Data Analytics Group 8

General Relativity Network

Elisa Spinelli, Rahul Shah

Data given:

The Arxiv GR-QC collaboration network dataset is from the e-print arXiv and covers scientific collaborations between authors' papers submitted to the General Relativity and Quantum Cosmology category.

The main goal is to explore and describe the data.

Figure 1.

1	# Collaboration network of Arxiv General Relat...
2	# Nodes: 5242 Edges: 28980
3	# FromNodeId\tToNodeId
4	3466\t937
...	...
28979	10154\t9224
28980	10154\t16830
28981	11113\t21723
28982	11113\t23836
28983	11113\t25050

28984 rows × 1 columns

Description of each block of code:

1. Read csv file

   The first thing we did was read and visualize the data as shown in *Figure 1*.

2. Saving elements in a Data Structure

   The second step was to extract the nodes and make a list of From nodes and a list of To nodes to facilitate the next step.

3. Plotting the Graph

   We used the `zip()` function on the two lists to build the edges for the undirected graph, that we then plot as shown in *Figure 2*, where each edge typically represents the connection between the i^{th} author and the j^{th} co-author. The resulting graph has a total of **5242** nodes and **14496** edges.

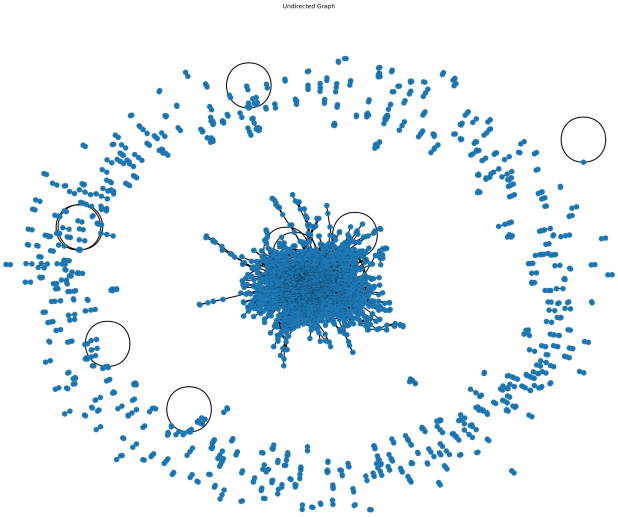


Figure 2.

4. Calculating and Plotting the Adjacency Matrix & **5.** Calculating and Plotting the Cliques

From the graph we calculated the adjacency matrix that we then plotted, *Figure 3*, to visualize how sparsely the graph is connected overall.

Additionally, we determined and plotted the cliques, *Figure 4*, which are sub-graphs of the original graph comprising nodes that are all mutually adjacent. The analysis revealed a total of **3906** cliques.

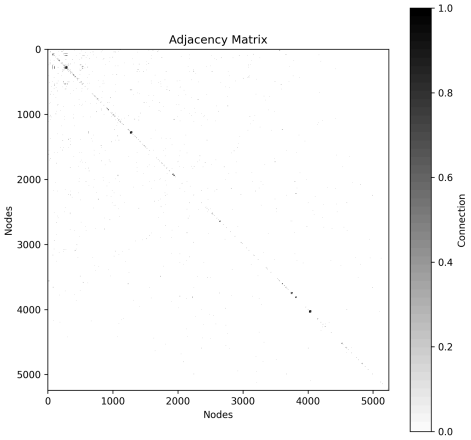


Figure 3.

Figure 4.

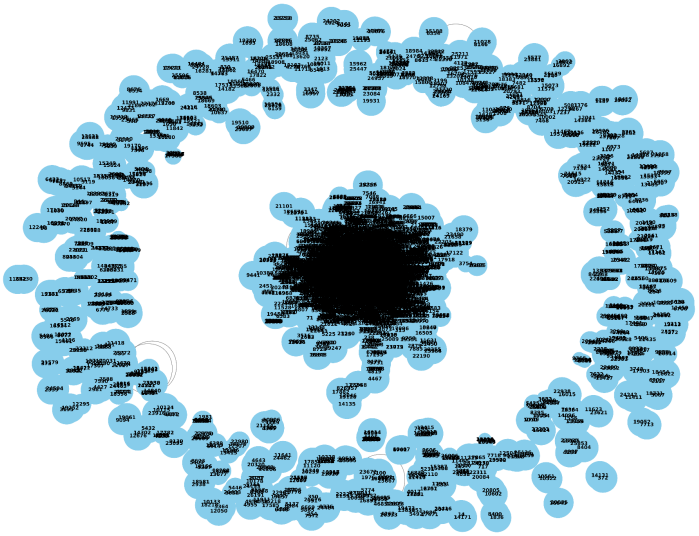
6. Calculating the triangles

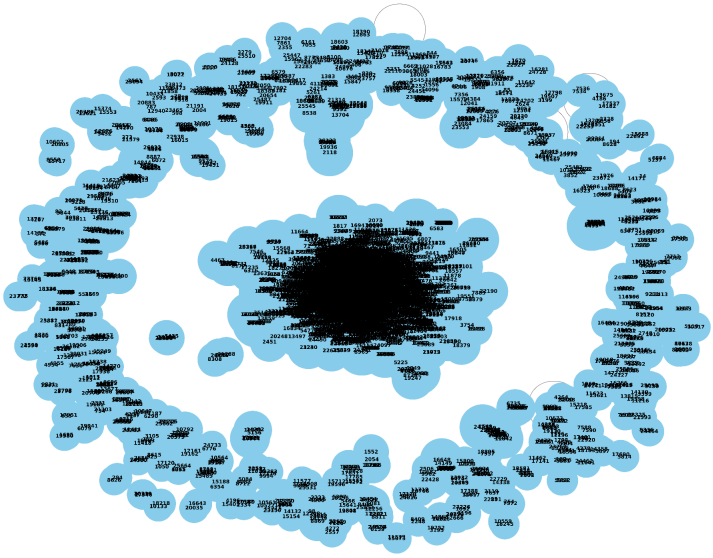
Successively we proceeded by counting the number of triangles associated with each node reflecting triadic closure, the tendency for nodes that share a connection with a common neighbor to form connections themselves. We later visualized this at point 12.

7. Calculating the Degree of Centrality and Plotting the Graph enlarging the authors that have most centrality & **8.** Calculating the PageRank and Plotting the Graph enlarging the authors with highest rank

Afterwards we calculated the Degree of Centrality and the PageRank. The Degree of Centrality measures the importance of a node based on the number of edges it has, while the PageRank considers not only immediate connections but also the global

structure of the network for ranking nodes. Node sizes in the visualizations are





proportional to their Centrality in *Figure 5* and to their PageRank in *Figure 6*.

Figure 5. *Figure 6.*

9. Storing the data in a dictionary & **10.** Storing only the numbers of Coauthors for each Author

We then organized the data into two structures: a dictionary mapping authors to arrays of their co-authors, and another dictionary named `mapped_keys_values` mapping authors to the number of their co-authors. This allowed us to efficiently identify the most prolific author, labeled as **21012**, with **81** co-authors.

11. Plot the Subgraph of the Most Prolific Author

To better visualize the inner part of the main graph (*Figure 1*), we opted to plot a subgraph (*Figure 7*), originating from the most prolific author, using a Breadth-First Search (BFS) strategy. This approach iterates over nodes in a breadth-first manner, exploring nodes up to a specified maximum distance from the starting node. After experimenting with different maximum distances, we decided that a maximum distance of 2 returned the best visualization, as increasing the distance resulted in an excessive number of nodes to be able to have a clear visualization.

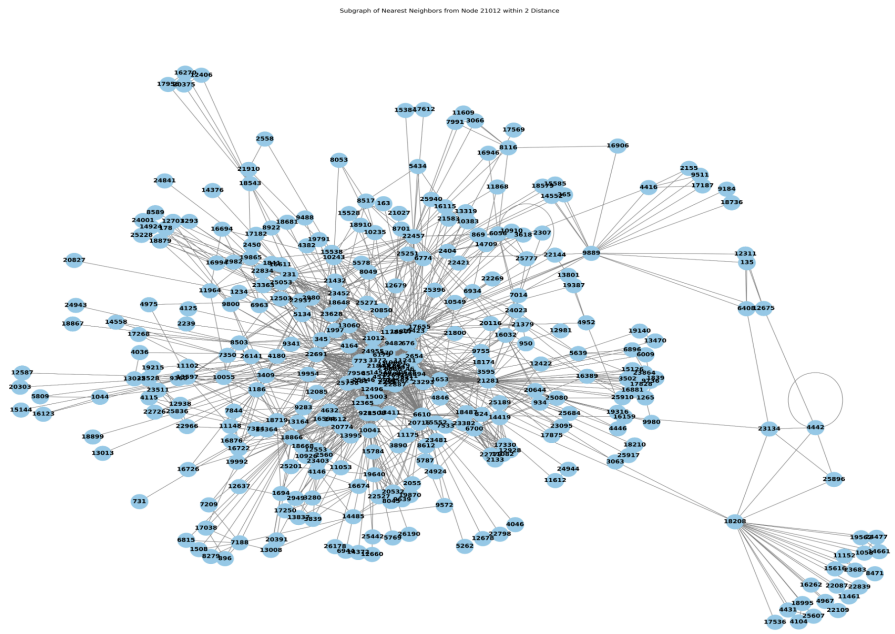


Figure 7.

12. Visualize the Triangles

At this point, we decided to visualize a graph highlighting only the triangles, as we now had the capability to select a suitable subgraph for their visualization, based on the previous steps. Given the complexity of visualizing triangles in a graph like *Figure 7*, we opted to focus on a less prolific author, specifically author **3466**, and employed BFS to extract a subgraph. Then, we computed the triangles within that subgraph and plotted the result, as shown in *Figure 8*.

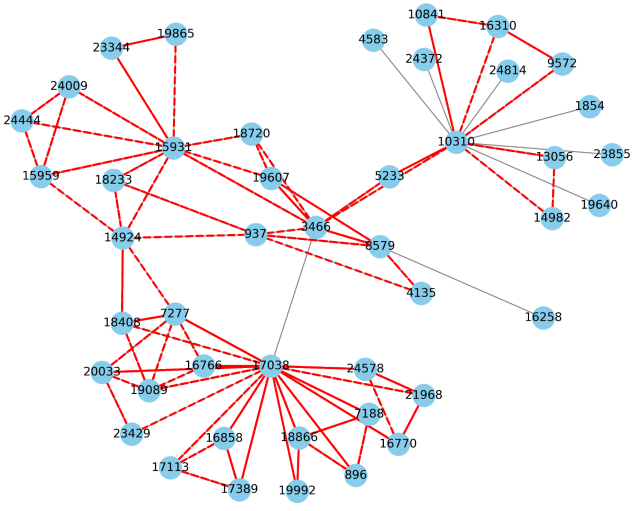
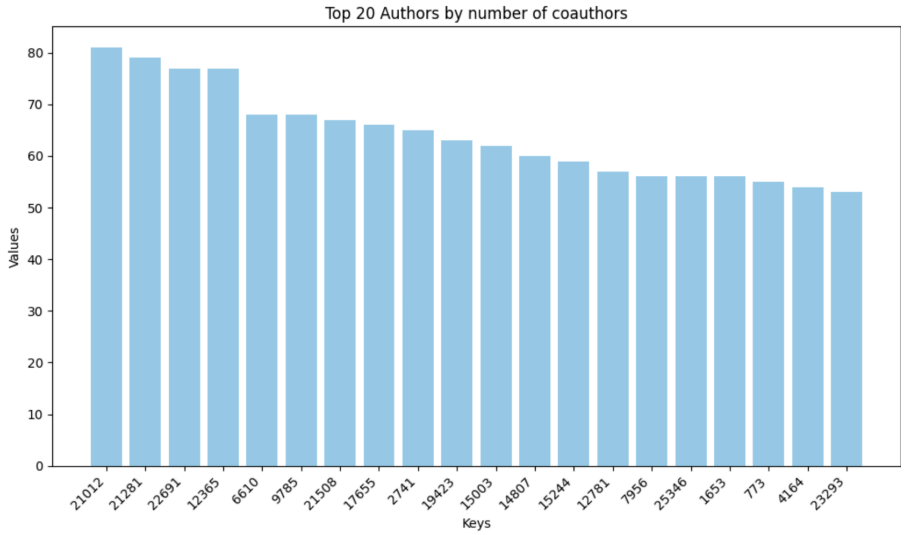


Figure 8.

13. Plot 20 Most Prolific Authors

Utilizing the `mapped_keys_values` data structure, we could easily plot the best and worst n authors. However, through experimentation plotting different graphs, we

found that visualizing the top authors was more interesting than visualizing the worst ones, so we plotted a bar graph to show the 20 most prolific authors, as



shown in *Figure 9*.

Figure 9.

14. Mapping Authors to Bins based on Co-Authoring and plotting them

Thanks to the previous step, we observed that many authors have only a few co-authors. To better illustrate the disparity in author prolificity, we thought of a method to categorize authors into bins based on the number of their co-authors. With the most prolific author having 81 co-authors, we opted for 8 bins, spanning from 0 to 10, 10 to 20, and so forth. However, after plotting it, we realized that a bar graph wasn't able to accurately visualize the differences between bin dimensions. So, we used a scatter graph, as depicted in *Figure 10*, to visualize it better.