Projet Analyse des réseaux

ALEXANDRA MILLOT et ULRICH KARL ODJO

Mai 2024

Table des matières

1 Introduction

Dans le cadre de l'Unité d'Enseignement "Analyse des réseaux", nous allons étudier les connexions existant au sein d'un groupe de chercheurs. Le but d'une telle analyse étant d'étudier l'effet d'une intervention politique sur cette communauté scientifique. Celle-ci a pour intention le renforcement et l'augmentation des collaborations au sein de ce réseau.

Ce réseau est réel, il s'agit d'un ensemble de 252 chercheurs travaillant au même endroit sur une thématique commune. Deux observations ont été réalisé, la première étant avant la mise en place de la politique et la seconde après son intervention.

Les données traitées dans ce projet sont constituées de deux fichiers .csv : "nodes" et "edges", pouvant être connectées grâce aux identifiants individuels (id). $nodes.csv$ fournit des informations invariables par rapport à l'intervention politique, tandis que $edges.csv$ fournit des informations variables sur les dyades.

Ce projet a deux objectifs principaux :

Premièrement, décrire et analyser la structure du réseau et son évolution entre les deux périodes (avant et après traitement) à l'aide de statistiques et de graphiques de réseau pertinents.

Deuxièmement, discuter des connexions de réseau en ce qui concerne les attributs fondamentaux et de genre. Cela implique d'explorer des hypothèses telles que :

– La densité et la centralité des réseaux de femmes par rapport à ceux des hommes.

– L'expansion des réseaux relatifs au genre et au traitement lors de la mise en place de l'intervension politique.

– L'identification de sous-communautés ou d'homophilie au sein des réseaux pré et post politique.

2 Modélisation

Avant d'aborder nos questions de recherche, il convient de contextualiser la manière dont nous avons structuré les informations lors de la construction du graphe.

Deux approches sont possibles pour construire le graph de notre réseau :

Réseaux séparés : cette approche consiste à créer deux réseaux distincts. L'un représentant les chercheurs avant la mise en place de la politique, et l'autre après la mise en place de la politique. Cette méthode permet de comparer directement la structure des deux réseaux sur les différentes périodes considérées.

Réseau unique : cette autre approche aurait été de construire un seul réseau où les arêtes sont pondérées en fonction des valeurs **pre_link** et **post_link** dans les données. Cette méthode consisterait donc à filtrer les arêtes en fonction de la période lors de l'analyse des structures avant et après la mise en place de la politique.

Pour répondre au mieux à nos objectifs d'analyse, nous avons opté pour la création d'un **réseau séparé**.

Avant de construire le graphe, nous avons cherché à déterminer si notre jeu de données comprenait des valeurs manquantes. Ce qui s'est avéré postif, deux des chercheurs appartenant au réseau n'avaient pas de genre qui leur étaient attribué. Nous avons donc effectué une imputation en utilisant le genre le plus représenté dans notre jeu de données, qui est le genre *masculin*.

Après la construction des deux graphes, nous avons décidé de supprimer les liens de ces graphes dont la pondération est à zéro. Ces liens représentent l'absence de collaboration entre deux chercheurs. Cette décision simplifie grandement le calcul des mesures de graphe, comme les mesures de centralité, en nous évitant de tenir compte de la pondération des liens à chaque calcul.

3 Objectif N°1

3.1 Analyse de la structure du réseau

Dans cette première sous-section, on se propose de décrire et d'analyser la structure de notre réseau puis son évolution dans le temps. Pour cela, nous ferons usage des mesures structurelles telles que :

– **L'ordre du réseau**, qui correspond au nombre de nœuds.

– **La taille du réseau**, qui représente le nombre de liens.

– **Le degré moyen du réseau**, qui indique le nombre moyen de collaborations par chercheur.

– **La densité du réseau**, qui est le nombre de liens dans le réseau rapporté au nombre total de liens possibles. En cas d'un graphe non dirigé, cette densité est calculée selon la formule suivante :

$$\frac{2E}{V(V-1)}$$

ou E correspond au nombre de liens et V au nombre de nœuds. Ces indicateurs sont calculés pour les deux périodes d'étude, c'est-à-dire, avant et après la mise en place de la politique. Les résultats de ces mesures sont consignés dans les tableaux suivants :

	Valeurs avant la politique	Valeurs après la politique
Nombre de nœuds (ordre)	252	252
Nombre de liens (taille)	693	739
Degré moyen du réseau	5.50	5.865
Densité du réseau	0.0219	0.0233

TABLE 1 – Statistiques calculées sur le réseau avant et après la mise en place de la politique

On observe grâce à la Table 1, que de façon générale, après la mise en place de la politique, les collaborations au sein du réseau se sont renforcées.

Premièrement, on note une augmentation du nombre de liens et donc de collaborations, passant de 693 à 739. De plus, le nombre moyen de collaborations par chercheur a également augmenté, passant d'environ 5 à 6 collaborateurs en moyenne

entre les deux périodes. Enfin, à travers le calcul de la densité du réseau avant la mise en place de la politique, on observe que 2.19% des liens sont présent dans le réseau sur la totalité des liens possibles pouvant existé dans ce réseau, ce qui montre un réseau relativement peu dense avec un nombre de collaborations limité. Dans la seconde période, cette densité augmente à 2.33%, indiquant une augmentation du nombre de collaborations entre les deux périodes.

3.2 Analyse de la distribution des collaborations dans le réseau

Nous continuons notre analyse, par une évaluation de la **distribution des degrés**. Nous permettant de comprendre comment le nombre de collaborations est réparti dans le réseau avant et après la mise en place de la politique.

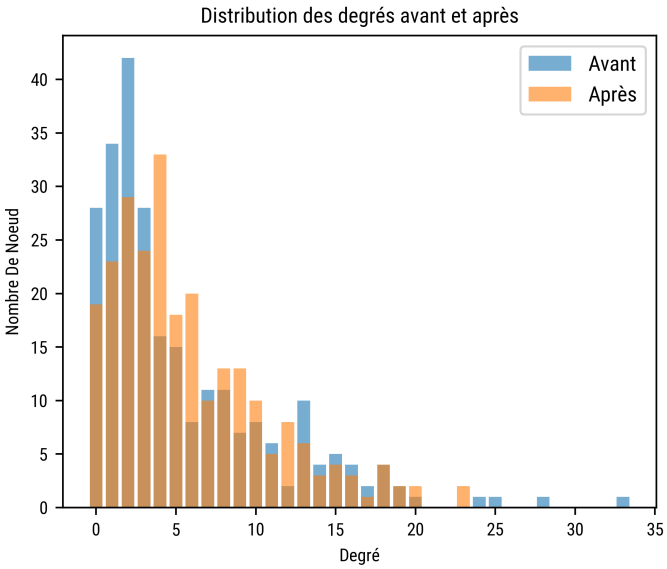

FIGURE 1 – Distribution des degrés avant (bleu) et après (orange) la mise en place de la politique

À partir de la Figure 1, nous observons que la distribution des degrés suit une loi de puissance, démontrant qu'il s'agit là, d'un **réseau centralisé**. Cela suggère que certains chercheurs importants contrôlent une grande partie des connexions ou sont à l'origine d'un nombre important de collaborations entre les chercheurs. De plus, cette caractéristique met en évidence qu'une proportion importante de chercheurs (nœuds) ont un faible nombre de collaborations (liens). Et inversement, un petit nombre de chercheurs ont un grand nombre de collaborations.

Nous constatons également que tant avant qu'après la mise en place de la poli-

tique, la plupart des chercheurs ont un réseau de collaborations relativement restreint (≤ 15 collaborations), avec un faible nombre de chercheurs ayant au-delà 15 collaborations.

Une autre remarque peut être faite sur cette différence de distribution des degrés. Lors de la mise en place de la politique, de nombreux chercheurs qui avaient un nombre élevé de collaborations se sont vu perdre environ une dizaine de collaborateurs, ce qui a conduit à une augmentation du nombre de chercheurs avec un nombre restreint de collaborations.

3.3 Analyse de l'importance de certains chercheurs

Nous passons maintenant à l'analyse de l'importance de certains individus au sein du réseau. On va donc chercher à déterminer quels sont les chercheurs considérés importants ainsi voir si ceux sont les mêmes avant et après la mise en place de l'intervention politique.

Parmi l'ensemble des mesures existantes, nous choisissons d'en utiliser deux qui nous semblent pertinentes pour notre analyse :

La **centralité de degré** (**degree centrality**) qui identifie un chercheur comme important s'il possède un grand nombre de collaborations (liens).

La **centralité d'intermédiarité** (**betweenness centrality**) qui identifie un chercheur comme important s'il permet la mise en relation avec d'autres collaborateurs. En d'autres termes, si pour avoir une collaboration entre deux chercheurs donnés dans notre réseau nous passons majoritairement par ce chercheur.

On pourra également comparer les résultats obtenus pour ces deux méthodes.

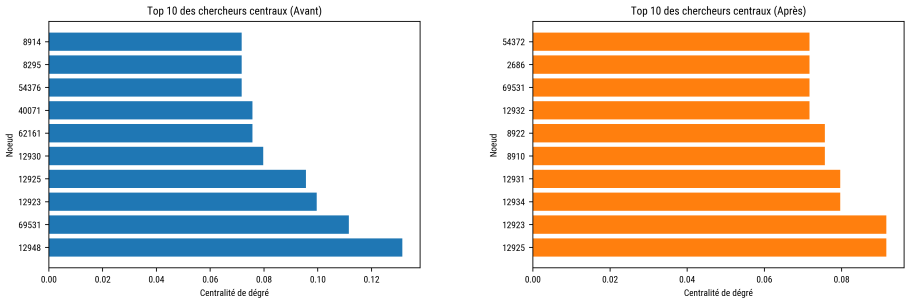

FIGURE 2 – Top 10 des individus les plus centraux (degree centrality) avant (bleu) et après (orange) la mise en place de la politique

Au travers de la Figure 2, montrant le résultat que nous avons obtenu pour la mesure de centralité de degré, on remarque que les 10 chercheurs ayant le plus grand nombre de collaborations diffèrent assez nettement entre les deux périodes, à l'exception de quelques chercheurs apparaissant dans les deux périodes (comme le chercheur avec l'identifiant 12923).

Cela suggère que certains chercheurs centraux (d'après cette mesure) avant la mise en place de la politique ont perdu en influence, au profit d'autres chercheurs qui ont gagné en influence.

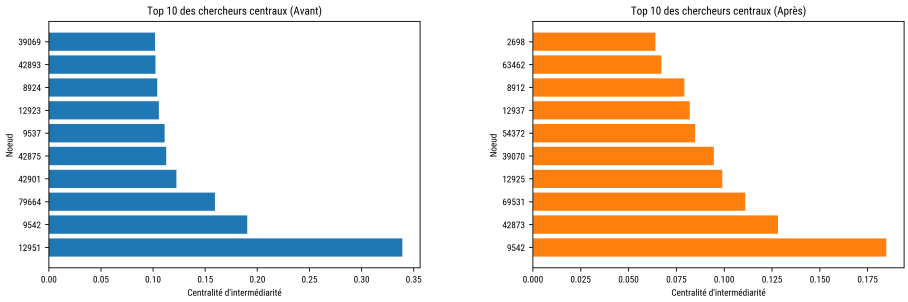

FIGURE 3 – Top 10 des individus les plus centraux (betweeness centrality) avant (bleu) et après (orange) la mise en place de la politique

Les résultats obtenues à partir de la mesure d'intermédiarité présentés sur la Figure 3, nous permettent de constater que les 10 chercheurs qui favorisent le plus la collaboration et le partage de connaissances entre les autres chercheurs sont totalement différents entre les deux périodes, à l'exception du chercheur 9542 qui a obtenu plus de collaborations. Cela indique un réarrangement des collaborations avant et après la mise en place de la politique.

En définitive, les résultats obtenus avec les deux mesures employées sont différents. Ainsi, nous pouvons dire que l'influence des individus dans le réseau est **pluridimensionnelle**. Certains individus sont influents en raison de leur grand nombre de collaborations directes (degré élevé), tandis que d'autres le sont en raison de leur position stratégique dans le réseau (betweenness élevée).

4 Objectif N°2

Dans cette deuxième partie, notre objectif est de mieux comprendre les connexions au sein de notre réseau avant et après la mise en place de la politique, en tenant compte du genre et de l'importance dans le partage de connaissances des chercheurs (atttribut **core**). Afin de répondre à cet objectif, nous chercherons à répondre aux questions suivantes :

1. Est-ce que les femmes ont un réseau plus dense que celui des hommes ?
2. Est-ce que les femmes sont plus centrales dans le réseau que les hommes ?
3. Élargissent-elles plus ou moins leur réseau par rapport aux hommes bénéficiant du traitement (au fil du temps) ?
4. Peut-on identifier des sous-communautés avant et après la mise en place de la politique ?

En répondant ainsi à ces questions, nous obtiendrons une meilleure vue d'ensemble de l'évolution de la collaboration au sein de notre réseau en fonction du genre des chercheurs.

4.1 Évolution de la densité du réseau en fonction du genre

Nous souhaitons déterminer si, après la mise en place de la politique, il y a eu un renforcement ou une émergence des collaborations dans le réseau des femmes par rapport à celui des hommes.

Pour répondre à cette question, nous avons d'abord réalisé une visualisation globale du réseau par genre avant et après la mise en place de la politique. Dans un second temps, de manière analytique, nous avons calculé la densité avant et après la politique pour chacun des sous-graphes constitués d'hommes et de femmes. Les résultats de ces analyses sont les suivants :

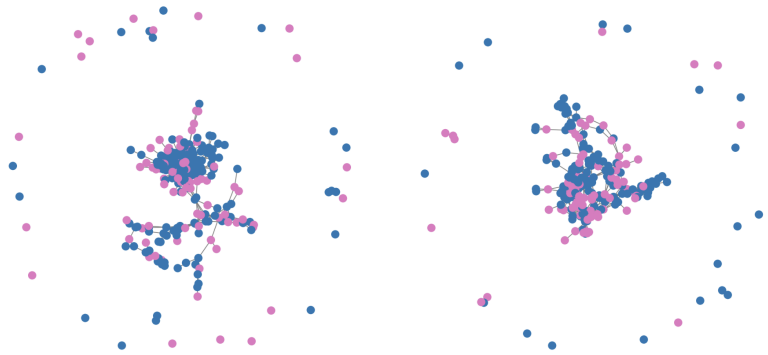

FIGURE 4 – Réseau des chercheurs par genre (Masculin = Bleu, Féminin = Rose) avant (à gauche) et après (à droite) la politique en place de la politique

La Figure 4 met en évidence que, après la mise en place de la politique, un nombre plus important de chercheuses ont été intégrées dans le réseau de collaborations. Cependant, cela s'est fait au détriment de certains chercheurs masculins qui ont quitté le réseau de collaboration. Cela suggère qu'entre les deux périodes, de nouvelles collaborations ont émergé au sein du réseau des femmes.

Pour confirmer cela de manière analytique, nous calculerons le degré moyen (correspondant au nombre de collaborations en moyenne) pour chaque genre sur l'ensemble du graphe et pour chacune des deux périodes considérées. Ainsi nous pourrons observer si entre les deux périodes les femmes possédaient plus de collaboration en moyenne.

	Valeurs avant la politique	Valeurs après la politique
Hommes	6.454	6.245
Femmes	3.752	5.168

TABLE 2 – Degré moyen pour les hommes et les femmes avant et après la mise en place de la politique

Analytiquement, on observe, Le Tableau 2, qu'avant tout comme après la mise de la politique, le réseau des hommes est toujours plus dense que celui des femmes. Néanmoins on peut constater qu'entre les deux périodes le nombre de collaboration moyenne pour le genre féminin a largement augmenté tandis que celui des hommes à subit une légère baisse. Cela confirme bien que la mise en place de la politique a favorisé l'émergence de la collaboration dans le réseau des chercheurs féminins contrairement à celui du réseau masculin.

4.2 Évolution de la centralité des chercheurs en fonction du genre

Afin d'observer l'évolution de la centralité des hommes et des femmes dans le réseau avant et après la politique, on effectue une comparaison de la moyenne de deux mesures de centralités choisies pour les deux genres.

	Valeurs avant la politique	Valeurs après la politique
Hommes	0.025	0.024
Femmes	0.015	0.020

TABLE 3 – Centralité de degré moyenne avant et après la mise en place de la politique en fonction du genre

Les résultats obtenus Figure 3 et 4 montrent que la mise en place de la politique ne modifie pas l'importance des chercheurs masculins dans le réseau, bien que

	Valeurs avant la politique	Valeurs après la politique
Hommes	0.017	0.012
Femmes	0.010	0.011

TABLE 4 – Centralité d'intermédiarité moyenne après la mise en place de la politique en fonction du genre

l'importance des chercheurs masculins entre les deux périodes ait baissé celui-ci reste légèrement supérieur à celui des femmes.

4.3 Évolution du réseau de collaboration en fonction du genre

Pour évaluer l'impact de la politique sur l'expansion du niveau de collaboration des chercheurs féminins par rapport à celui des chercheurs masculins, nous avons d'abord calculé le nombre de collaborations gagnées ou perdues entre les deux périodes pour chaque chercheur. Pour ce faire, nous avons pris la différence entre le degré du chercheur après la politique et son degré avant la politique. Ensuite, nous avons calculé la moyenne de ces différences de degré afin d'avoir une idée globale du gain ou de la perte de collaborateurs pour chaque sexe.

Nos calculs indiquent qu'en moyenne entre les deux périodes, les chercheurs de genre féminin ont obtenu environ 2 nouvelles collaborations supplémentaires, tandis que les chercheurs de genre masculin ont perdu en moyenne 1 collaboration.

4.4 Détection et évolution de communautés

Afin d'identifier les communautés présentent dans notre réseau, nous avons eu recours à l'algorithme de détection de communauté de Louvain. Cet algorithme se base l'optimisation d'une mesure nommé **modularité**. La **modularité** est une mesure qui évalue la qualité d'une partition de graphe en communautés. Elle compare la densité des liens à l'intérieur des communautés à la densité attendue dans un réseau aléatoire. Une modularité élevée indique une structure de communauté bien définie.

L'algorithme nous a permis d'identifier une trentaine de communautés à l'intérieur de notre réseau. Néanmoins, nous ne pouvons nous permettre d'étudier chacune de ces communautés trouvées sans outil de visualisation interactif nous permettant ainsi d'affiner notre analyse et d'identifier si oui non notre réseau contient de l'homophilie.

5 Conclusion

Au cours de la réalisation de notre projet, notre analyse structurelle a montré que la politique mise en place avait un effet sur les collaborations entre les chercheurs. La distribution des degrés a mis en évidence que la majorité des chercheurs avaient un nombre de collaborations limité, avec peu de chercheurs influents ayant un niveau élevé de collaborations. Par la suite, les résultats des mesures de centralité nous ont permis d'identifier les chercheurs centraux en matière de nombre élevé de collaborateurs et en matière d'intermédiation entre chercheurs. De plus, elles nous ont montré que les résultats de celles-ci étaient influencé par plusieurs facteurs, dont la politique mise en place.

Enfin, une analyse selon le genre a révélé que cette politique a favorisé l'émergence de collaborations chez les chercheuses, au détriment de certains chercheurs masculins qui ont perdu des collaborateurs.

Néanmoins, malgré cette analyse approfondie, certaines questions resten sans réponse, telles que l'étude de l'homophilie et la compréhension des communautés détectées dans notre réseau. Ces points constituent des pistes à explorer et à approfondir.

En conclusion, la politique mise en place dans ce réseau a eu un impact positif sur le niveau de collaborations entre les chercheuses, tout en contribuant à réduire le nombre de collaborations des chercheurs masculins. En considérant que cet impact a été positif pour le réseau dans sa globalité

A Code Source Python

```
#!/usr/bin/env python
# coding: utf-8

# # Projet Final d'Analyse De Réseaux
# Auteur : Ulrich Karl ODJO et Alexandra MILLOT

# In[1]:


# chargement des packages nécéessaires
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from matplotlib import ticker
import networkx as nx
from bokeh.io import output_notebook, show, save
from bokeh.models import Range1d, Circle, ColumnDataSource,
    MultiLine
from bokeh.plotting import figure, show, from_networkx
from bokeh.layouts import row


# In[2]:


# for display bokeh graph in notebook
output_notebook()


# In[3]:


# ajustement des paramètres graphique de matplotlib
plt.rc("font", family="Roboto Condensed")
plt.rc("xtick", labelsize="small")
plt.rc("ytick", labelsize="small")
plt.rc("axes", labelsize="small", titlesize="medium")


# ## Mise En Place et Pré Analyse

# In[4]:


# chargement des données
nodes_df = pd.read_csv("./data/nodes_cluster_4.csv", sep=";")
edges_df = pd.read_csv("./data/edges_cluster_4.csv", sep=";")


# In[5]:
```

```
# vérification
nodes_df.head(3)


# In[6]:


# vérification
edges_df.head(3)


# In[7]:


# vérification des valeurs manquantes dans les attributs de noeuds
nodes_df.isna().sum(axis = 0)


# In[8]:


# verification des valeurs manquantes dans les attributs de liens
edges_df.isna().sum(axis = 0)


# In[9]:


# on remplace les 2 valeurs manquantes de la variable gender
# par la valeur la plus fréquente du genre du jeu de données
nodes_df.gender.value_counts() # repartition des genres dans le
    jeu de données


# In[10]:


# imputation par la valeur modalité la plus fréquente
nodes_df['gender'] = nodes_df.gender.fillna(0.0)


# In[11]:


# vérification
nodes_df.isna().sum(axis = 0)


# In[12]:


# on vérifie que tous les noeuds sont présent dans les deux tables
# afin de voir comment construire notre graphe
print(nodes_df.id.nunique())
```

```
print(edges_df.source.nunique())
print(edges_df.target.nunique())


# In[13]:


# construction du graphe
# ======================
# graphe vide
G_before_policy = nx.Graph()
G_after_policy = nx.Graph()
# ensuite on ajoute des noeuds à chacun de nos graphes
G_before_policy.add_nodes_from(nodes_df.id.tolist())
G_after_policy.add_nodes_from(nodes_df.id.tolist())
# ajoutons les attributs de chacun de ces noeuds
coord_dict = {}
core_dict = {}
gender_dict = {}
# correspondance noeud <=> attribut
for index, rows in nodes_df.iterrows():
    coord_dict[rows['id']] = rows['coord']
    core_dict[rows['id']] = rows['core']
    gender_dict[rows['id']] = rows['gender']
# ajout des attributs dans les deux graphes
nx.set_node_attributes(G_before_policy, coord_dict, 'coord')
nx.set_node_attributes(G_before_policy, core_dict, 'core')
nx.set_node_attributes(G_before_policy, gender_dict, 'gender')
nx.set_node_attributes(G_after_policy, coord_dict, 'coord')
nx.set_node_attributes(G_after_policy, core_dict, 'core')
nx.set_node_attributes(G_after_policy, gender_dict, 'gender')
# on aurait aussi pue faire
# for _, rows in nodes_df.iterrows():
#     G_before_policy.add_node(rows["Id"], coord=rows["coord"],
    core=rows["core"], gender=rows["gender"])
#     G_after_policy.add_node(rows["Id"], coord=rows["coord"],
    core=rows["core"], gender=rows["gender"])
# ajoutons maintenant les liens entre nos différents noeuds
# chaque lien étant pondéré par pre_link et post_link
for index, rows in edges_df.iterrows():
    G_before_policy.add_edge(rows['source'], rows['target'],
    pre_link=rows['pre_link'])
    G_after_policy.add_edge(rows['source'], rows['target'],
    post_link=rows['post_link'])


# In[14]:


# vérification des attributs de noeud pour les 2 premiers noeuds
# doit contenir les mêmes informations (informations non variants
    dans le temps)
for node in list(G_before_policy.nodes)[:2]:
    print(f'node␣:␣{node},␣node␣data␣before␣policy␣:␣{
```

```
    G_before_policy.nodes[node]}')
    print(f'node : {node}, node data after policy : {
    G_after_policy.nodes[node]}')


# In[15]:


# vérification des attributs de liens pour les 2 premiers liens
# informations variants dans le temps peut-être différent d'une pé
    riode
# à l'autre
for edge in list(G_before_policy.edges)[:2]:
    print(f'edge : {edge}, edge data before policy : {
    G_before_policy.edges[edge]}')
    print(f'edge : {edge}, edge data after policy : {
    G_after_policy.edges[edge]}')


# In[16]:


# options pour la visualisation
options = {'node_size': 10, #'node_color': 'skyblue', #or 'teal'
          'width': .3, 'font_size': 5, 'edge_color': 'silver'}
# visualisation du graphe
plt.figure(figsize=(6, 4), dpi=400)
plt.subplot(1, 2, 1)
nx.draw(G_before_policy, with_labels=False, **options, node_color
    = 'tab:blue')
plt.title('Before Policy')
plt.subplot(1, 2, 2)
nx.draw(G_after_policy, with_labels=False, **options, node_color =
     'tab:orange')
plt.title('After Policy')
plt.savefig('graph01.png', transparent=None, dpi='figure', format=
    'png',
            metadata=None, bbox_inches=None, pad_inches=0.1,
            facecolor='auto', edgecolor='auto', backend=None)
plt.show()


# In[17]:


# visualisation interactive du réseau pour faciliter l'exploration
# titre du graphique
title_before = 'Network Before Policy'
title_after = 'Network After Policy'

# largeur et hauteur du graphe
WIDTH = 550
HEIGHT = 550
```

```
# quelle information afficher quand on passe le curseur sur un
    noeud
# HOVER_TOOLTIPS = [("@index")]
HOVER_TOOLTIPS = "id: @index, coord: @coord, core: @core, gender:
    @gender"

# création du graphique - dimension des axes, barre d'outil et
    titre
plot_before = figure(tooltips = HOVER_TOOLTIPS,
                     width = WIDTH, height = HEIGHT,
                     x_axis_location=None, y_axis_location=None,
                     tools="pan,wheel_zoom,save,reset",
    active_scroll='wheel_zoom',
                     x_range=Range1d(-10.1, 10.1), y_range=Range1d
    (-10.1, 10.1),
                     title=title_before)

plot_after = figure(tooltips = HOVER_TOOLTIPS,
                    width = WIDTH, height = HEIGHT,
                    x_axis_location=None, y_axis_location=None,
                    tools="pan,wheel_zoom,save,reset",
    active_scroll='wheel_zoom',
                    x_range=Range1d(-10.1, 10.1), y_range=Range1d
    (-10.1, 10.1),
                    title=title_after)

# création du graphe avec un spring layout
network_graph_before = from_networkx(G_before_policy, nx.
    spring_layout, scale=10, center=(0, 0))
network_graph_after = from_networkx(G_after_policy, nx.
    spring_layout, scale=10, center=(0, 0))

# taille des noeud et couleur à utiliser
network_graph_before.node_renderer.glyph = Circle(radius=0.2,
    fill_color='blue')
network_graph_after.node_renderer.glyph = Circle(radius=0.2,
    fill_color='orange')

# opacité des liens et largeur de chaques liens
network_graph_before.edge_renderer.glyph = MultiLine(line_alpha
    =0.5, line_width=1)
network_graph_after.edge_renderer.glyph = MultiLine(line_alpha
    =0.5, line_width=1)

# ajout du réseau au graphique pour l'affichage
plot_before.renderers.append(network_graph_before)
plot_after.renderers.append(network_graph_after)

show(row(plot_before, plot_after))
#save(plot, filename=f"{title}.html")


# In[18]:
```

```
# identitifions les liens avec un poids égale à zéro
# edges_to_remove = [(u, v) for u, v, data in G_before_policy.
    edges(data=True) if data.get("weight", 1) == 0]
edges_to_remove = [(u, v) for u, v, data in G_before_policy.edges(
    data=True) if data["pre_link"] == 0]
G_before = G_before_policy.copy() # copie du graphe initial
G_before.remove_edges_from(edges_to_remove)
edges_to_remove = [(u, v) for u, v, data in G_after_policy.edges(
    data=True) if data["post_link"] == 0]
G_after = G_after_policy.copy() # copie du graph initial
G_after.remove_edges_from(edges_to_remove)


# In[19]:


plt.figure(figsize=(6, 4), dpi=400)
plt.subplot(1, 2, 1)
nx.draw(G_before, with_labels=False, **options)
plt.title('Before Policy')
plt.subplot(1, 2, 2)
nx.draw(G_after, with_labels=False, **options, node_color='tab:
    orange')
plt.title('After Policy')
plt.savefig('graph02.png', transparent=None, dpi='figure', format=
    'png',
            metadata=None, bbox_inches=None, pad_inches=0.1,
            facecolor='auto', edgecolor='auto', backend=None)
plt.show()


# ## Objectif 1

# In[20]:


# calcul des statistiques descriptives du réseau pour chaque pé
    riode (analyse de structure)
def calculate_network_stats(G, weight: str = None) -> dict:
    """
    G : Graphe sur lequel les calculs seront effectuées
    weigth : Chaîne de caractère de l'attribut des poids du réseau
    """
    num_nodes = G.number_of_nodes() # information identique pour
    les deux périodes
    num_edges = G.number_of_edges()
    avg_degree = sum(dict(G.degree(weight=weight)).values()) /
    num_nodes # prise en compte des poids ou non
    return {
        'Nombre de noeuds': num_nodes,
        'Nombre d\'arêtes': num_edges,
        'Degré moyen': avg_degree,
        'Densité du réseau': nx.density(G) # information identique
```

```
     pour les deux périodes
    }

# calcul des statistiques pour chaque période
stats_before_policy = calculate_network_stats(G_before)
stats_after_policy = calculate_network_stats(G_after)


# In[21]:


# affichage des statistiques (avant)
print("Statistiques pour la période avant la mise en place de la
    politique :")
pd.DataFrame.from_dict(stats_before_policy, orient='index',
    columns=['Valeur'])


# In[22]:


# affichage des statistiques (après)
print("Statistiques pour la période après la mise en place de la
    politique :")
pd.DataFrame.from_dict(stats_after_policy, orient='index', columns
    =['Valeur'])


# In[23]:


# distribution des dégrés pondérés avant la mise en place de la
    politique (avec le graphe original)
weighted_degree = G_before_policy.degree(weight='pre_link')
weighted_degree = sorted((d for n, d in weighted_degree), reverse=
    True)
fig, ax = plt.subplots(figsize=(3, 2), dpi=400)
ax.bar(*np.unique(weighted_degree, return_counts=True))
ax.xaxis.set_major_locator(ticker.MultipleLocator(5))
ax.set_xlabel('Dégré')
ax.set_ylabel('Nombre de noeud')
ax.set_title('Distribution des dégrés (période pré-traitrement)')
plt.savefig('graph03.png', transparent=None, dpi='figure', format=
    'png',
            metadata=None, bbox_inches=None, pad_inches=0.1,
            facecolor='auto', edgecolor='auto', backend=None)
plt.show()


# In[24]:


# distribution des dégrés avant la mise en place de la politique
# (avec le graphe dont les liens ont été supprimé, doit fournir le
```

```
    même résultat)
degree_count = nx.degree_histogram(G_before)
degree_range = [_ for _ in range(0, len(degree_count))]
hist_data = (degree_range, degree_count)
plt.figure(figsize = (3, 2), dpi = 400)
plt.bar(*hist_data)
plt.xlabel('Dégré')
plt.ylabel('Nombre de noeud')
plt.title('Distribution des dégrés Avant')
plt.savefig('graph04.png', transparent=None, dpi='figure', format=
    'png',
            metadata=None, bbox_inches=None, pad_inches=0.1,
            facecolor='auto', edgecolor='auto', backend=None)
plt.show()


# In[25]:


N = nodes_df.shape[0] # nombres de noeuds
degree_count_normalise = [val/N for val in degree_count]

plt.figure(figsize=(7, 5), dpi=400)
plt.title('Distribution des degrés (log-log scale)')

plt.xscale("log", base=10)
plt.yscale("log", base=10)

plt.plot(degree_range, degree_count_normalise, 'o')

plt.xlabel('Degré\n(log scale)')
plt.ylabel('Fréquence des noeuds\n(log scale)')

plt.savefig('graph05.png', transparent=None, dpi='figure', format=
    'png',
            metadata=None, bbox_inches=None, pad_inches=0.1,
            facecolor='auto', edgecolor='auto', backend=None)

plt.show()


# In[26]:


# distribution des dégrés pondérés après la mise en place de la
    politique (avec le graphe original)
weighted_degree = G_after_policy.degree(weight='post_link')
weighted_degree = sorted((d for n, d in weighted_degree), reverse=
    True)
plt.figure(figsize = (3, 2), dpi = 400)
plt.bar(*np.unique(weighted_degree, return_counts=True), color = '
    tab:orange')
plt.xlabel('Dégré')
plt.ylabel('Nombre de noeud')
```

```
plt.title('Distribution␣des␣dégrés␣(période␣post-traitrement)')
plt.savefig('graph06.png', transparent=None, dpi='figure', format=
    'png',
            metadata=None, bbox_inches=None, pad_inches=0.1,
            facecolor='auto', edgecolor='auto', backend=None)
plt.show()


# In[27]:


# distribution des dégrés après la mise en place de la politique
# (avec le graphe dont les liens ont été supprimé, doit fournir le
    même résultat)
degree_count = nx.degree_histogram(G_after)
degree_range = [_ for _ in range(0, len(degree_count))]
hist_data = (degree_range, degree_count)
plt.figure(figsize = (3, 2), dpi = 400)
plt.bar(*hist_data, color = 'tab:orange')
plt.xlabel('Dégré')
plt.ylabel('Nombre␣de␣noeud')
plt.title('Distribution␣des␣dégrés␣Après')
plt.savefig('graph07.png', transparent=None, dpi='figure', format=
    'png',
            metadata=None, bbox_inches=None, pad_inches=0.1,
            facecolor='auto', edgecolor='auto', backend=None)
plt.show()


# In[28]:


degree_count_normalise = [val/N for val in degree_count]

plt.figure(figsize=(7, 5), dpi=400)
plt.title('Distribution␣des␣dégrés␣(log-log␣scale)')

plt.xscale("log", base=10)
plt.yscale("log", base=10)

plt.plot(degree_range, degree_count_normalise, 'o',
         color = 'tab:orange')

plt.xlabel('Degré\n(log␣scale)')
plt.ylabel('Fréquence␣des␣noeuds\n(log␣scale)')

plt.savefig('graph08.png', transparent=None, dpi='figure', format=
    'png',
            metadata=None, bbox_inches=None, pad_inches=0.1,
            facecolor='auto', edgecolor='auto', backend=None)

plt.show()
```

```
# In[29]:


# distribution des degrés dans le même graphique
degree_count_before = nx.degree_histogram(G_before)
degree_count_after = nx.degree_histogram(G_after)
degree_range_before = np.arange(len(degree_count_before))
degree_range_after = np.arange(len(degree_count_after))
fig, ax = plt.subplots(figsize=(5, 4), dpi=500)
ax.bar(degree_range_before, degree_count_before, label="Avant",
      alpha=0.6)
ax.bar(degree_range_after, degree_count_after, label="Après",
      alpha=0.6)
ax.legend(loc="best")
ax.set_xlabel("Degré")
ax.set_ylabel("Nombre De Noeud")
ax.set_title("Distribution des degrés avant et après")
plt.savefig('graph01Correction.png', transparent=None, dpi='figure
    ', format='png',
            metadata=None, bbox_inches=None, pad_inches=0.1,
            facecolor='auto', edgecolor='auto', backend=None)
plt.show()


# In[30]:


# centralité de dégrés avant la mise en place de la politique (
    Avant)
number_of_element_to_inspect = 10
degree_centra = sorted(nx.degree_centrality(G_before).items(), key
    =lambda x: x[1], reverse=True)[:number_of_element_to_inspect]
nodes = [str(n) for n, d in degree_centra]
degree = [d for n, d in degree_centra]
plt.figure(figsize=(6, 4), dpi=400)
plt.barh(nodes, degree)
plt.xlabel("Centralité de dégré")
plt.ylabel("Noeud")
# plt.title(f"Top {number_of_element_to_inspect} des chercheurs
    ayant le plus de colaborateurs (Avant)")
plt.title(f"Top {number_of_element_to_inspect} des chercheurs
    centraux (Avant)")
plt.savefig('graph09.png', transparent=None, dpi='figure', format=
    'png',
            metadata=None, bbox_inches=None, pad_inches=0.1,
            facecolor='auto', edgecolor='auto', backend=None)
plt.show()


# In[31]:


# centralité de dégrés après la mise en place de la politique (Apr
    ès)
```

```
degree_centra = sorted(nx.degree_centrality(G_after).items(), key=
    lambda x: x[1], reverse=True)[:number_of_element_to_inspect]
nodes = [str(n) for n, d in degree_centra]
degree = [d for n, d in degree_centra]
plt.figure(figsize=(6, 4), dpi=400)
plt.barh(nodes, degree, color='tab:orange')
plt.xlabel("Centralité de dégré")
plt.ylabel("Noeud")
# plt.title(f"Top {number_of_element_to_inspect} des chercheurs
    ayant le plus de colaborateurs (Après)")
plt.title(f"Top {number_of_element_to_inspect} des chercheurs
    centraux (Après)")
plt.savefig('graph10.png', transparent=None, dpi='figure', format=
    'png',
            metadata=None, bbox_inches=None, pad_inches=0.1,
            facecolor='auto', edgecolor='auto', backend=None)
plt.show()


# In[32]:


# centralité d'intermédiarité (Avant)
between_centra = sorted(nx.betweenness_centrality(G_before).items
    (), key=lambda x: x[1], reverse=True)[:
    number_of_element_to_inspect]
nodes = [str(n) for n, d in between_centra]
degree = [d for n, d in between_centra]
# visualisation des résultats
plt.figure(figsize=(6, 4), dpi=400)
plt.barh(nodes, degree)
plt.xlabel("Centralité d'intermédiarité")
plt.ylabel("Noeud")
# plt.title(f"Top {number_of_element_to_inspect} des chercheurs à
    l'origine des colaborations (Avant)")
plt.title(f"Top {number_of_element_to_inspect} des chercheurs
    centraux (Avant)")
plt.savefig('graph11.png', transparent=None, dpi='figure', format=
    'png',
            metadata=None, bbox_inches=None, pad_inches=0.1,
            facecolor='auto', edgecolor='auto', backend=None)
plt.show()


# In[33]:


# centralité d'intermédiarité (After)
between_centra = sorted(nx.betweenness_centrality(G_after).items()
    , key=lambda x: x[1], reverse=True)[:
    number_of_element_to_inspect]
nodes = [str(n) for n, d in between_centra]
degree = [d for n, d in between_centra]
# visualisation des résultats
```

```
plt.figure(figsize=(6, 4), dpi=400)
plt.barh(nodes, degree, color='tab:orange')
plt.xlabel("Centralité d'intermédiarité")
plt.ylabel("Noeud")
# plt.title(f"Top {number_of_element_to_inspect} des chercheurs à
    l'origine des colaborations (Après)")
plt.title(f"Top {number_of_element_to_inspect} des chercheurs
    centraux (Après)")
plt.savefig('graph12.png', transparent=None, dpi='figure', format=
    'png',
            metadata=None, bbox_inches=None, pad_inches=0.1,
            facecolor='auto', edgecolor='auto', backend=None)
plt.show()


# In[34]:


# est ce que les femmes ont un réseau plus dense que celui des
    hommes
couleurs_noeuds = ['tab:blue'] * N
# conversion en entier de la valeur du genre
gender_atr = [int(data['gender']) for node, data in G_before.nodes
    (data=True)]
# on recherche la position de tout les individus féminin
gender_atr = [idx for idx, val in enumerate(gender_atr) if  val ==
     1]
# on change la valeur de la couleur du noeud associé rose = pour
    les femmes et bleu pour les hommes
for idx in gender_atr:
    couleurs_noeuds[idx] = 'tab:pink'
# visualisation du réseau
new_options = {
    'node_size': 5,
    'edge_color': 'tab:grey', # 'silver'
    'width': .3,
    'font_size': 5}

plt.figure(figsize=(2, 2), dpi=400)
nx.draw(G_before, with_labels=False, **new_options, node_color =
    couleurs_noeuds,
       pos=nx.layout.fruchterman_reingold_layout(G_before))
plt.title('Réseau par genre (Avant Traitement)')
plt.savefig('graph13.png', transparent=None, dpi='figure', format=
    'png',
            metadata=None, bbox_inches=None, pad_inches=0.1,
            facecolor='auto', edgecolor='auto', backend=None)
plt.show()


# In[35]:


plt.figure(figsize=(2, 2), dpi=400)
```

```
nx.draw(G_after, with_labels=False, **new_options, node_color =
    couleurs_noeuds,
       pos=nx.layout.fruchterman_reingold_layout(G_after))
plt.title('Réseau␣par␣genre␣(Après␣Traitement)')
plt.savefig('graph14.png', transparent=None, dpi='figure', format=
    'png',
            metadata=None, bbox_inches=None, pad_inches=0.1,
            facecolor='auto', edgecolor='auto', backend=None)
plt.show()


# In[36]:


# calcul du degré avant
degree_before = dict(nx.degree(G_before))
# calcul du degré après
degree_after = dict(nx.degree(G_after))
# on identifie les noeuds de chaque genre
male_nodes = [n for n, data in G_before.nodes(data=True) if int(
    data['gender']) == 0]
female_nodes = [n for n, data in G_before.nodes(data=True) if int(
    data['gender']) == 1]
# nombre de noeud de chaque genre
N_male = len(male_nodes)
N_female = len(female_nodes)

# calcul du degré moyen pour les hommes et pour les femmes avant
### dans une première approche on peut construire une list
    contenant les degrés de chaque genre
### puis en faire la somme de ces valeurs et diviser par le nombre
     d'éléments mais cela nous
### à un cout au niveau de la mémoire nous obligeant à créer
    structure de donnée en plus pas forcément necessaire.
### Ainsi nous procederons de la manière suivante qui est plus
    direct et nous épargne une structure de données
### supplémentaire

# initialisation de la somme à 0 avant
degree_sum_male_before = 0
degree_sum_female_before = 0
# on parcours chaque noeud masculin et feminin puis on récupère le
     degré correspondant dans le dictionnaire de degré
for m_node in male_nodes:
    degree_sum_male_before += degree_before[m_node]
for f_node in female_nodes:
    degree_sum_female_before += degree_before[f_node]

# initialisation de la somme à 0
degree_sum_male_after = 0
degree_sum_female_after = 0
for m_node in male_nodes:
    degree_sum_male_after += degree_after[m_node]
for f_node in female_nodes:
```

```
    degree_sum_female_after += degree_after[f_node]

# affichage des résultats
print("-"*80)
print("Degré moyen avant la mise en place de la politique")
print(f"Degré moyen des hommes {round(degree_sum_male_before/
    N_male, 4)}")
print(f"Degré moyen des femmes {round(degree_sum_female_before/
    N_female, 4)}")
print("-"*80)
print("Degré moyen après la mise en place de la politique")
print(f"Degré moyen des hommes {round(degree_sum_male_after/N_male
    , 4)}")
print(f"Degré moyen des femmes {round(degree_sum_female_after/
    N_female, 4)}")
print("-"*80)


# In[37]:


# Est-ce que les femmes sont plus central dans le réseau que les
    hommes ?
degree_centra_before = nx.degree_centrality(G_before)
degree_centra_after = nx.degree_centrality(G_after)
between_centra_before = nx.betweenness_centrality(G_before)
between_centra_after = nx.betweenness_centrality(G_after)


# In[38]:


# centralité de dégré pour chaque genre
degree_centra_male_before = {node: degree_centra_before[node] for
    node in male_nodes}
degree_centra_female_before = {node: degree_centra_before[node]
    for node in female_nodes}
degree_centra_male_after = {node: degree_centra_after[node] for
    node in male_nodes}
degree_centra_female_after = {node: degree_centra_after[node] for
    node in female_nodes}
# centralité betweeness pour chaque genre
degree_between_male_before = {node: between_centra_before[node]
    for node in male_nodes}
degree_between_female_before = {node: between_centra_before[node]
    for node in female_nodes}
degree_between_male_after = {node: between_centra_after[node] for
    node in male_nodes}
degree_between_female_after = {node: between_centra_after[node]
    for node in female_nodes}


# In[39]:
```

```
# comparaison des résultat par la moyenne du score de centralité
    par genre
print("-"*80)
print(f"centralité de dégré moyenne pour les femmes avant le
    traitement {round(np.mean(list(degree_centra_female_before.
    values())), 4)}")
print(f"centralité de dégré moyenne pour les hommes avant le
    traitement {round(np.mean(list(degree_centra_male_before.
    values())), 4)}")
print(f"centralité de dégré moyenne pour les femmes après le
    traitement {round(np.mean(list(degree_centra_female_after.
    values())), 4)}")
print(f"centralité de dégré moyenne pour les hommes après le
    traitement {round(np.mean(list(degree_centra_male_after.values
    ())), 4)}")
print("-"*80)
print(f"betweenness centrality moyenne pour les femmes avant le
    traitement {round(np.mean(list(degree_between_female_before.
    values())), 4)}")
print(f"betweenness centrality moyenne pour les hommes avant le
    traitement {round(np.mean(list(degree_between_male_before.
    values())), 4)}")
print(f"betweenness centrality moyenne pour les femmes après le
    traitement {round(np.mean(list(degree_between_female_after.
    values())), 4)}")
print(f"betweenness centrality moyenne pour les hommes après le
    traitement {round(np.mean(list(degree_between_male_after.
    values())), 4)}")
print("-"*80)


# In[40]:


degree_before = dict(G_before.degree())
degree_after = dict(G_after.degree())

degree_before_female = [degree_before[node] for node in
    female_nodes]
degree_after_female = [degree_after[node] for node in female_nodes
    ]

degree_before_male = [degree_before[node] for node in male_nodes]
degree_after_male = [degree_after[node] for node in male_nodes]

diff_degree_female = [after - before for before, after in zip(
    degree_before_female, degree_after_female)]
diff_degree_male = [after - before for before, after in zip(
    degree_before_male, degree_after_male)]


# In[41]:
```

```
# affichage
print(f"changement␣moyen␣en␣nombre␣de␣connexion␣dans␣le␣groupe␣des
    ␣femmes␣{round(np.mean(diff_degree_female),␣4)}")
print(f"changement␣moyen␣en␣nombre␣de␣connexion␣dans␣le␣groupe␣des
    ␣hommes␣{round(np.mean(diff_degree_male),␣4)}")


# In[42]:


# Peut-on identifier des sous communauté ou detecter de l'
    homophilie à l'intérieur du réseaux avant et après la mise en
    place de la politique ?
partition_before = nx.community.louvain_communities(G_before,
    resolution=0.7, seed=123)
print("nb␣de␣communautés␣(louvain)␣:", len(partition_before))

# mesure de la modularité
print("modularité␣(louvain)␣:␣", round(nx.community.modularity(
    G_before, partition_before),2))


# In[43]:


partition_after =  nx.community.louvain_communities(G_after,
    resolution=0.7, seed=123)
print("nb␣de␣communautés␣(louvain)␣:", len(partition_after))

#mesure de la modularité
print("modularité␣(louvain)␣:␣", round(nx.community.modularity(
    G_after, partition_after),2))


# In[44]:


# algorithme maximisant la modularité
greed_before = nx.community.greedy_modularity_communities(G_before
    )
print("nb␣de␣communautés␣(greedy␣mod.␣before)␣:", len(greed_before
    ))
print("modularité␣(greedy␣mod.␣before)␣:␣", round(nx.community.
    modularity(G_after, greed_before),2))


# In[45]:


# algorithme maximisant la modularité
greed_after = nx.community.greedy_modularity_communities(G_after)
print("nb␣de␣communautés␣(greedy␣mod.)␣:", len(greed_after))
print("modularité␣(greedy␣mod.)␣:␣", round(nx.community.modularity
```

```
    (G_after, greed_after),2))


# In[46]:


# a list of colormap of 40 value
COLORS = ['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd',
          '#8c564b', '#e377c2', '#7f7f7f', '#bcbd22', '#17becf',
          '#aec7e8', '#ffbb78', '#98df8a', '#ff9896', '#c5b0d5',
          '#c49c94', '#f7b6d2', '#c7c7c7', '#dbdb8d', '#9edae5',
          '#c7e9c0', '#fdd0a2', '#bcbd22', '#ff9896', '#8c6d31',
          '#9c9ede', '#637939', '#e7cb94', '#6b6ecf', '#b5cf6b',
          '#7570b3', '#d95f02', '#1b9e77', '#d8b365', '#5ab4ac',
          '#7570b3', '#e7298a', '#66a61e', '#e6ab02', '#a6761d']
# couleur des noeuds avant
nodes_colors_before = [COLORS[idx] for idx in range(len(
    partition_before)) for com in partition_before[idx]]
nodes_colors_after = [COLORS[idx] for idx in range(len(
    partition_after)) for com in partition_after[idx]]


# In[47]:


plt.figure(figsize=(3, 3), dpi=400)
nx.draw(G_before, with_labels=False, node_color=
    nodes_colors_before, node_size=3, edge_color='silver',
        width=.3, pos=nx.layout.spring_layout(G_before))
plt.title("Communauté␣avant␣la␣mise␣en␣place␣de␣la␣politique")
plt.savefig('graph15.png', transparent=None, dpi='figure', format=
    'png',
            metadata=None, bbox_inches=None, pad_inches=0.1,
            facecolor='auto', edgecolor='auto', backend=None)
plt.show()


# In[48]:


plt.figure(figsize=(3, 3), dpi=400)
nx.draw(G_after, with_labels=False, node_color =
    nodes_colors_after, node_size=3, edge_color='silver',
        width=.3, pos=nx.layout.spring_layout(G_after))
plt.title("Communauté␣après␣la␣mise␣en␣place␣de␣la␣politique")
plt.savefig('graph16.png', transparent=None, dpi='figure', format=
    'png',
            metadata=None, bbox_inches=None, pad_inches=0.1,
            facecolor='auto', edgecolor='auto', backend=None)
plt.show()
```