An introduction to network inference and mining

Nathalie Villa-Vialaneix - nathalie.villa@toulouse.inra.fr
http://www.nathalievilla.org

INRA, UR 875 MIAT

Formation Biostatistique, Niveau 3

Outline

1. A brief introduction to networks/graphs
2. Network inference
3. Simple graph mining
   - Visualization
   - Global characteristics
   - Numerical characteristics calculation
   - Clustering

Outline

1 A brief introduction to networks/graphs

2 Network inference

3 Simple graph mining
- Visualization
- Global characteristics
- Numerical characteristics calculation
- Clustering

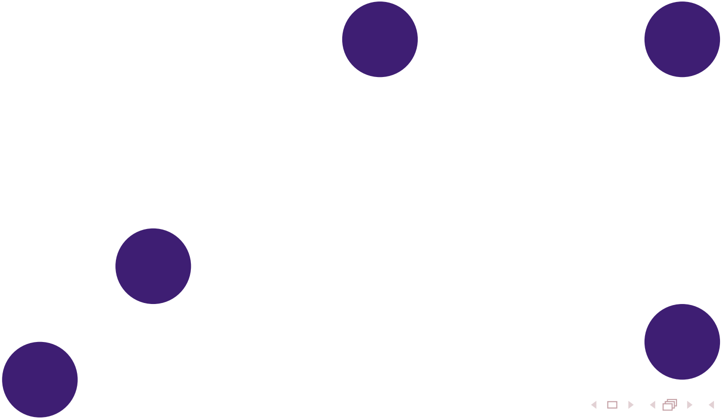
What is a network/graph? *réseau/graphe*

Mathematical object used to model **relational data between entities**.

What is a network/graph? *réseau/graphe*

Mathematical object used to model **relational data between entities**.

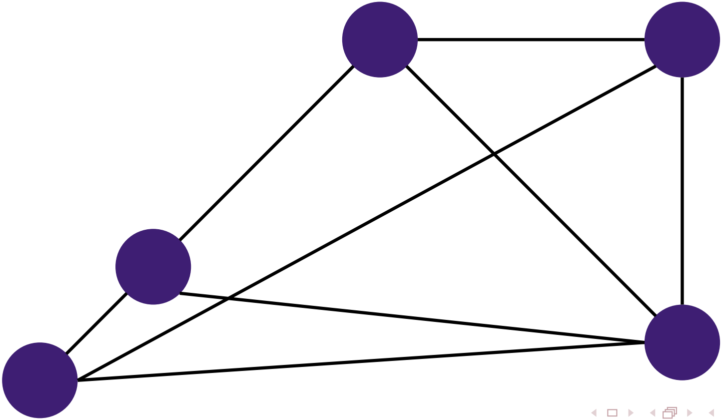
The entities are called the **nodes** or the **vertexes** (vertices in British) *nœuds/sommets*



What is a network/graph? *réseau/graphe*

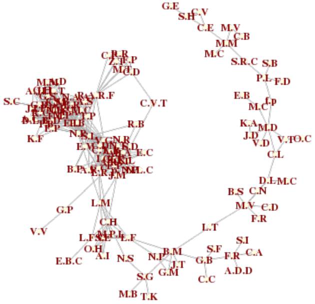
Mathematical object used to model **relational data between entities**.

A relation between two entities is modeled by an **edge** *arête*



(non biological) Examples

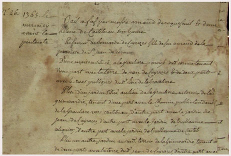
Social network: nodes: persons - edges: 2 persons are connected ("friends")



(Natty's facebook™[1] network)

(non biological) Examples

Modeling a large corpus of medieval documents

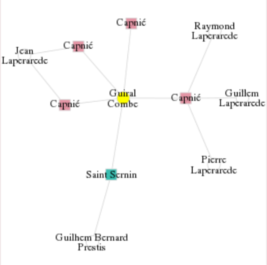


Notarial acts (mostly *baux à fief*, more precisely, land charters) established in a *seigneurie* named "Castelnau Montratier", written between 1250 and 1500, involving tenants and lords.[a]

[a] http://graphcomp.univ-tlse2.fr

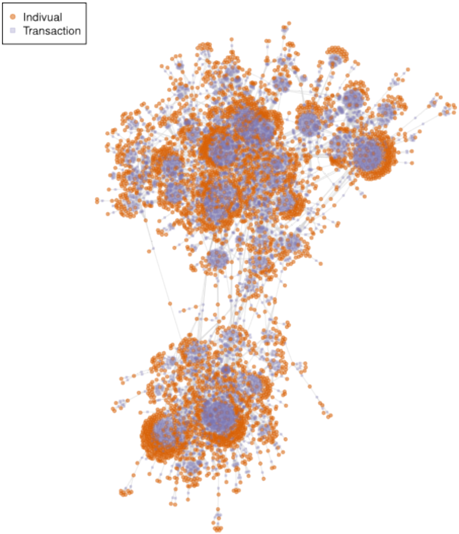
(non biological) Examples

Modeling a large corpus of medieval documents



- nodes: transactions and individuals (3 918 nodes)
- edges: an individual is directly involved in a transaction (6 455 edges)

(non biological) Examples



Standard issues associated with networks

Inference

Giving data, how to build a graph whose edges represent the **direct** links between variables?

Example: co-expression networks built from microarray data (nodes = genes; edges = significant "direct links" between expressions of two genes)

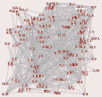
Standard issues associated with networks

Inference

Giving data, how to build a graph whose edges represent the **direct** links between variables?

Graph mining (examples)

1. **Network visualization**: nodes **are not** a priori associated to a given position. How to represent the network in a meaningful way?



Random positions

Positions aiming at representing connected nodes closer

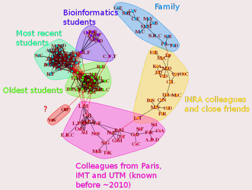
Standard issues associated with networks

Inference

Giving data, how to build a graph whose edges represent the **direct** links between variables?

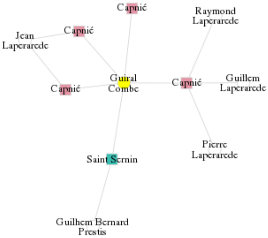
Graph mining (examples)

1. **Network visualization**: nodes **are not** a priori associated to a given position. How to represent the network in a meaningful way?
2. **Network clustering**: identify "communities" (groups of nodes that are densely connected and share a few links (comparatively) with the other groups)



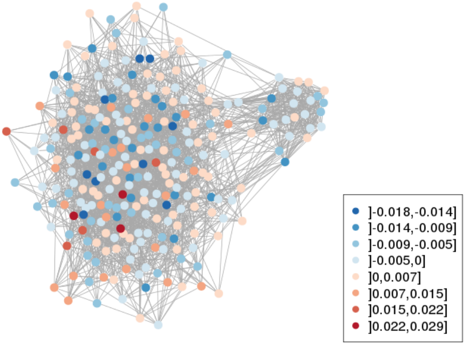
More complex relational models

Nodes may be **labeled** by a factor



More complex relational models

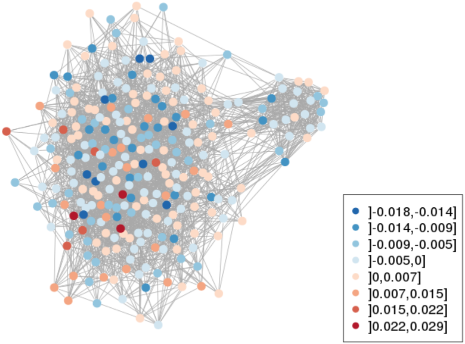
Nodes may be **labeled** by a factor



... or by a numerical information. [Laurent and Villa-Vialaneix, 2011]

More complex relational models

Nodes may be **labeled** by a factor



... or by a numerical information. [Laurent and Villa-Vialaneix, 2011]

Edges may also be labeled (type of the relation) or weighted (strength of the relation) or directed (direction of the relation).

Outline

1 A brief introduction to networks/graphs

2 Network inference

3 Simple graph mining
 Visualization
 Global characteristics
 Numerical characteristics calculation
 Clustering

Framework

Data: large scale gene expression data

$$\begin{array}{c} \text{individuals} \\ n \simeq 30/50 \end{array} \left\{ \underbrace{X = \begin{pmatrix} . & . & . & . & . & . \\ . & . & X_i^j & . & . & . \\ . & . & . & . & . & . \end{pmatrix}}_{\text{variables (genes expression)},\ p \simeq 10^{3/4}} \right.$$

What we want to obtain: a network with

- nodes: genes;
- edges: significant and direct co-expression between two genes (track transcription regulations)

Advantages of inferring a network from large scale transcription data

1. **over raw data**: **focuses on the strongest direct relationships**: irrelevant or indirect relations are removed (more robust) and the data are easier to visualize and understand.
   Expression data are **analyzed all together** and not by pairs.

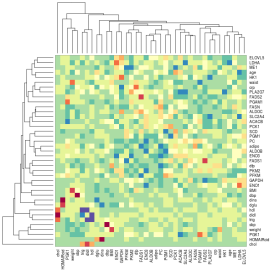
Advantages of inferring a network from large scale transcription data

1. **over raw data**: **focuses on the strongest direct relationships**: irrelevant or indirect relations are removed (more robust) and the data are easier to visualize and understand.
   Expression data are **analyzed all together** and not by pairs.

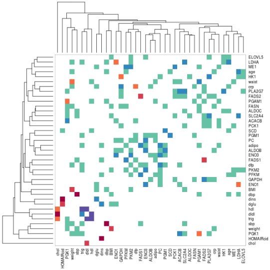
2. **over bibliographic network**: can handle **interactions with yet unknown** (not annotated) **genes** and deal with data collected in a particular condition.

Using *correlations*: relevance network [Butte and Kohane, 1999, Butte and Kohane, 2000]

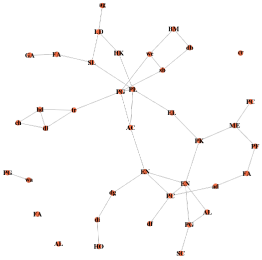
First (naive) approach: calculate correlations between expressions for all pairs of genes, threshold the smallest ones and build the network.



"Correlations"

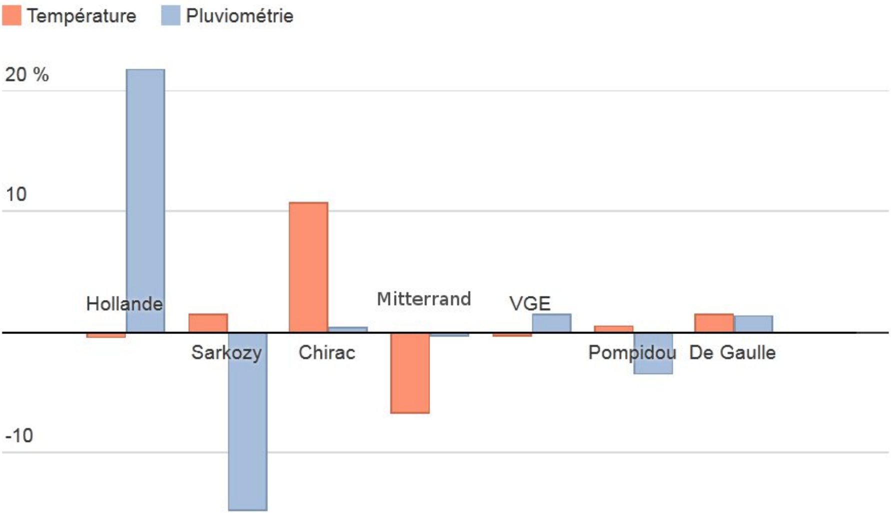


Thresholding

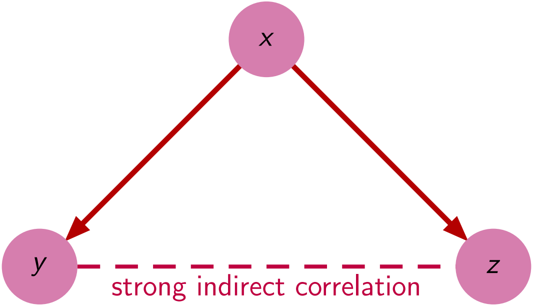


Graph

But correlation is not causality...

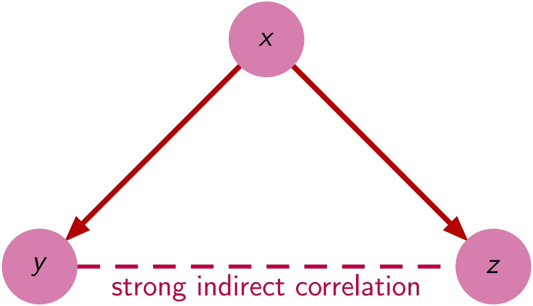


But correlation is not causality...



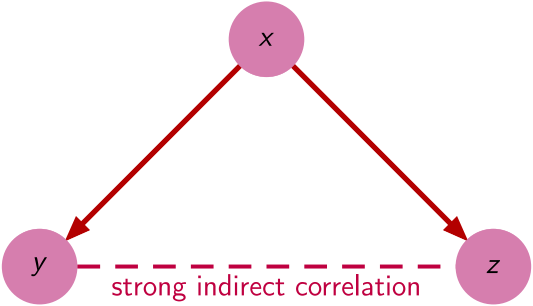

```
set.seed(2807); x <- runif(100)
y <- 2*x+1+rnorm(100,0,0.1); cor(x,y); [1] 0.9988261
z <- 2*x+1+rnorm(100,0,0.1); cor(x,z); [1] 0.998751
cor(y,z); [1] 0.9971105
```

But correlation is not causality...




```
set.seed(2807); x <- runif(100)
y <- 2*x+1+rnorm(100,0,0.1); cor(x,y); [1] 0.9988261
z <- 2*x+1+rnorm(100,0,0.1); cor(x,z); [1] 0.998751
cor(y,z); [1] 0.9971105
♯ Partial correlation
cor(lm(y∼x)$residuals,lm(z∼x)$residuals) [1] -0.1933699
```

But correlation is not causality...



Networks are built using **partial correlations**, i.e., correlations between gene expressions **knowing the expression of all the other genes** (residual correlations).

Various approaches (and R packages) to infer gene expression networks

- **Graphical Gaussian Model** $(X_i)_{i=1,\ldots,n}$ are i.i.d. Gaussian random variables $\mathcal{N}(0,\Sigma)$ (gene expression); then

$$j \longleftrightarrow j' \text{(genes } j \text{ and } j' \text{ are linked)} \Leftrightarrow \mathbb{C}\mathrm{or}\left(X^j, X^{j'} | (X^k)_{k\neq j,j'}\right) > 0$$

$\mathbb{C}\mathbf{or}\left(X^j, X^{j'} | (X^k)_{k\neq j,j'}\right) \simeq \left(\Sigma^{-1}\right)_{j,j'} \Rightarrow$ find the partial correlations by means of $(\widehat{\Sigma}^n)^{-1}$.

Various approaches (and R packages) to infer gene expression networks

- **Graphical Gaussian Model** $(X_i)_{i=1,\ldots,n}$ are i.i.d. Gaussian random variables $\mathcal{N}(0,\Sigma)$ (gene expression); then

$$j \longleftrightarrow j' (\text{genes } j \text{ and } j' \text{ are linked}) \Leftrightarrow \mathbb{C}\mathrm{or}\left(X^j, X^{j'} | (X^k)_{k\neq j,j'}\right) > 0$$

$\mathbb{C}\mathrm{or}\left(X^j, X^{j'} | (X^k)_{k\neq j,j'}\right) \simeq \left(\Sigma^{-1}\right)_{j,j'} \Rightarrow$ find the partial correlations by means of $(\widehat{\Sigma}^n)^{-1}$.

Problem: Σ is a p-dimensional matrix (with p large) and n is small compared to $p \Rightarrow (\widehat{\Sigma}^n)^{-1}$ is a poor estimate of Σ^{-1}!

Various approaches (and R packages) to infer gene expression networks

- **Graphical Gaussian Model**
  - seminal work:
    [Schäfer and Strimmer, 2005a, Schäfer and Strimmer, 2005b] (with bootstrapping or shrinkage and a proposal for a Bayesian test for significance); package GeneNet;

Various approaches (and R packages) to infer gene expression networks

- **Graphical Gaussian Model**
  - seminal work: **[Schäfer and Strimmer, 2005a, Schäfer and Strimmer, 2005b]** (with bootstrapping or shrinkage and a proposal for a Bayesian test for significance); package GeneNet;
  - sparse approaches **[Friedman et al., 2008]**: packages glasso, huge, GGMselect **[Giraud et al., 2009]**, SIMoNe **[Chiquet et al., 2009]**, JGL **[Danaher et al., 2014]** or therese **[Villa-Vialaneix et al., 2014]**... (with unsupervised clustering or able to handle multiple populations data)

Various approaches (and R packages) to infer gene expression networks

- **Graphical Gaussian Model**
  - seminal work: **[Schäfer and Strimmer, 2005a, Schäfer and Strimmer, 2005b]** (with bootstrapping or shrinkage and a proposal for a Bayesian test for significance); package GeneNet;
  - sparse approaches **[Friedman et al., 2008]**: packages glasso, huge, GGMselect **[Giraud et al., 2009]**, SIMoNe **[Chiquet et al., 2009]**, JGL **[Danaher et al., 2014]** or therese **[Villa-Vialaneix et al., 2014]**... (with unsupervised clustering or able to handle multiple populations data)
- **Other methods**: Bayesian network learning **[Pearl, 1998, Pearl and Russel, 2002, Scutari, 2010]** bnlearn, mutual information **[Meyer et al., 2008]** minet...

Outline

1 A brief introduction to networks/graphs

2 Network inference

3 Simple graph mining
   - Visualization
   - Global characteristics
   - Numerical characteristics calculation
   - Clustering

Settings

Notations

In the following, a **graph** $\mathcal{G} = (V, E, W)$ with:

- V: set of vertexes $\{x_1, \ldots, x_p\}$;
- E: set of edges;
- W: weights on edges s.t. $W_{ij} \geq 0$, $W_{ij} = W_{ji}$ and $W_{ii} = 0$.

Settings

Notations

In the following, a **graph** $\mathcal{G} = (V, E, W)$ with:

- V: set of vertexes $\{x_1, \ldots, x_p\}$;
- E: set of edges;
- W: weights on edges s.t. $W_{ij} \geq 0$, $W_{ij} = W_{ji}$ and $W_{ii} = 0$.

The graph is said to be **connected**/*connexe* if any node can be reached from any other node by a path/*un chemin*.

Settings

Notations

In the following, a **graph** $\mathcal{G} = (V, E, W)$ with:

- V: set of vertexes $\{x_1, \ldots, x_p\}$;
- E: set of edges;
- W: weights on edges s.t. $W_{ij} \geq 0$, $W_{ij} = W_{ji}$ and $W_{ii} = 0$.

The graph is said to be **connected**/*connexe* if any node can be reached from any other node by a path/*un chemin*.
The **connected components**/*composantes connexes* of a graph are all its connected subgraphs.

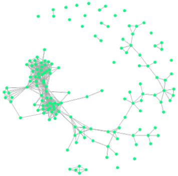
Settings

Notations

In the following, a **graph** $\mathcal{G} = (V, E, W)$ with:

- V: set of vertexes $\{x_1, \ldots, x_p\}$;
- E: set of edges;
- W: weights on edges s.t. $W_{ij} \geq 0$, $W_{ij} = W_{ji}$ and $W_{ii} = 0$.

Example 1: Natty's FB network has 21 connected components with 122 vertexes (professional contacts, family and closest friends) or from 1 to 5 vertexes (isolated nodes)



Settings

Notations

In the following, a **graph** $\mathcal{G} = (V, E, W)$ with:

- V: set of vertexes $\{x_1, \ldots, x_p\}$;
- E: set of edges;
- W: weights on edges s.t. $W_{ij} \geq 0$, $W_{ij} = W_{ji}$ and $W_{ii} = 0$.

Example 2: Medieval network: 10 542 nodes and the largest connected component contains 10 025 nodes ("giant component" / *composante géante*).

Visualization tools help understand the graph macro-structure

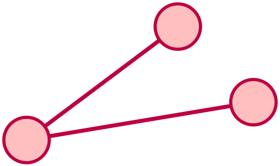
Purpose: How to display the nodes in a **meaningful** and **aesthetic** way?

Visualization tools help understand the graph macro-structure

Purpose: How to display the nodes in a **meaningful** and **aesthetic** way?

Standard approach: **force directed placement** algorithms (FDP)

algorithmes de forces (e.g., [Fruchterman and Reingold, 1991])

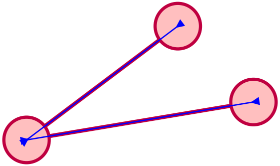


Visualization tools help understand the graph macro-structure

Purpose: How to display the nodes in a **meaningful** and **aesthetic** way?

Standard approach: **force directed placement** algorithms (FDP)

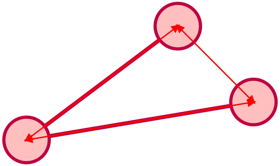
algorithmes de forces (e.g., [Fruchterman and Reingold, 1991])



- attractive forces: similar to springs along the edges

Visualization tools help understand the graph macro-structure

Purpose: How to display the nodes in a **meaningful** and **aesthetic** way?
Standard approach: **force directed placement** algorithms (FDP)
algorithmes de forces (e.g., **[Fruchterman and Reingold, 1991]**)

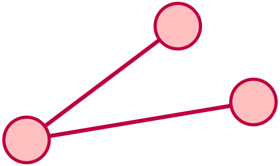


- attractive forces: similar to springs along the edges
- repulsive forces: similar to electric forces between all pairs of vertexes

Visualization tools help understand the graph macro-structure

Purpose: How to display the nodes in a **meaningful** and **aesthetic** way?

Standard approach: **force directed placement** algorithms (FDP) *algorithmes de forces* (e.g., [Fruchterman and Reingold, 1991])



- attractive forces: similar to springs along the edges
- repulsive forces: similar to electric forces between all pairs of vertexes

iterative algorithm until stabilization of the vertex positions.

Visualization software

- R package igraph[1] [Csardi and Nepusz, 2006] (static representation with useful tools for graph mining)

[1] http://igraph.sourceforge.net/

[2] http://gephi.org

Visualization software

- R package `igraph`[1] [Csardi and Nepusz, 2006] (static representation with useful tools for graph mining)
- free software **Gephi**[2] (interactive software, supports zooming and panning)

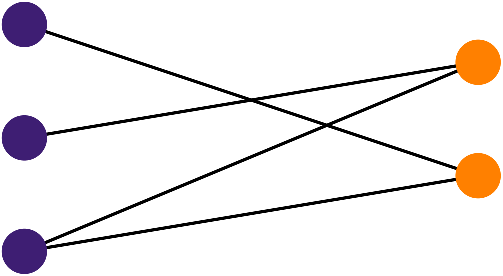
1 http://igraph.sourceforge.net/
2 http://gephi.org

Peculiar graphs

Medieval network (largest connected component):

- 10 025 vertexes: transactions or persons;
- edges model the active involvement of a person in a transaction.

⇒ **Bipartite graph** / *graphe biparti*



Peculiar graphs

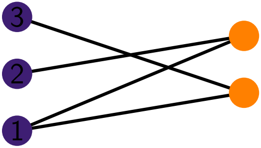
Medieval network (largest connected component):

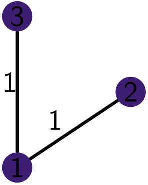
- 10 025 vertexes: transactions or persons;
- edges model the active involvement of a person in a transaction.

$\Rightarrow$ **Bipartite graph** / *graphe biparti*

Projected graphs:

- individuals: nodes are the 3 755 individuals and edges weighted by the number of common transactions;
- transactions (not used): nodes are the 6 270 transactions and edges are weighted by the number of common actively involved persons.





Density / Transitivity *Densité / Transitivité*

Density: Number of edges divided by the number of pairs of vertexes. *Is the network densely connected?*

Density / Transitivity *Densité / Transitivité*

Density: Number of edges divided by the number of pairs of vertexes. *Is the network densely connected?*

Examples

Example 1: Natty's FB network

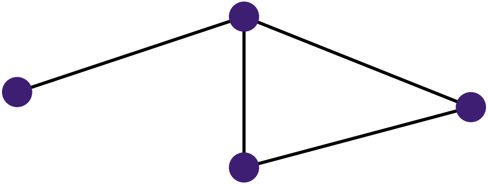
- 152 vertexes, 551 edges $\Rightarrow$ density $= \frac{551}{152 \times 151/2} \simeq 4.8\%$;
- largest connected component: 122 vertexes, 535 edges $\Rightarrow$ density $\simeq 7.2\%$.

Example 2: Medieval network (largest connected component): 10 025 vertexes, 17 612 edges $\Rightarrow$ density $\simeq 0.035\%$.
Projected network (individuals): 3 755 vertexes, 8 315 edges $\Rightarrow$ density $\simeq 0.12\%$.

Density / Transitivity *Densité / Transitivité*

Density: Number of edges divided by the number of pairs of vertexes. *Is the network densely connected?*

Transitivity: Number of triangles divided by the number of triplets connected by at least two edges. *What is the probability that two friends of mine are also friends?*

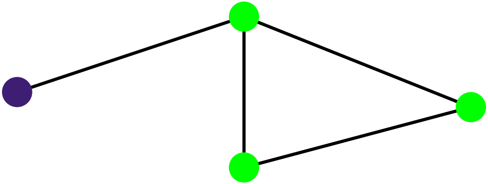


Density is equal to $\frac{4}{4\times 3/2} = 2/3$; Transitivity is equal to $1/3$.

Density / Transitivity *Densité / Transitivité*

Density: Number of edges divided by the number of pairs of vertexes. *Is the network densely connected?*

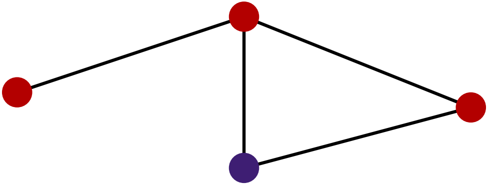
Transitivity: Number of triangles divided by the number of triplets connected by at least two edges. *What is the probability that two friends of mine are also friends?*



Density / Transitivity *Densité / Transitivité*

Density: Number of edges divided by the number of pairs of vertexes. *Is the network densely connected?*

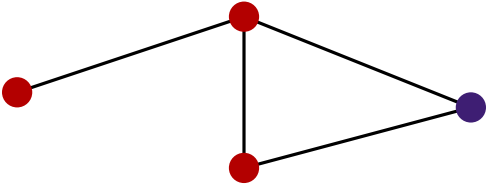
Transitivity: Number of triangles divided by the number of triplets connected by at least two edges. *What is the probability that two friends of mine are also friends?*



Density / Transitivity *Densité / Transitivité*

Density: Number of edges divided by the number of pairs of vertexes. *Is the network densely connected?*

Transitivity: Number of triangles divided by the number of triplets connected by at least two edges. *What is the probability that two friends of mine are also friends?*



Density / Transitivity *Densité / Transitivité*

Density: Number of edges divided by the number of pairs of vertexes. *Is the network densely connected?*

Transitivity: Number of triangles divided by the number of triplets connected by at least two edges. *What is the probability that two friends of mine are also friends?*

Examples

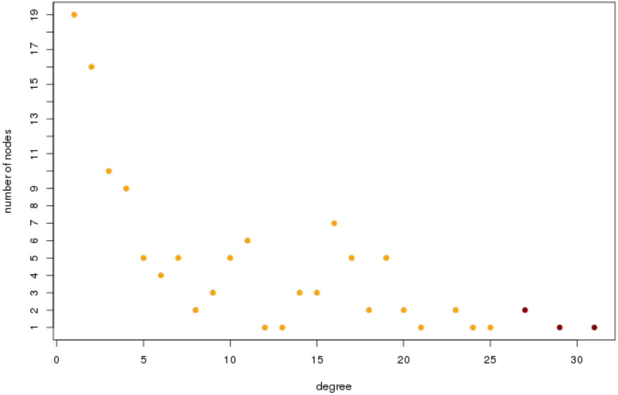
Example 1: Natty's FB network

- density $\simeq$ 4.8%, transitivity $\simeq$ 56.2%;
- largest connected component: density $\simeq$ 7.2%, transitivity $\simeq$ 56.0%.

Example 2: Medieval network (projected network, individuals): density $\simeq$ 0.12%, transitivity $\simeq$ 6.1%.

Extracting important nodes

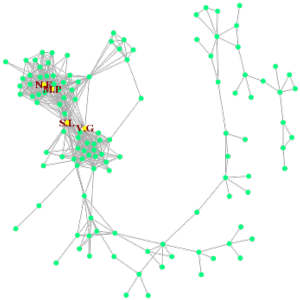
1. **vertex degree** *degré*: number of edges adjacent to a given vertex or $d_i = \sum_j W_{ij}$. Vertexes with a high degree are called **hubs**: measure of the vertex popularity.



Number of nodes (y-axis) with a given degree (x-axis)

Extracting important nodes

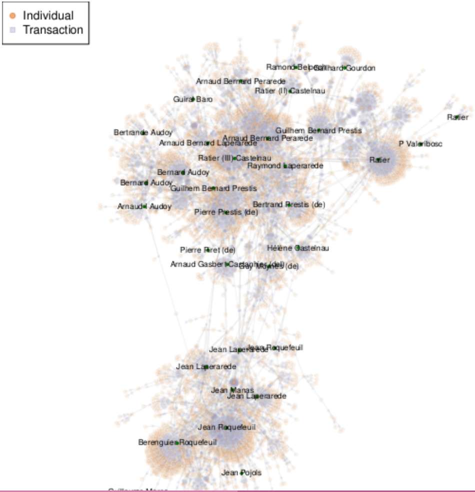
1. **vertex degree** *degré*: number of edges adjacent to a given vertex or $d_i = \sum_j W_{ij}$.
Vertexes with a high degree are called **hubs**: measure of the vertex popularity.



Two hubs are students who have been hold back at school and the other two are from my most recent class.

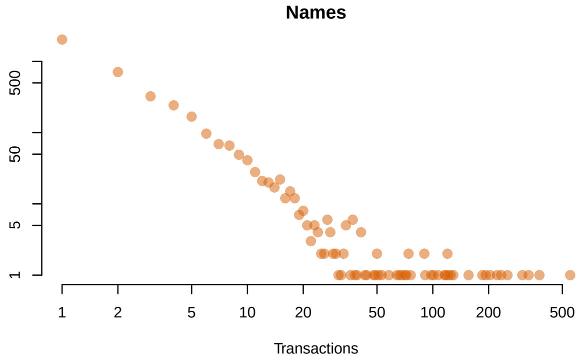
Extracting important nodes

1. **vertex degree** *degré*: number of edges adjacent to a given vertex or $d_i = \sum_j W_{ij}$. Vertexes with a high degree are called **hubs**: measure of the vertex popularity.



Extracting important nodes

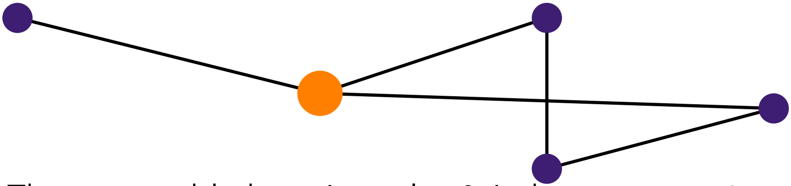
1. **vertex degree** *degré*: number of edges adjacent to a given vertex or $d_i = \sum_j W_{ij}$.
The **degree distribution** is known to fit a **power law** *loi de puissance* in most real networks:



This distribution indicates **preferential attachement** *attachement préférentiel*.

Extracting important nodes

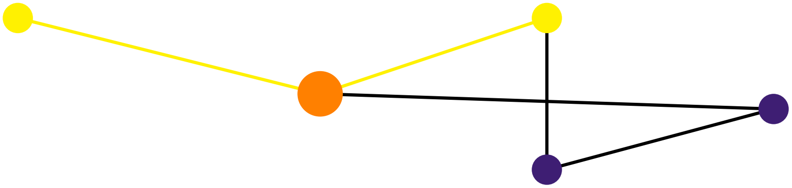
1. **vertex degree** *degré*: number of edges adjacent to a given vertex or $d_i = \sum_j W_{ij}$. The **degree distribution** is known to fit a **power law** *loi de puissance* in most real networks:
2. **vertex betweenness** *centralité*: number of shortest paths between all pairs of vertexes that pass through the vertex. Betweenness is a centrality measure (vertexes that are likely to disconnect the network if removed).



The orange node's degree is equal to 2, its betweenness to 4.

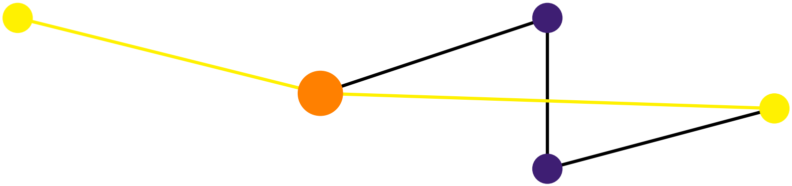
Extracting important nodes

1. **vertex degree** *degré*: number of edges adjacent to a given vertex or $d_i = \sum_j W_{ij}$.
   The **degree distribution** is known to fit a **power law** *loi de puissance* in most real networks:
2. **vertex betweenness** *centralité*: number of shortest paths between all pairs of vertexes that pass through the vertex. Betweenness is a centrality measure (vertexes that are likely to disconnect the network if removed).



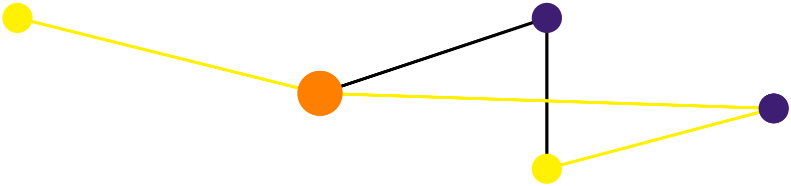
Extracting important nodes

1. **vertex degree** *degré*: number of edges adjacent to a given vertex or $d_i = \sum_j W_{ij}$.
   The **degree distribution** is known to fit a **power law** *loi de puissance* in most real networks:
2. **vertex betweenness** *centralité*: number of shortest paths between all pairs of vertexes that pass through the vertex. Betweenness is a centrality measure (vertexes that are likely to disconnect the network if removed).



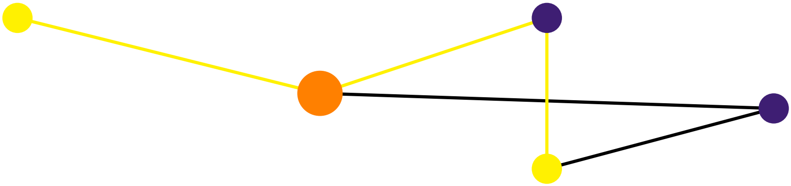
Extracting important nodes

1. **vertex degree** *degré*: number of edges adjacent to a given vertex or $d_i = \sum_j W_{ij}$.
   The **degree distribution** is known to fit a **power law** *loi de puissance* in most real networks:
2. **vertex betweenness** *centralité*: number of shortest paths between all pairs of vertexes that pass through the vertex. Betweenness is a centrality measure (vertexes that are likely to disconnect the network if removed).



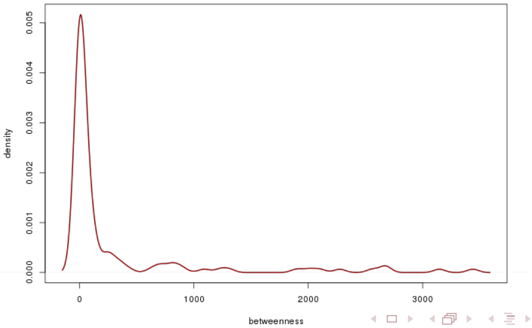
Extracting important nodes

1. **vertex degree** *degré*: number of edges adjacent to a given vertex or $d_i = \sum_j W_{ij}$. The **degree distribution** is known to fit a **power law** *loi de puissance* in most real networks:
2. **vertex betweenness** *centralité*: number of shortest paths between all pairs of vertexes that pass through the vertex. Betweenness is a centrality measure (vertexes that are likely to disconnect the network if removed).



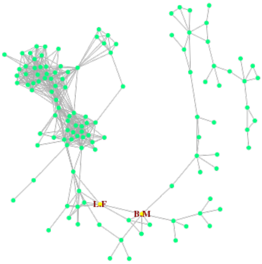
Extracting important nodes

1. **vertex degree** *degré*: number of edges adjacent to a given vertex or $d_i = \sum_j W_{ij}$. The **degree distribution** is known to fit a **power law** *loi de puissance* in most real networks:
2. **vertex betweenness** *centralité*: number of shortest paths between all pairs of vertexes that pass through the vertex. Betweenness is a centrality measure (vertexes that are likely to disconnect the network if removed).



Extracting important nodes

1. **vertex degree** *degré*: number of edges adjacent to a given vertex or $d_i = \sum_j W_{ij}$.
   The **degree distribution** is known to fit a **power law** *loi de puissance* in most real networks:
2. **vertex betweenness** *centralité*: number of shortest paths between all pairs of vertexes that pass through the vertex. Betweenness is a centrality measure (vertexes that are likely to disconnect the network if removed).



Vertexes with a high betweenness ($> 3\ 000$) are 2 political figures.

Extracting important nodes

1. **vertex degree** *degré*: number of edges adjacent to a given vertex or $d_i = \sum_j W_{ij}$.
   The **degree distribution** is known to fit a **power law** *loi de puissance* in most real networks:
2. **vertex betweenness** *centralité*: number of shortest paths between all pairs of vertexes that pass through the vertex. Betweenness is a centrality measure (vertexes that are likely to disconnect the network if removed).
   Example 2: In the medieval network: moral persons such as the "Chapter of Cahors" or the "Church of Flaugnac" have a high betweenness despite a low degree.

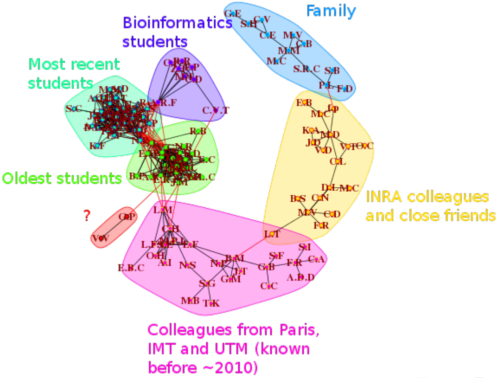
Vertex clustering *classification*

Cluster vertexes into groups that are **densely connected** and share **a few links** (comparatively) **with the other groups**. Clusters are often called **communities** *communautés* (social sciences) or **modules** *modules* (biology).

Vertex clustering *classification*

Cluster vertexes into groups that are **densely connected** and share **a few links** (comparatively) **with the other groups**. Clusters are often called **communities** *communautés* (social sciences) or **modules** *modules* (biology).

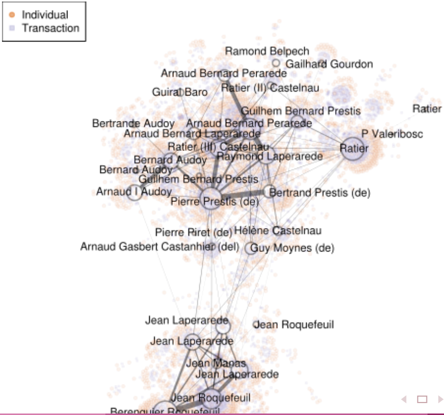
Example 1: Natty's facebook network



Vertex clustering *classification*

Cluster vertexes into groups that are **densely connected** and share **a few links** (comparatively) **with the other groups**. Clusters are often called **communities** *communautés* (social sciences) or **modules** *modules* (biology).

Example 2: medieval network



Vertex clustering *classification*

Cluster vertexes into groups that are **densely connected** and share **a few links** (comparatively) **with the other groups**. Clusters are often called **communities** *communautés* (social sciences) or **modules** *modules* (biology).

Several clustering methods:

- min cut minimization minimizes the number of edges between clusters;
- spectral clustering [von Luxburg, 2007] and kernel clustering uses eigen-decomposition of the **Laplacian**/*Laplacien*

$$L_{ij} = \begin{cases} -w_{ij} & \text{if } i \neq j \\ d_i & \text{otherwise} \end{cases}$$

  (matrix strongly related to the graph structure);
- Generative (Bayesian) models [Zanghi et al., 2008];
- Markov clustering simulate a flow on the graph;
- **modularity maximization**
- ... (clustering jungle... see e.g., [Fortunato and Barthélémy, 2007, Schaeffer, 2007, Brohée and van Helden, 2006])

Find clusters by modularity optimization *modularité*

The **modularity** [Newman and Girvan, 2004] of the partition $(\mathcal{C}_1, \ldots, \mathcal{C}_K)$ is equal to:

$$\mathcal{Q}(\mathcal{C}_1, \ldots, \mathcal{C}_K) = \frac{1}{2m} \sum_{k=1}^{K} \sum_{x_i, x_j \in \mathcal{C}_k} (W_{ij} - P_{ij})$$

with P_{ij}: weight of a "null model" (graph with the same degree distribution but no preferential attachment):

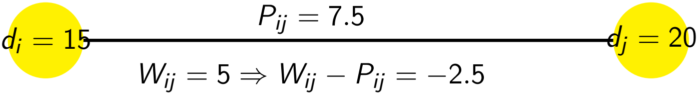
$$P_{ij} = \frac{d_i d_j}{2m}$$

with $d_i = \frac{1}{2} \sum_{j \neq i} W_{ij}$.

Interpretation

A good clustering should **maximize the modularity**:

- $\mathcal{Q} \nearrow$ when (x_i, x_j) are in the **same cluster** and $W_{ij} \gg P_{ij}$
- $\mathcal{Q} \searrow$ when (x_i, x_j) are in **two different clusters** and $W_{ij} \gg P_{ij}$ $(m = 20)$



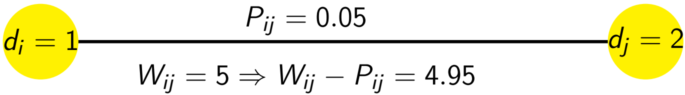
i and j in the same cluster decreases the modularity

Interpretation

A good clustering should **maximize the modularity**:

- $\mathcal{Q} \nearrow$ when (x_i, x_j) are in the **same cluster** and $W_{ij} \gg P_{ij}$
- $\mathcal{Q} \searrow$ when (x_i, x_j) are in **two different clusters** and $W_{ij} \gg P_{ij}$

  $(m = 20)$



i and j in the same cluster increases the modularity

Interpretation

A good clustering should **maximize the modularity**:

- $\mathcal{Q} \nearrow$ when (x_i, x_j) are in the **same cluster** and $W_{ij} \gg P_{ij}$
- $\mathcal{Q} \searrow$ when (x_i, x_j) are in **two different clusters** and $W_{ij} \gg P_{ij}$
- Modularity
    - **helps separate hubs** ($\neq$ spectral clustering or min cut criterion);
    - is not an increasing function of the number of clusters: useful to **choose the relevant number of clusters** (with a grid search: several values are tested, the clustering with the highest modularity is kept) but modularity has a **small resolution default** (see **[Fortunato and Barthélémy, 2007]**)

Interpretation

A good clustering should **maximize the modularity**:

- $\mathcal{Q} \nearrow$ when (x_i, x_j) are in the **same cluster** and $W_{ij} \gg P_{ij}$
- $\mathcal{Q} \searrow$ when (x_i, x_j) are in **two different clusters** and $W_{ij} \gg P_{ij}$
- Modularity
  - **helps separate hubs** ($\neq$ spectral clustering or min cut criterion);
  - is not an increasing function of the number of clusters: useful to **choose the relevant number of clusters** (with a grid search: several values are tested, the clustering with the highest modularity is kept) but modularity has a **small resolution default** (see [Fortunato and Barthélémy, 2007])

Main issue: Optimization = **NP-complete problem** (exhaustive search is not not usable)

Different solutions are provided in [Newman and Girvan, 2004, Blondel et al., 2008, Noack and Rotta, 2009, Rossi and Villa-Vialaneix, 2011] (among others) and some of them are implemented in the R package igraph.

Open issues with clustering (not addressed)

- overlapping communities *communautés recouvrantes*;
- hierarchical clustering **[Rossi and Villa-Vialaneix, 2011]** provides an approach;
- "organized" clustering (projection on a small dimensional grid) and clustering for visualization **[Boulet et al., 2008, Rossi and Villa-Vialaneix, 2010, Rossi and Villa-Vialaneix, 2011]**;
- ...

References

Blondel, V., Guillaume, J., Lambiotte, R., and Lefebvre, E. (2008).
Fast unfolding of communites in large networks.
Journal of Statistical Mechanics: Theory and Experiment, P10008:1742–5468.

Boulet, R., Jouve, B., Rossi, F., and Villa, N. (2008).
Batch kernel SOM and related Laplacian methods for social network analysis.
Neurocomputing, 71(7-9):1257–1273.

Brohée, S. and van Helden, J. (2006).
Evaluation of clustering algorithms for protein-protein interaction networks.
BMC Bioinformatics, 7(488).

Butte, A. and Kohane, I. (1999).
Unsupervised knowledge discovery in medical databases using relevance networks.
In *Proceedings of the AMIA Symposium*, pages 711–715.

Butte, A. and Kohane, I. (2000).
Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements.
In *Proceedings of the Pacific Symposium on Biocomputing*, pages 418–429.

Chiquet, J., Smith, A., Grasseau, G., Matias, C., and Ambroise, C. (2009).
SIMoNe: Statistical Inference for MOdular NEtworks.
Bioinformatics, 25(3):417–418.

Csardi, G. and Nepusz, T. (2006).
The igraph software package for complex network research.
InterJournal, Complex Systems.

Danaher, P., Wang, P., and Witten, D. (2014).
The joint graphical lasso for inverse covariance estimation accross multiple classes.
Journal of the Royal Statistical Society Series B, 76(2):373–397.

Fortunato, S. and Barthélémy, M. (2007).
Resolution limit in community detection.
In *Proceedings of the National Academy of Sciences*, volume 104, pages 36–41.
doi:10.1073/pnas.0605965104; URL: http://www.pnas.org/content/104/1/36.abstract.

Friedman, J., Hastie, T., and Tibshirani, R. (2008).
Sparse inverse covariance estimation with the graphical lasso.
Biostatistics, 9(3):432–441.

Fruchterman, T. and Reingold, B. (1991).
Graph drawing by force-directed placement.
Software, Practice and Experience, 21:1129–1164.

Giraud, C., Huet, S., and Verzelen, N. (2009).
Graph selection with ggmselect.
Technical report, preprint arXiv.
http://fr.arxiv.org/abs/0907.0619.

Laurent, T. and Villa-Vialaneix, N. (2011).
Using spatial indexes for labeled network analysis.
Information, Interaction, Intelligence (I3), 11(1).

Meyer, P., Lafitte, F., and Bontempi, G. (2008).
minet: A R/Bioconductor package for inferring large transcriptional networks using mutual information.
BMC Bioinformatics, 9(461).

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

Noack, A. and Rotta, R. (2009).
Multi-level algorithms for modularity clustering.

In *SEA 2009: Proceedings of the 8th International Symposium on Experimental Algorithms*, pages 257–268, Berlin, Heidelberg. Springer-Verlag.

Pearl, J. (1998).
Probabilistic reasoning in intelligent systems: networks of plausible inference.
Morgan Kaufmann, San Francisco, California, USA.

Pearl, J. and Russel, S. (2002).
Bayesian Networks.
Bradford Books (MIT Press), Cambridge, Massachussets, USA.

Rossi, F. and Villa-Vialaneix, N. (2010).
Optimizing an organized modularity measure for topographic graph clustering: a deterministic annealing approach.
Neurocomputing, 73(7-9):1142–1163.

Rossi, F. and Villa-Vialaneix, N. (2011).
Représentation d'un grand réseau à partir d'une classification hiérarchique de ses sommets.
Journal de la Société Française de Statistique, 152(3):34–65.

Schaeffer, S. (2007).
Graph clustering.
Computer Science Review, 1(1):27–64.

Schäfer, J. and Strimmer, K. (2005a).
An empirical bayes approach to inferring large-scale gene association networks.
Bioinformatics, 21(6):754–764.

Schäfer, J. and Strimmer, K. (2005b).
A shrinkage approach to large-scale covariance matrix estimation and implication for functional genomics.
Statistical Applications in Genetics and Molecular Biology, 4:1–32.

Scutari, M. (2010).
Learning Bayesian networks with the bnlearn R package.

Journal of Statistical Software, 35(3):1–22.

Villa-Vialaneix, N., Vignes, M., Viguerie, N., and San Cristobal, M. (2014).
Inferring networks from multiple samples with concensus LASSO.
Quality Technology and Quantitative Management, 11(1):39–60.

von Luxburg, U. (2007).
A tutorial on spectral clustering.
Statistics and Computing, 17(4):395–416.

Zanghi, H., Ambroise, C., and Miele, V. (2008).
Fast online graph clustering via erdös-rényi mixture.
Pattern Recognition, 41:3592–3599.