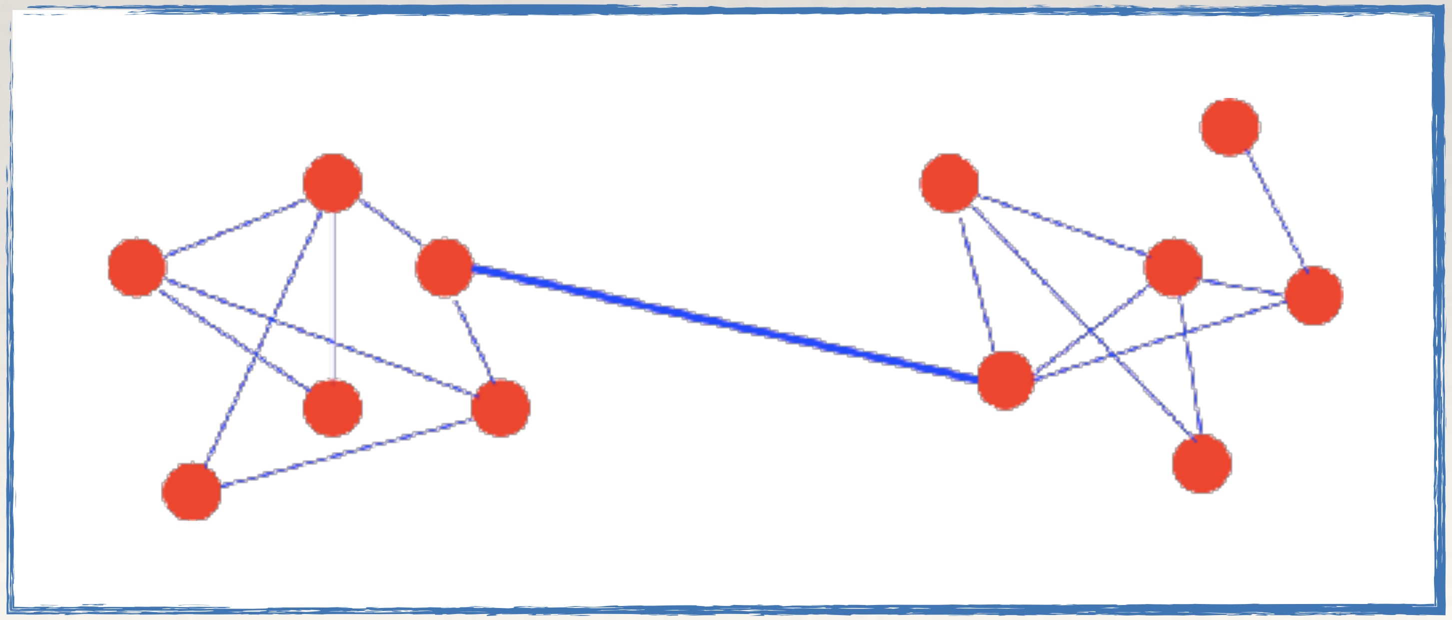


Image from [Fortunato 2009]

Structural Analysis

Clustering Techniques:

Agglomerative (bottom-up)

Divisive (top-dow)

approach based on

Centrality Measures

Atomic Communities

Quality Functions

Clique Percolation:

k-adjacency: two clique of size *k* are *k*-adjacent if they share *k-1* nodes

k-clique community: maximal union of cliques of size *k* pairwise connected by a sequence of *k*-adjacent cliques

Decomposition in *k*-clique communities

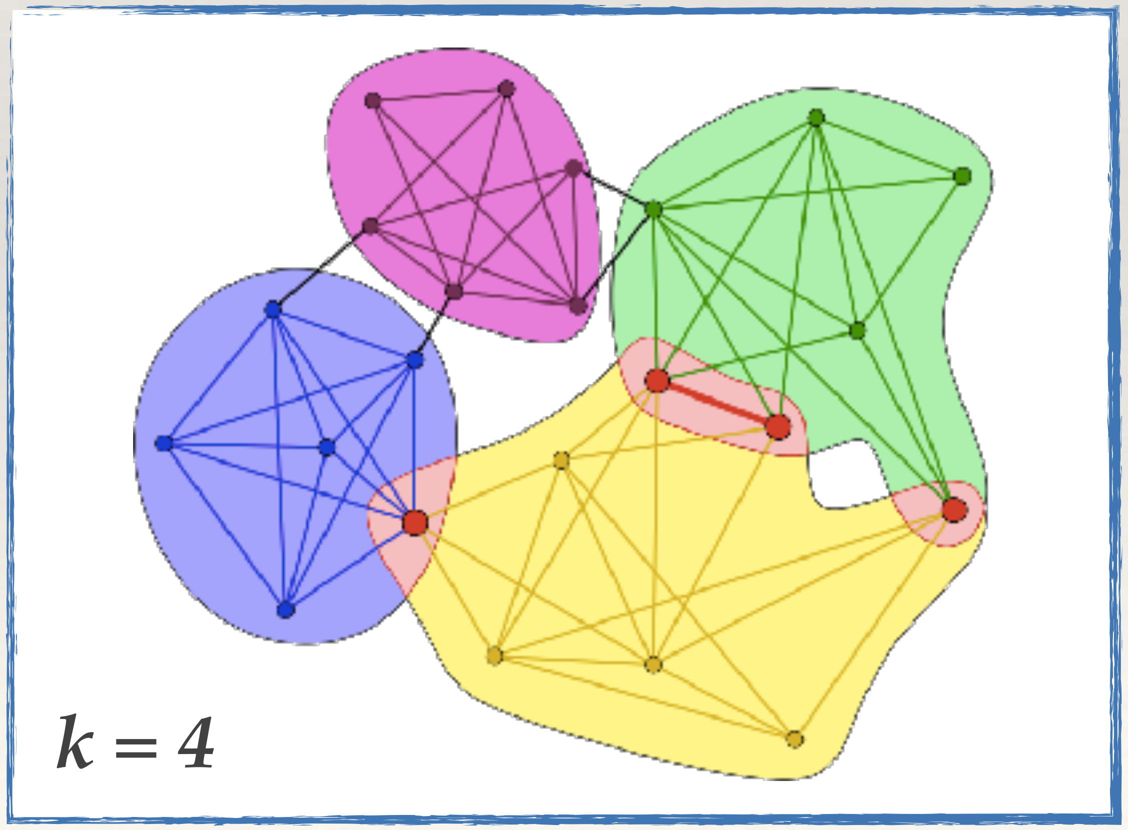


Image from [Palla et al. 2005]

Structural Analysis

Clustering Techniques:

Agglomerative (bottom-up)

Divisive (top-dow)

approach based on

Centrality Measures

Atomic Communities

Quality Functions

Modularity-based Algorithm:

Modularity: measure for clustering quality

Iterated aggregation of communities of nodes whose merging *increases modularity*

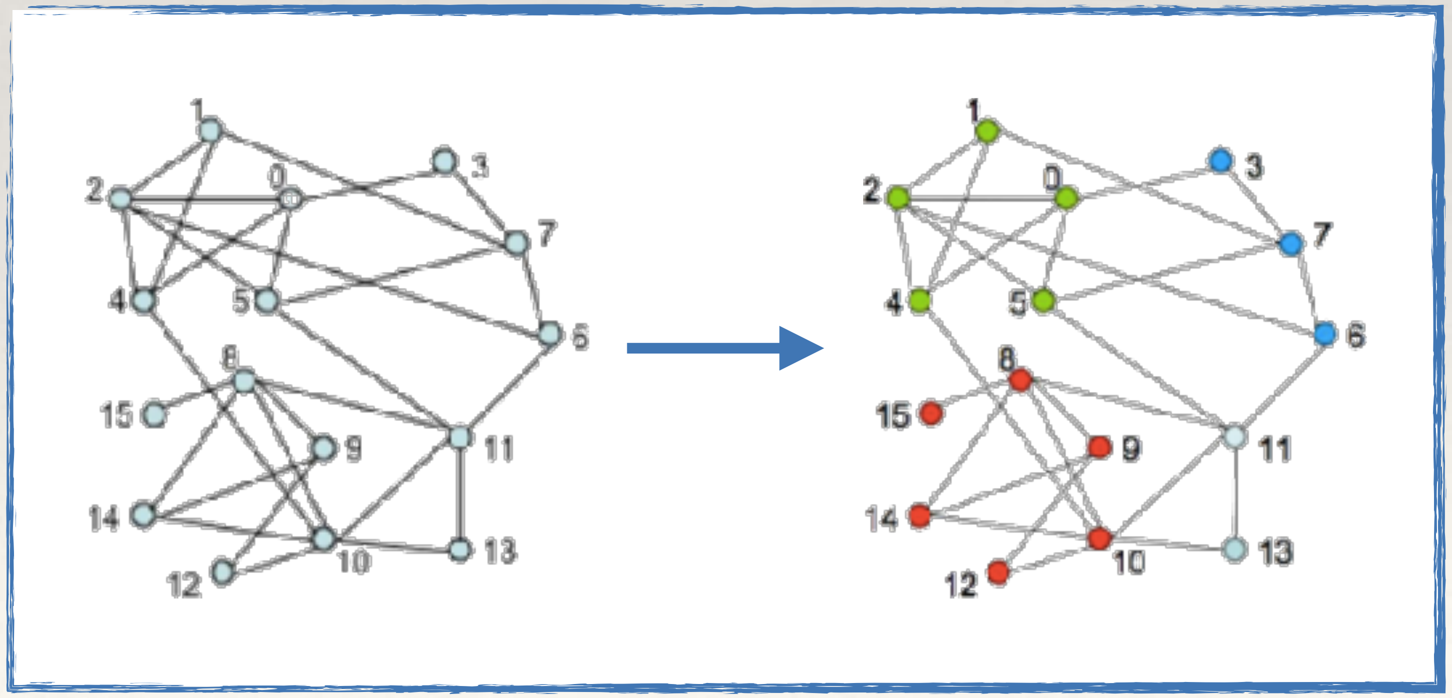


Image from [Blondel et al. 2008]

Outline

Brief Introduction to Complex Network Analysis

Persistence-based Network Analysis

Persistence-based Network Analysis

Several Application based on Persistent Homology:

- *Sensor* Networks [De Silva 2013]
- *Brain* Networks [Lee et al. 2012]
- *Collaborative/Co-occurence* Networks [Carstens et al. 2013; Rieck et al. 2016]
- *Geolocalized* Networks [Fellegara et al. 2016]
- …

Simplicial Complex Representation:

A network is represented through:

- Simplicial complex *Flag(G)* induced by G
  - *simplices of Flag(G)* ⟷ *cliques of G*

Persistence-based Network Analysis

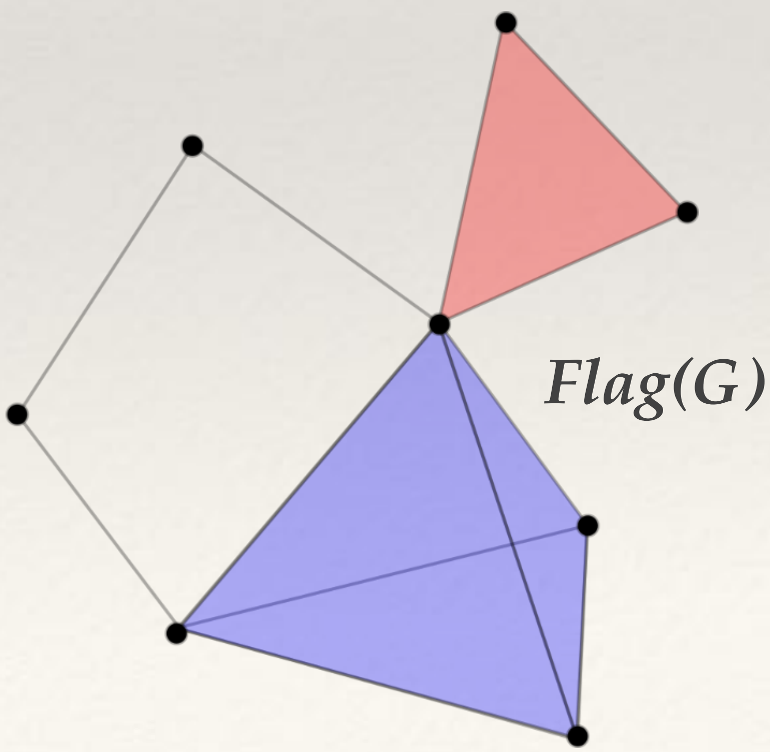
Several Application based on Persistent Homology:

- *Sensor* Networks [De Silva 2013]
- *Brain* Networks [Lee et al. 2012]
- *Collaborative/Co-occurence* Networks [Carstens et al. 2013; Rieck et al. 2016]
- *Geolocalized* Networks [Fellegara et al. 2016]
- …

Simplicial Complex Representation:

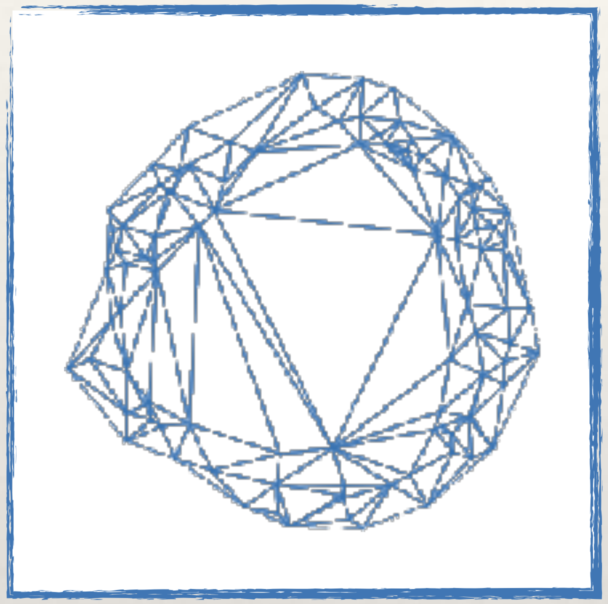
A network is represented through:

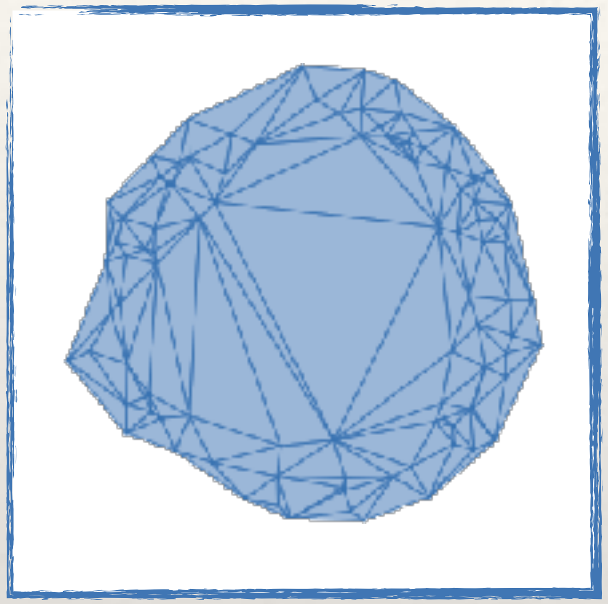
- Simplicial complex *Flag(G)* induced by G
  - *simplices of Flag(G)* ⟷ *cliques of G*

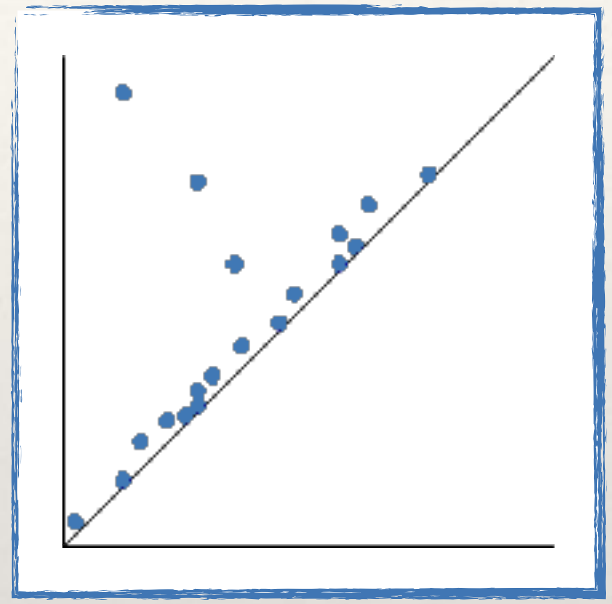


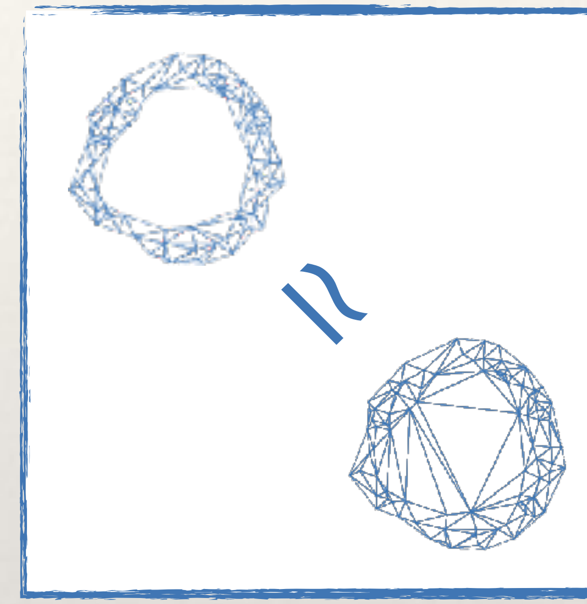
Persistence-based Network Analysis

A Common Pipeline in TDA:









Complex Construction *PH_k module* *Distance Computation*

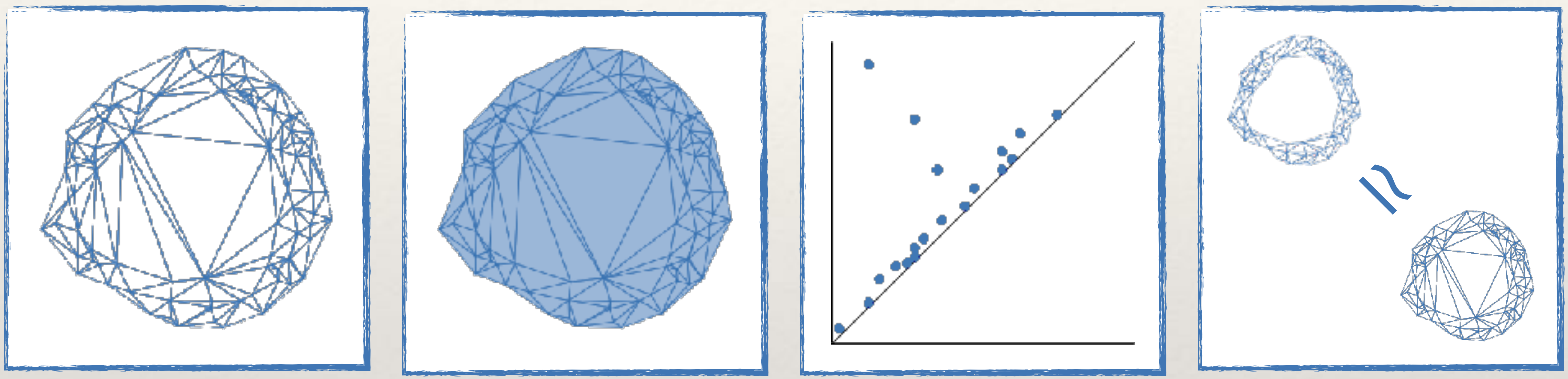
Topological Summaries have proven to be particularly effective to **distinguish shapes**

but

It's still hard to give a **meaningful interpretation** of what homological cycles represent

Persistence-based Network Analysis

A Common Pipeline in TDA:



Complex Construction *PH_k module* *Distance Computation*

What if …

we replace the PH_k module with something having a more natural interpretation?

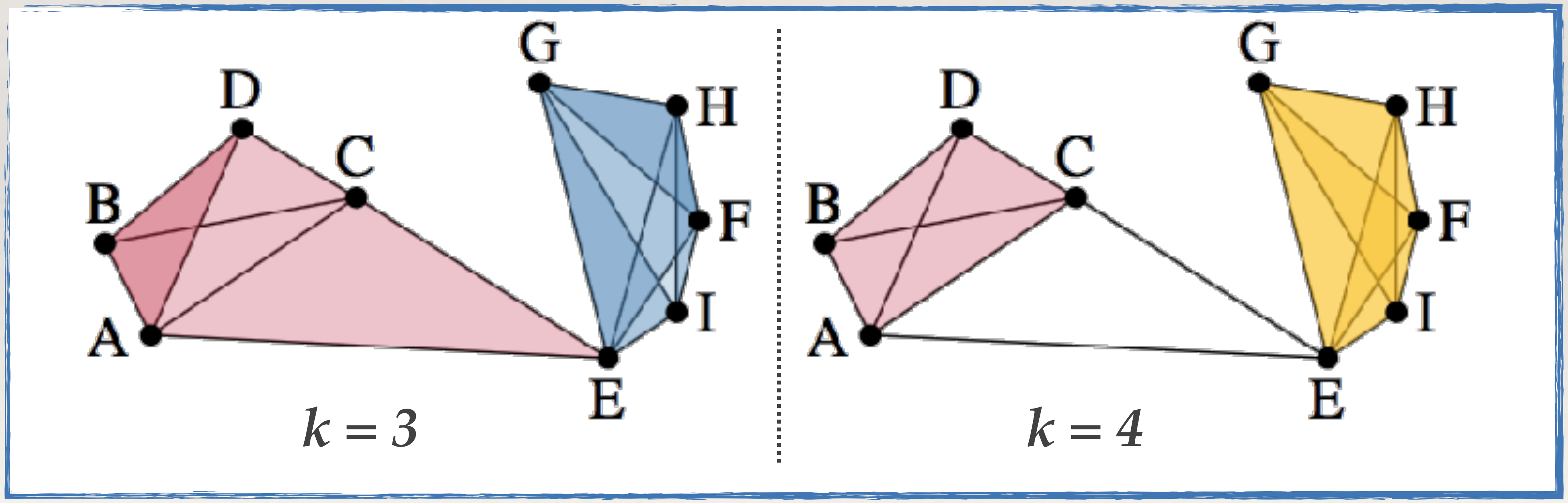
Persistence-based Network Analysis

k-Clique Communities:

Given a network $G=(V, E)$,

***Clique of size k*:**

a complete subgraph of k vertices of G ⇔ *a (k-1)-simplex of Flag(G)*



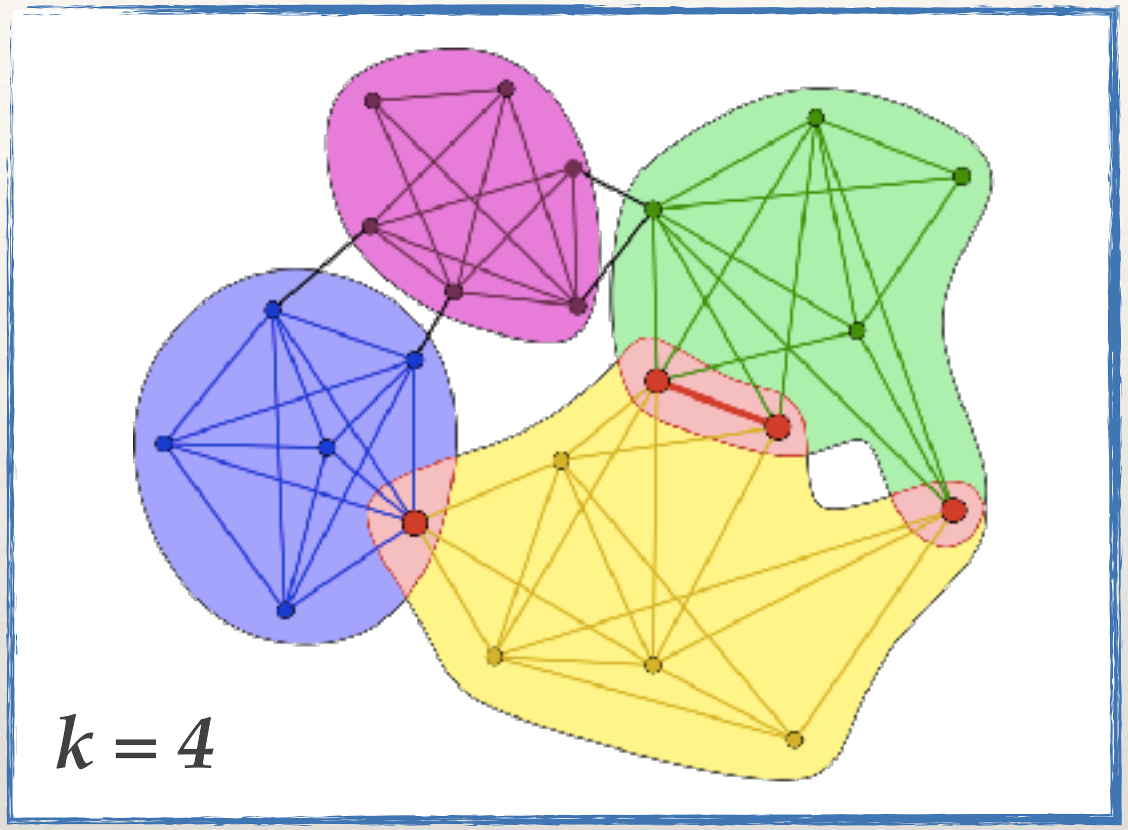
***k-adjacency*:** two clique of size k are k-adjacent if they share k-1 nodes

***k-clique community*:** maximal union of cliques of size k pairwise connected by a sequence of k-adjacent cliques

Persistence-based Network Analysis

k-Clique Community Decomposition:

- *Reveal **Highly Connected** Communities*
- *Allow **Overlaps***
- *Have a **Hierarchical Structure***

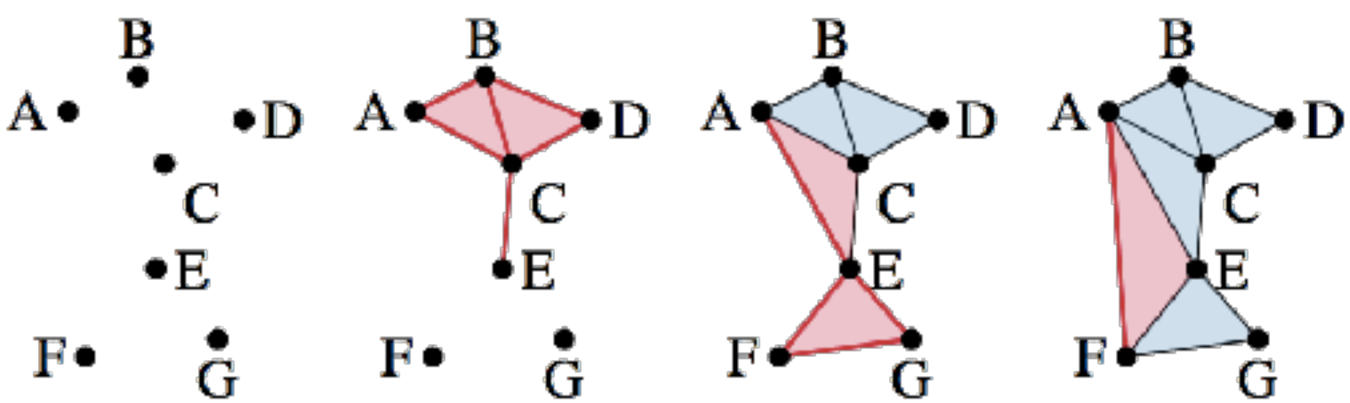


- *Difficult to be Computed*
- *What is the right k?*
- *How to Manage Weighted Networks?*

Persistence-based Network Analysis

Clique Community Persistence:

Use **Persistent Homology** *to keep track of the evolution of these communities*

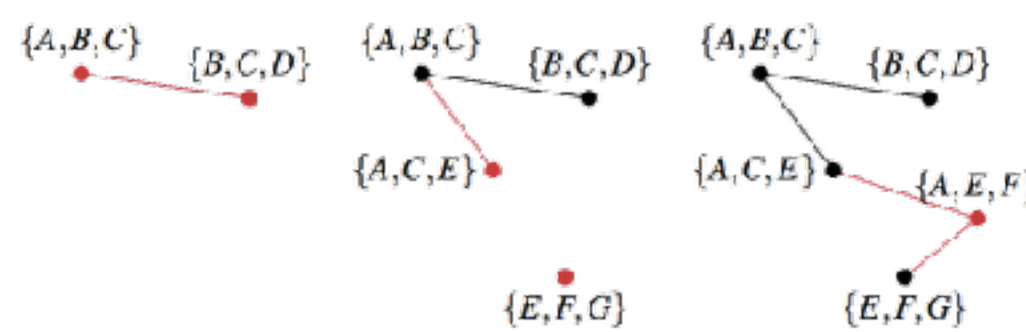


(a) $\varepsilon = 0.0$ (b) $\varepsilon = 1.0$ (c) $\varepsilon = 2.0$ (d) $\varepsilon = 3.0$

k-clique communities of G

0-homology of G_k

(its connected components)



(a) $\varepsilon = 0.0$ (b) $\varepsilon = 1.0$ (c) $\varepsilon = 2.0$ (d) $\varepsilon = 3.0$

Persistence-based Network Analysis

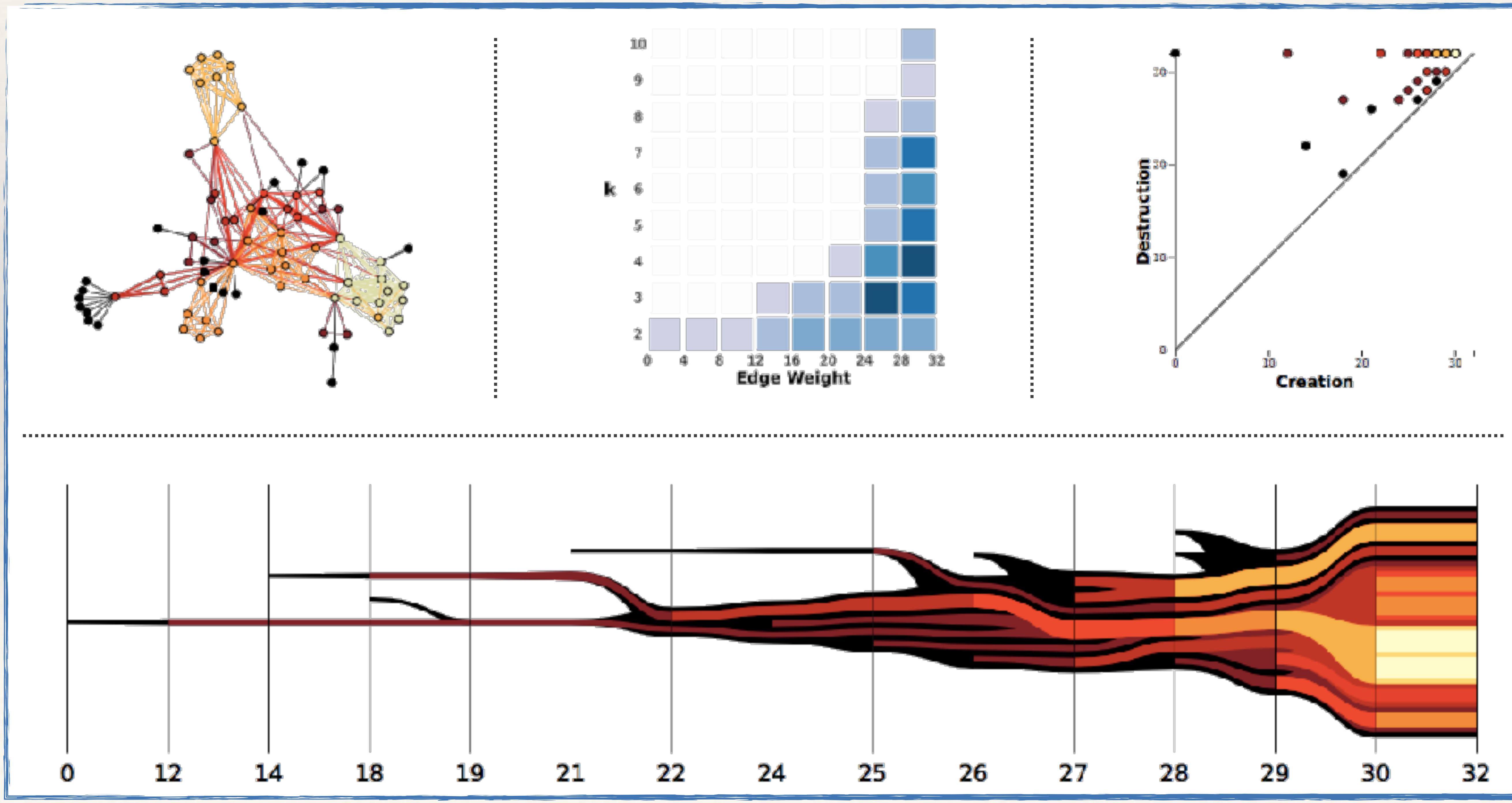
Clique Community Persistence:

The presented approach allows for designing:

- **Nested Graph Visualization**
- Definition of a new **Centrality Measure**
- Introduction of a **Persistence Indicator Function**
- …

Persistence-based Network Analysis

Nested Graph Visualization:

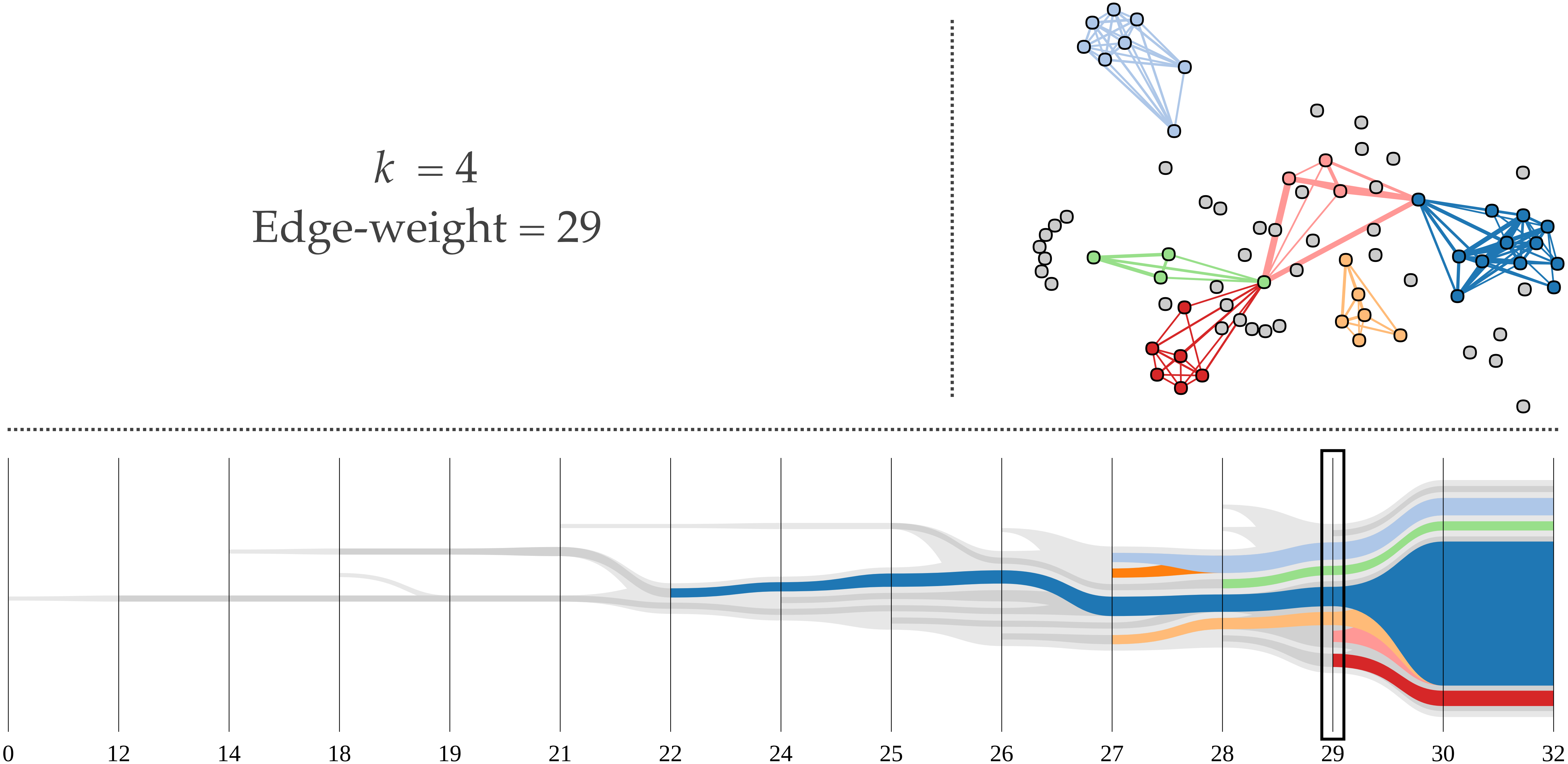


Persistence-based Network Analysis

Nested Graph Visualization:

$k = 4$

Edge-weight = 29



Outline

Brief Introduction to Complex Network Analysis

Persistence-based Network Analysis

Thank you

Ulderico Fugacci

***TU Kaiserslautern**, Dept. of Computer Science*