

UNIVERSITÉ DE BORDEAUX



Master Mathématiques Appliquées et Statistiques  
Parcours Ingénierie des Risques Économiques et Financiers

---

# Dynamiques de collaboration scientifique et inégalités de genre

Analyse d'un réseau avant et après une intervention politique

---

Analyse des réseaux

Réalisé par : **Karama Fabarika Tiebele**  
**Barkiré Douramane Moussa**  
Encadrant : **Pascale Roux & Nicolas Carayol**

*Année académique 2024–2025*

# Table des matières

Introduction . . . . .	2
Modélisation et Préparation des Données . . . . .	3
Objectif n°1 : Analyse de la structure du réseau et de son évolution . . . . .	4
Objectif n°2 : Analyse des dynamiques selon le genre et les communautés . . . . .	6
.1    Évolution de la Densité du Réseau et Expansion par Genre . . . . .	7
.2    Évolution de la Centralité des Chercheurs en Fonction du Genre . . . . .	8
.3    Détection et Évolution de Communautés et de l'Homophilie . . . . .	9
Conclusion . . . . .	10
I    Code Source Python . . . . .	12

## Introduction

Le présent rapport s'inscrit dans le cadre de l'analyse d'un réseau de collaborations scientifiques, portant sur une communauté de 310 chercheurs partageant une thématique de recherche et une localisation communes. L'objectif principal de cette étude est d'évaluer l'évolution de la structure relationnelle de ce réseau, ainsi que d'analyser l'impact d'un traitement introduit entre deux périodes d'observation. Ce traitement visait explicitement à renforcer et à augmenter les collaborations scientifiques au sein de cette communauté.

L'analyse repose sur deux jeux de données distincts, fournis au format `.csv`. Le premier, `nodes_cluster_F.csv`, contient des informations individuelles sur les chercheurs, considérées comme stables pour cette étude : leur identifiant unique (`id`), leur genre (`gender`), leur appartenance préalable au noyau d'expertise du cluster (`core`), et leur éventuel statut de coordinateur (`coord`). Le second fichier, `edges_cluster_F.csv`, décrit les liens de collaboration scientifique bilatéraux et non dirigés entre les chercheurs, en spécifiant leur état (actifs ou non) avant (`pre_link`) et après (`post_link`) le traitement. Une analyse préliminaire de ces données révèle d'ailleurs une évolution quantitative notable : le nombre de collaborations actives est passé de 869 dans la période pré-traitement à 1258 dans la période post-traitement, suggérant une intensification marquée de la structure relationnelle.

Dans ce contexte, ce rapport poursuit deux objectifs analytiques complémentaires. Premièrement, il s'agit de décrire et d'analyser la structure globale du réseau et son évolution temporelle à travers des mesures et visualisations appropriées (telles que la densité, la distribution des degrés, le clustering, ou encore les centralités de degré et d'intermédiarité). Deuxièmement, l'étude vise à examiner l'impact du traitement sur les dynamiques de connexion en fonction des attributs individuels, en portant une attention particulière au genre. Cet axe inclut l'analyse comparative des métriques de réseau entre hommes et femmes, l'étude de l'homophilie par genre, et l'identification de l'évolution des structures communautaires via l'algorithme de Louvain. L'influence de l'appartenance au groupe `core` sera également brièvement considérée.

Pour répondre à ces objectifs, une approche méthodologique basée sur la modélisation de deux graphes statiques distincts (`G_pre` et `G_post`), représentant respectivement les états avant et après le traitement, a été adoptée. Cette démarche, mise en œuvre en `Python` à l'aide des bibliothèques `NetworkX` et `community`, facilite une comparaison directe des caractéristiques structurelles entre les deux périodes.

La structure de ce rapport suivra logiquement ces objectifs : la section 2 détaillera la modélisation et la préparation des données ; la section 3 sera consacrée à l'analyse structurelle globale et à son évolution (Objectif 1) ; la section 4 abordera les dynamiques spécifiques liées au genre et aux communautés (Objectif 2). Enfin, la section 5 présentera une synthèse

des résultats et une discussion critique, incluant les limites de l'étude et des perspectives de recherche.

## Modélisation et Préparation des Données

Avant de procéder à l'analyse détaillée du réseau, cette section présente la méthode de construction des graphes utilisés, en mettant en évidence les étapes clés de préparation des données et les choix de modélisation adoptés. Les données initiales ont été extraites de deux fichiers au format `.csv` : `nodes_cluster_F.csv`, listant les 310 chercheurs et leurs attributs (genre, appartenance au `core`, statut de coordinateur), et `edges_cluster_F.csv`, décrivant les collaborations bilatérales avant (`pre_link`) et après (`post_link`) l'intervention. Une vérification initiale a confirmé l'absence de valeurs manquantes critiques. Étant donné que les collaborations sont non dirigées, le fichier `edges` contenait potentiellement des doublons pour chaque lien. Un nettoyage a donc été effectué en créant une identification unique pour chaque collaboration et en supprimant ces doublons afin d'assurer la cohérence de la structure du réseau.

Pour analyser les effets de l'intervention, deux options de modélisation étaient envisageables : un graphe unique pondéré ou deux graphes distincts représentant les états avant et après. Nous avons retenu la seconde approche, créant ainsi les graphes `G_pre` (avant intervention) et `G_post` (après intervention). Ce choix de modélisation en réseaux séparés a été privilégié car il permet une comparaison directe et indépendante des structures et des métriques globales du réseau aux deux périodes. Il simplifie également l'interprétation des résultats en évitant la complexité liée à la pondération des arêtes dans un graphe unique.

La construction effective des graphes a été réalisée avec la bibliothèque Python `NetworkX`. Les 310 chercheurs identifiés dans les données `nodes` ont été ajoutés comme nœuds dans les deux graphes, `G_pre` et `G_post`. Chaque nœud s'est vu attribuer ses caractéristiques fixes (`gender`, `core`, `coord`). Par la suite, les liens de collaboration ont été ajoutés conditionnellement : dans `G_pre`, une arête relie deux chercheurs si leur collaboration était active avant l'intervention (`pre_link == 1`), tandis que dans `G_post`, une arête est créée si la collaboration était active après l'intervention (`post_link == 1`). Seules les collaborations effectives (valeur 1) ont ainsi été retenues pour représenter les relations actives pertinentes pour l'analyse, excluant les liens à valeur nulle.

Cette démarche de modélisation et de préparation a abouti aux deux graphes non dirigés qui forment la base de notre étude : `G_pre` (310 nœuds, 869 liens) et `G_post` (310 nœuds, 1258 liens). Les analyses structurelles et relationnelles détaillées dans les sections suivantes s'appuieront sur ces deux représentations distinctes du réseau de collaboration.

## Objectif n°1 : Analyse de la structure du réseau et de son évolution

### Analyse de la Structure Globale du Réseau

La première étape de l'analyse consiste à caractériser la structure globale du réseau avant et après traitement, à travers des indicateurs classiques en théorie des graphes, calculés sur  $G_{pre}$  et  $G_{post}$ . Les principales mesures utilisées sont : l'ordre (nombre de nœuds), la taille (nombre d'arêtes), le degré moyen par chercheur et la densité du réseau. Ces indicateurs sont complétés par les coefficients de clustering et des mesures de connectivité (taille de la plus grande composante connexe et chemin moyen).

Les résultats obtenus sont résumés dans le tableau suivant :

	Valeurs avant le traitement	Valeurs après le traitement
Nombre de nœuds (ordre)	310	310
Nombre de liens (taille)	869	1258
Degré moyen du réseau	5.606	8.116
Densité du réseau	0.0181	0.0263

Tableau 1 – Statistiques calculées sur le réseau avant et après le traitement.

L'analyse de ces mesures montre une évolution nette du réseau. Le nombre de collaborations augmente fortement, passant de 869 à 1258 liens. Le degré moyen s'élève de 5.606 à 8.116, indiquant qu'en moyenne, chaque chercheur collabore avec un nombre plus élevé de collègues après traitement. La densité du réseau croît également (de 1.81% à 2.63%), traduisant une connectivité renforcée. L'augmentation des coefficients de clustering suggère une tendance accrue à la formation de groupes cohésifs. La connectivité globale s'améliore également, avec une composante principale plus grande et des distances moyennes réduites. Ces résultats suggèrent que l'intervention a contribué à dynamiser les interactions scientifiques, rendant le réseau plus dense, plus intégré et plus « clusterisé ».

### Analyse de la Distribution des Collaborations (Degrés)

L'étude de la distribution du degré, c'est-à-dire le nombre de collaborations par chercheur, est essentielle pour comprendre la structure interne du réseau et identifier si les collaborations sont réparties équitablement ou concentrées autour de quelques individus clés.

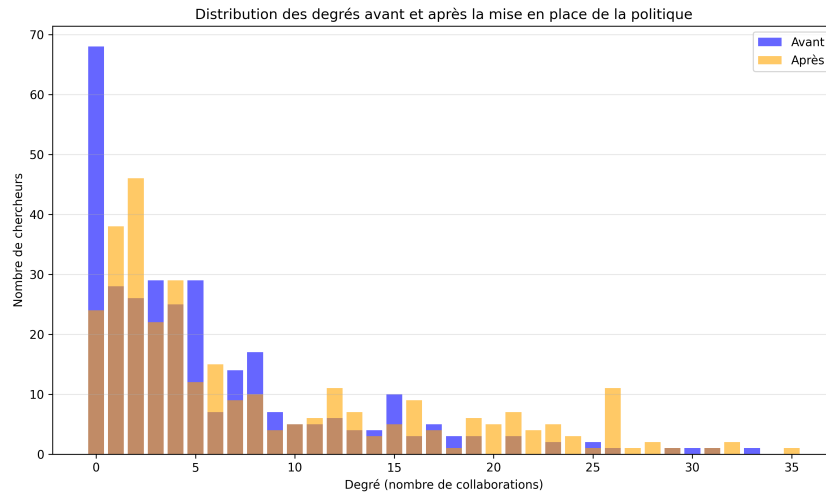


FIGURE 1 – Distribution des degrés avant (bleu) et après (orange) l'intervention

L'analyse de la distribution avant traitement montre que la majorité des chercheurs avaient un nombre de collaborations relativement faible (inférieur à 10). Quelques chercheurs très connectés émergent, créant une structure hétérogène typique des réseaux sociaux scientifiques. La forme de la distribution suit une décroissance caractéristique des réseaux à loi de puissance.

Après traitement, on observe une augmentation générale du nombre de collaborations. Le nombre de chercheurs ayant plus de 10 collaborateurs augmente, traduisant une intensification des interactions. La structure globale reste asymétrique, mais la queue de distribution est légèrement étendue, indiquant que le traitement a encouragé de nouveaux liens sans bouleverser l'architecture centralisée du réseau.

Globalement, avant comme après traitement, le réseau reste centré autour de quelques individus très connectés, bien que l'intervention ait permis une mobilisation accrue de l'ensemble des chercheurs.

## Analyse de l'Importance de Certains Chercheurs (Centralités)

Au sein d'un réseau scientifique, certains individus occupent des positions stratégiques, soit par le nombre de leurs collaborations (centralité de degré), soit par leur rôle d'intermédiaire dans la transmission d'informations (centralité d'intermédiation).

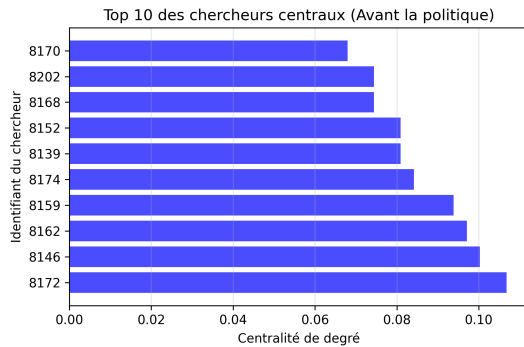


FIGURE 2 – Top 10 Degré Avant

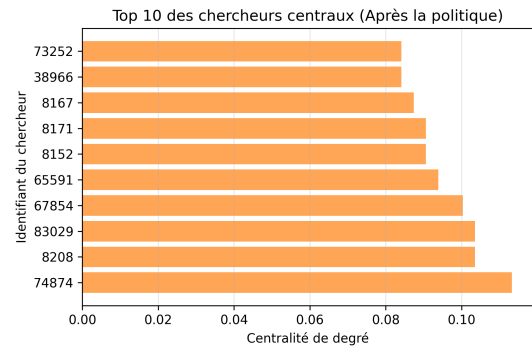


FIGURE 3 – Top 10 Degré Après

La centralité de degré mesure directement l'activité de collaboration. Avant traitement (Figure 2), le réseau est dominé par quelques individus extrêmement connectés. Après traitement (Figure 3), de nouveaux chercheurs émergent parmi les plus connectés, indiquant un rééquilibrage partiel des collaborations et une ouverture des réseaux existants.

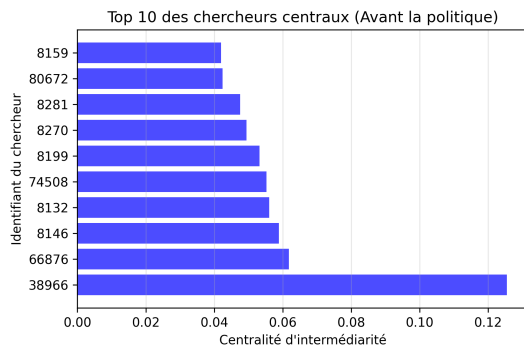


FIGURE 4 – Top 10 Intermédiation Avant

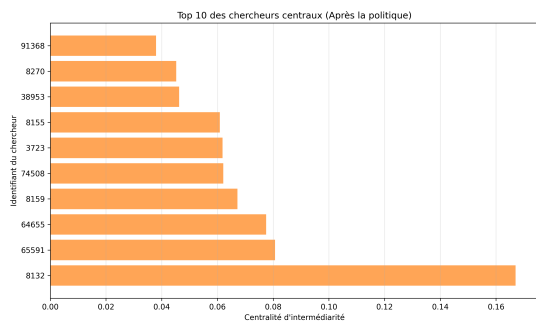


FIGURE 5 – Top 10 Intermédiation Après

La centralité d'intermédiation révèle les chercheurs qui agissent comme « passeurs » d'information. Avant intervention (Figure 4), le réseau repose sur quelques intermédiaires clés. Après intervention (Figure 5), la hiérarchie évolue : certains anciens intermédiaires perdent leur influence, tandis que de nouveaux chercheurs, comme l'identifiant 8132, émergent comme acteurs stratégiques.

## Objectif n°2 : Analyse des dynamiques selon le genre et les communautés

Dans cette deuxième partie, notre objectif est de mieux comprendre l'évolution des collaborations scientifiques en fonction du genre des chercheurs, ainsi que d'analyser l'émer-

gence de sous-communautés et les dynamiques d’homophilie au sein du réseau suite à l’intervention politique. Nous nous appuyons sur des indicateurs globaux et par genre afin de comprendre les effets produits par la politique.

## .1 Évolution de la Densité du Réseau et Expansion par Genre

Afin d’étudier l’impact du genre sur les dynamiques de collaboration, nous analysons ici l’évolution du niveau moyen d’interaction des chercheurs masculins et féminins, avant et après le traitement. Les Figures 6 et 7 (regroupées dans la Figure 8) illustrent visuellement les réseaux avant et après traitement, avec les nœuds colorés selon le genre (bleu pour hommes, rouge pour femmes). Ces visualisations suggèrent qu’avant traitement, les collaborations semblent davantage structurées autour des chercheurs masculins, les chercheuses étant moins nombreuses et parfois plus en périphérie. Après traitement, une meilleure intégration des chercheuses au réseau central est perceptible, et leur nombre de liens semble visiblement augmenter, tout comme l’interconnexion générale du réseau.

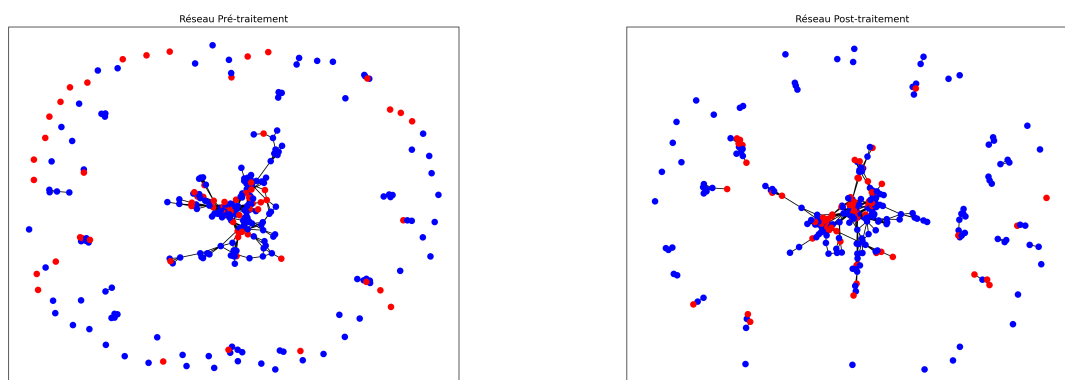


FIGURE 6 – Réseau avant traitement (genre en couleur)      FIGURE 7 – Réseau après traitement (genre en couleur)

FIGURE 8 – Visualisation des réseaux avant et après traitement, colorés par genre.

L’analyse quantitative du degré moyen par genre, présentée dans le Tableau 1, confirme et précise cette tendance.

Genre	Degré moyen avant	Degré moyen après	Évolution (%)
Hommes	5.7265	7.6026	+32.76
Femmes	5.2368	9.6974	+85.18

TABLE 1 – Évolution du degré moyen par genre avant et après la mise en place de la politique.



Ces résultats montrent une augmentation significative du degré moyen pour les deux genres, mais cette augmentation est beaucoup plus marquée pour les femmes (+85,18% contre +32,76% pour les hommes). Les chercheuses surpassent ainsi les chercheurs masculins en termes de connectivité moyenne après la politique. Cette dynamique est également confirmée par l'analyse de la variation nette des collaborations individuelles : en moyenne, les femmes ont gagné près de 4,5 nouvelles collaborations, contre environ 1,9 pour les hommes. Ces résultats indiquent clairement que le traitement a bénéficié de manière plus prononcée aux chercheuses, en augmentant leur intégration au sein du réseau scientifique.

## .2 Évolution de la Centralité des Chercheurs en Fonction du Genre

Après avoir étudié l'évolution du degré moyen, nous examinons l'évolution de la centralité (position structurelle) des chercheurs masculins et féminins. Deux types de centralités sont analysés : la centralité de degré et la centralité d'intermédiarité (Tableaux 2 et 3). Ces mesures permettent d'évaluer non seulement l'importance directe d'un chercheur (nombre de liens), mais aussi son rôle en tant que passeur d'information entre différentes parties du réseau.

Concernant la centralité de degré moyenne (Tableau 2), avant traitement, les hommes étaient légèrement plus centraux (0.0185) que les femmes (0.0169). Après traitement, les femmes augmentent leur centralité de degré de +85,18%, contre +32,76% pour les hommes, rattrapant et dépassant en moyenne les chercheurs masculins.

Genre	Centralité degré avant	Centralité degré après	Évolution (%)
Hommes	0.0185	0.0246	+32.76
Femmes	0.0169	0.0314	+85.18

TABLE 2 – Évolution de la centralité de degré moyenne par genre.

Pour la centralité d'intermédiarité (Tableau 3), les femmes, qui avaient déjà une centralité moyenne légèrement supérieure avant la politique (0.0052 contre 0.0046), voient cet avantage s'accroître : leur centralité moyenne augmente fortement de +35,53%, tandis que celle des hommes progresse plus modestement de +11,28%.

Genre	Centralité intermédiarité avant	Centralité intermédiarité après	Évolution (%)
Hommes	0.0046	0.0051	+11.28
Femmes	0.0052	0.0070	+35.53

TABLE 3 – Évolution de la centralité d'intermédiarité moyenne par genre.

Cette évolution confirme que les femmes ont significativement renforcé leur position stratégique dans la circulation de l'information après le traitement. En résumé, le traitement a contribué à améliorer nettement la position structurelle des femmes dans le réseau, réduisant les inégalités initiales.

### .3 Détection et Évolution de Communautés et de l'Homophilie

Une dernière étape importante consiste à identifier les sous-communautés présentes dans le réseau et à évaluer l'évolution de l'homophilie par genre (Tableau 4). Pour cela, l'algorithme de Louvain (résolution = 0.7) a été utilisé pour détecter les communautés naturelles.

Indicateur	Avant politique	Après politique
Nombre de communautés	93	53
Homophilie de genre	0.659	0.595

TABLE 4 – Évolution du nombre de communautés et de l'homophilie par genre.

Avant traitement, le réseau était divisé en 93 communautés, illustrant une fragmentation importante. Après traitement, le nombre de communautés diminue fortement à 53, traduisant une consolidation notable du réseau et suggérant une meilleure cohésion globale. Parallèlement, l'analyse de l'homophilie de genre révèle une diminution de la proportion de liens intra-genre (de 0,659 à 0,595). Bien que la collaboration entre personnes du même genre reste majoritaire, cette baisse indique une ouverture croissante des collaborations inter-genres, cohérente avec la consolidation observée des communautés. Ces résultats quantitatifs, couplés aux impressions visuelles des Figures 6 et 7, suggèrent que l'intervention a favorisé une plus grande interconnexion et mixité entre les chercheurs.

### Impact de l'Appartenance au Groupe "Core"

L'analyse de l'influence spécifique de l'attribut *core* sur ces dynamiques n'a pas été l'objet d'une investigation approfondie dans ce travail et constitue une piste pour des recherches futures, notamment pour comprendre si cet attribut a modulé les effets observés sur le genre.

L'ensemble de ces analyses par attribut met donc en lumière un impact significatif et particulièrement bénéfique de l'intervention politique sur l'intégration des chercheuses, la consolidation structurelle du réseau et l'accroissement de la mixité des collaborations. Ces effets positifs traduisent une dynamique d'inclusion accrue et une meilleure connectivité du réseau scientifique après intervention.

## Conclusion

Au terme de ce projet d'analyse du réseau de collaborations scientifiques, notre étude a permis d'évaluer avec précision l'impact d'une intervention politique visant à dynamiser les interactions entre 310 chercheurs d'une même communauté scientifique. Deux axes principaux ont structuré notre démarche : la caractérisation de l'évolution globale du réseau et l'étude de l'effet différencié de l'intervention selon le genre des chercheurs.

L'analyse structurelle du réseau a révélé une intensification notable des interactions après l'intervention. Le réseau est devenu plus dense, le degré moyen par chercheur a significativement augmenté, et les coefficients de clustering ont mis en évidence un renforcement de la cohésion locale. Bien que la distribution des degrés conserve une forme asymétrique typique des réseaux sociaux, la hausse du nombre de collaborations traduit une dynamique relationnelle plus active. L'examen des centralités de degré et d'intermédiarité montre une redistribution partielle des rôles d'influence, signe d'une reconfiguration des flux d'information et d'un réseau devenu moins dépendant de quelques acteurs-clés.

Au-delà des tendances générales, l'analyse des dynamiques genrées constitue un apport majeur de ce travail. Les chercheuses ont bénéficié de manière particulièrement marquée de l'intervention : leur degré moyen a crû de 85,2% contre 32,8% pour leurs homologues masculins, et leur rôle stratégique en tant qu'intermédiaires s'est renforcé de manière significative (+35,5% en centralité d'intermédiarité). Ces résultats indiquent que l'intervention n'a pas simplement accru le volume global des collaborations, mais a aussi contribué à une meilleure intégration des chercheuses dans les circuits d'information stratégiques du réseau.

Parallèlement, l'évolution de la structure communautaire corrobore cette dynamique de renforcement de la cohésion : le nombre de sous-communautés a diminué de 93 à 53, traduisant une consolidation du réseau. De plus, la réduction de l'homophilie de genre (de 0.659 à 0.595) témoigne d'une ouverture accrue des collaborations intergenres, et donc d'une réduction des barrières implicites au sein du cluster scientifique.

En conclusion, l'intervention politique semble avoir produit des effets globalement positifs : densification des interactions, émergence de nouvelles figures centrales, meilleure intégration des chercheuses, et consolidation structurelle du réseau. Ces évolutions suggèrent la constitution d'une communauté scientifique plus cohésive, plus égalitaire et potentiellement plus résiliente face aux défis de la recherche collective.

## Limites et Perspectives

Néanmoins, certaines limites méthodologiques doivent être soulignées. L'analyse repose uniquement sur la structure observable des collaborations, sans intégrer d'information sur la qualité, la fréquence ou les thématiques de ces échanges. De même, d'autres va-

riables individuelles, notamment l'appartenance au groupe **core** ou le niveau d'ancienneté scientifique, n'ont pas été explorées dans cette étude. Pour approfondir, des analyses longitudinales intégrant des mesures temporelles fines, une étude de la nature des collaborations, ainsi que des indicateurs de performance scientifique permettraient d'enrichir la compréhension des mécanismes à l'œuvre et de mieux évaluer la durabilité des effets observés.

# I Code Source Python

```

# Chargement des bibliothèques
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from matplotlib import ticker
import networkx as nx

# Chargement des données
nodes = pd.read_csv("nodes_cluster_F.csv" , sep = ";")
nodes.head()

edges = pd.read_csv("edges_cluster_F.csv" , sep = ";")
edges.head()

# Identifier les liens bilatéraux et créer une colonne unique pour
chaque lien
edges['link_id'] = edges.apply(lambda row: tuple(sorted([row['source'],
row['target']])), axis=1)

# Compter les occurrences de chaque lien
link_counts = edges['link_id'].value_counts()
print(f"Nombre de liens uniques: {link_counts.size}")
print(f"Nombre de liens bilatéraux (apparaissant 2 fois): {sum(
    link_counts == 2)}")

# Filtrer pour ne garder qu'une version unique de chaque lien
edges_unique = edges.drop_duplicates(subset=['link_id']).drop(columns=['
link_id'])
print(f"Nombre de lignes après suppression des doublons: {len(
edges_unique)}")

# Créer une version canonique des liens (source < target)
edges_canonical = edges.assign(
    source=edges['link_id'].apply(lambda x: x[0]),
    target=edges['link_id'].apply(lambda x: x[1])
).drop_duplicates(subset=['source', 'target']).drop(columns=['link_id'])
print(f"Nombre de lignes dans la version canonique: {len(edges_canonical
)}")

# Vérifier les liens unidirectionnels
single_links = link_counts[link_counts == 1]
if not single_links.empty:
    print(f"\nAttention: {len(single_links)} liens n'apparaissent qu'une
seule fois!")
    print("Exemples de ces liens unidirectionnels:")
    print(edges[edges['link_id'].isin(single_links.index)].head())

```

---

```

# Cr ation du r seau pr -traitement
G_pre = nx.Graph()
G_pre.add_nodes_from(nodes['id'])
pre_edges = edges_unique[edges_unique['pre_link'] == 1]
G_pre.add_edges_from([(row['source'], row['target']) for _, row in
    pre_edges.iterrows()])

# Cr ation du r seau post-traitement
G_post = nx.Graph()
G_post.add_nodes_from(nodes['id'])
post_edges = edges_unique[edges_unique['post_link'] == 1]
G_post.add_edges_from([(row['source'], row['target']) for _, row in
    post_edges.iterrows()])

# Ajout des attributs aux n uds
for _, row in nodes.iterrows():
    node_id = row['id']
    if node_id in G_pre.nodes():
        G_pre.nodes[node_id]['gender'] = row['gender']
        G_pre.nodes[node_id]['core'] = row['core']
        G_pre.nodes[node_id]['coord'] = row['coord']
    if node_id in G_post.nodes():
        G_post.nodes[node_id]['gender'] = row['gender']
        G_post.nodes[node_id]['core'] = row['core']
        G_post.nodes[node_id]['coord'] = row['coord']

# V rification des r seaux cr s
print("\nR seau pr -traitement:")
print(f"Nombre de n uds: {G_pre.number_of_nodes()}")
print(f"Nombre de liens: {G_pre.number_of_edges()}")

print("\nR seau post-traitement:")
print(f"Nombre de n uds: {G_post.number_of_nodes()}")
print(f"Nombre de liens: {G_post.number_of_edges()}")

# V rification concise des attributs des deux premiers n uds
first_two_ids = list(nodes['id'])[:2]

print("Dans le dataframe nodes:")
print(nodes[nodes['id'].isin(first_two_ids)])

print("\nDans les r seaux:")
for period, G in [("Pr ", G_pre), ("Post", G_post)]:
    print(f"\n{period}-traitement:")
    for node_id in first_two_ids:
        if node_id in G.nodes():
            attrs = G.nodes[node_id]

```

---

---

```

        print(f"N ud {node_id}: gender={attrs.get('gender')}, core
              ={attrs.get('core')}, "
              f"coord={attrs.get('coord')}, degr ={G.degree(node_id
              )}")
def calculate_network_metrics(G, name):
    n_nodes = G.number_of_nodes()
    n_edges = G.number_of_edges()

    # Préparation pour le calcul du chemin moyen
    largest_cc = max(nx.connected_components(G), key=len)
    largest_G = G.subgraph(largest_cc).copy()

    # Toutes les m triques en une seule fois
    metrics = {
        'R seau': name,
        'Nombre de n uds (ordre)': n_nodes,
        'Nombre de liens (taille)': n_edges,
        'Degr moyen du r seau': round(2 * n_edges / n_nodes, 3),
        'Densit du r seau': round(nx.density(G), 4),
        'Average local clustering coefficient': round(nx.
            average_clustering(G), 4),
        'Global clustering coefficient': round(nx.transitivity(G), 4),
        'Average shortest path (plus grande composante)': round(nx.
            average_shortest_path_length(largest_G), 4),
        '% n uds dans plus grande composante': round(len(largest_cc) /
            n_nodes * 100, 2)
    }

    return metrics

# Calcul et affichage des m triques
metrics_df = pd.DataFrame([
    calculate_network_metrics(G_pre, "Valeurs avant la politique"),
    calculate_network_metrics(G_post, "Valeurs apr s la politique")
]).set_index('R seau')

print("\nStatistiques calcul es sur le r seau avant et apr s la mise
      en place de la politique:")
print(metrics_df)

# Distribution des degr s avant la mise en place de la politique
pre_degrees = [d for _, d in G_pre.degree()]
degree_counts = np.unique(pre_degrees, return_counts=True)

# Cr ation du graphique
fig, ax = plt.subplots(figsize=(8, 5), dpi=300)
ax.bar(degree_counts[0], degree_counts[1], color='blue', alpha=0.7)
ax.set_xlabel('Degr (nombre de collaborations)')
ax.set_ylabel('Nombre de chercheurs')

```

---

---

```

ax.set_title('Distribution des degr s avant la mise en place de la
    politique')
ax.xaxis.set_major_locator(ticker.MultipleLocator(5))
ax.grid(axis='y', alpha=0.3)

plt.tight_layout()
plt.savefig('degree_distribution_pre.png', dpi=300)
plt.show()
# Distribution des degr s apr s la mise en place de la politique
post_degrees = [d for _, d in G_post.degree()]
degree_counts = np.unique(post_degrees, return_counts=True)

# Cr ation du graphique
plt.figure(figsize=(8, 5), dpi=300)
plt.bar(degree_counts[0], degree_counts[1], color='tab:orange', alpha
    =0.7)
plt.xlabel('Degr (nombre de collaborations)')
plt.ylabel('Nombre de chercheurs')
plt.title('Distribution des degr s apr s la mise en place de la
    politique')
plt.grid(axis='y', alpha=0.3)
plt.tight_layout()
plt.savefig('degree_distribution_post.png', dpi=300)
plt.show()
# Distribution des degr s dans le m me graphique
degree_count_pre = nx.degree_histogram(G_pre)
degree_count_post = nx.degree_histogram(G_post)

# Assurer que les deux listes ont la m me longueur
max_len = max(len(degree_count_pre), len(degree_count_post))
degree_count_pre = degree_count_pre + [0] * (max_len - len(
    degree_count_pre))
degree_count_post = degree_count_post + [0] * (max_len - len(
    degree_count_post))

# Cr ation du graphique
fig, ax = plt.subplots(figsize=(10, 6), dpi=300)
degree_range = np.arange(max_len)

ax.bar(degree_range, degree_count_pre, label="Avant", alpha=0.6, color='
    blue')
ax.bar(degree_range, degree_count_post, label="Apr s", alpha=0.6, color
    ='orange')

ax.legend(loc="best")
ax.set_xlabel("Degr (nombre de collaborations)")
ax.set_ylabel("Nombre de chercheurs")

```

---



---

```

ax.set_title("Distribution des degr s avant et apr s la mise en place
    de la politique")
ax.grid(axis='y', alpha=0.3)

plt.tight_layout()
plt.savefig('degree_distribution_comparison.png', dpi=300)
plt.show()

# Calcul des centralit s
def get_centrality_df(G):
    return pd.DataFrame({
        'node': list(G.nodes()),
        'degree': [d for _, d in G.degree()],
        'betweenness': list(nx.betweenness centrality(G).values()),
        'closeness': list(nx.closeness centrality(G).values())
    })

pre_cent = get_centrality_df(G_pre)
post_cent = get_centrality_df(G_post)

# Fusion avec attributs
pre_cent = pre_cent.merge(nodes, left_on='node', right_on='id')
post_cent = post_cent.merge(nodes, left_on='node', right_on='id')
pre_cent
post_cent.head()

# Top 10 des chercheurs centraux avant la politique
number_of_element_to_inspect = 10
degree Centrality = nx.degree Centrality(G_pre)
top_nodes = sorted(degree Centrality.items(), key=lambda x: x[1],
    reverse=True)[:number_of_element_to_inspect]

# Pr paration des donn es pour le graphique
nodes_ids = [str(n) for n, _ in top_nodes]
centrality_values = [c for _, c in top_nodes]

# Cr ation du graphique
plt.figure(figsize=(6, 4), dpi=300)
plt.barh(nodes_ids, centrality_values, color='blue', alpha=0.7)
plt.xlabel("Centralit de degr ")
plt.ylabel("Identifiant du chercheur")
plt.title(f"Top {number_of_element_to_inspect} des chercheurs centraux (
    Avant la politique)")
plt.grid(axis='x', alpha=0.3)
plt.tight_layout()
plt.savefig('top_central_researchers_pre.png', dpi=300)
plt.show()

# Top 10 des chercheurs centraux apr s la politique

```

---

---

```

number_of_element_to_inspect = 10
degree Centrality = nx.degree Centrality(G_post)
top_nodes = sorted(degree Centrality.items(), key=lambda x: x[1],
                    reverse=True)[:number_of_element_to_inspect]

# Préparation des données pour le graphique
nodes_ids = [str(n) for n, _ in top_nodes]
centrality_values = [c for _, c in top_nodes]

# Création du graphique
plt.figure(figsize=(10, 6), dpi=300)
plt.barh(nodes_ids, centrality_values, color='tab:orange', alpha=0.7)
plt.xlabel("Centralité de degré ")
plt.ylabel("Identifiant du chercheur")
plt.title(f"Top {number_of_element_to_inspect} des chercheurs centraux (
Après la politique)")
plt.grid(axis='x', alpha=0.3)
plt.tight_layout()
plt.savefig('top_central_researchers_post.png', dpi=300)
plt.show()

# Top 10 des chercheurs par centralité d'intermédiation avant la
politique
number_of_element_to_inspect = 10
betweenness Centrality = nx.betweenness Centrality(G_pre)
top_nodes = sorted(betweenness Centrality.items(), key=lambda x: x[1],
                    reverse=True)[:number_of_element_to_inspect]

# Préparation des données pour le graphique
nodes_ids = [str(n) for n, _ in top_nodes]
centrality_values = [c for _, c in top_nodes]

# Création du graphique
plt.figure(figsize=(6, 4), dpi=300)
plt.barh(nodes_ids, centrality_values, color='blue', alpha=0.7)
plt.xlabel("Centralité d'intermédiation ")
plt.ylabel("Identifiant du chercheur")
plt.title(f"Top {number_of_element_to_inspect} des chercheurs centraux (
Avant la politique)")
plt.grid(axis='x', alpha=0.3)
plt.tight_layout()
plt.savefig('top_betweenness_researchers_pre.png', dpi=300)
plt.show()

# Top 10 des chercheurs par centralité d'intermédiation après la
politique
number_of_element_to_inspect = 10
betweenness Centrality = nx.betweenness Centrality(G_post)

```

---

---

```

top_nodes = sorted(betweenness centrality.items(), key=lambda x: x[1],
                    reverse=True)[:number_of_element_to_inspect]

# Préparation des données pour le graphique
nodes_ids = [str(n) for n, _ in top_nodes]
centrality_values = [c for _, c in top_nodes]

# Création du graphique
plt.figure(figsize=(10, 6), dpi=300)
plt.barh(nodes_ids, centrality_values, color='tab:orange', alpha=0.7)
plt.xlabel("Centralité d'intermédiation")
plt.ylabel("Identifiant du chercheur")
plt.title(f"Top {number_of_element_to_inspect} des chercheurs centraux (
    Après la politique)")
plt.grid(axis='x', alpha=0.3)
plt.tight_layout()
plt.savefig('top_betweenness_researchers_post.png', dpi=300)
plt.show()

# Calcul du degré avant et après
degree_pre = dict(nx.degree(G_pre))
degree_post = dict(nx.degree(G_post))

# Identification des nœuds par genre
male_nodes = [n for n, data in G_pre.nodes(data=True) if data.get('
    gender') == 0]
female_nodes = [n for n, data in G_pre.nodes(data=True) if data.get('
    gender') == 1]

# Nombre de nœuds 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 la
    politique
degree_sum_male_pre = sum(degree_pre[m_node] for m_node in male_nodes)
degree_sum_female_pre = sum(degree_pre[f_node] for f_node in
    female_nodes)

# Calcul du degré moyen pour les hommes et pour les femmes après la
    politique
degree_sum_male_post = sum(degree_post[m_node] for m_node in male_nodes)
degree_sum_female_post = sum(degree_post[f_node] for f_node in
    female_nodes)

# Calcul des moyennes
avg_degree_male_pre = degree_sum_male_pre / N_male
avg_degree_female_pre = degree_sum_female_pre / N_female

```

---

---

```

avg_degree_male_post = degree_sum_male_post / N_male
avg_degree_female_post = degree_sum_female_post / N_female

# Affichage des r sultats
print("-"*80)
print("Degr  moyen avant la mise en place de la politique")
print(f"Degr  moyen des hommes: {round(avg_degree_male_pre, 4)}")
print(f"Degr  moyen des femmes: {round(avg_degree_female_pre, 4)}")
print("-"*80)
print("Degr  moyen apr s la mise en place de la politique")
print(f"Degr  moyen des hommes: {round(avg_degree_male_post, 4)}")
print(f"Degr  moyen des femmes: {round(avg_degree_female_post, 4)}")
print("-"*80)

# Cr ation d'un tableau r capitulatif
degree_by_gender = pd.DataFrame({
    'Genre': ['Hommes', 'Femmes'],
    'Degr  moyen avant': [round(avg_degree_male_pre, 4), round(
        avg_degree_female_pre, 4)],
    'Degr  moyen apr s': [round(avg_degree_male_post, 4), round(
        avg_degree_female_post, 4)],
    'volution (%)': [
        round((avg_degree_male_post - avg_degree_male_pre) /
            avg_degree_male_pre * 100, 2),
        round((avg_degree_female_post - avg_degree_female_pre) /
            avg_degree_female_pre * 100, 2)
    ]
})

print("\nTableau r capitulatif des degr s moyens par genre:")
print(degree_by_gender)

# Visualisation des r sultats
plt.figure(figsize=(10, 6))
bar_width = 0.35
index = np.arange(2)

plt.bar(index - bar_width/2, [avg_degree_male_pre, avg_degree_female_pre
    ],
        bar_width, label='Avant politique', color='blue', alpha=0.7)
plt.bar(index + bar_width/2, [avg_degree_male_post,
    avg_degree_female_post],
        bar_width, label='Apr s politique', color='orange', alpha=0.7)

plt.xlabel('Genre')
plt.ylabel('Degr  moyen')
plt.title('volution  du degr  moyen par genre avant et apr s la
    politique')

```

---

---

```

plt.xticks(index, ('Hommes', 'Femmes'))
plt.legend()
plt.grid(axis='y', alpha=0.3)

plt.tight_layout()
plt.savefig('degree_by_gender.png', dpi=300)
plt.show()

# Calcul des centralit s avant et apr s la politique
degree_centra_pre = nx.degree_centrality(G_pre)
degree_centra_post = nx.degree_centrality(G_post)
between_centra_pre = nx.betweenness_centrality(G_pre)
between_centra_post = nx.betweenness_centrality(G_post)

# Identification des n uds par genre
male_nodes = [n for n, data in G_pre.nodes(data=True) if data.get('
    gender') == 0]
female_nodes = [n for n, data in G_pre.nodes(data=True) if data.get('
    gender') == 1]

# Calcul des centralit s moyennes par genre
# Centralit  de degr 
avg_degree_cent_male_pre = np.mean([degree_centra_pre[n] for n in
    male_nodes])
avg_degree_cent_female_pre = np.mean([degree_centra_pre[n] for n in
    female_nodes])
avg_degree_cent_male_post = np.mean([degree_centra_post[n] for n in
    male_nodes])
avg_degree_cent_female_post = np.mean([degree_centra_post[n] for n in
    female_nodes])

# Centralit  d'interm diarit 
avg_between_cent_male_pre = np.mean([between_centra_pre[n] for n in
    male_nodes])
avg_between_cent_female_pre = np.mean([between_centra_pre[n] for n in
    female_nodes])
avg_between_cent_male_post = np.mean([between_centra_post[n] for n in
    male_nodes])
avg_between_cent_female_post = np.mean([between_centra_post[n] for n in
    female_nodes])

# Cr ation d'un tableau r capitulatif
centrality_by_gender = pd.DataFrame({
    'Genre': ['Hommes', 'Femmes'],
    'Centralit  de degr  (Avant)': [round(avg_degree_cent_male_pre, 4),
        round(avg_degree_cent_female_pre, 4)],
    'Centralit  de degr  (Apr s)': [round(avg_degree_cent_male_post,
        4), round(avg_degree_cent_female_post, 4)],

```

---

```

    'volution degr (%)': [
        round((avg_degree_cent_male_post - avg_degree_cent_male_pre) /
              avg_degree_cent_male_pre * 100, 2),
        round((avg_degree_cent_female_post - avg_degree_cent_female_pre)
              / avg_degree_cent_female_pre * 100, 2)
    ],
    'Centralit d\'interm diarit (Avant)': [round(
        avg_between_cent_male_pre, 4), round(avg_between_cent_female_pre,
        4)],
    'Centralit d\'interm diarit (Apr s)': [round(
        avg_between_cent_male_post, 4), round(
        avg_between_cent_female_post, 4)],
    'volution interm diarit (%)': [
        round((avg_between_cent_male_post - avg_between_cent_male_pre) /
              avg_between_cent_male_pre * 100, 2),
        round((avg_between_cent_female_post -
              avg_between_cent_female_pre) / avg_between_cent_female_pre *
              100, 2)
    ]
}

print("Analyse de la centralit par genre:")
print(centrality_by_gender)

# Visualisation des r sultats - Centralit de degr
plt.figure(figsize=(12, 5))

# Graphique pour la centralit de degr
plt.subplot(1, 2, 1)
bar_width = 0.35
index = np.arange(2)

plt.bar(index - bar_width/2, [avg_degree_cent_male_pre,
    avg_degree_cent_female_pre],
        bar_width, label='Avant politique', color='blue', alpha=0.7)
plt.bar(index + bar_width/2, [avg_degree_cent_male_post,
    avg_degree_cent_female_post],
        bar_width, label='Apr s politique', color='orange', alpha=0.7)

plt.xlabel('Genre')
plt.ylabel('Centralit de degr moyenne')
plt.title('Centralit de degr par genre')
plt.xticks(index, ('Hommes', 'Femmes'))
plt.legend()
plt.grid(axis='y', alpha=0.3)

# Graphique pour la centralit d\'interm diarit
plt.subplot(1, 2, 2)

```

---

```

plt.bar(index - bar_width/2, [avg_between_cent_male_pre,
    avg_between_cent_female_pre],
    bar_width, label='Avant politique', color='blue', alpha=0.7)
plt.bar(index + bar_width/2, [avg_between_cent_male_post,
    avg_between_cent_female_post],
    bar_width, label='Apr s politique', color='orange', alpha=0.7)

plt.xlabel('Genre')
plt.ylabel('Centralit d\'interm diarit moyenne')
plt.title('Centralit d\'interm diarit par genre')
plt.xticks(index, ('Hommes', 'Femmes'))
plt.legend()
plt.grid(axis='y', alpha=0.3)

plt.tight_layout()
plt.savefig('centrality_by_gender.png', dpi=300)
plt.show()

# Test statistique pour comparer les centralit s entre hommes et femmes
from scipy import stats

# Test pour la centralit de degr avant la politique
t_stat_degree_pre, p_value_degree_pre = stats.ttest_ind(
    [degree_centra_pre[n] for n in male_nodes],
    [degree_centra_pre[n] for n in female_nodes]
)

# Test pour la centralit de degr apr s la politique
t_stat_degree_post, p_value_degree_post = stats.ttest_ind(
    [degree_centra_post[n] for n in male_nodes],
    [degree_centra_post[n] for n in female_nodes]
)

print("\nTests statistiques pour comparer les centralit s entre hommes
    et femmes:")
print(f"Centralit de degr avant: p-value = {p_value_degree_pre:.4f}"
    )
print(f"Centralit de degr apr s: p-value = {p_value_degree_post:.4f}
    ")

IGNORE_WHEN_COPYING_START
Use code with caution. Python
IGNORE_WHEN_COPYING_END

# Calcul des degr s avant et apr s

```

---

```

degree_pre = dict(G_pre.degree())
degree_post = dict(G_post.degree())

# Identification des n uds par genre
male_nodes = [n for n, data in G_pre.nodes(data=True) if data.get('
    gender') == 0]
female_nodes = [n for n, data in G_pre.nodes(data=True) if data.get('
    gender') == 1]

# Extraction des degr s par genre
degree_pre_female = [degree_pre[node] for node in female_nodes]
degree_post_female = [degree_post[node] for node in female_nodes]

degree_pre_male = [degree_pre[node] for node in male_nodes]
degree_post_male = [degree_post[node] for node in male_nodes]

# Calcul des diff rences de degr pour chaque chercheur
diff_degree_female = [post - pre for pre, post in zip(degree_pre_female,
    degree_post_female)]
diff_degree_male = [post - pre for pre, post in zip(degree_pre_male,
    degree_post_male)]

# Affichage des changements moyens en nombre de connexions par genre
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)}")

from networkx.algorithms import community

# D tecton de communaut s avec l'algorithme de Louvain
pre_comm = community.louvain_communities(G_pre, seed=123, resolution
    =1.2)
post_comm = community.louvain_communities(G_post, seed=123, resolution
    =1.2)

# Cr ation d'un dictionnaire pour associer chaque n ud sa
    communaut
pre_comm_dict = {}
for i, comm in enumerate(pre_comm):
    for node in comm:
        pre_comm_dict[node] = i

post_comm_dict = {}
for i, comm in enumerate(post_comm):
    for node in comm:
        post_comm_dict[node] = i

```



---

```

# Ajout des attributs de communauté aux nœuds
nx.set_node_attributes(G_pre, pre_comm_dict, 'community')
nx.set_node_attributes(G_post, post_comm_dict, 'community')

# Fonction pour calculer l'homophilie
def homophily(G, attr):
    same_attr = sum(1 for u, v in G.edges() if G.nodes[u][attr] == G.
        nodes[v][attr])
    return same_attr / G.number_of_edges()

# Calcul et affichage de l'homophilie par genre
print(f"Homophilie genre pr : {homophily(G_pre, 'gender'):.3f}")
print(f"Homophilie genre post: {homophily(G_post, 'gender'):.3f}")

# Calcul et affichage du nombre de communautés détectées
N_pre = len(pre_comm)
N_post = len(post_comm)
print("Le nombre de communautés détectées avant la politique est:",
    N_pre)
print("Le nombre de communautés détectées après la politique est:",
    N_post)

# Visualisation des réseaux
def plot_network(G, title, attr='gender'):
    plt.figure(figsize=(10, 8))
    pos = nx.spring_layout(G, seed=42)

    # Coloration par attribut
    colors = ['blue', 'red']
    node_colors = [colors[G.nodes[n][attr]] for n in G.nodes()]

    nx.draw_networkx(G, pos, node_color=node_colors,
        node_size=50, with_labels=False)
    plt.title(title)
    plt.savefig(f"{title.lower().replace(' ', '_')}.png", dpi=300)
    plt.show()

plot_network(G_pre, "Réseau Pré-traitement")
plot_network(G_post, "Réseau Post-traitement")

```