Social Network Analysis in the 1492 Papal Election Simulation

Nolan Shaffer
HIST 12203 Italian Renaissance: Petrarch, Machiavelli, and the Wars of Popes and Kings
Professor Ada Palmer
28 May 2024

Methods

This project was divided into four phases: parsing, visualization, research, and analysis.

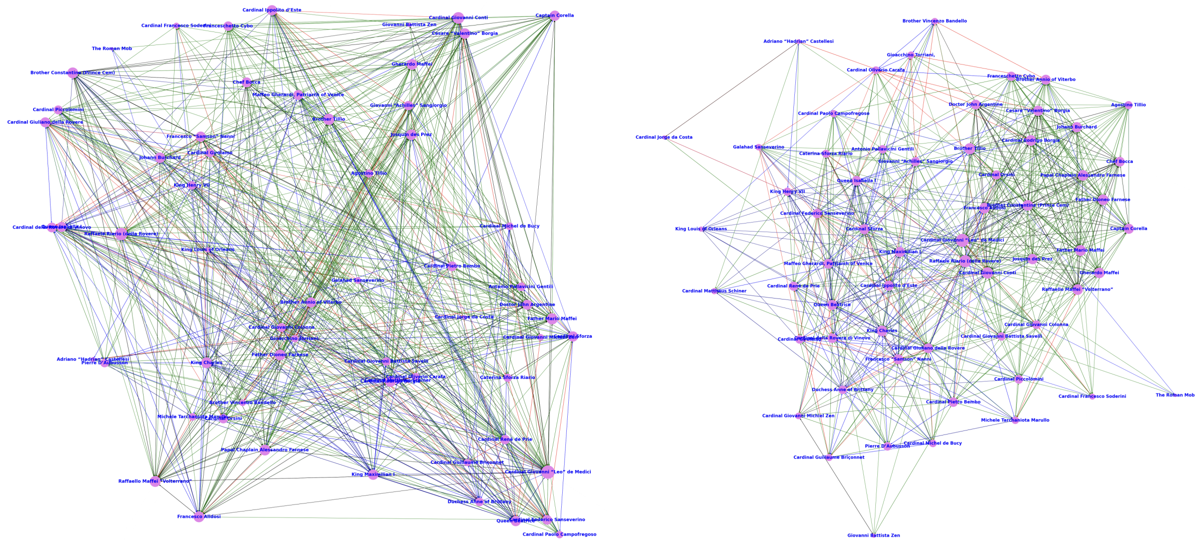
The first phase, parsing, (source code in mymain.py) involved taking in as input the 63 character sheets and outputting the connections between characters. Connections were sorted as one of "close ally", "ally", "allies", "confidant", "client", "correspondent", "patron", "boss", and "colleague", and needed to be between any two of the 63 characters. This meant that connections to NPCs had to be filtered out.

There were a number of obstacles in this phase, which turned out to be significantly more difficult than I anticipated. This largely came from the fact that the character sheets were not standardized–in any given allies table, which was what I was parsing, character names were frequently abbreviated, misspelled, put in a list, shortened, or replaced entirely with phrases like “all crowned heads of Europe”. This was a problem because I needed to find a match between a listed ally on a given character sheet and one of the 62 other characters. I also knew that manually going through all 63 character sheets was not an option, especially given the ambiguity of some of the character names and the large amount of connections that some characters had.

My solution to this problem was to compare a potential match based on the number of shared words in the two names. This required filtering of the words "of", "the", "in", "and", "de", "da", a", and "della" which appeared too frequently and were causing incorrect matches. My process involved iteratively running my matching algorithm, looking for errors in matches, and correcting them by adding more filters and adjusting the shared word threshold. Then, I searched through the large list of unfound matches and manually added filters to replace typos or add more context to an entry so that the algorithm would find its match. Through sampling, I estimate that after these filters, my parsing has an error rate of less than 1%.

The second phase, visualization, involved using the NetworkX Python library to create a graph from the parsed data. This was also a more challenging problem than I realized, since random positioning of characters in a visualization graph makes it difficult to visualize data and relationships. After experimenting with a number of different visualization algorithms, I settled on the Kamada-Kawai algorithm, which uses iterative optimization to minimize the ideal position of nodes based on graph distances (Kobourov 2012) (See Fig. 1).

Figure 1



A random layout (left) versus a Kamada-Kawai algorithm based layout (right). Cardinal Della Porta and Duke Louis of Orleans not pictured.

For the third phase, I conducted a review of a variety of literature on network theory. I searched for techniques to analyze my constructed graph and learned about a variety of methods for making inferences about a network given a graph. In the **Results** section, I'll connect the literature I used with my analysis.

The final phase, analysis, involved implementing the algorithms I found in the research phase (source code in grapher.ipynb). Two important modifications were made to the graph for analysis–both Cardinal Della Porta and Duke Louis of Orleans and all of their relationships were removed. For Della Porta, this was done because the Cardinal is allied with every other character, and hence made many of the analyses trivial. Duke Louis of Orleans was removed because he had only one ally and according to his character sheet, was only meant to be played in the event that King Charles VIII died. For almost all of my analysis, I also only looked at relationships that were either "close ally", "ally", "allies", "confidant", or "patron", since these are the most meaningful and significant. Due to the small size of the network (61 nodes, 1018 edges), complex and computationally expensive algorithms that would be otherwise unusable on a large graph were able to run in a trivial amount of time, and so algorithm time complexity was of no concern.

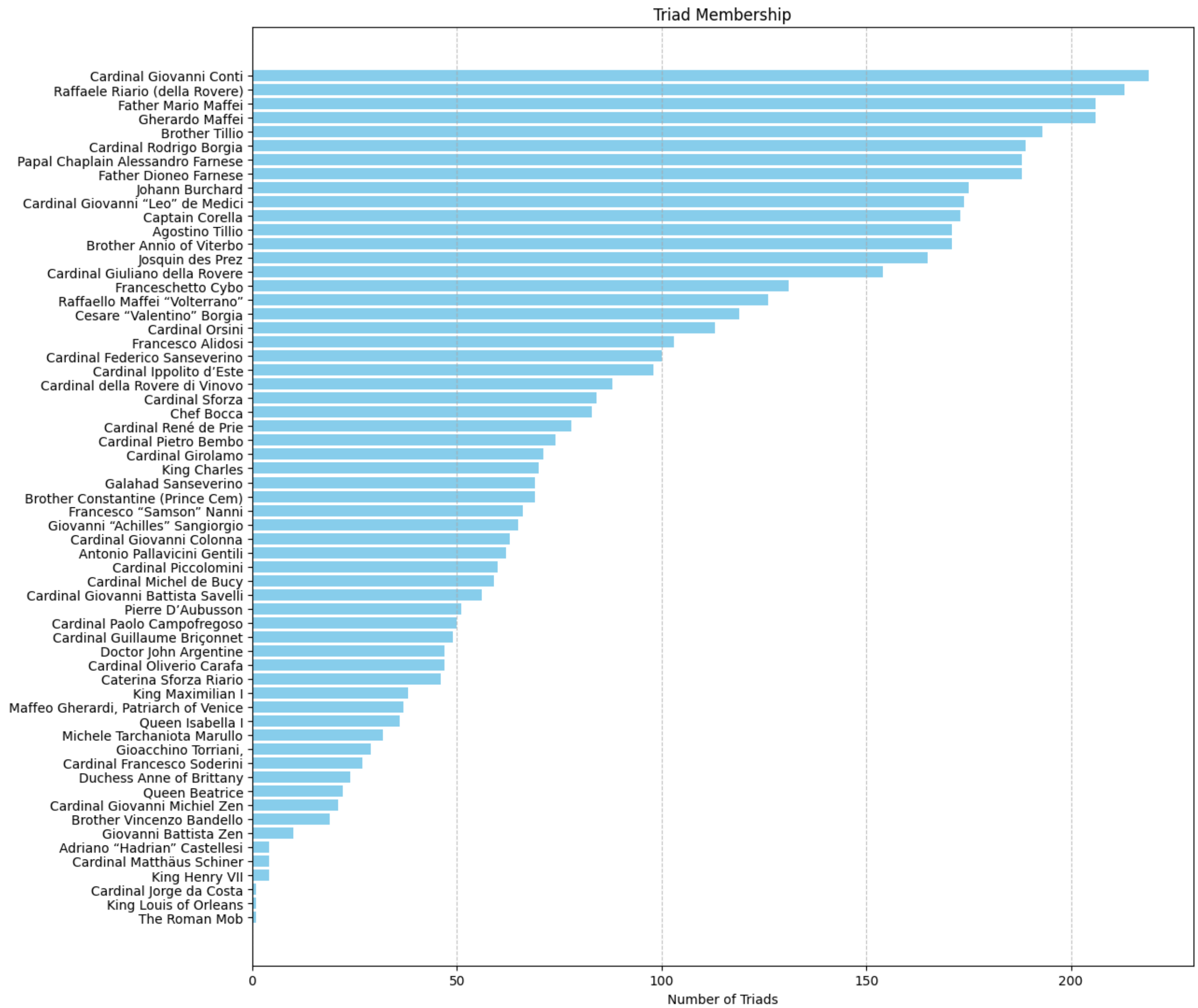
Results

See **Additional Figures** section for a complete set of visualization graphs. All code can be found on the associated GitHub repository.

My analysis fell into two categories: strongest and most influential individuals, and subgroups and communities.

Determining the influential individuals in a network can take a variety of forms. One strategy was to use triangle-based cohesive subgraphs. The idea behind this is that triads, or 3-way relationships, have long been identified as significant and influential in social networks. This dates back to George Simmel, who theorized in 1902 that triads were a fundamental unit of sociological analysis (Simmel 1902). Hence using an algorithm to determine the number of triads that a given character is a part of can illuminate patterns in a network, identifying individuals by the number of stable relationship groups they are in (fig. 2).

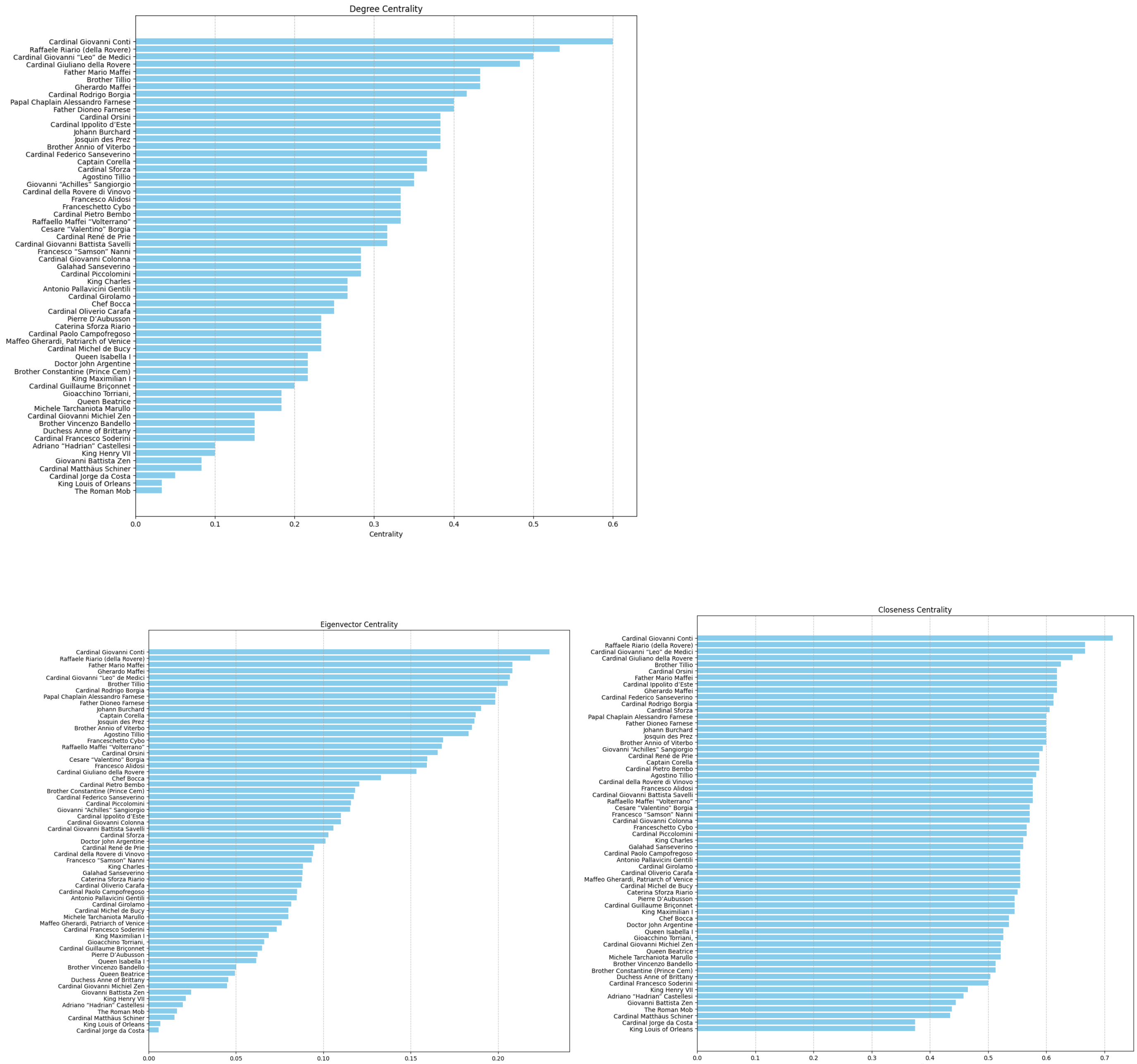
Figure 2



Sorting characters by the amount of triads they are in

Another method to consider in character influence is centrality. Centrality has been a major focus of network analysis for decades, and it indicates those who occupy critical positions in a network (Valente et al. 2008). There are a variety of measures that have been proposed, but I focused on three of them: eigenvector centrality, degree centrality, and closeness centrality. These three algorithms are distinctly different (Bonacich 2007) and are optimal in different types of networks (Perez and Germon 2016), making them good choices for a uniquely small and well connected graph such as ours (fig 3.).

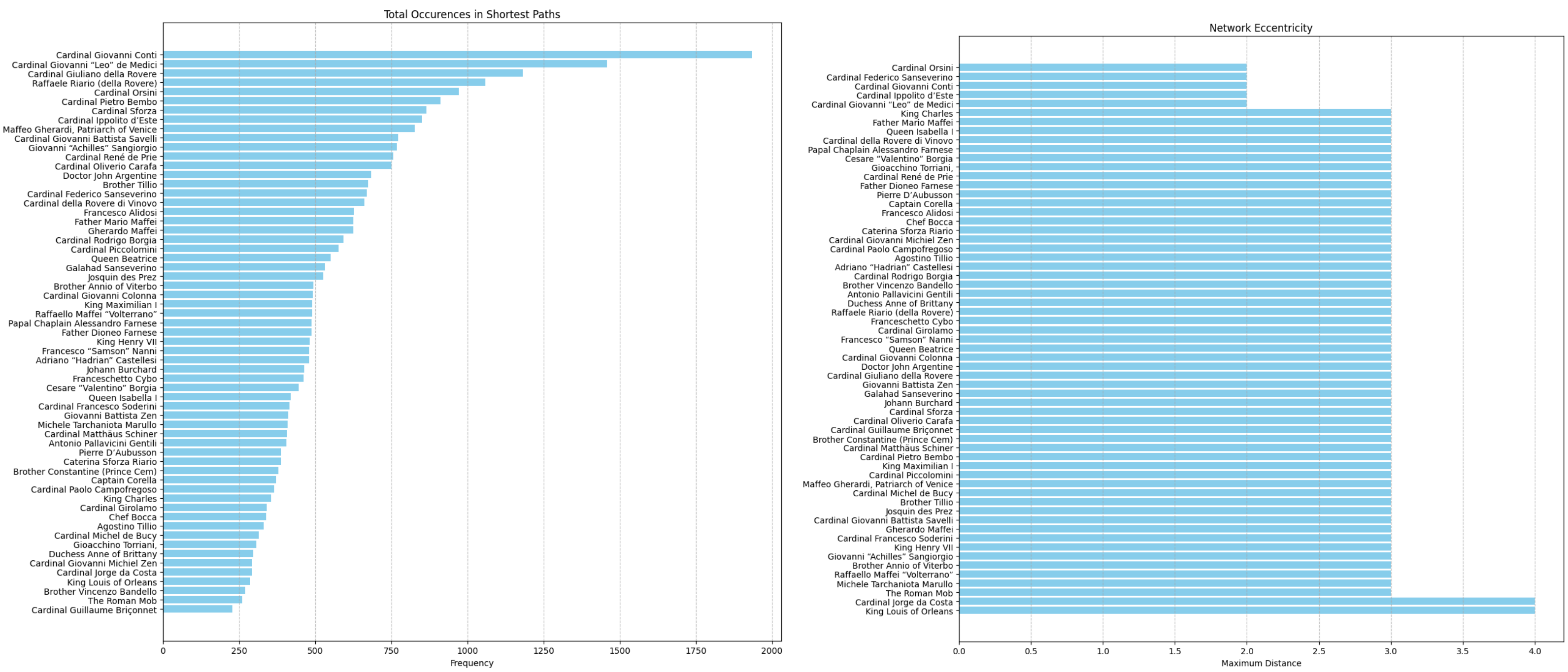
Figure 3



Degree centrality, eigenvector centrality, closeness centrality, counter-clockwise from top. Results were largely similar but varied slightly on an individual level.

Two more metrics were used to analyze individual influence (fig. 4). Both of these were picked specifically with the nature of our network in mind. The first, shortest path occurrence, tracks the frequency of each character in a shortest path. A shortest path between two characters is defined as the least number of character hops needed to get from one character to another. Tracing the number of shortest paths that an individual is in can therefore yield insight into the influence of that character . The more shortest paths a character is in, the more "essential" they are to the efficient spread of information, and the more connected they are. The second, network eccentricity, uses the above idea of centrality to determine the maximum distance of a character to other characters (Hage and Harary 1995). This reveals the most and least well connected characters in our network.

Figure 4



Shortest path frequency (left) and network eccentricity (right).

Communities play an important role in network analysis, and given the importance of relationships and factions in our network, they are especially relevant. A community is a subset of nodes (in this case characters) whose connections are denser than connections with the rest of the network (Radicchi et al. 2004). This can reveal a helpful analysis of relationships and perhaps detect latent factions that are inherent in the structure of our network.

Two algorithms were used to detect communities in our network. The first was k-clique percolation. Cliques are complete subgraphs, meaning that in a clique, every character has a relationship with every other character (Palla et al. 2008). Identifying the maximum clique can therefore identify the largest closely connected faction. We can do even better than this however using percolation. After finding the maximum clique, we can use percolation to find overlapping communities and expand this strongest faction to include adjacent strong factions that are connected. This produces a set of characters who we can think about as being in the strongest community (fig. 5). Note that a community only means being more densely connected than the rest of the graph, and not everyone in the community is allied with one another.

Figure 5

Cardinal Giovanni "Leo" de Medici Gherardo Maffei Franceschetto Cybo Father Mario Maffei Papal Chaplain Alessandro Farnese Agostino Tillio Josquin des Prez Captain Corella Brother Tillio Father Dioneo Farnese Brother Annio of Viterbo Cesare "Valentino" Borgia Cardinal Rodrigo Borgia Johann Burchard Cardinal Giovanni Conti

The strongest community in our network, found using k-clique percolation with a k-value of 15.

The other method I used to identify communities was the Louvain algorithm, one of the most popular algorithms for uncovering community structure in network theory (Smith et al. 2020). The algorithm is optimization-based and able to incorporate weights, a helpful trait that allows us to encode additional data in our analysis. Weights were determined according to relationships, with close allies being weighted as 10, allies 6, patrons and clients 5, confidants 3, clients 2, and correspondents and colleagues 1. The result is a complete list of cuts, or partitions, of our graph, so that every character is in exactly one community (fig. 6).

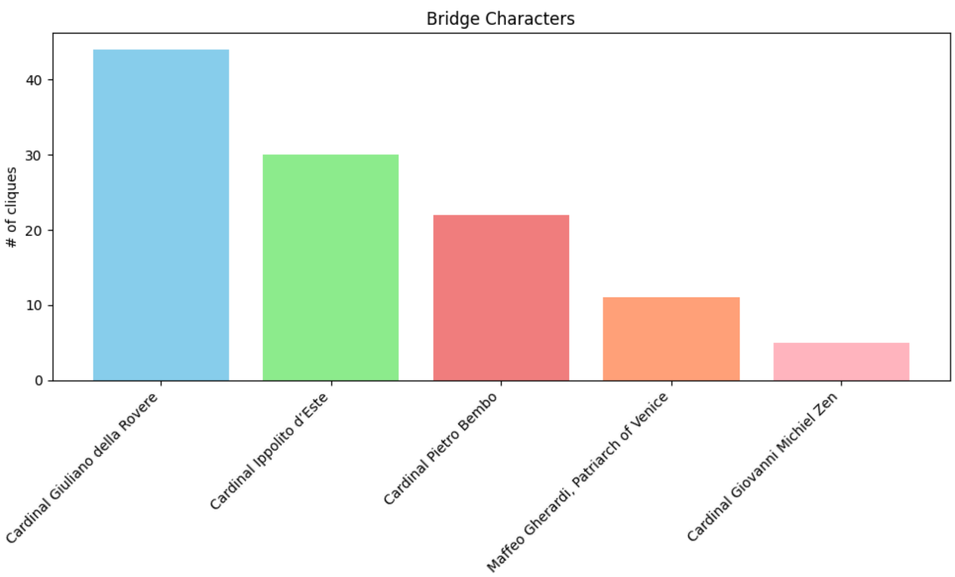
Figure 6

Brother Vincenzo Bandello Queen Beatrice Caterina Sforza Riario Cardinal Sforza Cardinal Paolo Campofregoso Antonio Pallavicini Gentili Cardinal Ippolito d'Este Cardinal Oliverio Carafa Galahad Sanseverino Giovanni "Achilles" Sangiorgio King Maximilian I Gioacchino Torriani, King Louis of Orleans Cardinal Federico Sanseverino Queen Isabella I	Giovanni Battista Zen King Charles Pierre D'Aubusson Cardinal Pietro Bembo King Henry VII Cardinal René de Prie Francesco "Samson" Nanni Cardinal Jorge da Costa Cardinal Michel de Bucy Duchess Anne of Brittany Cardinal Giovanni Michiel Zen Cardinal Girolamo Maffeo Gherardi, Patriarch of Venice Cardinal Giuliano della Rovere Michele Tarchaniota Marullo Cardinal Matthäus Schiner Cardinal della Rovere di Vinovo Adriano "Hadrian" Castellesi Cardinal Guillaume Briçonnet
Raffaele Riario (della Rovere) Father Mario Maffei Josquin des Prez Captain Corella Father Dioneo Farnese Johann Burchard Brother Tillio Raffaello Maffei "Volterrano" Gherardo Maffei Franceschetto Cybo Papal Chaplain Alessandro Farnese Agostino Tillio Chef Bocca Brother Constantine (Prince Cem) Doctor John Argentine Francesco Alidosi Brother Annio of Viterbo Cesare "Valentino" Borgia Cardinal Rodrigo Borgia	Cardinal Giovanni "Leo" de Medici Cardinal Giovanni Battista Savelli Cardinal Giovanni Colonna Cardinal Piccolomini The Roman Mob Cardinal Francesco Soderini Cardinal Orsini Cardinal Giovanni Conti

The four communities found using the Louvain method for community detection.

Finally, I used maximum clique detection to find "bridge" individuals in overlapping cliques. Recall from above that a clique in the context of our network is a group of individuals where every individual has a relationship with each other. Maximum cliques are hence the largest possible subgroups of closely related people in a social network. It is therefore a useful tool in network analysis that corresponds to searching for the most powerful group of people in a network (Belik 2014). In our case, maximum cliques are not very useful for determining large communities due to the clustering of our network (hence why we used percolation above), but they can provide valuable insight into characters who are in multiple cliques. These characters, whom I call bridges, can be thought of as important connectors between closely allied groups (fig. 7).

Figure 7



Cardinal Giuliano della Rovere: 44
Cardinal Ippolito d'Este: 30
Cardinal Pietro Bembo: 22
Maffeo Gherardi, Patriarch of Venice: 11
Cardinal Giovanni Michiel Zen: 5

"Bridge" characters and the number of cliques they're a part of. Tabular data on the right.

References

Belik, Ivan. 2014. “The Analysis of Split Graphs in Social Networks Based on the K-Cardinality Assignment Problem.” SSRN Scholarly Paper. Rochester, NY. https://doi.org/10.2139/ssrn.2405010.

Blondel, Vincent D., Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. “Fast Unfolding of Communities in Large Networks.” *Journal of Statistical Mechanics: Theory and Experiment* 2008 (10): P10008. https://doi.org/10.1088/1742-5468/2008/10/P10008.

Bonacich, Phillip. 2007. “Some Unique Properties of Eigenvector Centrality.” *Social Networks* 29 (4): 555–64. https://doi.org/10.1016/j.socnet.2007.04.002.

Golbeck, Jennifer. 2015. “Chapter 21 - Analyzing Networks.” In *Introduction to Social Media Investigation*, edited by Jennifer Golbeck, 221–35. Boston: Syngress. https://doi.org/10.1016/B978-0-12-801656-5.00021-4.

Hage, Per, and Frank Harary. 1995. “Eccentricity and Centrality in Networks.” *Social Networks* 17 (1): 57–63. https://doi.org/10.1016/0378-8733(94)00248-9.

Kobourov, Stephen G. 2012. “Spring Embedders and Force Directed Graph Drawing Algorithms.” arXiv. https://doi.org/10.48550/arXiv.1201.3011.

McMillan, Cassie, and Diane Felmlee. 2020. “Beyond Dyads and Triads: A Comparison of Tetrads in Twenty Social Networks.” *Social Psychology Quarterly* 83 (4): 383–404. https://doi.org/10.1177/0190272520944151.

Palla, Gergely, Dániel Ábel, Illés J. Farkas, Péter Pollner, Imre Derényi, and Tamás Vicsek. 2008. “K-Clique Percolation and Clustering.” In *Handbook of Large-Scale Random Networks*, edited by Béla Bollobás, Robert Kozma, and Dezső Miklós, 369–408. Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-540-69395-6_9.

Pan, Rusheng, Yunhai Wang, Jiashun Sun, Hongbo Liu, Ying Zhao, Jiazhi Xia, and Wei Chen. 2023. “Simplifying Social Networks via Triangle-Based Cohesive Subgraphs.” *Visual Informatics* 7 (4): 84–94. https://doi.org/10.1016/j.visinf.2023.07.003.

Perez, Charles, and Rony Germon. 2016. “Chapter 7 - Graph Creation and Analysis for Linking Actors: Application to Social Data.” In *Automating Open Source Intelligence*, edited by Robert Layton and Paul A. Watters, 103–29. Boston: Syngress. https://doi.org/10.1016/B978-0-12-802916-9.00007-5.

Radicchi, Filippo, Claudio Castellano, Federico Cecconi, Vittorio Loreto, and Domenico Parisi. 2004a. “Defining and Identifying Communities in Networks.” *Proceedings of the National Academy of Sciences* 101 (9): 2658–63. https://doi.org/10.1073/pnas.0400054101.

———. 2004b. “Defining and Identifying Communities in Networks.” *Proceedings of the National Academy of Sciences of the United States of America* 101 (9): 2658–63. https://doi.org/10.1073/pnas.0400054101.

Sabade, Juyee. 2020. “Community Detection through Clique Percolation Method on the Game of Thrones Universe.” *Medium* (blog). June 8, 2020. https://medium.com/@sabadejuyee21/community-detection-through-clique-percolation-method-on-the-game-of-thrones-universe-b12efb188d99.

Simmel, Georg. 1902. “The Number of Members as Determining the Sociological Form of the Group. I.” *American Journal of Sociology* 8 (1): 1–46. https://doi.org/10.1086/211115.

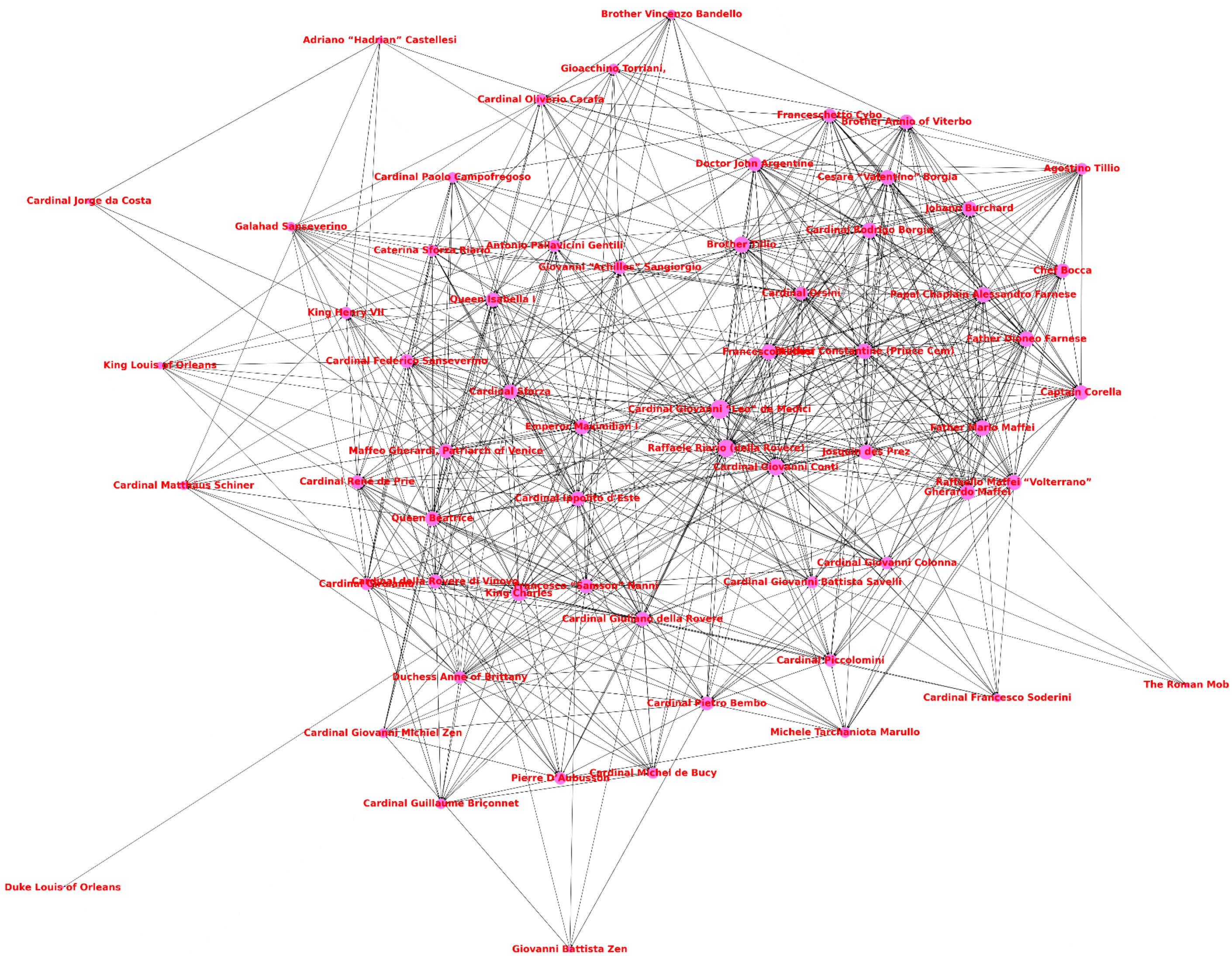
Smith, Natalie R, Paul N Zivich, Leah Frerichs, James Moody, and Allison E Aiello. 2020. “A Guide for Choosing Community Detection Algorithms in Social Network Studies: The Question-Alignment

Approach." *American Journal of Preventive Medicine* 59 (4): 597–605. https://doi.org/10.1016/j.amepre.2020.04.015.

Valente, Thomas W., Kathryn Coronges, Cynthia Lakon, and Elizabeth Costenbader. 2008. "How Correlated Are Network Centrality Measures?" *Connections (Toronto, Ont.)* 28 (1): 16–26. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2875682/.

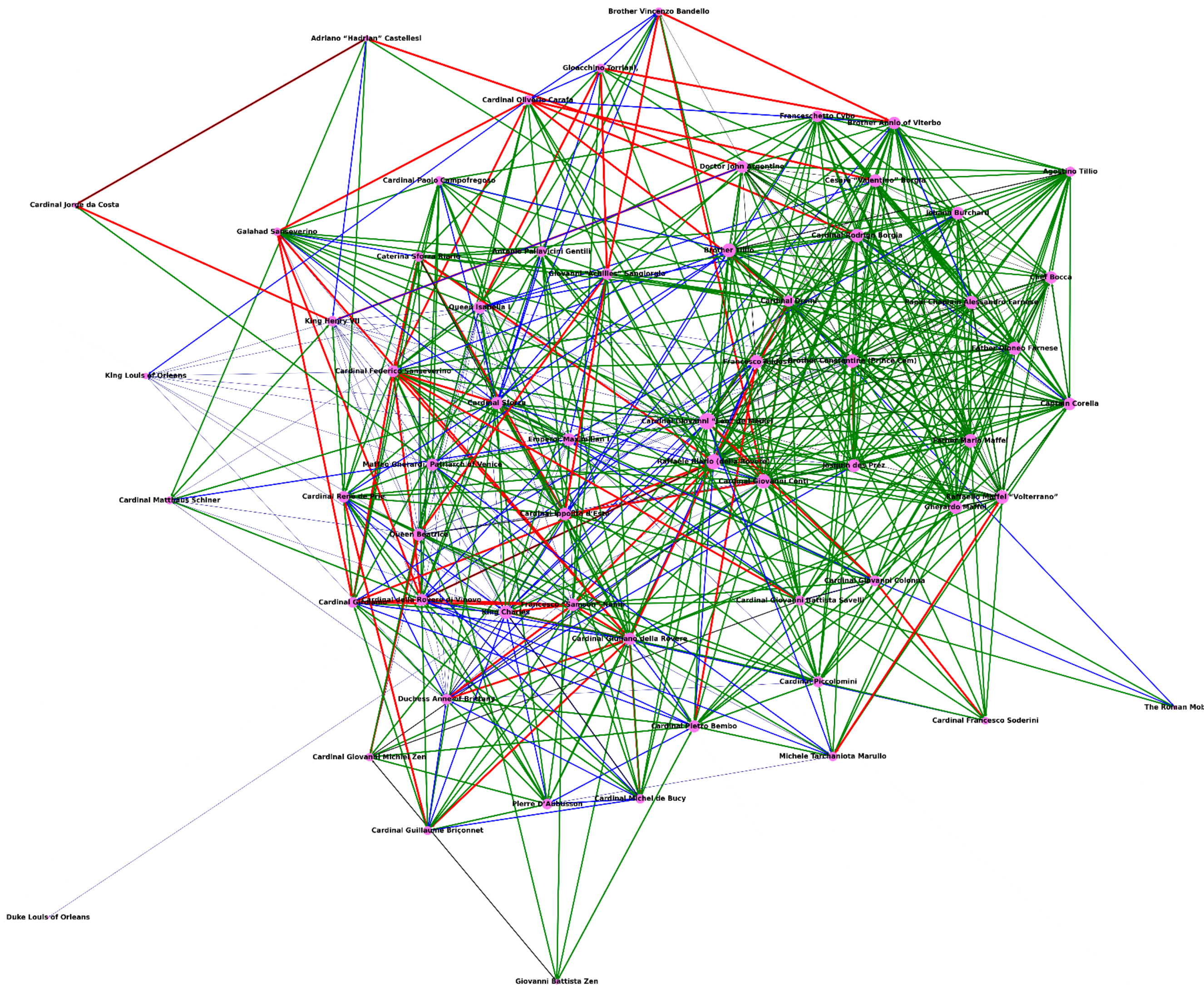
Additional Figures

1A



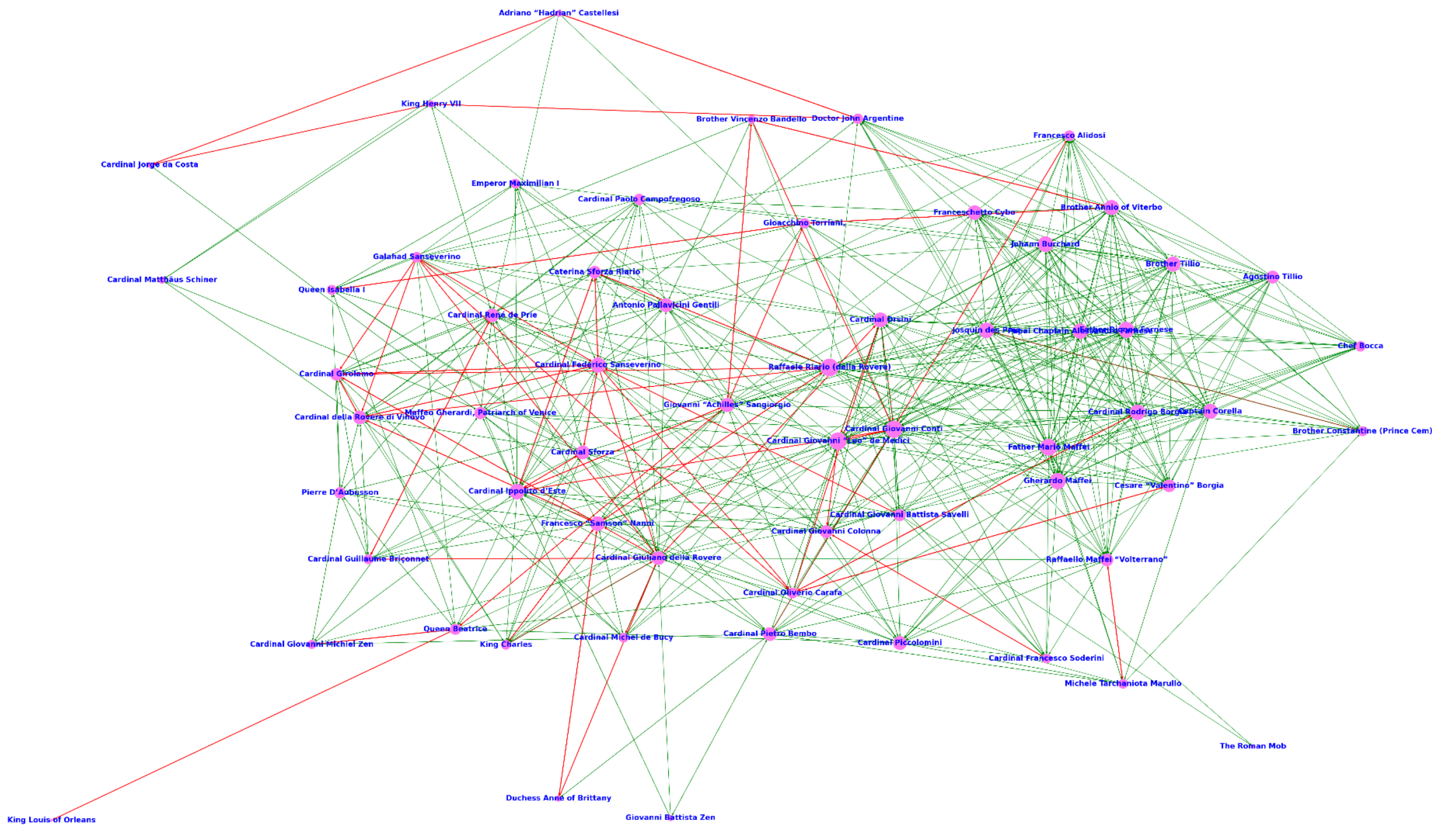
All characters and connections without any processing or relationship distinctions. Cardinal Della Porta and relationships removed.

1B



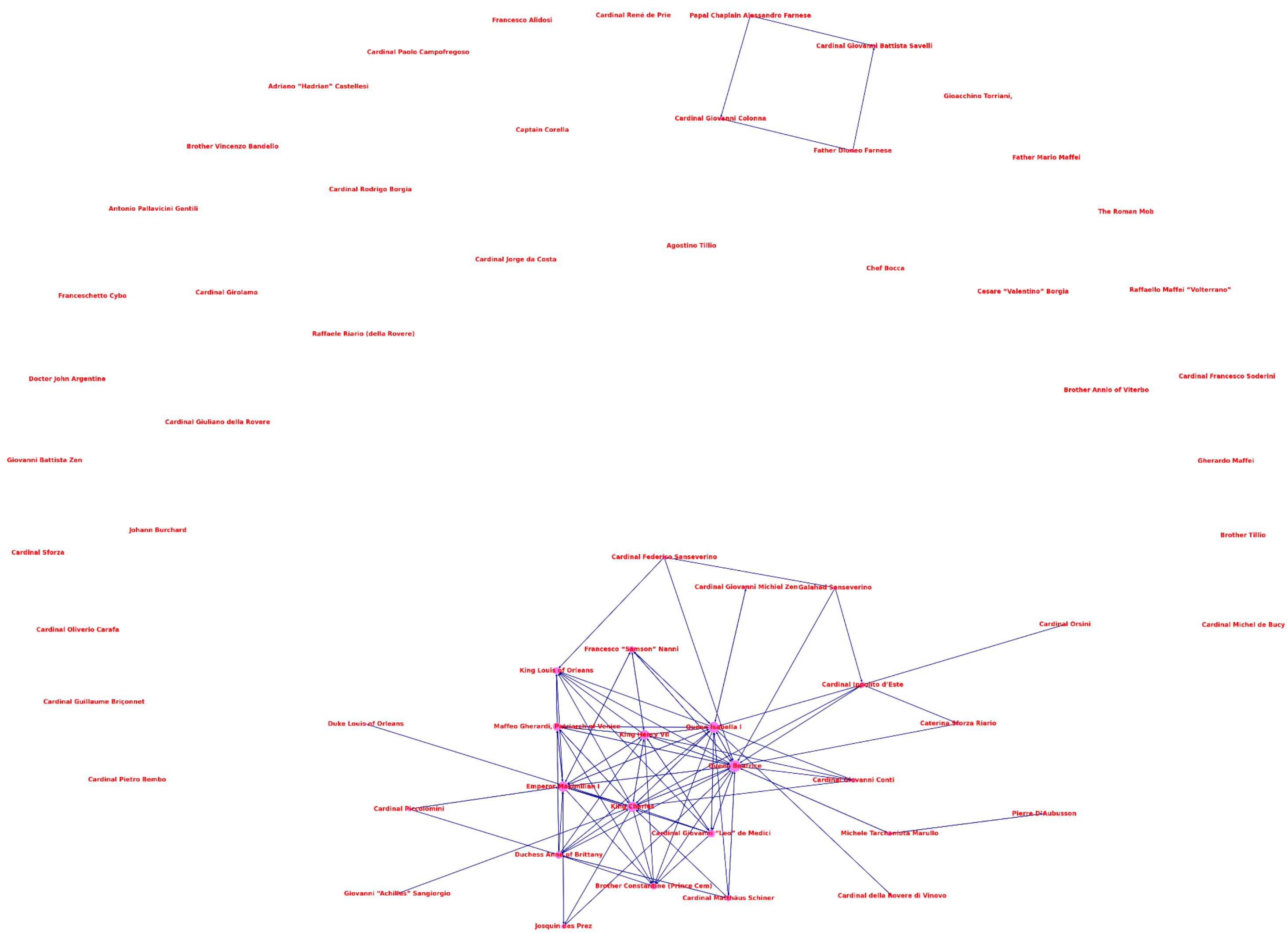
All data, with relationships distinguished by weight and color (Cardinal Della Porta not pictured). Close allies are red, allies green, confidants orange, clients and patrons blue, correspondents maroon, boss green, colleagues black.

2A



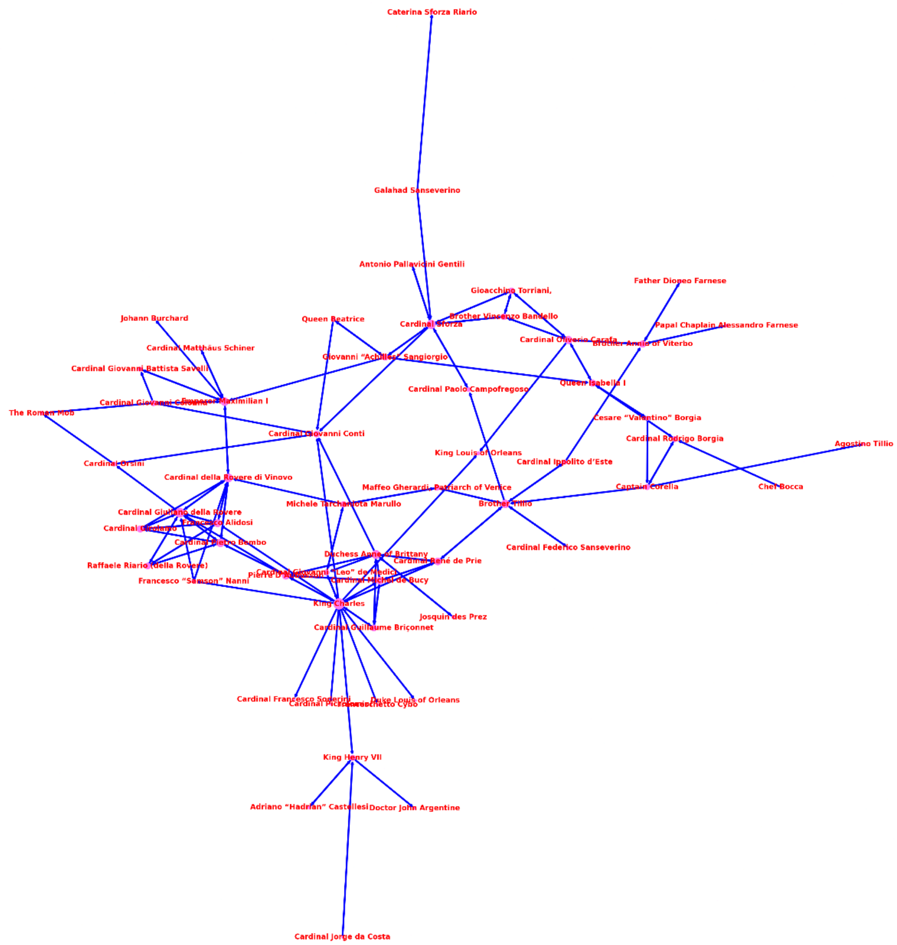
Close allies (red) and allies (green). Cardinal Della Porta and Duke Louis of Orleans not pictured.

2B



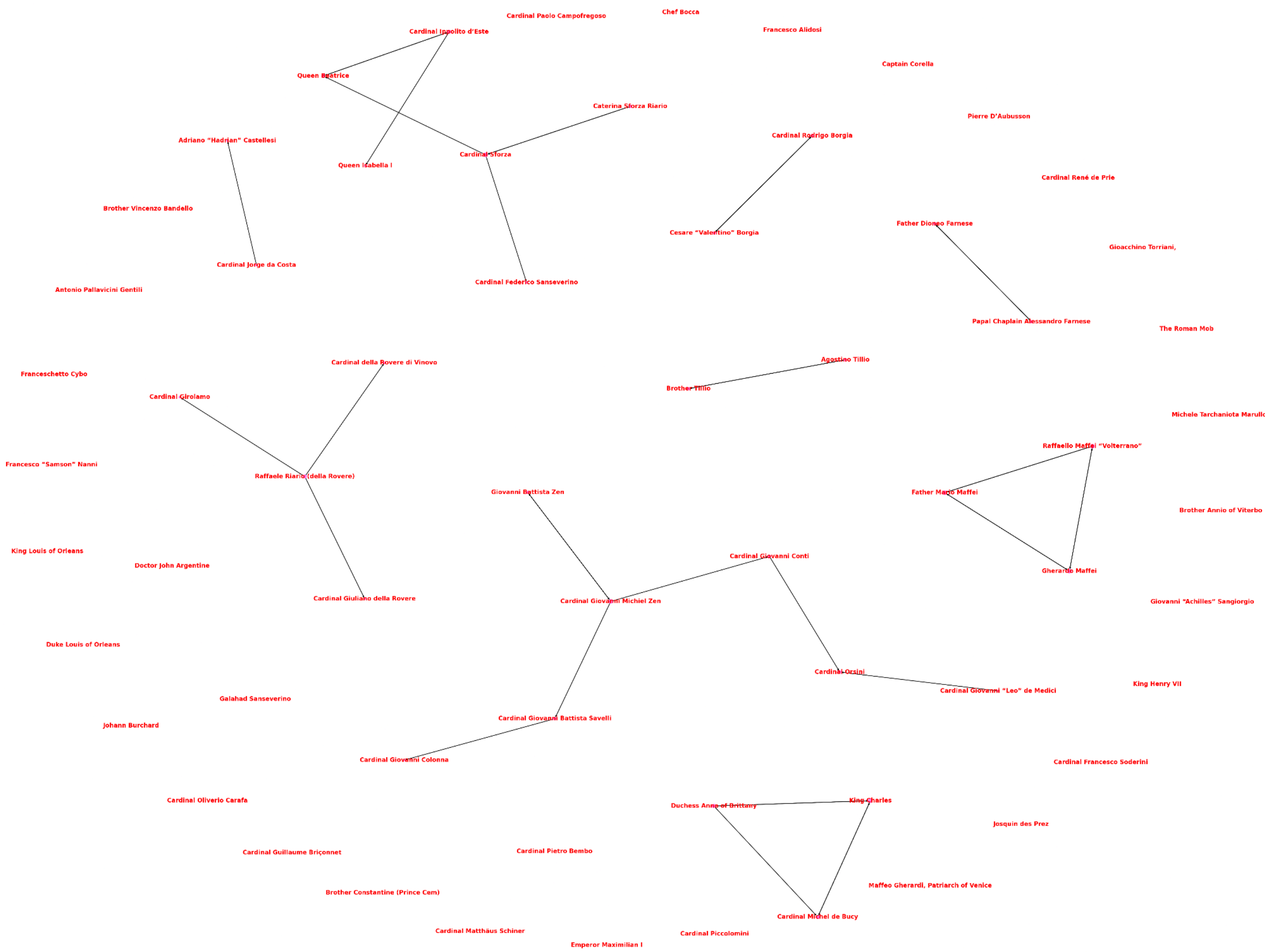
Correspondent network (Cardinal Della Porta not pictured).

2C



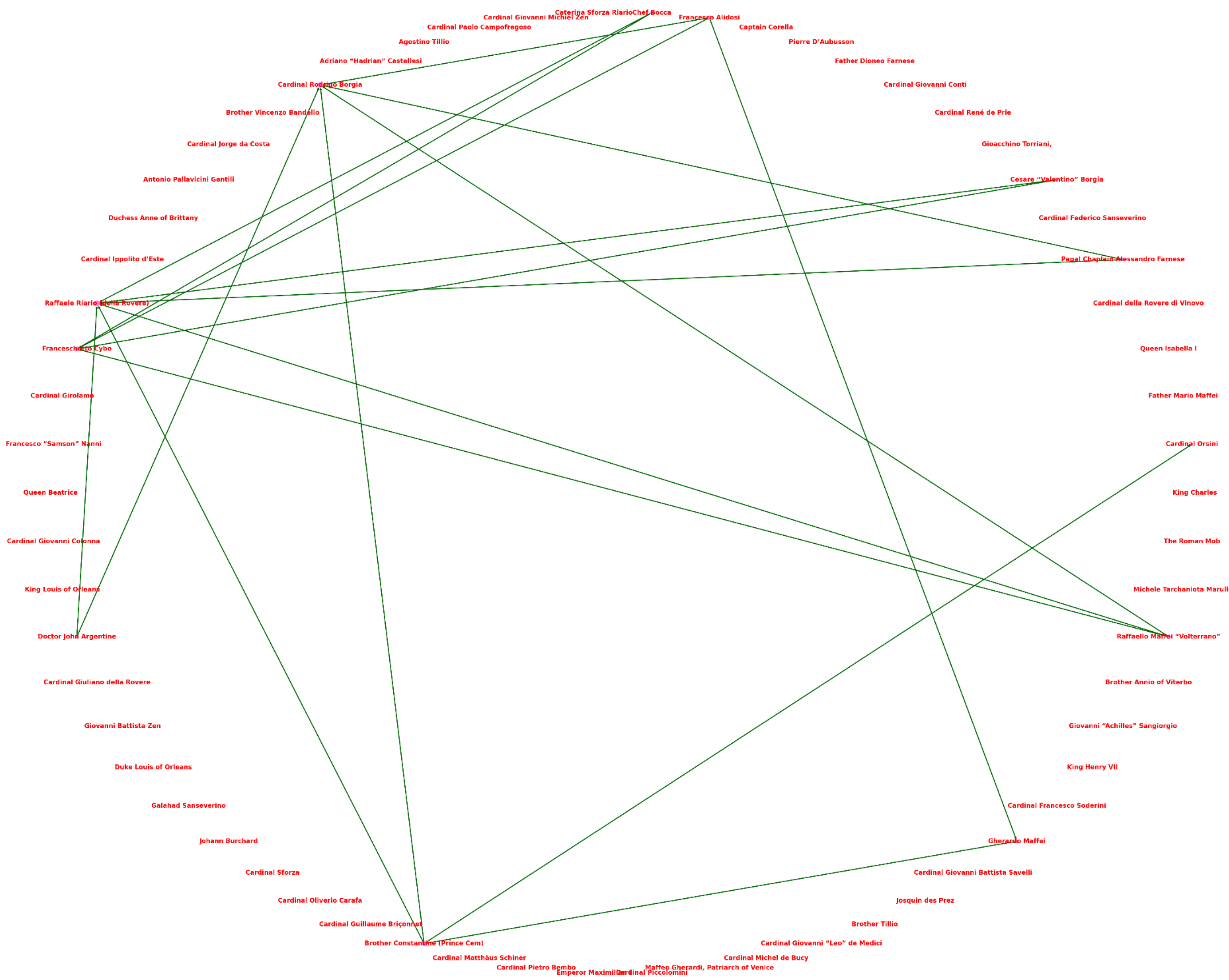
Client-patron network. Individuals not in a client-patron relationship not pictured.

2D



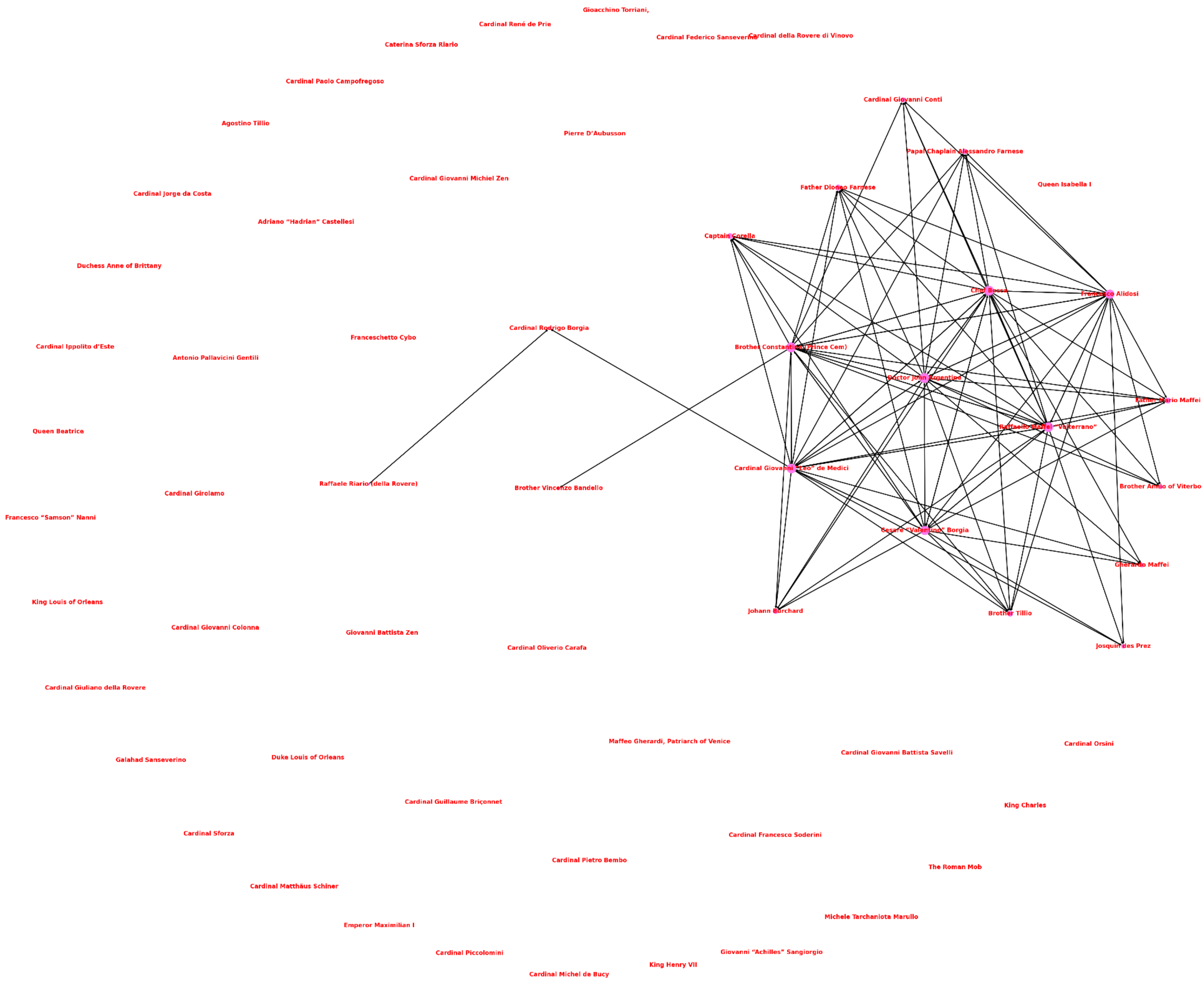
Confidants. Cardinal Della Porta not pictured.

2D



Boss relationships. Cardinal Della Porta not pictured.

2E



Colleagues. Cardinal Della Porta not pictured.