Network Analysis of Mortality-Associated Disease Co-Occurrences in EHRSHOT Patients Under 30

Patrick Aabram
Vivekanandhan Kathirvel
Justin Scandale
Samiuddin Ahmed Meenam
Department of Computer Science, University of South Florida
Tampa, FL, United States

Abstract

We analyze disease co-occurrence patterns among individuals under 30 years old who experienced mortality using the EHRSHOT dataset. We construct an undirected weighted network where nodes represent diseases and edges represent the number of patients who had both diseases recorded. We compute degree and betweenness centralities to identify highly-connected diseases and bridging diseases, respectively, and apply the Louvain community detection algorithm to identify clusters of comorbid conditions. This paper provides the analysis pipeline, visualizations, and interpretation guidelines for clinical and epidemiological follow-up.

CCS Concepts

• **Networks** → **Network analysis**; • **Information systems** → *Data mining*.

Keywords

disease co-occurrence, network analysis, centrality, Louvain, EHRSHOT

1 Introduction

1.1 Motivation

Understanding the network of co-occurring diseases among patients who die before age 30 can reveal critical health patterns contributing to early mortality. Traditional analyses often focus on individual diseases, but young-age mortality is frequently driven by interacting comorbidities. Leveraging electronic health records (EHRs) enables a systems-level perspective, allowing the identification of disease interactions that may inform preventive care and targeted interventions.

1.2 Related Works

To situate our research, we reviewed recent studies that apply network-based approaches to comorbidity analysis:

- **Koskinen et al. [10]** constructed a disease–disease co-occurrence network from large hospital datasets, identified significant disease clusters, and linked them to mortality risk. Their work follows a similar analytical framework to ours—building weighted networks, applying community detection, and interpreting clusters clinically. However, our analysis focuses specifically on deaths under age 30 and explores bridging diseases through betweenness centrality.
- **Jensen et al. [11]** used nationwide registry data covering 6.2 million patients to construct directed, time-ordered disease networks, revealing how one condition increases risk for another. While they focused on temporal dynamics across populations, our approach centers on undirected co-occurrence patterns within a young deceased population (under 30), emphasizing community structure and bridging conditions.
- **Jiang et al. [5]** derived an epidemiological human disease network from disease co-occurrence data in Taiwan, constructing an undirected weighted network where edge weights represented co-occurrence frequency. Their methodology of building disease networks from population-level health records directly informed our edge weighting approach, though our study targets a younger population.
- **Li et al. [6]** applied graphical models to electronic health records for comorbidity network analysis, emphasizing the distinction between direct and indirect disease associations. Their work highlights the importance of betweenness centrality in identifying bridging diseases that connect otherwise separate comorbidity clusters—a key focus of our third research question.
- **Fotouhi et al. [3]** provided a comprehensive review of statistical methods for constructing disease comorbidity networks from longitudinal inpatient data. Their discussion of network construction choices and validation approaches informed our methodological decisions, particularly regarding edge weighting and community detection algorithm selection.
- **Dervic et al. [2]** analyzed cradle-to-grave disease trajectories using multilayer comorbidity networks, incorporating temporal information to reveal disease progression patterns. While our study uses static co-occurrence networks, their work demonstrates the value of network analysis in understanding disease relationships and motivates our future work on incorporating temporal dynamics.

Our study builds on these established methodologies while uniquely focusing on mortality-associated comorbidities in patients under 30 years old. We combine degree centrality, betweenness centrality, and community detection (Louvain algorithm) to identify both highly connected hub diseases and critical bridging conditions within this population.

1.3 Background on Network Analysis

In a disease co-occurrence network, each node represents a diagnosis, and edges connect diseases that appear together in patient records, revealing potential relationships or shared risk factors.

Analyzing these connections helps identify tightly linked groups of conditions—known as communities—that may share biological, environmental, or demographic similarities. Some diseases act as central nodes or bridges between clusters, highlighting key comorbidities and pivotal points in disease progression. Studying such networks provides valuable insights into disease mechanisms, healthcare planning, and risk prediction.

1.4 Research Questions

In this project, we analyze the **EHRShot** dataset to address three main questions:

(1) Which diseases have high **degree centrality** (i.e., co-occur with many other conditions)?
(2) What clusters emerge from **Girvan–Newman community detection** on the disease co-occurrence network?
(3) Which nodes act as **bridges** (high betweenness centrality) that connect clusters?

1.5 Summary

Our study constructs a weighted disease co-occurrence network using the EHRShot dataset, focusing on patients under 30. Each edge weight represents how frequently two diseases co-occur, capturing comorbidity patterns in young populations. Through centrality measures, we identify diseases that link different parts of the network and may serve as early indicators of broader health complications. Community detection uncovers clusters of related conditions, revealing shared health domains. Overall, this approach highlights key bridging diseases associated with early mortality and supports data-driven strategies for preventive care and early intervention.

2 Dataset and Preprocessing

2.1 EHRSHOT Subset

We worked with a subset of the EHRSHOT dataset consisting of conditions exhibited by patients whose recorded age at death was less than 30 years. We identified the relevant tables for our research question and their relationship as per the diagram in figure 1. Each record in *person* is associated with a *person_id* which relates that person to records in *death* and *condition_occurrence*. Each condition is labeled with a *condition_concept_id*, which can then be associated with a record in *concept* identifying the standard vocabulary used to describe that condition.

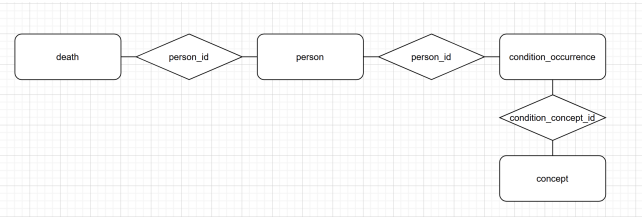

Figure 1: Entity-Relationship diagram of table subset

2.2 Table Joining and Filtering

To select the data used to construct our network, we completed the following steps:

(1) The first step was to find people whose age at the time of death was less than 30 years old. To do this, we first created a date of birth column in the table *person*, joining the $year\_of\_birth$, $month\_of\_birth$, and day_of_birth columns into a single datetime object.
(2) We then left-join *person* to *death* and calculate an age-at-death column as the difference between $death\_date$ and date of birth.
(3) Finally, we filter this resulting table to rows where age-at-death is less than 30 years.
(4) Next, we could identify what conditions occurred in persons who died under 30. We created a mask of $condition\_occurrence$, keeping only rows with a $person\_id$ that had a matching record in our filtered deaths table. This reduced the number of unique conditions from over 6,000 to 615.
(5) Finally, we left-joined *concept* to the filtered condition occurrences, allowing each condition to be identified by its $concept\_name$.

2.3 Edge construction and weighting

Following established practices in comorbidity network construction [5, 9], we define an undirected weighted graph $G = (V, E)$:

- V: unique condition concepts (according to standard vocabulary)
- E: undirected edges (u, v) where u and v co-occur in the same patient record at least once
- Weight $w(u, v)$: number of distinct patients in which both u and v appear

We realized this by creating an edge list as a dataframe structure with three columns: $concept\_a$, $concept\_b$, and n_people. Each row describes the frequency of a condition co-occurrence, i.e., that condition A and condition B each occurred at some point in n persons. This yielded over 66,000 edges. Finally, edges with a weight of 1 (i.e., comorbidities that only occurred in one person) were removed, leaving 5,677 edges.

3 Metrics and Analysis Methods

3.1 Overview

To quantify the structure and connectivity of the disease co-occurrence network, we computed several network-level and node-level metrics. These measures characterize how diseases are interconnected, how information may flow through the network, and how clusters of related conditions emerge. The selected metrics align directly with our research questions, focusing on identifying highly connected diseases, bridge conditions, and community structures. All code was implemented in Python using pandas [7], NetworkX [4], and Plotly [8] libraries. The NetworkX library provides well-established implementations of network analysis algorithms including centrality measures and the Louvain community detection method.

3.2 Degree Centrality

Degree centrality measures the number of direct connections a node has within the network. For an undirected network, the degree centrality of a node i is defined as:

$$C_D(i) = \frac{k_i}{N-1}$$

where k_i is the number of edges incident to node i and N is the total number of nodes. In this study, diseases with high $C_D(i)$ represent core comorbidities that frequently co-occur with many other conditions. This directly addresses our first research question by identifying the most interconnected diseases within the young mortality subset.

3.3 Betweenness Centrality

Betweenness centrality quantifies how often a node lies on the shortest paths between other nodes, capturing its role as a connector between clusters. For a node i, it is defined as:

$$C_B(i) = \sum_{s \neq i \neq t} \frac{\sigma_{st}(i)}{\sigma_{st}}$$

where σ_{st} is the total number of shortest paths between nodes s and t, and $\sigma_{st}(i)$ is the number of those paths that pass through node i. Diseases with high $C_B(i)$ values serve as bridges between communities, linking otherwise separate disease clusters. These nodes are clinically important, as they may represent conditions that mediate comorbidity progression.

3.4 Community Detection

To identify groups of diseases that frequently co-occur, we applied the **Louvain community detection algorithm**, which optimizes network modularity through an iterative process of node aggregation and community refinement. Unlike edge-removal methods such as Girvan–Newman, the Louvain method efficiently partitions large networks by maximizing the modularity score, making it well-suited for weighted disease co-occurrence graphs.

Formally, the modularity Q quantifies the strength of the community structure as:

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)$$

where A_{ij} represents the edge weight between nodes i and j, k_i and k_j are the weighted degrees of nodes i and j, m is the total edge weight in the network, and $\delta(c_i, c_j)$ equals 1 if nodes i and j belong to the same community and 0 otherwise.

Each detected community corresponds to a cluster of diseases that tend to co-occur within similar patient groups, reflecting shared biological pathways or risk factors. By revealing these modular structures, the Louvain algorithm helps uncover clinically meaningful comorbidity patterns within the under-30 population.

3.5 Visualization

We generate an interactive Plotly figure following visualization best practices for disease networks [2]:

- Edges drawn as lines with transparency to show network density
- Nodes colored by Louvain community and sized by degree centrality
- Hover text showing node name, degree centrality, betweenness centrality, and community assignment

3.6 Relevance to Research Objectives

These metrics collectively provide a multi-scale understanding of the disease network. Degree centrality identifies the most connected diseases, betweenness centrality highlights bridging conditions, and community detection uncovers modular disease groups. Together, they reveal structural patterns that shed light on how interacting comorbidities contribute to early mortality in patients under 30.

3.7 Research Standards

Our analytical framework follows established standards in network epidemiology and comorbidity research [10, 11]. These studies emphasize using centrality and modularity-based measures to interpret disease relationships from EHR data. By adopting these well-validated methods, our study maintains methodological rigor while tailoring the analysis to a younger patient cohort.

4 Results

4.1 Network Statistics

The disease co-occurrence network contains 615 conditions (nodes) connected by 66,653 co-occurrence relationships (edges). The network has a density of 0.3530, meaning approximately 35% of all possible disease pairs appear together in at least one patient. The network is fully connected, showing that diseases that are key to mortality rates are highly interconnected.

4.2 Top Diseases by Degree Centrality

Table 1 shows the top 10 diseases ranked by degree centrality. These are the diseases that co-occur with the most other conditions in the network. We observe that breathing problems (Dyspnea, Pleural effusion), heart-related issues (Electrocardiogram abnormal, Cardiac arrhythmia), and general symptoms (Anemia, Fever) have the highest degree centrality values. This suggests these conditions frequently appear alongside many fatal diseases in patients under 30.

Table 1: Top 10 diseases by degree centrality

Disease	Centrality	Connections
Dyspnea	0.9593	589
Pleural effusion	0.9593	589
Electrocardiogram abnormal	0.9430	579
Abdominal pain	0.8925	548
Cardiac arrhythmia	0.8925	548
Anemia	0.8909	547
Constipation	0.8730	536
Nausea	0.8664	532
Chest pain	0.8648	531
Nausea and vomiting	0.8648	531

4.3 Top Diseases by Betweenness Centrality

Table 2 shows the top 10 diseases ranked by betweenness centrality. These diseases act as bridges connecting different disease groups. Interestingly, there is substantial overlap with the degree centrality rankings, which may suggest that in this dense network, the most highly connected diseases also serve as key bridges. However, conditions like Thrombocytopenic disorder and Acute renal failure syndrome appear more prominently in betweenness rankings, indicating their role in connecting distinct disease groups [6].

Table 2: Top 10 diseases by betweenness centrality

Disease	Betweenness
Dyspnea	0.0126
Pleural effusion	0.0126
Electrocardiogram abnormal	0.0125
Anemia	0.0120
Abdominal pain	0.0116
Cardiac arrhythmia	0.0116
Thrombocytopenic disorder	0.0110
Nausea	0.0109
Retention of urine	0.0109
Fever	0.0101

4.4 Louvain Community Detection

The Louvain algorithm identified 5 main communities with a modularity score of 0.3194. Following Blondel et al.'s guidelines, a modularity score above 0.3 indicates that the network has reasonably clear community structure (Table 3) [1]. The communities are fairly balanced in size, with the two largest containing 175 and 168 conditions.

Table 3: Main community sizes detected by Louvain algorithm

Community	Size (nodes)
1	175
2	168
0	119
3	90
4	63

Each main community was further divided into smaller sub-communities to see more detailed patterns, taking advantage of the hierarchical nature of the Louvain method [1]:

Community 1 (175 nodes, 3 sub-communities): This community has a highly connected core (sub-community 1.1) with conditions like abdominal pain, cardiac arrhythmia, sepsis, and thrombocytopenic disorder. Sub-community 1.0 contains more specific conditions like secondary hypertension and intestinal disorders, while sub-community 1.2 focuses on critical conditions like septic shock and intracranial hemorrhage.

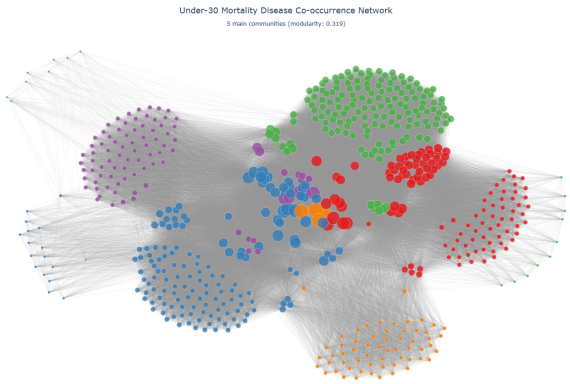

Figure 2: Disease co-occurrence network visualization. Nodes are colored by Louvain community (5 communities detected) and sized by degree centrality. The network exhibits high density (0.3530) with 615 nodes and 66,653 edges. Gray edges represent co-occurrence relationships, with transparency indicating connection density.

Community 2 (168 nodes, 3 sub-communities): The largest sub-community (2.0) contains digestive and metabolic conditions including malnutrition, dehydration, and hypothyroidism. Sub-community 2.2 represents bleeding and post-surgery conditions, while sub-community 2.1 includes liver disease and immunodeficiency.

Community 0 (119 nodes, 2 sub-communities): This community splits into medical device complications and infections (0.0) versus highly-connected critical conditions (0.1) including fever, retention of urine, and cancer-related pain.

Community 3 (90 nodes, 3 sub-communities): Contains brain and psychiatric conditions, organ failures, and metabolic problems. Sub-community 3.1 includes critical care conditions like acute renal failure and acidosis.

Community 4 (63 nodes, 2 sub-communities): The smallest main community includes heart structure problems and birth defects. Notably, sub-community 4.1 has only 5 nodes but extremely high centrality scores—these are the most universally co-occurring conditions in the entire network (Dyspnea, Pleural effusion, Electrocardiogram abnormal, Anemia, Scoliosis).

4.5 Visualizations

4.5.1 Network Topology. Figure 2 shows the complete disease co-occurrence network. Nodes are colored by which community they belong to and sized by their degree centrality (bigger nodes = more connections). We can see that nodes of the same color tend to cluster together, which confirms the community structure detected by the Louvain algorithm. The largest nodes (like Dyspnea and Pleural effusion) are positioned near the center, which makes sense given their role as highly connected hubs [6].

4.5.2 Degree Distribution. Figure 3 shows how many connections each disease has. The left panel shows a histogram with two peaks—one around 150 connections and another around 250 connections. This

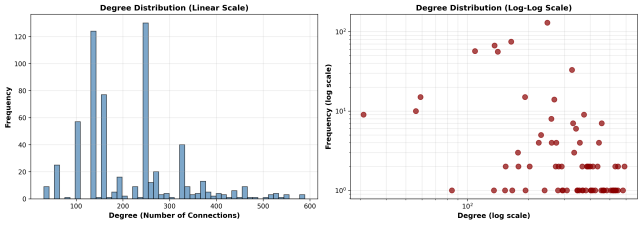

Figure 3: Degree distribution of the disease co-occurrence network. Left: Linear scale histogram showing two peaks around 150 and 250 connections. Right: Log-log scale plot showing the network does not follow a power-law distribution.

tells us there are two groups of diseases: moderately connected ones and highly connected ones. The right panel uses a log-log scale to check if the network follows a "power law" pattern. Our network doesn't show this pattern, suggesting relatively even connectivity across diseases rather than the scale-free topology seen in some biological networks [3].

4.5.3 Community Size Distribution. Figure 4 shows how many diseases are in each of the 5 communities. Communities 1 and 2 are about the same size (175 and 168 nodes), making them the two largest groups. The other communities get progressively smaller, with Community 4 being the smallest at 63 nodes.

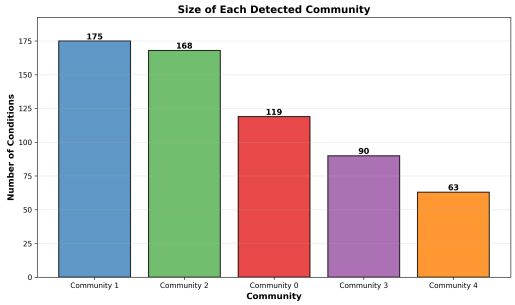

Figure 4: Size distribution of the 5 main communities detected by Louvain algorithm. Bar colors correspond to community colors in the network visualization.

4.5.4 Centrality Relationship. Figure 5 shows how degree centrality and betweenness centrality relate to each other. The plot shows a clear positive correlation—diseases with high degree centrality also tend to have high betweenness centrality. This pattern tells us that in our dense network, the most connected diseases naturally end up being bridges between different groups of diseases, consistent with findings in other dense healthcare networks [9].

5 Analysis and Interpretation

5.1 Comparing Degree vs Betweenness

The strong correlation between degree and betweenness centrality (Figure 5) shows that in our dense network, the diseases that connect

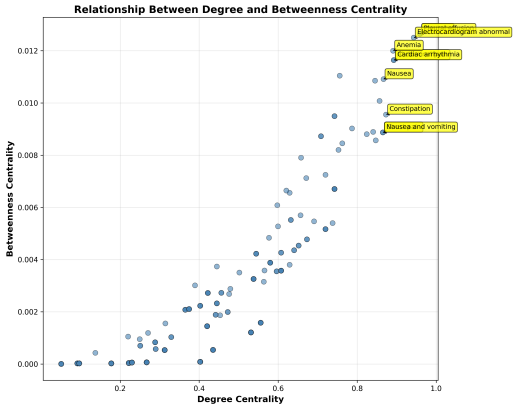

Figure 5: Relationship between degree and betweenness centrality. The strong positive correlation indicates that highly connected diseases also serve as bridges. Top 10 diseases by degree centrality are annotated.

to many other diseases also serve as bridges between different groups. This is consistent with patterns observed in other dense comorbidity networks [9], but differs from sparser networks where hub nodes (highly connected) and bridge nodes might be separate [6]. The overlap tells us that conditions like dyspnea, anemia, and heart problems are both very common and serve as connections between different types of diseases.

5.2 Clinical Patterns

The community structure shows some interpretable groupings, similar to patterns observed in other disease network studies [2, 5]:

- **Sudden vs Long-term:** Communities seem to separate sudden, critical conditions (Community 1) from longer-term nutritional and metabolic problems (Community 2).
- **Widespread vs Specific:** Community 4's high-centrality conditions (dyspnea, pleural effusion, anemia) represent symptoms that affect multiple body systems rather than being specific to one area.
- **Hospital-related:** Community 0 includes many medical device complications and hospital-acquired infections, suggesting that healthcare-related conditions play a role in mortality for this age group.
- **Connecting Conditions:** Some conditions like thrombocytopenic disorder and acute renal failure rank high in betweenness, likely because they connect primary diseases to other complications, serving as bridging conditions in disease progression. [6]

5.3 Unexpected Findings

The very high centrality of breathing and heart symptoms (dyspnea, pleural effusion, electrocardiogram abnormal) suggests these might be common end-stage symptoms rather than starting conditions. It's also surprising to see digestive issues like constipation and nausea in the top 10 most connected conditions—this might reflect side effects from medications.

5.4 Limitations

- **Coding variability:** Differences in how medical diagnoses are recorded or how similar conditions are grouped together can affect the structure of the network [3].
- **Causal inference:** The fact that two conditions appear together does not necessarily mean one causes the other. As noted by Li et al., pairwise co-occurrence methods may recover indirect associations rather than direct causal relationships [6]. Factors like hospital practices or coding habits might create connections that are not truly causal.
- **Community detection sensitivity:** The Louvain algorithm's results depend on the chosen resolution parameter [1]. Using other methods, such as hierarchical clustering, might produce different community structures.
- **Sample selection:** Focusing only on deaths under age 30 limits how well the results apply to broader or non-fatal comorbidity patterns [2].
- **Temporal information:** The network does not include time-related data, so it cannot show whether certain conditions occurred before or after others, limiting our ability to understand disease progression pathways [2].

6 Conclusion and Future Work

This project identifies the most connected diseases and groupings in the mortality network for patients under 30. The diseases with highest centrality scores (difficulty breathing, anemia, heart problems) appear frequently across many patients, while the five communities we found show natural groupings of conditions that tend to occur together.

Future work could include:

- **Adding time information:** Include when each disease was diagnosed to see the order diseases appear and build networks showing progression over time, as demonstrated by Dervic et al. in their multilayer comorbidity network approach [2]
- **Analyzing different subgroups:** Split the data by factors like age ranges, gender, or main cause of death to see if network patterns differ between groups, following approaches used in stratified disease network studies [9]
- **Checking against existing classifications:** Compare our detected communities to standard disease category systems to validate if our groupings make medical sense [6]
- **Predicting with Machine Learning:** Use the network features we calculated (like centrality scores and community membership) as inputs to machine learning models that predict patient outcomes
- **Comparing different algorithms:** Run other community detection methods to see if they find similar patterns or reveal different insights, as suggested by Fotouhi et al.'s comparative analysis [3]

Acknowledgments

Thanks to the course instructor and dataset custodians for access to EHRSHOT. This work was completed as part of the Network Analysis / ML course project 1.

References

[1] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, 2008.
[2] E. Dervic, J. Sorger, W. Gall, A. Rauber, M. Kuhn, M. Pohl, and S. Thurner, "Unraveling cradle-to-grave disease trajectories from multilayer comorbidity networks," *npj Digital Medicine*, vol. 7, no. 1, p. 55, 2024.
[3] B. Fotouhi, N. M. Taramsari, M. A. Riolo, and D. L. Buckeridge, "Statistical methods for constructing disease comorbidity networks from longitudinal inpatient data," *Applied Network Science*, vol. 3, no. 1, p. 46, 2018.
[4] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart, "Exploring network structure, dynamics, and function using NetworkX," in *Proceedings of the 7th Python in Science Conference (SciPy2008)*, Gäel Varoquaux, Travis Vaught, and Jarrod Millman (Eds.), Pasadena, CA, USA, pp. 11–15, Aug 2008.
[5] Y. Jiang, S. Ma, B.-C. Shia, and T.-S. Lee, "An epidemiological human disease network derived from disease co-occurrence in Taiwan," *Scientific Reports*, vol. 8, no. 1, p. 4557, 2018.
[6] J. Li, Y. Wu, X. Zhang, and D. Wang, "Comorbidity network analysis using graphical models for electronic health records," *Frontiers in Big Data*, vol. 6, p. 846202, 2023.
[7] W. McKinney and others, "Data structures for statistical computing in Python," in *Proceedings of the 9th Python in Science Conference*, vol. 445, pp. 51–56, 2010.
[8] Plotly Technologies Inc., "Collaborative data science," Montréal, QC, 2015. Available: https://plot.ly
[9] Z. Sun, Z. Dong, H. Lv, and X. Wang, "Phenotypic disease network analysis to identify comorbidity patterns in hospitalized patients with ischemic heart disease using large-scale administrative data," *Healthcare*, vol. 10, no. 1, p. 80, 2022.
[10] S. Koskinen, M. Niiranen, and J. Saramäki, "Data-driven comorbidity analysis of 100 common disorders," *Scientific Reports*, vol. 12, no. 1, p. 16685, 2022.
[11] A. B. Jensen, P. L. Moseley, T. Oprea, S. Gebruers, J. G. Gardeux, and S. Brunak, "Temporal disease trajectories condensed from population-wide registry data covering 6.2 million patients," *Nature Communications*, vol. 5, no. 1, p. 4022, 2014.

A Representative Python Code Snippet

Below is a concise but complete snippet that mirrors the pipeline used to build the graph and compute centrality and communities.

```
import pandas as pd
import networkx as nx
from networkx.algorithms.community import louvain_communities
import plotly.graph_objects as go
import numpy as np

# Load CSV
df = pd.read_csv('edges.csv')

# Build weighted undirected graph
G = nx.Graph()
edges = [(row['concept_A'], row['concept_B'],
          row['n_people'])
         for _, row in df.iterrows()]
G.add_weighted_edges_from(edges)

# Centralities
deg_cent = nx.degree_centrality(G)
betw_cent = nx.betweenness_centrality(G,
                                      weight='weight')

# Louvain community detection with resolution optimization
best_R = 0.0
best_mod = -1.0
for R in np.arange(0.0, 3.1, 0.1):
    comms = louvain_communities(G, weight='weight',
                                seed=42, resolution=R)
    mod = nx.algorithms.community.modularity(
```

```
        G, comms, weight='weight')
    if mod > best_mod:
        best_mod = mod
        best_R = R

communities = louvain_communities(G, weight='weight',
                                  seed=42,
                                  resolution=best_R)

# Create community map
community_map = {}
for idx, comm in enumerate(communities):
    for n in comm:
        community_map[n] = idx

nx.set_node_attributes(G, deg_cent,
                       'degree_centrality')
nx.set_node_attributes(G, betw_cent,
                       'betweenness_centrality')
nx.set_node_attributes(G, community_map,
                       'community')

# Layout and visualization
pos = nx.spring_layout(G, seed=42, k=1.0,
                       iterations=50)
```