

TP sur les Communautés

Le but de ce travail est de mieux comprendre l'analyse d'un réseau social par le biais des communautés et des relations structurellement équivalentes.

A cette fin, nous nous appuyons sur le jeu de données du réseau de la classe et celui des femmes-événements.

Étape 1 : Identification des groupes basée sur l'interaction entre les nœuds

Q1. Identifier les communautés de chacun des deux réseaux en utilisant au moins deux méthodes ?

Réponse

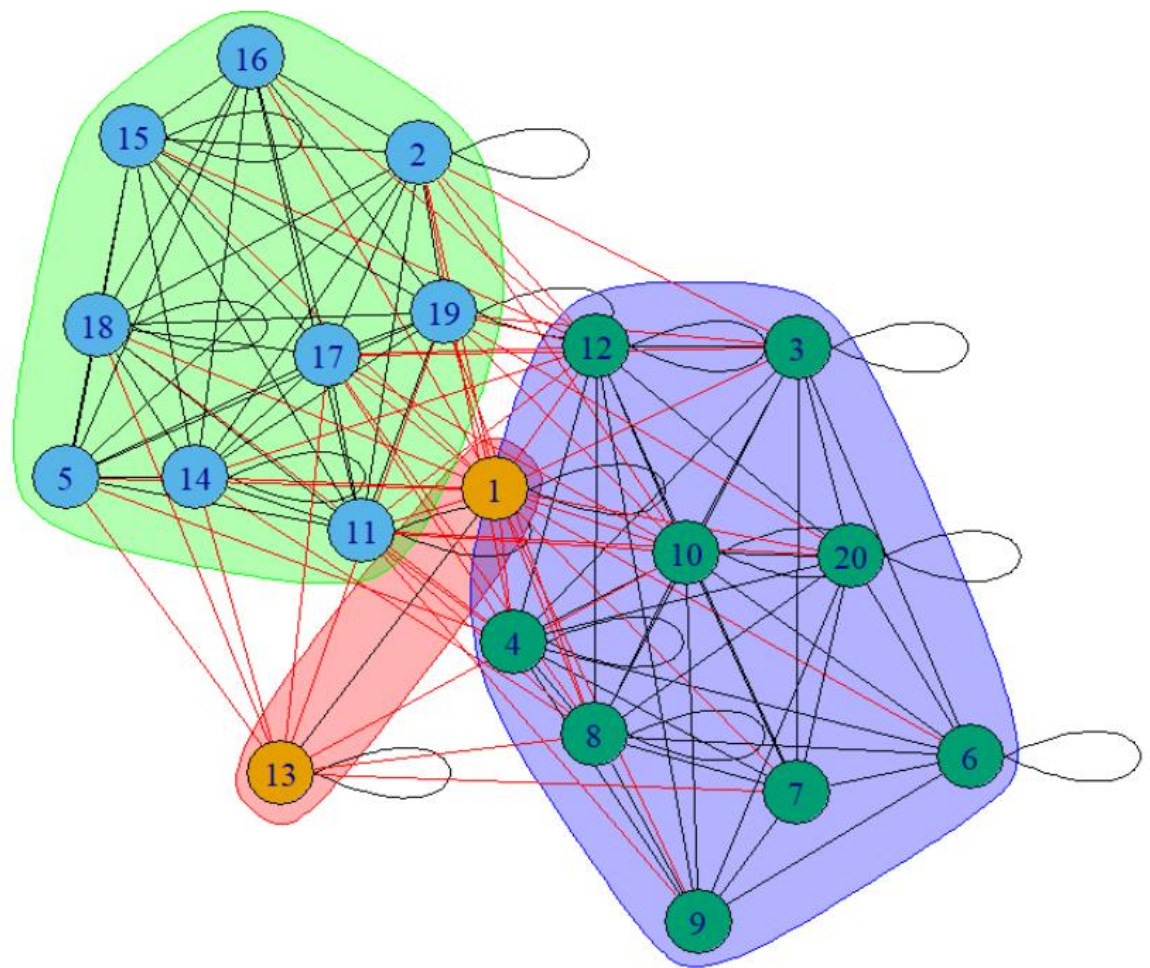
Nous allons utiliser ces deux méthodes de détection de communauté : Maximisation de la modularité et Edge Betweenness.

Pour le réseau de la classe

Maximisation de la modularité

Code

```
library(igraph)
library(readxl)
classe <- read_xlsx("C:\\RS\\TP-COMMUNAUTE\\Fichier_ARS_2023.xlsx",
sheet=1, col_names = TRUE)
View(classe)
anorm <- subset(classe,select = -c(1))
netBI <- as.matrix(anorm)
G<- graph_from_adjacency_matrix(netBI)
cfg <- cluster_fast_greedy(as.undirected(G))
plot(cfg, as.undirected(G))
modularity(cfg)
```



Edge Betweenness

Code

```
library(igraph)
```

```
library(readxl)
```

```
classe <- read_xlsx("C:\\RS\\TP-COMMUNAUTE\\Fichier_ARS_2023.xlsx",  
sheet=1)
```

```
View(classe)
```

```
anorm <- subset(classe, select = -c(1))
```

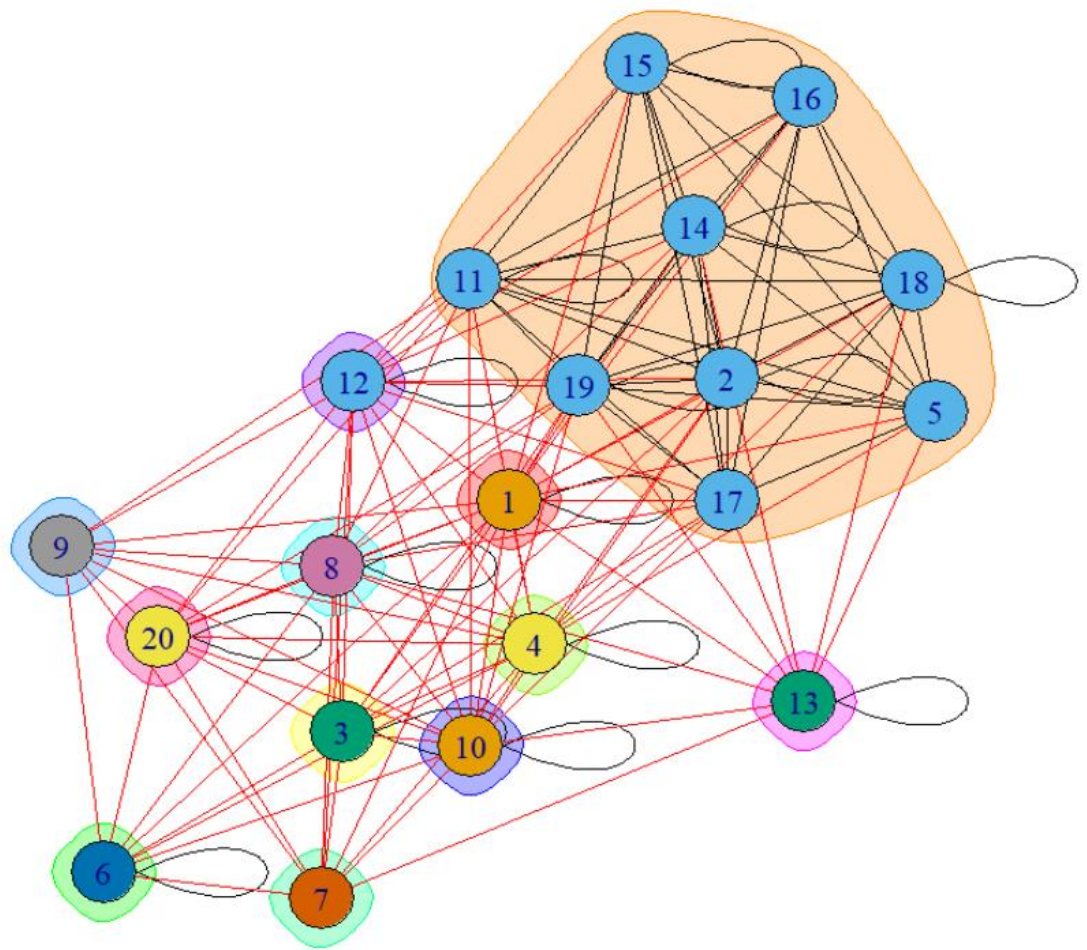
```
netBI <- as.matrix(anorm)
```

```
G<- graph_from_adjacency_matrix(netBI)
```

```
ceb <- cluster_edge_betweenness(as.undirected(G))
```

```
dendPlot(ceb)
```

```
plot(ceb, as.undirected(G))
```



Pour le réseau `women_liste_binaire`

J'ai converti le fichier CSV en fichier Excel.

Edge Betweenness

Code

```
library(readxl)
```

```
library("igraph")
```

```
fem_Ev<- read_xlsx("C:\\RS\\TP-COMMUNAUTE\\women_liste_binaire.xlsx",
```

```
col_names = TRUE)
```

```
MatriceFem_Ev <- as.matrix(fem_Ev)
```

```
Graphe<- graph_from_incidence_matrix(MatriceFem_Ev)
```

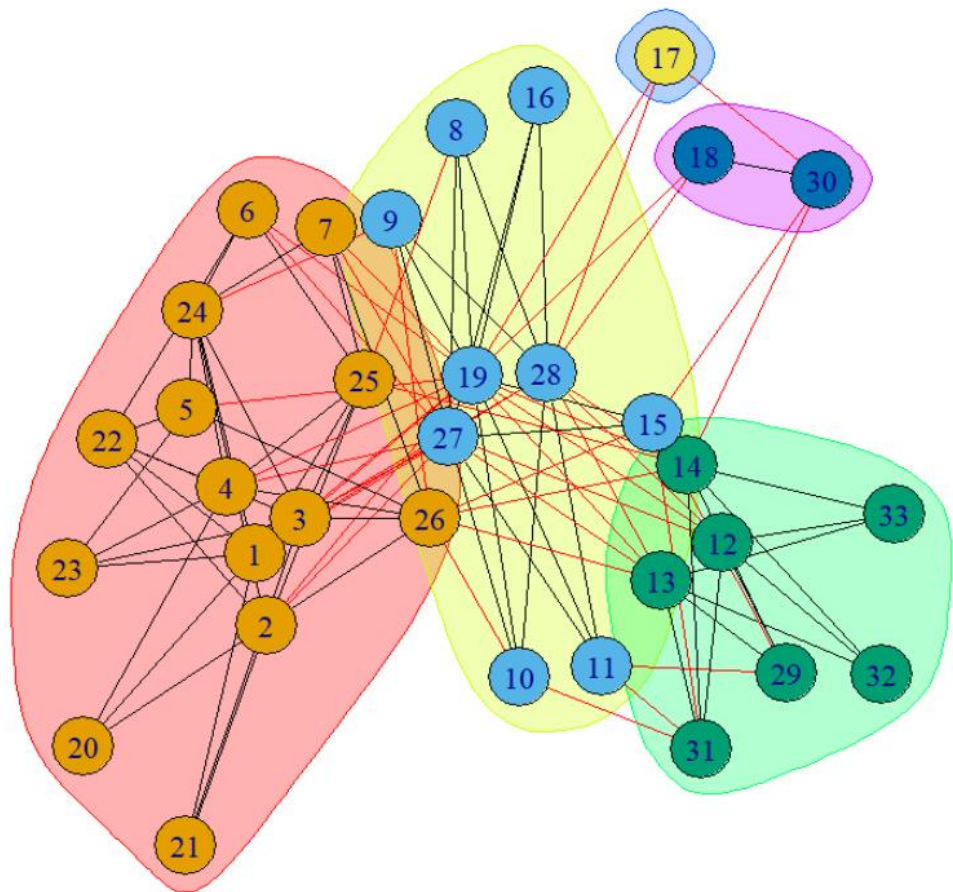
```
bipartite.mapping(Graphe)
```

```
ceb <- cluster_edge_betweenness(as.undirected(Graphe))
```

```
dendPlot(ceb)
```

```
plot(ceb, as.undirected(Graphe))
```

Résultat



Maximisation de la modularité

Code

```
library(readxl)
```

```
library("igraph")
```

```
fem_Ev<- read_xlsx("C:\\RS\\TP-COMMUNAUTE\\women_liste_binaire.xlsx",
```

```
col_names = TRUE)
```

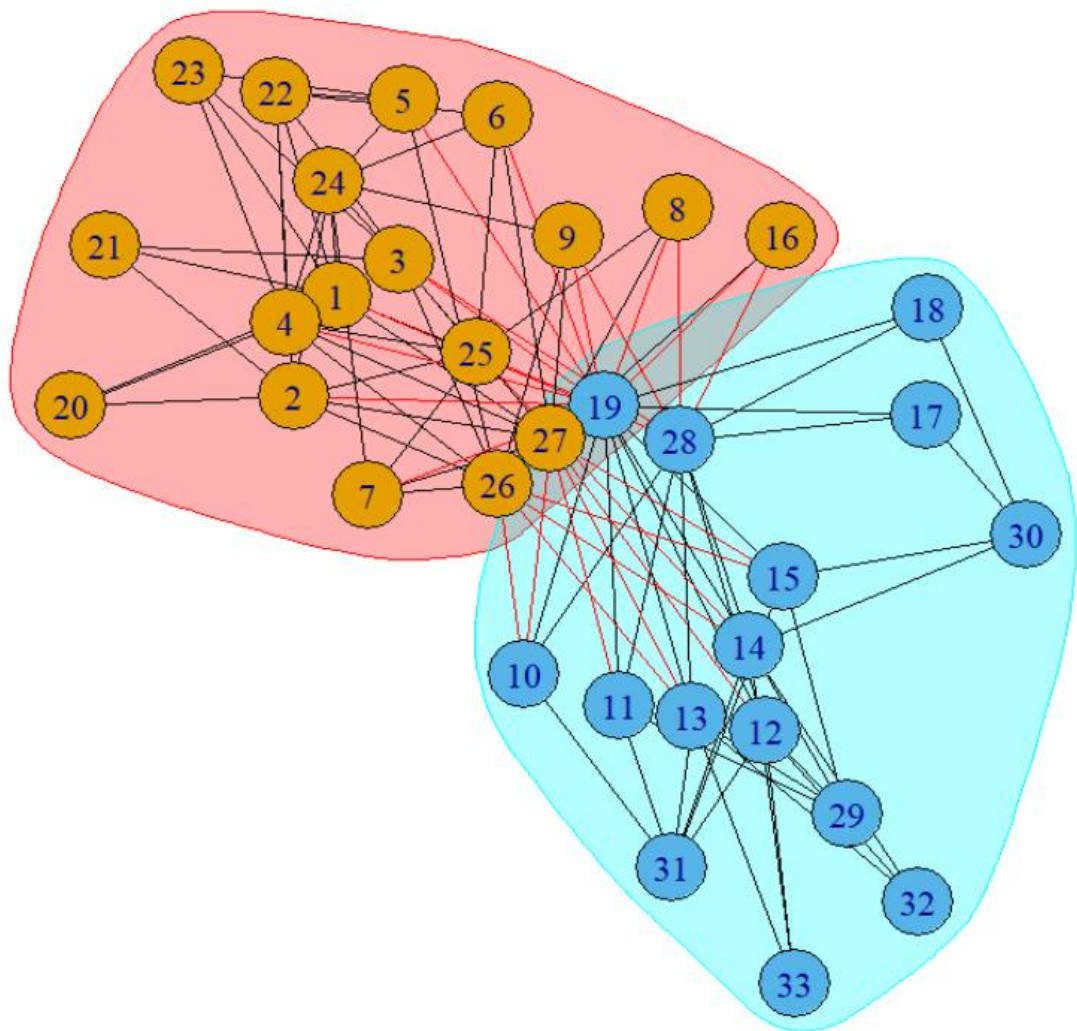


```

MatriceFem_Ev <- as.matrix(fem_Ev)
Graphe<- graph_from_incidence_matrix(MatriceFem_Ev)
bipartite.mapping(Graphe)
#plot(Graphe)
cfg <- cluster_fast_greedy(as.undirected(Graphe))
plot(cfg, as.undirected(Graphe))
modularity(cfg)

```

Resultat



Q2. Pour chaque réseau, est-ce que les communautés obtenues sont les mêmes peu importe la méthode ? Expliquez pourquoi ?

Réponse

Non car les critères de détection de communauté ne sont pas les même.

Étape 2 : Identification des groupes basée sur l'équivalence structurelle des nœuds

Q3. Cette fois, on essaie de faire nos communautés sur la base de la similarité des nœuds du réseau. Ainsi, on définit une communauté comme étant l'ensemble des nœuds structurellement équivalents à un certain degré près. Autrement, tous les nœuds dont la similarité dépasse un seuil, seront considérés comme faisant partie d'une même communauté.

Définissez dans R/Igraph un algorithme pour détecter les communautés avec cette approche.

Indication : il y a une fonction *similarity* avec plusieurs variantes qui permet de calculer la similarité des nœuds d'un graphe. Le résultat est une matrice que vous pouvez parcourir avec des boucles pour trouver les nœuds qui doivent former un groupe.

Réponse Q 3

```
library(igraph)
library(igraph)
library(readxl)
classe <- read_xlsx("C:\\RS\\TP-COMMUNAUTE\\Fichier_ARS_2023.xlsx", sheet=1)
View(classe)
anorm <- subset(classe, select = -c(1))
netBI <- as.matrix(anorm)
g<- graph_from_adjacency_matrix(netBI)
# Calculate Jaccard similarity matrix
sim_matrix <- similarity(g, method = "jaccard")

# Set the similarity threshold
threshold <- 0.8

# Create a vector to store community assignments
communities <- rep(1, vcount(as.undirected(g)))
```

```

# Assign nodes to communities based on similarity threshold
for (i in 1:vcount(g)) {
  for (j in 1:vcount(g)) {
    if (sim_matrix[i, j] >= threshold && is.na(communities[i])) {
      communities[i] <- communities[j]
    } else if (sim_matrix[i, j] >= threshold && is.na(communities[j])) {
      communities[j] <- communities[i]
    } else if (sim_matrix[i, j] > threshold && communities[i] != communities[j]) {
      # Merge communities if they have overlapping nodes
      communities[communities == communities[j]] <- communities[i]
    }
  }
}

# Visualize the communities
plot(g, vertex.color = communities, vertex.label = NA, main = "Communities based on
Jaccard Similarity (Threshold = 0.8)")

```

Q4. Est-ce que les groupes obtenus avec le Q3 sont identiques à ceux obtenus avec le Q4 ?
Je pense cette y’a une erreur.

Étape 3 : Approfondissement et synthèse

Q5. On reprend le réseau de la classe auquel on ajoute un à deux attributs avec des valeurs différentes. Par exemple, les attributs âge et sexe. Proposez un algorithme de détection de communautés qui intègre au-delà de la connectivité entre les nœuds les deux attributs précédents.