UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

RODRIGO PARANHOS BASTOS

Evolved VizColab: Enhancing the Visualization of Brazilian Academic Collaboration Networks

Work presented in partial fulfillment of the requirements for the degree of Bachelor in Computer Engineering

Advisor: Prof. Dr. Juliano Araújo Wickboldt

Porto Alegre
August 2023

“Great discoveries and improvements invariably
involve the cooperation of many minds.”
— ALEXANDER GRAHAM BELL

ACKNOWLEDGEMENTS

I would like to express my profound gratitude to my family. Their unwavering support, understanding, and encouragement have been my foundation throughout this academic journey. In particular, I would like to thank my parents, Antonio Carlos de Sampaio Bastos and Claúdia da Cunha Paranhos, for doing everything they could to help me flourish in my aspirations, for being always willing to listen and trying to help even when they knew nothing about the subjects I was studying, which they would gladly hear me talk about.

I would like to acknowledge my colleagues and fellow students at UFRGS, especially Eduardo Fischer, Gabriel Lando, Gabriel Probst, Lucca Milano, Matheus Woeffel and Maurício Ohse, for their friendship I intend to keep forever, the shared experiences, the help and support through every challenge the university life, and life outside it, threw at us. Eduardo Fischer specially, since he developed the first VizColab.

All my love to my girlfriend, Bruna, who made me company and put up with my anxiety and the weekends where I had the need to sacrifice our personal time in favor of this project. And for her input and suggestions on details of the features implemented here.

Special thanks to my advisor, Professor Juliano Araújo Wickboldt, for his unending patience, invaluable advice, and continuous support throughout the course of this thesis. Your expertise and insights have been instrumental in shaping this work.

ABSTRACT

Academic collaboration networks, graphs that depict researchers as nodes and collaborative works as links, prove challenging to analyze and visualize at a large-scale. This study improves upon an existing tool, VizColab, originally designed for visualizing a large-scale Brazilian academic collaboration network. VizColab originally visualized these networks in three-dimensional space across three hierarchic levels: universities, graduate programs, and intellectual production authors. The study first enriches VizColab with a two-dimensional visualization capability, expanding the user's viewing options and improving the tool's versatility. A search functionality has been implemented, allowing users to easily locate nodes of interest within the extensive network. The study also introduces color-coding based on centrality metrics, offering a more insightful visual representation of network dynamics. To further support data analysis, tables displaying this centrality information have been added. Finally, the upgraded VizColab now facilitates the sharing of customized visualization setups, promoting collaborative exploration of the academic network. This enhanced version of VizColab delivers an enriched, interactive, and intuitive experience, effectively addressing the complexity of large-scale academic collaboration networks' visualization.

Keywords: Academic collaborations. academic co-authorships. visualization. graphs. graph analysis. co-authorship networks. graduate studies. CAPES.

LIST OF ABBREVIATIONS AND ACRONYMS

CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CSS	Cascading Style Sheets
JSON	JavaScript Object Notation
MD5	MD5 Message-Digest Algorithm hash function
RAM	Random-Access Memory
SPA	Single Page Application
SVG	Scalable Vector Graphics

LIST OF FIGURES

Figure 3.1 Visualization of an individual's co-authorship network using the InterRing technique, sourced from (HUANG; HUANG, 2006).....17
Figure 3.2 Summary of the information contained in a node, extracted from (KUROSAWA; TAKAMA, 2012). The colors of the sectors represent keywords, with size proportional to their usage. The division into concentric rings of different saturations indicates whether the author published only at the beginning of the analyzed period (two de-saturated rings), continuously throughout the period (central de-saturated ring, outer saturated ring), or only recently (two saturated rings).....18
Figure 3.3 Overview of the tool, extracted from (KUROSAWA; TAKAMA, 2012). (a) Panel with the co-authorship network; (b) Node details, where it's possible to filter keywords; (c) Operations panel, where one can configure the analysis interval, among other things. 19

Figure 4.1 Screenshots of different hierarchical levels in the original VizColab.....21
Figure 4.2 Comparison of different connection densities, sourced from (FISCHER, 2022). 22

Figure 5.1 Slider for selecting a year range of data to display on the graphs. "FAIXA DE ANOS" means "YEAR RANGE", and "DENSIDADE DE CONEXÕES" means "CONNECTION DENSITY" 26
Figure 5.2 The adapted InterRing visualization showcasing an individual's co-authorship history. Hovering over sections displays the corresponding author names at the top right, as is being shown with *Lucas Bondan*. The full list of coauthors is shown if the user scrolls the panel. 32
Figure 5.3 A profiling of the use of CPU (yellow) and heap memory (blue) resources. At around the 20000ms mark the view switches from 3D to 2D, the drop in resource usage is noticeable. 32
Figure 5.4 The 2D view of the university graph. Nodes are transparent and have borders to aid inspection when they're close to each others. 33
Figure 5.5 The Universidade Federal do Rio Grande do Sul node focused with it's detail panel openend to the side. 34
Figure 5.6 The share button added to the header. A tooltip that's shown on hover is displayed.....35
Figure 5.7 The two states of the share feature modal, the last state displays the shareable URL. The logo at the top right spins whilst the URL is being created to give feedback to the user that the operation is still ongoing. 35
Figure 5.8 The table view applied to universities, some columns are hidden (a feature available in the eye menu in the top left) and rows are ordered by betweenness centrality. The table, or "ranking" view is displayed by clicking the trophy icon in the header. 38
Figure 5.9 Authors from the Computer Science program in UFRGS colored according to their betweenness centrality in both logarithmic (top) and linear (bottom) scales. 40
Figure 5.10 Post-graduate programs from the UNOPAR university colored according to Degree Centrality ("GRAU" means "DEGREE") in both a logarithmic scale (top) and a linear scale (bottom). 41

Figure 6.1 UFRGS Computer Science authors, seen colored according to research area and betweenness centrality. Betweenness centrality flags authors that connect the different hubs around the research areas. 43
Figure 6.2 At the top right, an isolated cluster is made apparent by turning into a strong red island when coloring nodes by proximity. Most other nodes are similar shades of orange....44
Figure 6.3 The node and his students are diconnected from the greater computer science network of UFRGS if we take into consideration only the top 7 connections of every author. 45
Figure 6.4 A researcher's own collaboration network usually shows them with relatively high centrality. That is contrasted with student graphs which are more homogeneous in that regard. 46
Figure 6.5 Two researchers that publish very often together. 47
Figure 6.6 Researchers that participate in many programs can have their specialties highlighted by the centrality metrics. They're relative participation in each program may be quantatively compared considering such metrics. 48
Figure 6.7 The top five researchers by betweenness centrality for each year, from top to bottom 2017 to 2020. 49

CONTENTS

1 INTRODUCTION 10
2 DEGREE, CLOSENESS, BETWEENNESS AND EIGENVECTOR CENTRALITIES 12
2.1 Degree Centrality 12
2.2 Closeness Centrality 13
2.3 Betweenness Centrality 13
2.4 Eigenvector Centrality 15
3 GRAPH VISUALIZATIONS 17
3.1 InterRing 17
3.2 Co-Authorship Networks Visualization System for Supporting Survey of Researchers' Future Activities 18
4 THE ORIGINAL VIZCOLAB 20
4.1 Backend Architecture: Neo4j Database 22
4.1.1 Nodes in VizColab's Neo4j Database 23
4.1.2 Relationships in VizColab's Neo4j Database 23
5 ENHANCEMENTS TO VIZCOLAB 25
5.1 Kubernetes Cluster 25
5.2 Time Slice Filtering 26
5.2.1 Time Slice Filtering Implementation 27
5.3 Adapted InterRing Visualization for Author Networks 30
5.3.1 Coloring Arcs 31
5.4 Two-dimensional View 31
5.5 Node Search and Focus 33
5.5.1 Node Search and Selection 34
5.5.2 Camera Focus and Exploration 34
5.6 Sharing Visualizations 35
5.6.1 Backend Implementation 36
5.6.2 Database Choice 36
5.6.3 Deployment 37
5.6.4 Limitation on Camera Rotation Recreation 37
5.7 Table View 37
5.8 Coloring Nodes by Centrality Measures 39
6 ENHANCEMENT USE CASES 42
6.1 Highlighting nodes that connect hubs 42
6.2 Flagging isolated clusters 42
6.3 Identifying unique and common traits of authors and students networks 45
6.4 Quantifying the relative participation of researchers in different programs 47
6.5 Identifying outstanding data for each year 47
7 FINAL CONSIDERATIONS AND FUTURE WORK 50
REFERENCES 52

1 INTRODUCTION

The growing intricacy of modern scientific pursuits requires both increased specialization and multidisciplinarity, making collaboration an essential pillar of science (UTZERATH; FERNáNDEZ, 2017; WANG; WU; PAN, 2014). Despite its importance, objective discussions about "collaboration" can be challenging due to the nebulous nature of the concept (WOOLGAR, 2012). In this context, metrics and visualizations can provide valuable insights.

Co-authorships in academia are one of the most visible and easily accessible indicators of collaboration between researchers (ABBASI; ALTMANN; HWANG, 2010), making them an appealing object of study in analyzing scientific collaboration (MILOJEVIć, 2010). In Brazil, CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) maintains and publishes data sets related to the intellectual production of researchers from all stricto sensu graduate programs (CAPES, 2023). Leveraging these factors, the VizColab (FISCHER, 2022) tool was developed as a web application that allows for the visualization of a three-dimensional graph of Brazilian academic collaborations at three hierarchical levels: universities, graduate programs, and academic production authors.

The original VizColab tool achieved significant success in enabling structured data visualization and offering a swift means to qualitatively explore various research questions related to regional, institutional, and individual collaborations in Brazil (FISCHER, 2022). Despite its virtues, VizColab had some limitations, such as the lack of temporal visualization and more elaborate filters, absence of co-authorship details, and the unavailability of metrics and auxiliary visualization methods to enrich the analysis.

Addressing these limitations, this study sought to enhance VizColab, adding several features: two-dimensional visualization, node search functionality, node color-coding based on centrality values, dynamic tables with centrality information, a temporal filter to select a specific range of years, and a feature to save and share the current state of the graph. These improvements were inspired by suggestions from the evaluation committee of the original VizColab work, input from other UFRGS professors and researchers, discussions on media appearances[1], and original ideas from the authors of the current study.

The following sections are organized as follows: Chapter 2 discusses the centrality

[1] VizColab: Vizualização de colaborações - <https://youtu.be/0VYuaq7pn-Y>

metrics available in the VizColab tool, citing relevant previous work that used these metrics to analyze academic collaboration networks. Chapter 3 contains an in-depth exploration of two papers that approached the problem of visualizing co-authorship networks. That serves as context to the discussion of the new features added to VizColab further into the article. Chapter 4 describes the original VizColab app, before the enhancements. Chapter 5 discusses the implementation of the enhancements to the tool, with a section covering each one. Following this, Chapter 6 explores some use cases that demonstrate the utility of these enhancements. The thesis concludes with final considerations and directions for future work in chapter 7.

The emphasis of this study is on improving the VizColab tool, equipping researchers, analysts, and enthusiasts with enhanced functionalities to more comprehensively visualize and explore the landscape of Brazilian academic collaborations.

2 DEGREE, CLOSENESS, BETWEENNESS AND EIGENVECTOR CENTRALITIES

This section presents the centrality metrics made available in VizColab by this project, how they are calculated and previous work on exploring their interpretations. Their visualization is detailed chapter 5.8.

2.1 Degree Centrality

The degree centrality of a node is the number of other nodes it connects to. In other words, it is the count of edges that contain the node at one of its endpoints. Let's denote the degree of a node n as d and the set of graph edges as E. An edge where one endpoint is node n and another is node m is denoted as (n, m). The degree of a node can be calculated with the following formula:

$$d(n) = \sum_{(n,m)\in E} 1$$

If we understand the degree centrality of a node simply as the result of this calculation, using this metric to compare different graphs can be problematic. For instance, a node of degree 9 in a graph with a total of 10 edges is present in 90% of the graph's links, whereas a node with the same degree in a graph with 100 edges is only part of 9% of the connections, being much less connected in relative terms. For this reason, a node's degree centrality is normalized by dividing the node's degree by the total number of edges in the graph. In a graph where a node cannot link to itself, this self-link is subtracted from the total. Thus, for the co-authorship graph, the degree centrality of a node is given by:

$$C_D(n) = \frac{d(n)}{total\ nodes - 1}$$

Nodes with high degree centrality can be considered "popular" and, in the context of co-authorship networks, have the potential to guide the topics studied by what can be understood as "research groups": the set of nodes connected to the high centrality node. However, these groups can be isolated from the network as a whole. This highlights that degree centrality is a local measure and does not, by itself, position the node within the network as a whole. In comparison, a measure that does this is, for example, the closeness

centrality.

2.2 Closeness Centrality

The closeness centrality of a node pertains to how close, on average, a node is to all others in terms of the number of edges that must be traversed to reach them. One way to calculate it is by summing the reciprocal of the number of edges traversed from the node to all other nodes. This calculation is performed in this way to account for disconnected networks. In the case of two disconnected nodes, the number of edges between them is considered infinite and the reciprocal of this "distance" is considered zero.

If we denote the closeness centrality of a node n as C_C and the distance between this node and another node m as $d(n, m)$, considering the set of all nodes N, the formula for closeness centrality is:

$$C_C(n) = \frac{N - 1}{\sum_{m \in N} d(n, m)^{-1}}$$

According to (YAN; DING, 2009), closeness centrality "focuses on the breadth of a node's influence over the entire network". (LI-CHUN et al., 2006) posits this centrality as "a measure of how long it will take for information at a node in the network to reach others". Yet another source, (ABBASI; HOSSAIN; LEYDESDORFF, 2012), argues that a node closer to all others (on average) can more easily obtain information and disseminate it throughout the network. They conclude from this that the measure approximates how independent and efficient a node is in terms of communicating with others. It's easy to see how these insights can be applied to co-authorship networks, in analyzing the flow of information and the influence of researchers and the capillarity of their communications.

2.3 Betweenness Centrality

Betweenness centrality refers to the proportion of shortest paths between two nodes that pass through the node under analysis. More precisely, if we refer to the betweenness centrality of node n as $C_B(n)$, the quantity of shortest paths between two nodes j and k as $\sigma_{d(j,k)}$, and the quantity of shortest paths between j and k that pass through n as $\sigma_{d(j,n,k)}$, the formula for betweenness centrality is:

$$C_B(n) = \sum_{j \neq n \neq k} \frac{\sigma_{d(j,n,k)}}{\sigma_{d(j,k)}} = \sum_{j \neq n \neq k} C_B^{j,k}(n)$$

Where $C_B^{j,k}(n)$ is the betweenness centrality of the node in relation to the pair of nodes (j, k).

For disconnected graphs, it is possible that $\sigma_{d(j,k)}$ is equal to 0. In these cases, it is conventionally accepted that $C_B^{j,k}(n)$ equals 0. Another point to note is that each pair of nodes (j, k) contributes to the sum with a value $C_B^{j,k}(n) \in [0, 1]$. This means that, calculated in this way, betweenness centrality tends to be higher for nodes in graphs with more nodes. As will be shown in following chapters, we are interested in calculating these metrics for graphs of different sizes, corresponding to collaboration networks for universities, post-graduate programs and individual researchers, so normalizing the values is useful. This can be done by dividing the value of the sum by the total possible number of pairs (j, k) that do not include n. If we take the number of nodes in the graph as N, for an undirected graph, like the co-authorship one, this value is $(N-1)(N-2)/2$. The final calculation is:

$$C_B(n) = \frac{\sum_{j \neq n \neq k} C_B^{j,k}(n)}{(N-1)(N-2)/2} \begin{cases} C_B^{j,k}(n) = 0, & \text{if } \sigma_{d(j,k)} = 0 \\ C_B^{j,k}(n) = \frac{\sigma_{d(j,n,k)}}{\sigma_{d(j,k)}}, & \text{if } \sigma_{d(j,k)} \neq 0 \end{cases}$$

(YAN; DING, 2009) cites (LI-CHUN et al., 2006) stating that nodes with high betweenness are "pivots of information flows in the network." According to (YAN; DING, 2009), authors with high betweenness connect groups of researchers with common interests and, because they connect different groups, often operate in different research areas, demonstrating interdisciplinarity. Due to their mediating role, high betweenness nodes can be seen as those whose removal has the most potential to affect the flow of information in the network.

(ABBASI; HOSSAIN; LEYDESDORFF, 2012), having studied scientific articles published in fifteen of the most prestigious journals in the steel structures field from 1999 to 2009, assessed the correlation between the three centrality metrics discussed thus far and the likelihood of authors forming collaborations with new authors. Their findings indicated that, for the studied works, betweenness centrality outperformed the other centralities as a predictor of to which nodes new entrants to the co-authorship network would connect. However, the authors acknowledged that further studies were required to deter-

mine whether this observation would hold for other fields of knowledge. In this context, the resources developed in this work, which include the calculation of centrality metrics for co-authorship networks using multidisciplinary data from CAPES, and tools to visualize these networks' evolution over time, may be instrumental in addressing such question, as well as aiding CAPES decision-makers in understanding who contributes most significantly to the growth of co-authorship networks.

2.4 Eigenvector Centrality

Eigenvector centrality is a measure of influence in a network that assigns relative scores to nodes considering not only their direct connections but also the significance of their neighbours (BONACICH, 1987). This concept is well-applied in the context of scientific co-authorship networks, where a researcher's influence is shaped not merely by their quantity of collaborations, but also by the prominence of their co-authors.

The eigenvector centrality of a node in the adjacency matrix $\mathbf{A}$ of a graph, where $A_{ij} = 1$ if nodes i and j are connected and $A_{ij} = 0$ otherwise, can be calculated by the following equation:

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x} \tag{2.1}$$

Here, $\mathbf{x}$ is the vector of centrality scores of all nodes and λ is the principal eigenvalue of the adjacency matrix $\mathbf{A}$. This equation signifies that the centrality of a node is proportional to the sum of the centralities of its neighbours. As such, a node connected to many high-centrality nodes will have a high centrality itself.

The reason why λ is the largest eigenvalue of $\mathbf{A}$ comes from the Perron–Frobenius theorem, which assures that a positive, square matrix like $\mathbf{A}$ will have a single principal eigenvector (corresponding to the largest eigenvalue) that consists of all positive entries. This is important, as the centrality measures of all nodes should be non-negative.

The scores are normalized such that the sum of squares equals one, $\sum_i x_i^2 = 1$. As indicated before, this is particularly important when comparing networks with significant differences in size and connectivity.

(ABBASI; ALTMANN; HOSSAIN, 2011) utilized Eigenvector centrality in their analysis of co-authorship networks. They found that researchers who are connected to many other distinct scholars often have better citation-based performance (g-index) than

those with fewer connections. However, interestingly, they found that Eigenvector centrality had a negative significant influence on the g-index, suggesting that scholars may benefit more from working with many students instead of other high-performing scholars.

3 GRAPH VISUALIZATIONS

This sections explores two papers related to the task of visualizing co-authorship networks, InterRing (HUANG; HUANG, 2006) and Co-Authorship Networks Visualization System for Supporting Survey of Researchers' Future Activities (KUROSAWA; TAKAMA, 2012). The in-depth investigation the articles serves to contextualise VizCo-lab among previous work in graph visualizations, specially in the context of academic collaboration.

3.1 InterRing

(HUANG; HUANG, 2006) developed a co-authorship visualization called ***InterRing*** (figure 3.1). The technique involves representing an author's co-authorship history as a series of concentric rings. Each ring represents a year in which the author was involved in collaborative academic publications and rings closer to the center denote earlier periods. The rings are divided into colored sections, each corresponding to a co-author. The space occupied by each section is proportional to the relevance of that co-author within the co-authorships of that particular year. This relevance is determined by taking into account not only the number of partnerships with the author under analysis, but also the relative contribution in each publication. This is inferred from the order in which the authors' names appear on papers, relying on the commonly recognized convention that authors are listed by the magnitude of their contributions.

The authors emphasize that this visualization method places more emphasis on analyzing the temporal dimension, in contrast to graph visualizations which typically

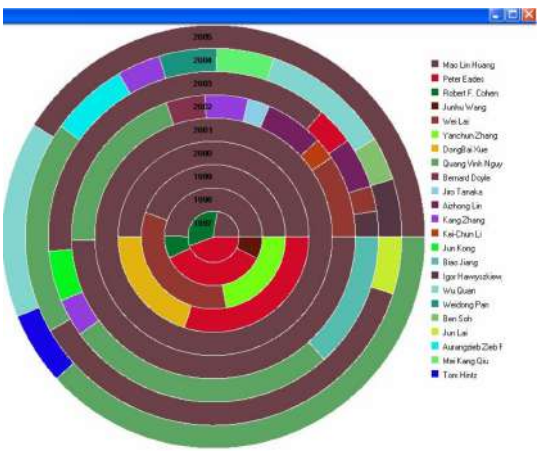


Figure 3.1 – Visualization of an individual's co-authorship network using the InterRing technique, sourced from (HUANG; HUANG, 2006)

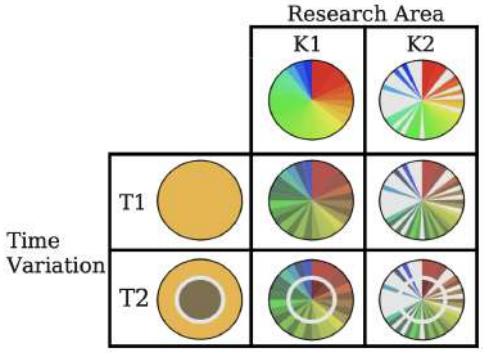


Figure 3.2 – Summary of the information contained in a node, extracted from (KUROSAWA; TAKAMA, 2012). The colors of the sectors represent keywords, with size proportional to their usage. The division into concentric rings of different saturations indicates whether the author published only at the beginning of the analyzed period (two de-saturated rings), continuously throughout the period (central de-saturated ring, outer saturated ring), or only recently (two saturated rings).

represent only the aggregate outcome at the end of a time period.

Along with the main visualization, the authors also provided panels in the developed tools listing the articles co-authored by each author. This allowed for a more detailed understanding of each author's research and its progression.

3.2 Co-Authorship Networks Visualization System for Supporting Survey of Researchers' Future Activities

(KUROSAWA; TAKAMA, 2012) sought visualizations that could be used to predict future activities of researchers. In particular, they aimed for tools capable of identifying rising authors, those most likely to "write an article of interest" in the future, and the emergence of new research areas. The authors argue that co-authorship networks are an apt data source to underpin this analysis. This is because scientific papers are often collaborative endeavors, and new study areas typically arise from the collaboration of researchers from diverse fields of knowledge.

The proposed system consists of a two-dimensional network of nodes representing authors and edges representing co-authorships. The way the nodes are colored conveys the authors' publication cadence within the analyzed time span, as well as the distribution of keywords in their academic output (figure 3.2). Additionally, in the graph visualization, it was possible to select automatic clusterings based on the use of similar keywords.

The researchers evaluated the effectiveness of their tool by providing it to a group of students who were tasked with using it to identify rising researchers and those who

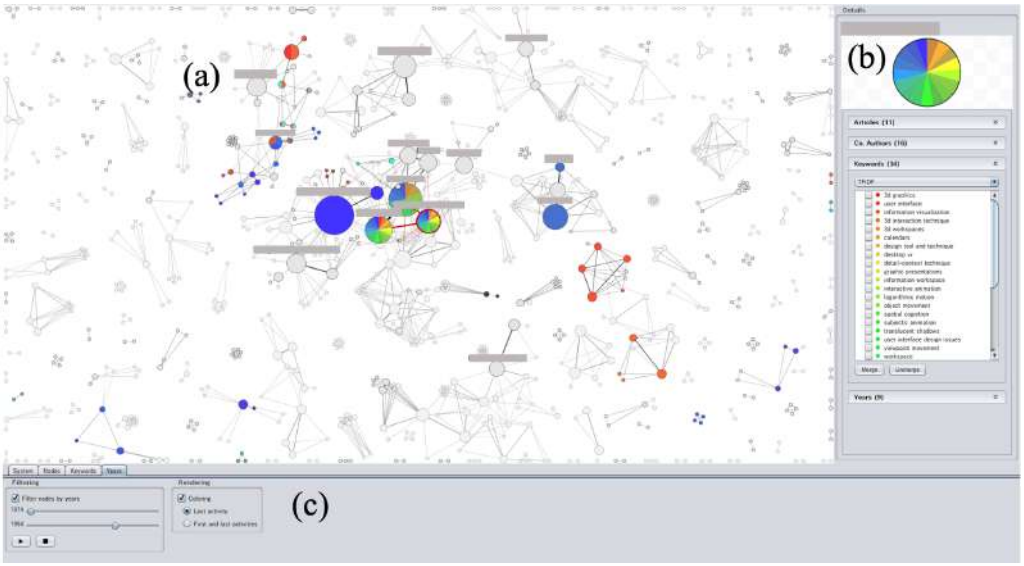


Figure 3.3 – Overview of the tool, extracted from (KUROSAWA; TAKAMA, 2012). (a) Panel with the co-authorship network; (b) Node details, where it's possible to filter keywords; (c) Operations panel, where one can configure the analysis interval, among other things.

would act more as "supervisors", having past achievements but whose future contributions might be less prolific. To assess the impact of specific features, the authors supplied different versions of the tool, with varying enabled features, and evaluated the outcomes. Their conclusion was that the tool was effective in aiding analyses, and all proposed features were deemed relevant.

4 THE ORIGINAL VIZCOLAB

VizColab emerged as a tool designed to explore Brazilian academic collaboration networks in a highly visual and interactive way. It offers a comprehensive visualization of co-authorship relations, presenting data in a structured and user-friendly manner.

For the co-authorship networks and academic collaborations visualized in VizColab, nodes and edges have different meanings across multiple hierarchical visualization levels (see Figure 4.1):

1. **Collaborations Among Universities:** The main page presents a visualization of academic collaborations in Brazilian postgraduate programs at the university level. Nodes denote universities, color-coded by regions (North, Northeast, Central-West, Southeast, South). Their sizes reflect the aggregate academic outputs of each university, while edges denote co-authorships among their faculty members. The diameter of these edges signifies the number of co-authorship ties.
2. **Collaborations Among Postgraduate Programs of a Given University:** For each university's data, one can delve into a new graph mapping collaborations among its postgraduate programs. In this tier, nodes represent the programs, color-coded according to their primary knowledge domain, and edges symbolize co-authorships of their respective authors. Like the previous level, the size of the elements scales with the number of publications and collaborations.
3. **Collaborations Among Authors of a Specific Program:** At this level, nodes embody authors, distinguished by their publication count (node diameter), and edges denote co-authorships.
4. **Collaborations of a Specific Author:** One can further *explore* a node from the previous tier, representing an author. The resulting visualization uses the same symbols, but the dataset is trimmed to only display co-authorship relations involving the chosen author.

With the vast dataset it handles, containing 14,883,507 co-authorship relations derived from the Open Data Portal of CAPES (FISCHER, 2022), VizColab has incorporated several strategies to facilitate analysis:

1. **Hierarchical Node Segmentation:** A high number of nodes and connections might result in overlapping visualization elements in VizColab's three-dimensional display, hindering discernment of network sections and patterns. That's why, to miti-

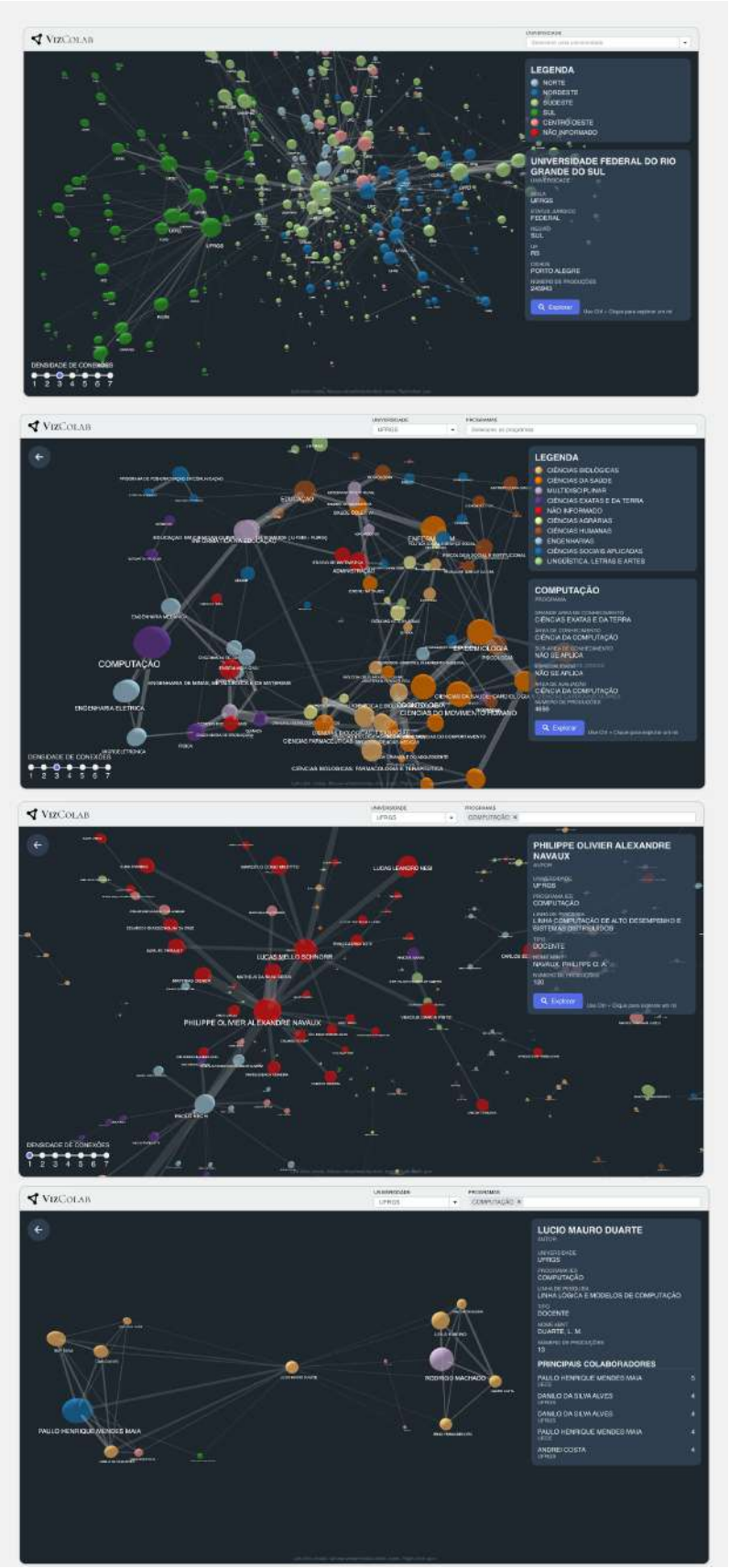


Figure 4.1 – Screenshots of different hierarchical levels in the original VizColab.

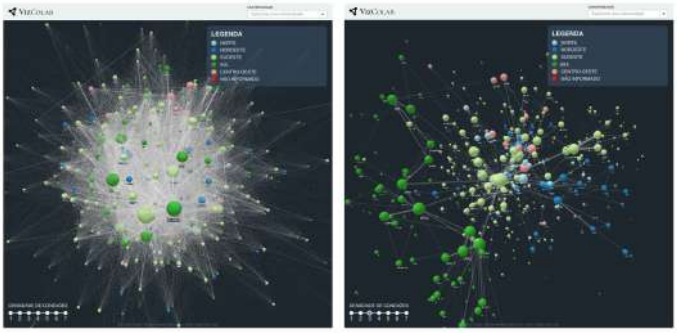


Figure 4.2 – Comparison of different connection densities, sourced from (FISCHER, 2022).

gate this, VizColab partitions data across the multiple aforementioned graphs, which categorize information into hierarchical levels, reducing the concurrently displayed node and edge count.

2. **Variable Connection Density:** Even with fewer nodes, the number of edges could sometimes obscure graph exploration. To remedy this flexibly, a feature was incorporated to dynamically adjust the connection density on screen, hiding less relevant edges if necessary. The criterion for "relevance" used here is: the most relevant edges linked to a node are those connecting it with the nodes with which it collaborated the most. By ranking edges in descending order based on collaboration numbers, and selecting a density d, the system can return a graph that contains only the top d edges from each node.

While these techniques, combined with the use of color and size differentiation, already enabled a deep analysis of the data and the extraction of numerous insights (FISCHER, 2022), the literature review on co-authorship network visualization and analysis, of which particular tools and studies were detailed in chapters 2 and 3, highlighted potential areas for improvement. Chapter 5 will detail the specific improvements made, providing a comprehensive overview of the enhanced interactive experience now available for users exploring Brazilian academic collaboration networks.

4.1 Backend Architecture: Neo4j Database

A crucial aspect of VizColab's ability to visualize and navigate vast co-authorship networks stems from its backend data architecture. To ensure efficient access to highly interconnected data, VizColab leverages Neo4j, a native graph database management system (NEO4J, 2022b).

This database management system is specifically designed to process and store

intricately related data. Its graph-centric architecture facilitates highly efficient queries, making it the most popular graph database worldwide. Within this architecture, there are two primary entities: nodes and relationships.

4.1.1 Nodes in VizColab's Neo4j Database

- **Author:** Denotes an intellectual production's contributor.
- **Program:** Represents a specific post-graduate program.
- **University:** Symbolizes a higher education institution.
- **Production:** Signifies a published article.

4.1.2 Relationships in VizColab's Neo4j Database

- **Authorship:** Between Author and Production, marking an individual's contribution to an intellectual output.
- **Collaboration:** Between University and University, showcasing their collaborative engagements and counting the collaborations.
- **Co-authorship:** Between Author and Author, signifying a shared contribution, detailing the collaboration count, and itemizing the shared academic works.
- **Affiliation to a Program:** Associates an Author with a Program, marking the author's involvement with a particular post-graduate course.
- **Affiliation to a University:** Links an Author to a University, signifying the author's tie to the educational institution.

This structure allowed VizColab to host a sprawling academic collaboration network, comprising:

- 1,275,852 authors
- 1,708,666 intellectual productions
- 4,685 postgraduate programs
- 532 higher education institutions
- 6,072,199 authorship relations ([author] -> [production])
- 14,883,507 co-authorship relations between authors

- 50,003 collaboration relations between educational institutions
- 1,090,185 collaboration relations between postgraduate programs
- 1,275,852 affiliation relations to postgraduate programs ([author] -> [program])
- 1,275,852 affiliation relations to higher education institutions ([author] -> [university])

These staggering numbers underscore the depth and breadth of Brazilian academic collaborations. The ability to dynamically interact with this expansive network, without significant delays, stands as a testament to both Neo4j's efficiency and the careful configuration of its parameters, detailed in more depth in (FISCHER, 2022).

5 ENHANCEMENTS TO VIZCOLAB

VizColab underwent significant enhancements to further facilitate and enrich the exploration of Brazilian academic collaboration networks. These improvements aim to offer users more granular control over the data, diversify visualization modes, display novel information and provide new ways of interacting with and sharing the network insights. The following sections elucidate each of these features, highlighting their importance and the underlying motivation for their inclusion.

5.1 Kubernetes Cluster

The original neo4j database ran in a simple container managed through Portainer[1], a lightweight management UI that allows users to easily manage Docker containers. In order to improve the reliability, scalability and control over the backend resources, that deployment was substituted for a deployment of Neo4j in a Kubernetes cluster of the Federal University of the State of Rio Grande do Sul (UFRGS) Institute of Informatics (INF). That cluster has more than 100GB of RAM freely available for academic projects and over 80 cores in total, some of them of high-performance Intel Xeon processors. Hosting the database on owned infrastructure was thought to enable finer tuning of performance characteristics and to allow for more streamlined testing of ideias and implementation of new features, without incurring in extra costs associated with managed cloud solutions. Whilst it freed the current authors from having to use the limited resources available on their personal computers, due to difficulties in parallelizing some of the data processing steps, such as author grouping, and the powerful single core performance of the original personal computer used to process it (a Macbook with a M1 Pro processor (FISCHER, 2022)) the cluster deployment was only able to match the original total time of importing and processing the data, at around 33 minutes, and kept the response time of the dynamic queries issued during use of the tool below the 2 second threshold, which was found to be satisfactory for interactivity. The achievement of this effort is thus allowing any future work on the database to be carried on by students with less powerful computers, and further tuning might be performed in future works.

To aid in providing some of the new features implemented in this work the neo4j deployment was enhanced with the installation of the APOC library. APOC (Awesome

[1] Available in https://www.portainer.io/

Figure 5.1 – Slider for selecting a year range of data to display on the graphs. "FAIXA DE ANOS" means "YEAR RANGE", and "DENSIDADE DE CONEXÕES" means "CONNECTION DENSITY"

Procedures on Cypher) is a powerful addition to neo4j that provides access to user-defined procedures and functions which extend the use of the Cypher query language into areas such as data integration, graph algorithms, and data conversion. Particularly in this work, extensive use was made of it's list manipulation features. For the implementation of the time slice filtering feature to be detailed in section 5.2, the aggregates of production counts in nodes and in co-autorship and collaboration relationships were partitioned into lists with values for each year using APOC.

5.2 Time Slice Filtering

One of the most powerful ways to understand and interpret data is through the lens of time. Temporal analysis allows researchers and analysts to identify trends, patterns, and anomalies that may only become apparent when observing data chronologically.

The data from CAPES available in VizColab spans the years 2017 to 2020 (FISCHER, 2022). To harness the potential of this temporal dimension, a range slider was added, positioned above the "connection density"(2) selector. This slider provides users the ability to filter the data based on a specific year or a range of years within that interval. When a time slice is selected using this feature, the displayed graph is dynamically updated to show only the data relevant to that specified timeframe.

The significance of temporal analysis in academic collaboration networks was underlined by the works presented in Chapter 3. Both (HUANG; HUANG, 2006) and (KUROSAWA; TAKAMA, 2012) emphasized the importance of considering the temporal dimension when visualizing and analyzing collaboration networks. By allowing users

to select specific time slices, VizColab facilitates a more nuanced exploration of how collaboration patterns evolve over time, how certain collaborations might be more prevalent during specific years, and how the dynamics of the academic world might shift and adapt with changing contexts.

5.2.1 Time Slice Filtering Implementation

To support the time slice functionality, the data in the application database required some preliminary transformations. The aggregate counts of academic productions of nodes, and the number of total articles pertaining to each collaboration relationship was partitioned in different values for each year, which were stored in nodes and edges as *prod_counts_per_year* and *collab_counts_per_year* properties, respectively. Those properties are arrays constructed with the help of list operations offered by the APOC library, referenced in 5.1. Below is an example of the queries used to annotate collaboration relationships between universities with the required information, and to annotate universities themselves with the respective *prod_count_per_year* arrays:

```
MATCH (u1:University)<-[:WORKS_AT]-(a1:Author)-[:AUTHOR]-(p:
   Production)-[:AUTHOR]-(a2:Author)-[:WORKS_AT]->(u2:University
   )
WHERE u1.name <> u2.name
WITH u1, u2, p.year as year, COUNT(DISTINCT p) AS
   collabs_count_year
WITH u1, u2, year - 2017 as year_index, collabs_count_year
MATCH (u1)-[r:COLLABORATES_WITH]-(u2)
SET r.collab_counts_per_year = apoc.coll.set(coalesce(r.
   collab_counts_per_year, [0,0,0,0]), year_index,
   collabs_count_year);
```

Listing 5.1 – Set collab_counts_per_year for university collaboration edges.

```
MATCH (u:University)<-[:WORKS_AT]-(:Author)-[:AUTHOR]->(p:
   Production)
WITH u, p.year as year, COUNT(DISTINCT p) AS prod_count_year
WITH u, year - 2017 as year_index, prod_count_year
SET u.prod_counts_per_year = apoc.coll.set(coalesce(u.
   prod_counts_per_year, [0,0,0,0]), year_index, prod_count_year
```

```
    )
```

Listing 5.2 – Set prod_counts_per_year for universities.

The queries for the Program nodes and relationships were similar and are available in the GitHub repository for the project[2]. Getting the arrays of prod_counts_per_year and collab_counts_per_year for Author nodes and CO_AUTHOR relationships was, however, more involved, due to the sheer number of Author nodes (1,275,852) and co-authorship relations (14,883,507). The number of collaborations for each year for every pair of authors in the dataset was computed with the help of a python script whose results were then imported into the database.

```
# Load data
co_authorships = pd.read_csv('output/co_authorships.csv',
    delimiter=';')
productions = pd.read_csv('output/processed_productions.csv',
    delimiter=';')

# Convert year column to integer
productions['AN_BASE'] = productions['AN_BASE'].astype(int)

# Merge dataframes on production id
merged = pd.merge(co_authorships, productions, left_on='PROD_ID'
    , right_on='ID_ADD_PRODUCAO_INTELECTUAL')

# Group by year and co-authors pair, count collaborations
collabs_per_year = merged.groupby(['AUTHOR_1', 'AUTHOR_2', '
    AN_BASE']).size().reset_index(name='collabs_count')

# Pivot this DataFrame to have years as columns, fill missing
    values with 0
collabs_per_year_pivoted = collabs_per_year.pivot_table(index=['
    AUTHOR_1', 'AUTHOR_2'], columns='AN_BASE', values='
    collabs_count', fill_value=0).reset_index()

# Now collabs_per_year_pivoted contains each author pair along
    with collaboration counts per year.
```

[2]https://github.com/ComputerNetworks-UFRGS/vizcolab

```
collabs_per_year_pivoted.to_csv('output/
    co_author_collabs_per_year.csv', index=False)
```

Listing 5.3 – Script to generate a CSV with the articles jointly published by each author pair aggregated by year.

The results were imported using the periodic commits feature, which allows writing intermediate results to disk instead of doing the full computation in-memory and eventually exhausting the resources of any one pod in the cluster:

```
:auto LOAD CSV WITH HEADERS FROM 'file:///
    co_author_collabs_per_year.csv' AS row
CALL {
     WITH row
  MATCH (a1:Author {id: toInteger(row.AUTHOR_1)})-[coauthor:
    CO_AUTHOR]-(a2:Author {id: toInteger(row.AUTHOR_2)})
  SET coauthor.collab_counts_per_year = [toInteger(row.`2017`),
    toInteger(row.`2018`), toInteger(row.`2019`), toInteger(row
    .`2020`)]
} IN TRANSACTIONS OF 20000 ROWS;
```

Listing 5.4 – Set collab_counts_per_year for co-authorships.

Setting prod_counts_per_year for Author nodes, in turn, made use of an APOC facility again: it's parallel batching iteration features, which made good use of the cluster's pods:

```
CALL apoc.periodic.iterate(
  "MATCH (a:Author)-[:AUTHOR]->(p:Production) RETURN a, p.year
    as year, COUNT(DISTINCT p) AS prod_count_year",
  "WITH a, year - 2017 as year_index, prod_count_year SET a.
    prod_counts_per_year = apoc.coll.set(coalesce(a.
    prod_counts_per_year, [0,0,0,0]), year_index, prod_count_year
    )",
  {batchSize:10000, parallel:true}
)
```

Listing 5.5 – Set prod_counts_per_year for authors.

The UI for the feature had a more straight-forward implementation. The front-end of VizColab is developed using React[3], a popular JavaScript library for building user

[3]Documentation available in https://react.dev/

interfaces. React's component-based architecture allows for modular development, and it's popularity means that a number of ready-made UI components are available from public packages. For the time slice feature, we incorporated a slider component from the MUI material library. This particular component was chosen due to its ease of integration, user-friendly nature, and compatibility with the application's existing style.

Modern React development makes use of functional programming techniques, using so called *hooks*, special functions that let you "hook into" React features, such as state and lifecycle methods[4]. The graph visualisation was made to dynamically update by defining the year range as a reactive data point through the use of the *useState()* hook, in combination with *useEffect()*, which tied the *"effect"* of re-fetching graph data to changes of that state. The collaboration counts for edges and production counts for nodes were then computed in the data fetching query using APOC to sum the values in the arrays corresponding to the selected range.

```
1 \\ ...
2 apoc.coll.sum(r.collab_counts_per_year[${yearStartIndex}..${
     yearEndIndex + 1}]) as collabs_count
3 \\ ...
```

Listing 5.6 – Computing collab_counts_per_year dinamically using cypher query langue strings interpolated with Javascript

5.3 Adapted InterRing Visualization for Author Networks

An adapted version of the InterRing technique (described in 3.1) was implemented for the detail panels of author nodes inside VizColab. Detail panels are cards that show to the side of the screen displaying additional information when the user clicks a node. The primary aim remained the same: offering a deep dive into an individual's co-authorship history by visualizing it through a series of concentric rings. However, we found that the convention of listing the most important authors first was not widely adopted in Brazil. The variety of institutions and fields of study encompassed in the data-set meant there were multiple conventions at play, with some groups listing the project financiers first, others the less senior members first, others the most senior, some those that performed advisory roles but often did not perform the bulk of the research. With that in mind,

[4]Documentation available at https://react.dev/.

instead of, like the original InterRing, making the arcs taken by coauthors take into account the positions of their names on papers, we chose the simpler method of making the coauthor arcs proportional to the number of partnerships with the analyzed author each year.

5.3.1 Coloring Arcs

A notable challenge, for which no solution was presented in the original paper, was making sure author arcs had preferably unique or at least well varied colors, so that they wouldn't get mixed up when inspecting the visualization. A clever approach was devised taking into account the fact that colors in CSS can be represented in the so called hexadecimal short form by three hexadecimal digits, which line up nicely with characters produced by the inexpensive and readily available MD5 hash algorithm. Colors for each coauthor were made to be the first three characters from the hash of their full names. This approach guaranteed three main advantages:

- **Uniqueness**: Each co-author, based on their full name, would have a distinct color.
- **Stability**: The color assigned to a co-author remains consistent across different views or sessions.
- **Variety**: With the capability to produce colors for up to 16^3 co-authors, the palette remains diverse, ensuring clarity and reducing the potential for confusion.

The visual elements, namely the rings, were constructed using SVG elements and were drawn using the D3 library[5]. SVG ensures that the visualization retains its quality across different screen sizes and resolutions. Leveraging D3 allowed for efficient data binding, enabling the dynamic creation and modification of the visualization based on the inspected author.

5.4 Two-dimensional View

The introduction of a two-dimensional view in our visualization tool is primarily motivated by two key factors. First, the 2D view requires fewer computational resources compared to the 3D view, making it a more accessible and efficient option for users with

[5]Data-Driven Documents, created by Mike Bostock. <https://d3js.org/>

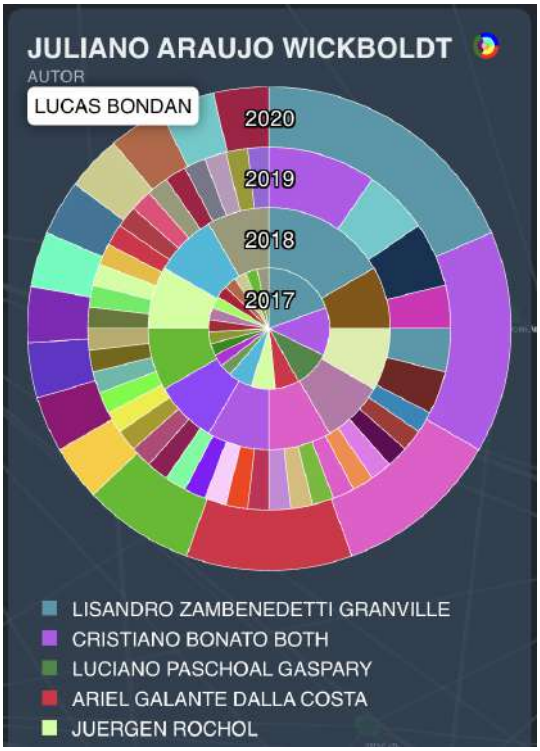


Figure 5.2 – The adapted InterRing visualization showcasing an individual's co-authorship history. Hovering over sections displays the corresponding author names at the top right, as is being shown with *Lucas Bondan*. The full list of coauthors is shown if the user scrolls the panel.

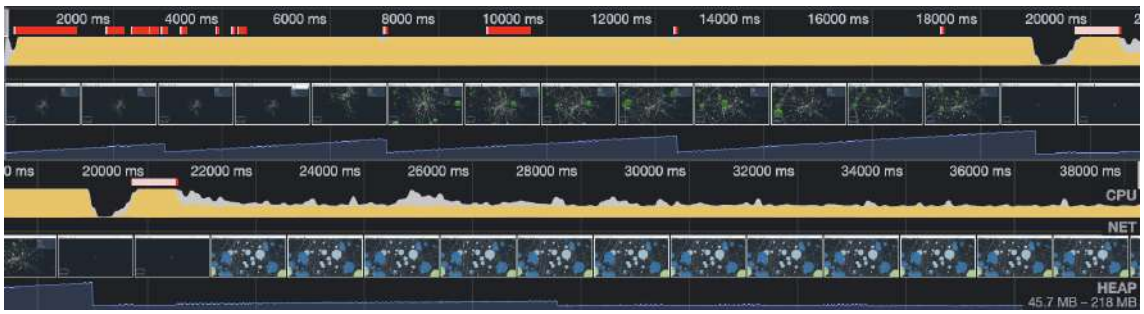


Figure 5.3 – A profiling of the use of CPU (yellow) and heap memory (blue) resources. At around the 20000ms mark the view switches from 3D to 2D, the drop in resource usage is noticeable.

less powerful computing equipment. Second, the 2D view can be more easily captured and reproduced in documents, such as research papers or presentations, offering a more familiar and straightforward representation of the network.

The adoption of a 2D view aligns with the approach taken in many existing network visualizations, such as the previously mentioned (KUROSAWA; TAKAMA, 2012) and (HUANG; HUANG, 2006), where two-dimensional visualizations are utilized. The 2D view also makes it easier to interpret the relationships and patterns present in the network without the need for complex interactions, like rotating or zooming in a 3D space.

However, moving to a 2D view comes with its own set of challenges. The loss of a dimension means that nodes are more likely to overlap and pile up on top of each other, making it difficult to distinguish individual nodes and their connections. To address

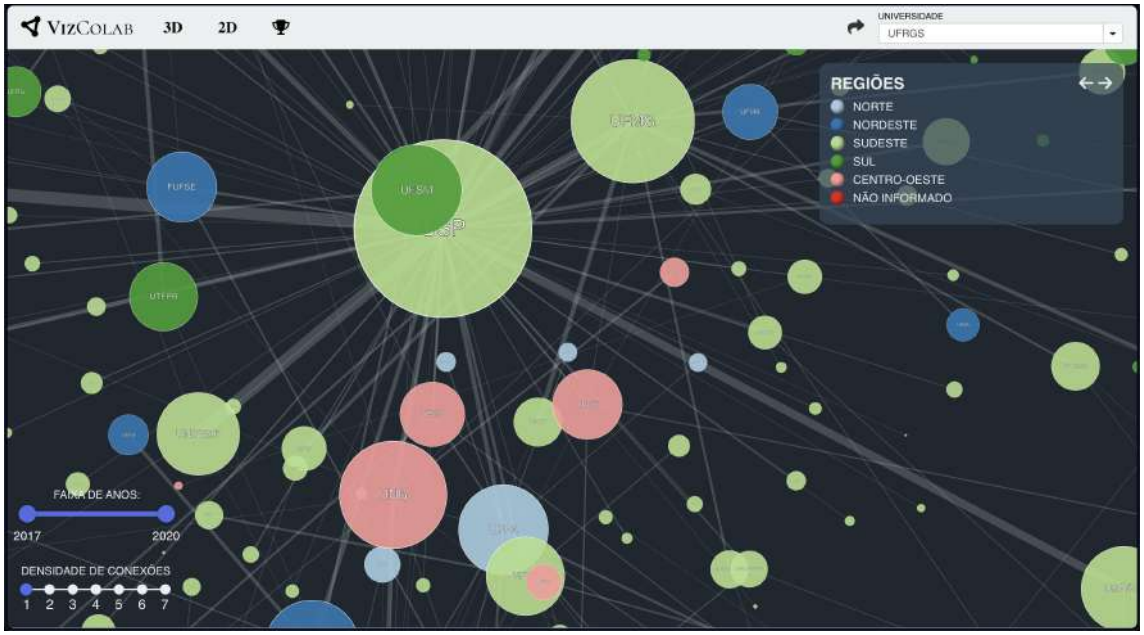


Figure 5.4 – The 2D view of the university graph. Nodes are transparent and have borders to aid inspection when they're close to each others.

this issue, we made nodes semi-transparent in this view and added borders around them and their labels. This allows users to visually separate overlapping nodes, providing better clarity and readability of the network. Additionally, the semi-transparency helps to reveal the connections between nodes even when they are overlapping, allowing users to understand the underlying relationships more effectively.

In conclusion, the introduction of the 2D view serves as a valuable addition to our visualization tool, making it more versatile, accessible, and efficient. It provides an alternative representation of the network that can be easily reproduced in documents, while also being more lightweight in terms of computational requirements. The enhancements made to the nodes and their labels in this view help to mitigate the challenges posed by the loss of a dimension, offering a more clear and interpretable visualization.

5.5 Node Search and Focus

A common challenge that users encountered in the initial version of the tool was the difficulty in locating specific nodes within the graph. Based on this feedback, a key enhancement in the revised tool is the ability to easily search for and focus on a particular node.

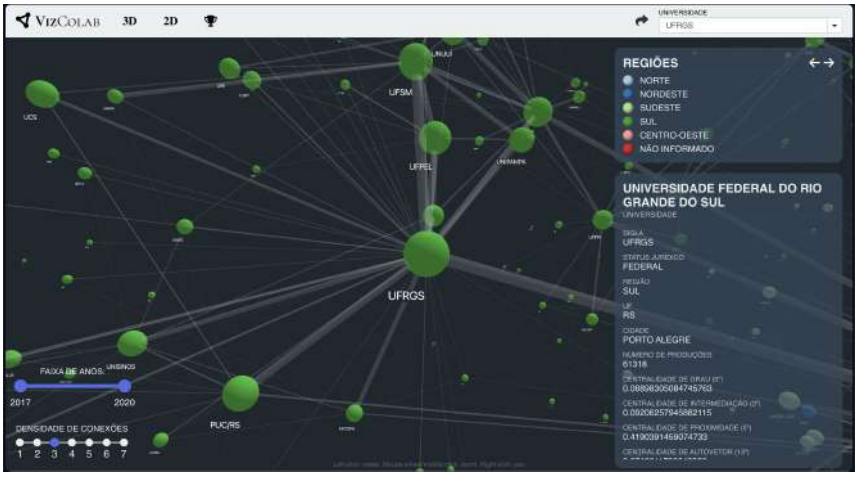


Figure 5.5 – The Universidade Federal do Rio Grande do Sul node focused with it's detail panel openend to the side.

5.5.1 Node Search and Selection

Users can now easily search for nodes of interest using a selector in the header. As they type the node's name into the selector, a list of matching nodes is dynamically presented, allowing the user to select the desired node. Unlike the previous version, selecting a node's name in this updated tool does not immediately transition to the graph of that node's data. Instead, the camera's center coordinates are focused on the selected node, providing users with a clear view of the node within the broader graph context.

5.5.2 Camera Focus and Exploration

The distance between the camera and the focused node is calculated to be proportional to the node's size, preventing the camera from ending up inside large nodes. This ensures that the focused node is comfortably visible within the screen. Users can explore the focused node by either "Ctrl clicking" on it or using the "Explore" button in the details panel (double click on the node).

In the default 3D view, the distance between the camera and the node center is scaled based on the position and size of the node, ensuring that the camera is always positioned outside of it, regardless of its diameter. In the 2D view detailed in 5.4, the focused node is made to fit completely within the canvas with a 10 pixel padding to the nearest sides. This guarantees that the entire node is visible and ensures a consistent visual experience across both 2D and 3D views.

Figure 5.6 – The share button added to the header. A tooltip that's shown on hover is displayed.

Figure 5.7 – The two states of the share feature modal, the last state displays the shareable URL. The logo at the top right spins whilst the URL is being created to give feedback to the user that the operation is still ongoing.

5.6 Sharing Visualizations

In the latest version of VizColab, we have added a feature that allows users to easily share the state of the visualizations they create with others. This sharing feature enables users to generate a link that, when accessed, loads the same graph that was shared, including the node positions, the hierarchy level (universities, programs, or authors, as detailed in 4), the selected connection density (2), year range (5.2), and node color key.

5.6.1 Backend Implementation

To implement this sharing feature, we created a backend whose main responsibility is to provide endpoints to save and retrieve the application states. When a user clicks the sharing button on the front-end, the current application state is sent to the backend in JSON form. This state is then inserted into a JSONB column in a PostgreSQL database. Each state is assigned a generated ID, which is returned to the front-end. The front-end constructs a URL using this ID, which is shared by the user. When the URL is accessed, the front-end retrieves the saved state from the backend and reconstructs the graph accordingly.

5.6.2 Database Choice

At first glance, it may seem that a NoSQL database, such as MongoDB, would be a suitable fit for this application given that we are saving JSON objects directly and not performing any complex queries. Moreover, the format of the saved data might change as new features are added to the tool, requiring more state to be kept. However, we chose PostgreSQL for this task due to the availability of the JSONB column type in newer versions of the database. Introduced in PostgreSQL 9.4 in 2014, JSONB columns allow the database to act as a "document store" similar to document-only databases like MongoDB, and were implemented partially in response to the growing popularity of such solutions. They encode a binary representation of JSON and are very performatic.

One key advantage of using PostgreSQL's JSONB column type is that it gives us all the features of common document-only databases, such as flexibility in storing semi-structured data and efficient storage of JSON documents, but in addition, it provides the ability to query specific JSON fields and to use this data in tandem with more traditional, normalized relational tables, should the need arise in the future. Moreover, as the entire infrastructure is managed using Kubernetes, as described in section 5.1, deploying a PostgreSQL instance is straightforward.

5.6.3 Deployment

The backend server was implemented using Express.js, a popular Node.js web application framework. This server was packaged into a Docker container, providing a self-contained and easily deployable unit. This container was then deployed to the Kubernetes cluster described in section 5.1, where it runs alongside the graph database and the NGINX deployment that serves the React SPA.

5.6.4 Limitation on Camera Rotation Recreation

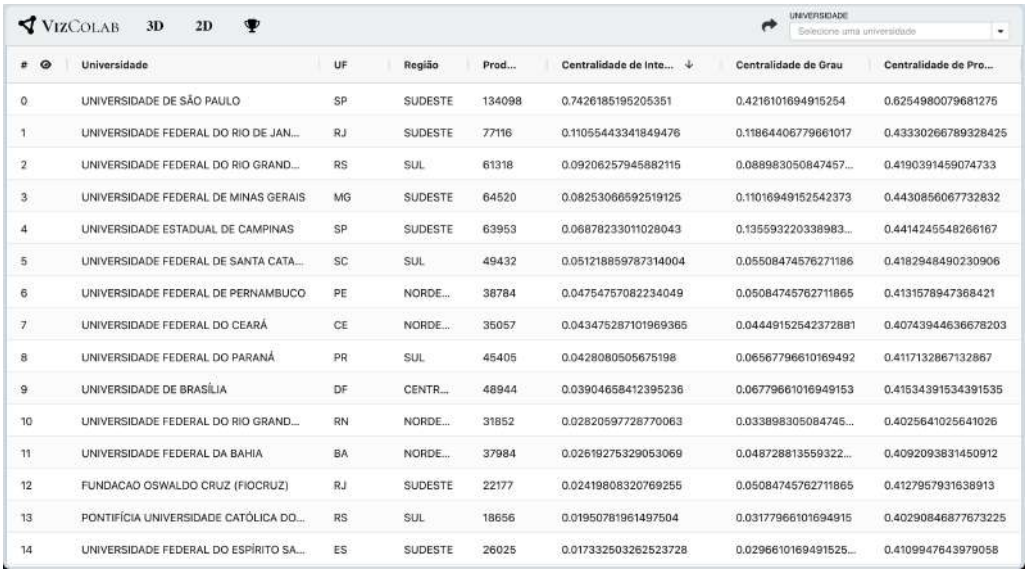
It is important to note a limitation regarding the sharing feature's ability to recreate camera rotations in the 3D view. While zoom and "look at" positions can be reconstructed successfully, we encountered difficulties in accurately reproducing camera rotations using the available APIs in the 3D graph rendering library utilized. As a result, when a user rotates the camera and shares the graph state, the recipient will not be able to view the nodes at the same angle as the original user. We acknowledge that this discrepancy may affect the accuracy of the reconstructed view and have documented it as a known issue in the project's repository. This limitation is a potential area for future work, and we hope it gets addressed in subsequent versions of the tool by exploring alternative approaches or leveraging future updates to the rendering library.

5.7 Table View

In addition to the visual graph views, we introduced a Table View feature to enable users to interact with the data in a more structured and detailed format. This feature is designed to facilitate deeper exploration and analysis of the underlying numbers.

The Table View allows users to perform various operations to customize their data presentation. Key functionalities include:

1. **Sorting Rows:** Users can sort rows based on specific columns to help prioritize or organize the information according to their interests or requirements.
2. **Collapsing and Expanding Rows:** Users can collapse or expand rows to better manage the displayed data. This feature is particularly useful for exploring hierarchical relationships within the data or focusing on particular sections.



#	Universidade	UF	Região	Prod...	Centralidade de Inte...	Centralidade de Grau	Centralidade de Pro...
0	UNIVERSIDADE DE SÃO PAULO	SP	SUDESTE	134098	0.7426185195205351	0.4216101694915254	0.6254980079681275
1	UNIVERSIDADE FEDERAL DO RIO DE JAN...	RJ	SUDESTE	77116	0.11055443341849476	0.11864406779661017	0.433302667893284 25
2	UNIVERSIDADE FEDERAL DO RIO GRAND...	RS	SUL	61318	0.09206257945882115	0.088983050847457...	0.4190391459074733
3	UNIVERSIDADE FEDERAL DE MINAS GERAIS	MG	SUDESTE	64520	0.08253066592519125	0.11016949152542373	0.4430856067732832
4	UNIVERSIDADE ESTADUAL DE CAMPINAS	SP	SUDESTE	63953	0.06878233011028043	0.135593220338983...	0.4414245548266167
5	UNIVERSIDADE FEDERAL DE SANTA CATA...	SC	SUL	49432	0.051218859787314004	0.05508474576271186	0.4182948490230906
6	UNIVERSIDADE FEDERAL DE PERNAMBUCO	PE	NORDE...	38784	0.04754757082234049	0.05084745762711865	0.4131578947368421
7	UNIVERSIDADE FEDERAL DO CEARÁ	CE	NORDE...	35057	0.043475287101969365	0.04449152542372881	0.40743944636678203
8	UNIVERSIDADE FEDERAL DO PARANÁ	PR	SUL	45405	0.0428080505675198	0.06567796610169492	0.4117132867132867
9	UNIVERSIDADE DE BRASÍLIA	DF	CENTR...	48944	0.03904658412395236	0.06779661016949153	0.41534391534391535
10	UNIVERSIDADE FEDERAL DO RIO GRAND...	RN	NORDE...	31852	0.02820597728770063	0.033898305084745...	0.4025641025641026
11	UNIVERSIDADE FEDERAL DA BAHIA	BA	NORDE...	37984	0.02619275329053069	0.048728813559322...	0.4092093831450912
12	FUNDACAO OSWALDO CRUZ (FIOCRUZ)	RJ	SUDESTE	22177	0.02419808320769255	0.05084745762711865	0.4127957931638913
13	PONTIFÍCIA UNIVERSIDADE CATÓLICA DO...	RS	SUL	18656	0.01950781961497504	0.03177966101694915	0.402908468776732 25
14	UNIVERSIDADE FEDERAL DO ESPÍRITO SA...	ES	SUDESTE	26025	0.017332503262523728	0.0296610169491525...	0.4109947643979058

Figure 5.8 – The table view applied to universities, some columns are hidden (a feature available in the eye menu in the top left) and rows are ordered by betweenness centrality. The table, or “ranking" view is displayed by clicking the trophy icon in the header.

3. **Showing or Hiding Columns:** Users have the flexibility to customize the table by showing or hiding specific columns, making it easier to focus on relevant information and reduce visual clutter.
4. **Filtering Rows by Text:** For text-based columns, such as author names, cities, or university affiliations, users can apply text filters to narrow down the displayed rows to match their search criteria.
5. **Filtering Rows by Value or Interval:** For numeric columns, such as academic production counts or node centralities, users can filter rows based on specific values or value intervals to focus on relevant data ranges.
6. **Filtering Data by Time Period:** Filtering the data by time period, as described in section 5.2, is also possible to do when visualizing the table.

The Table View was implemented using the widely-adopted AgGrid library, specifically its community version. As a result, certain features like showing or hiding columns were custom implemented, as they were not natively supported in this version of the library.

The Table View complements the existing visualization tools and offers an alternative approach for users who prefer a more granular or quantitative perspective on the data. It provides users with a customizable interface for exploring and analyzing the data, catering to a broad range of use cases and analytical needs.

5.8 Coloring Nodes by Centrality Measures

The centrality measures discussed in Chapter 2 provide valuable insights into the roles and importance of nodes within the network. To visualize these measures, we implemented a feature that allows users to color nodes according to their centrality values. This feature offers an intuitive way to identify influential nodes or observe patterns in the network based on the selected centrality measure.

The centrality measures available for coloring include betweenness centrality, degree centrality, closeness centrality, and eigenvector centrality. Each centrality measure offers a different perspective on the significance of a node within the network, as described in Chapter 2. The exact centrality values for each node are displayed in the node's detail panel, which can be accessed by clicking on the node.

To color the graph according to node centralities, we use a gradient to represent the range of centrality values. Users can choose between two scaling options for mapping centrality values to positions in the gradient: a logarithmic scale or a linear scale. Each scaling method offers distinct advantages:

- **Linear Scale:** The linear scale assigns colors to centrality values based on their absolute magnitude. It provides a straightforward visualization of the differences between nodes. This scale is appropriate for cases where the centrality measures are more evenly distributed. That is often the case with degree centrality, but can vary depending on the specific network.
- **Logarithmic Scale:** The logarithmic scale assigns colors based on the order of magnitude of centrality values. It is especially useful for highlighting nodes with extremely high or low centrality values in networks with skewed centrality distributions. This scale is particularly suitable for betweenness centrality, which often exhibits a power-law distribution where a few nodes have exceptionally high values compared to the majority of nodes.

The choice between linear and logarithmic scaling depends on the centrality measure being visualized and the specific distribution of centrality values in the network. Users can toggle between the two scales to explore the data from different perspectives and gain a deeper understanding of the network's structure and the roles of its nodes.

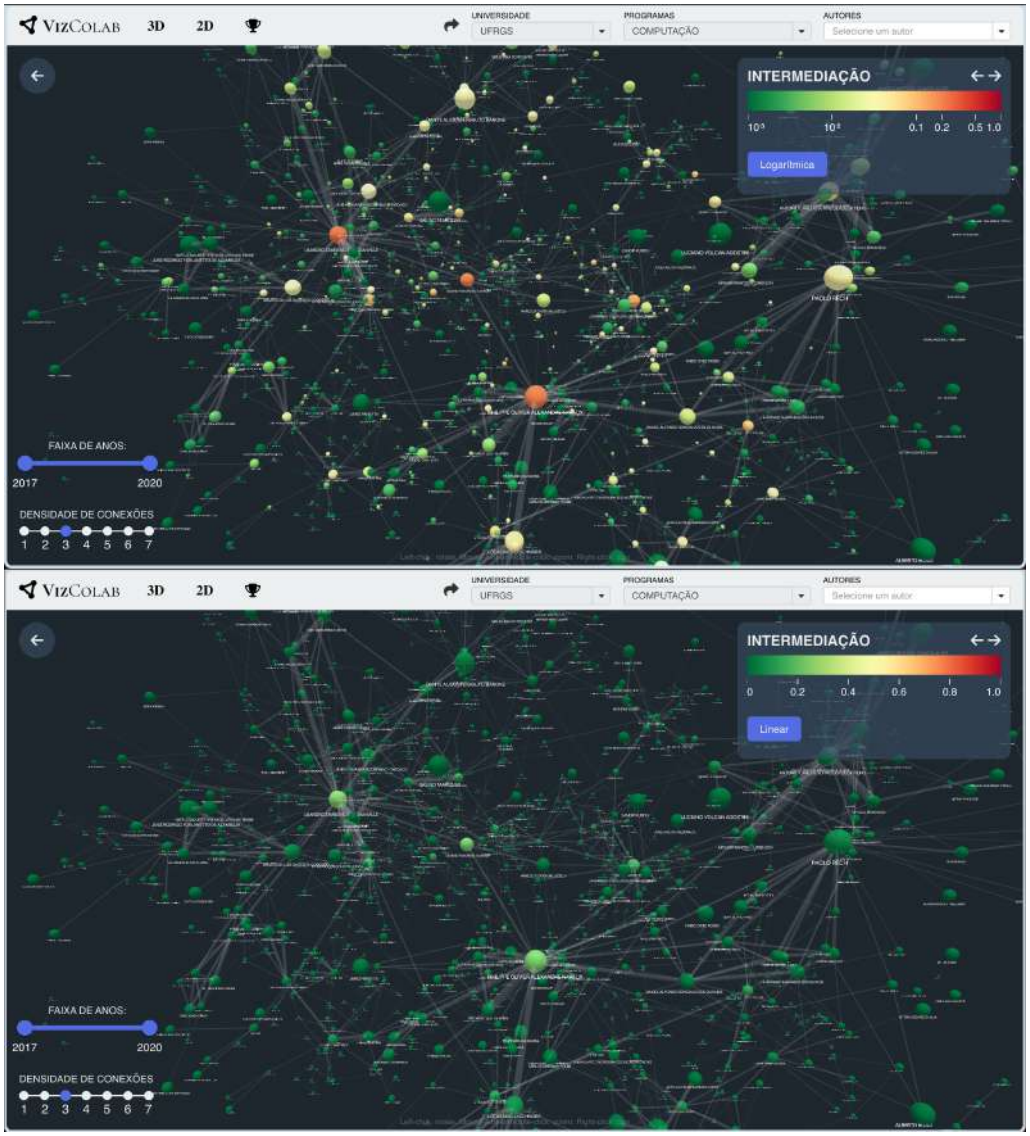


Figure 5.9 – Authors from the Computer Science program in UFRGS colored according to their betweenness centrality in both logarithmic (top) and linear (bottom) scales.

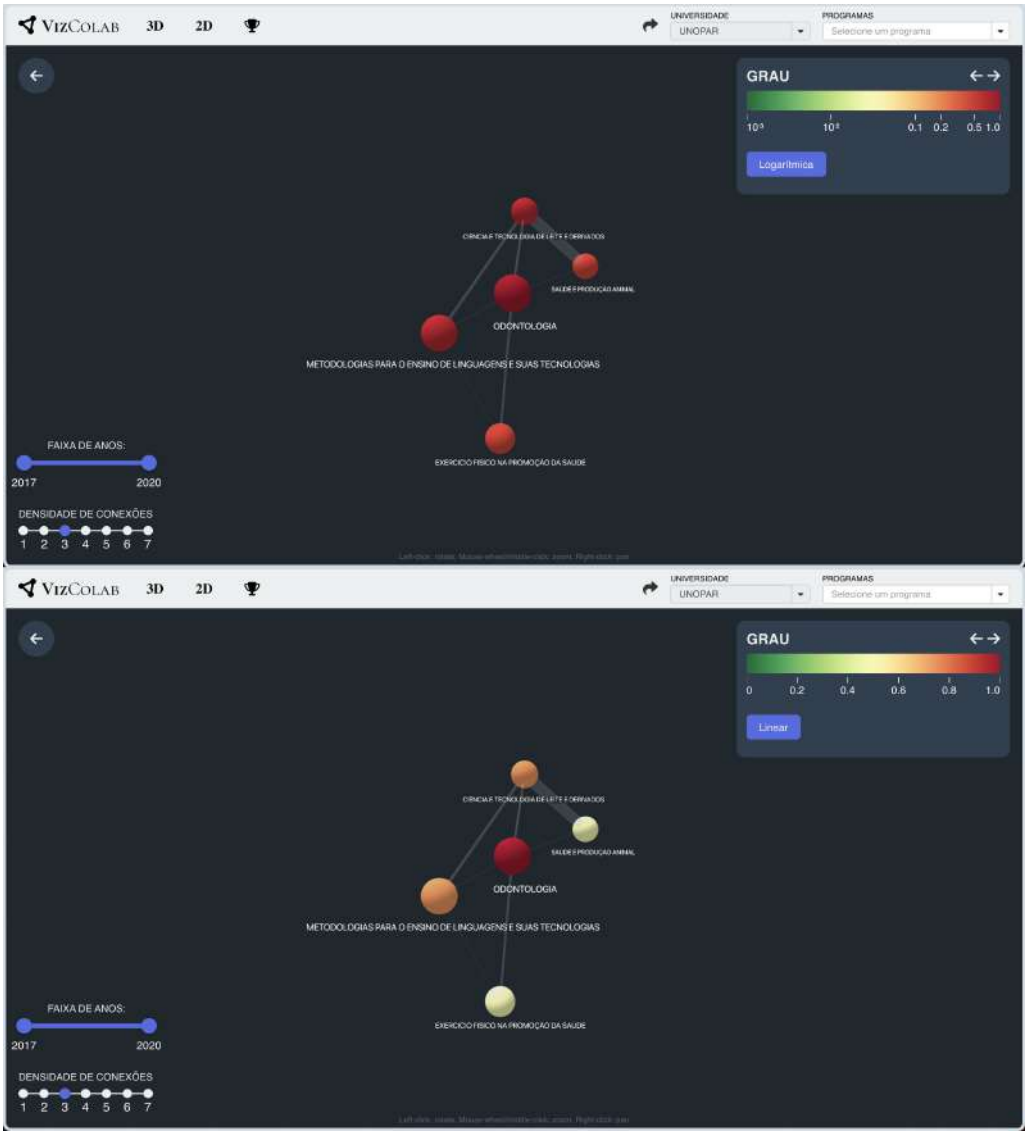


Figure 5.10 – Post-graduate programs from the UNOPAR university colored according to Degree Centrality ("GRAU" means "DEGREE") in both a logarithmic scale (top) and a linear scale (bottom).

6 ENHANCEMENT USE CASES

To give an idea of the sort of use the features made available might lend themselves to, the authors of this thesis have developed brief explorations for demonstrative purposes. They may serve as starting points for deeper dives into the data by researchers or enthusiasts.

6.1 Highlighting nodes that connect hubs

Looking at the authors from the computer science postgraduate program from UFRGS with the research area coloring, it's already possible to see that among the greater network there are smaller clusters, usually formed by researches acting in the same research areas. It is vital that those clusters do not isolate completely, making collaborations between their members and those from other clusters very valuable. Changing the node colors to represent betweenness centrality makes researchers that do so more apparent. Figure 6.1 shows an example of this. Even though it was already possible to see the importance of some nodes, the new coloring makes it obvious and quantitatively ascertainable.

6.2 Flagging isolated clusters

Another thing the centrality metrics make pop is isolated clusters. Such clusters usually have a single very important node that gets strongly colored. In figure 6.2 we can see how the isolated cluster around a researcher at the top right stands out in a strong red whilst most nodes are tinted a shade of orange. That's a position where that wouldn't be easily spotted otherwise. The camera focus feature, in turn, makes it possible to easily inspect such cluster in detail, which would be hard to do before because the camera was locked into looking at the center of the graph and that micro cluster is in the periphery. The central node is seen focused in figure 6.3. Furthermore, if we check its details panel we can quickly see that its closeness centrality ("CENTRALIDADE DE PROXIMIDADE") is 1, which is the highest value possible since all centralities are normalized. That immediately confirms that every node in that cluster has directly collaborated with the node, without the need to carefully inspect the graph. A look into the details of every node in

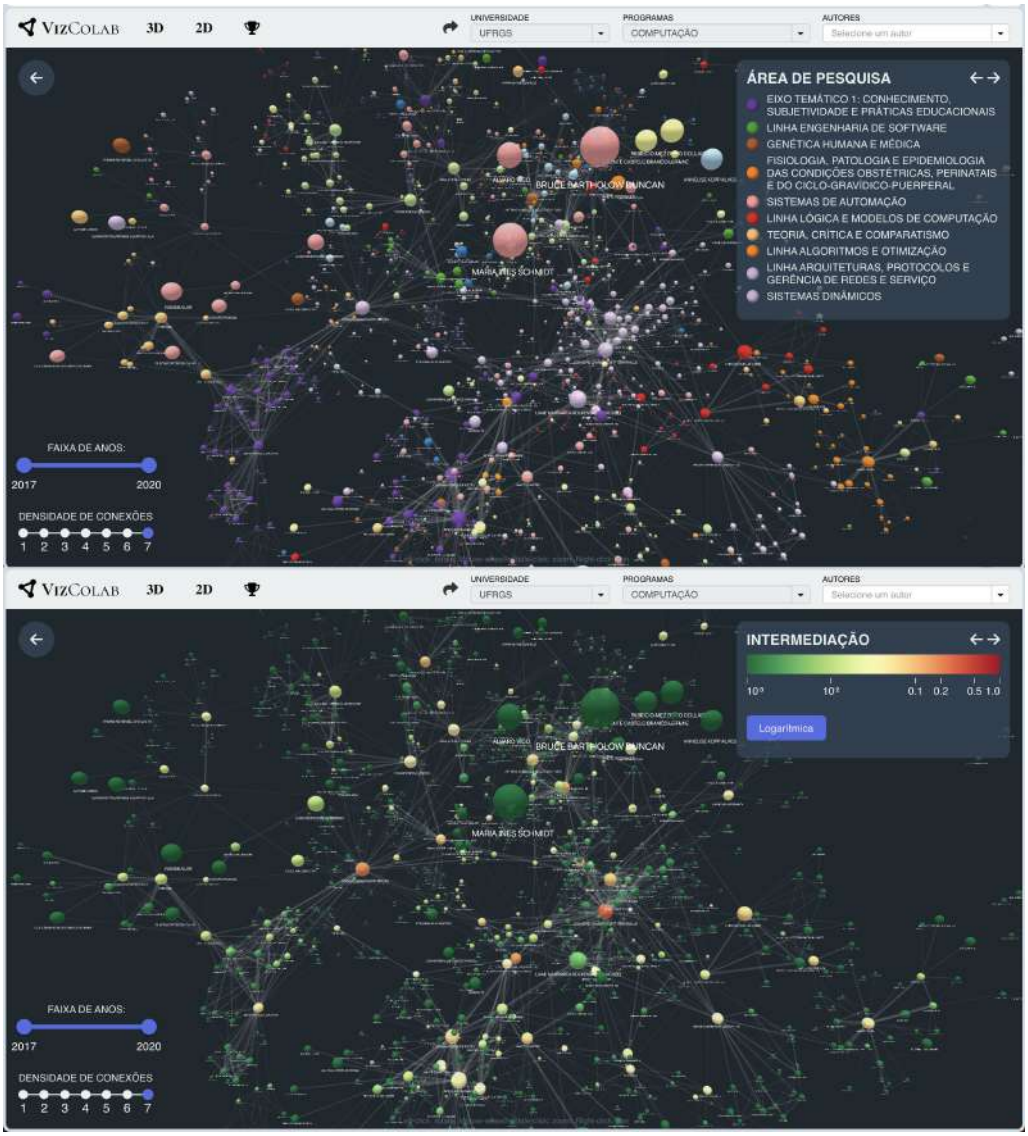


Figure 6.1 – UFRGS Computer Science authors, seen colored according to research area and betweenness centrality. Betweenness centrality flags authors that connect the different hubs around the research areas.

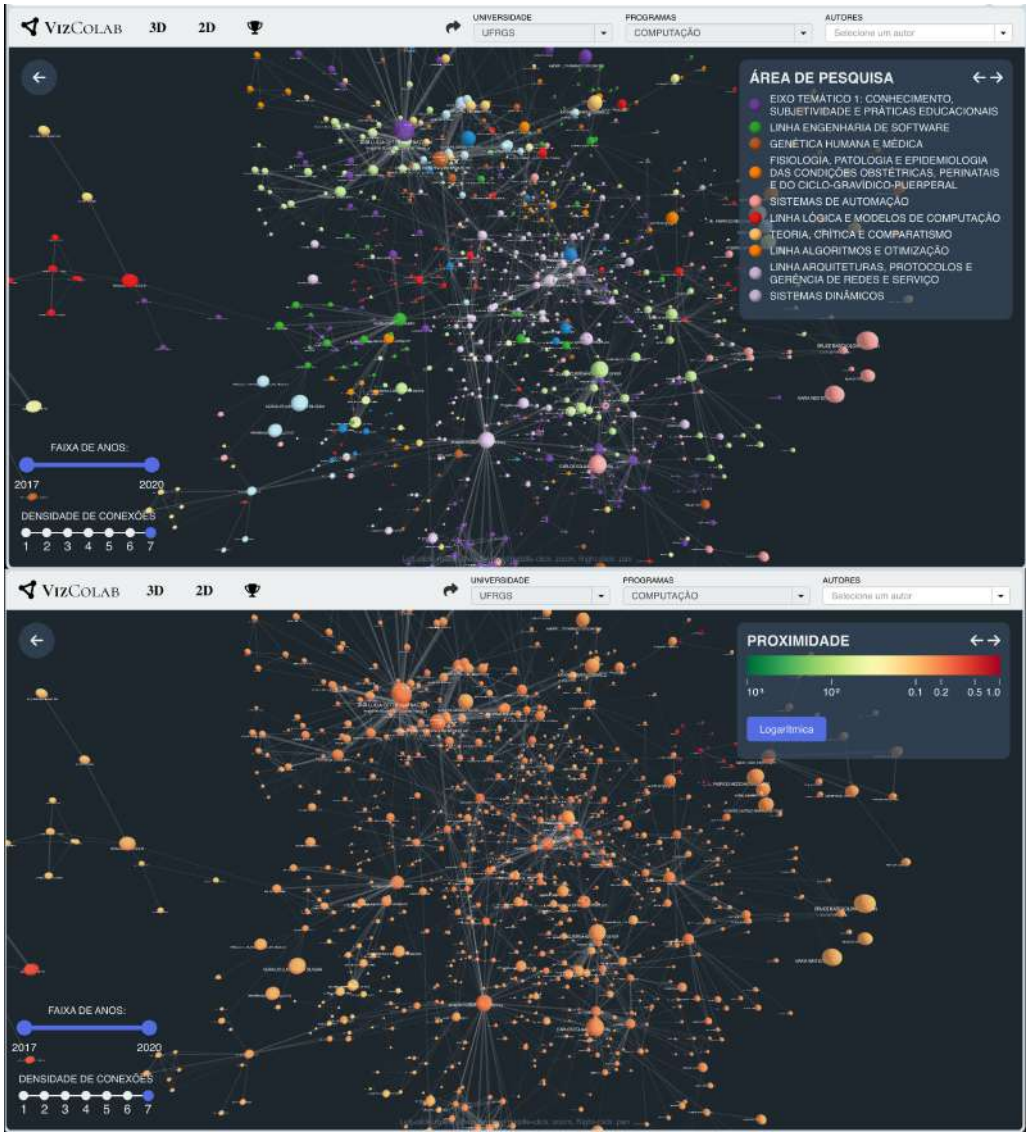


Figure 6.2 – At the top right, an isolated cluster is made apparent by turning into a strong red island when coloring nodes by proximity. Most other nodes are similar shades of orange.

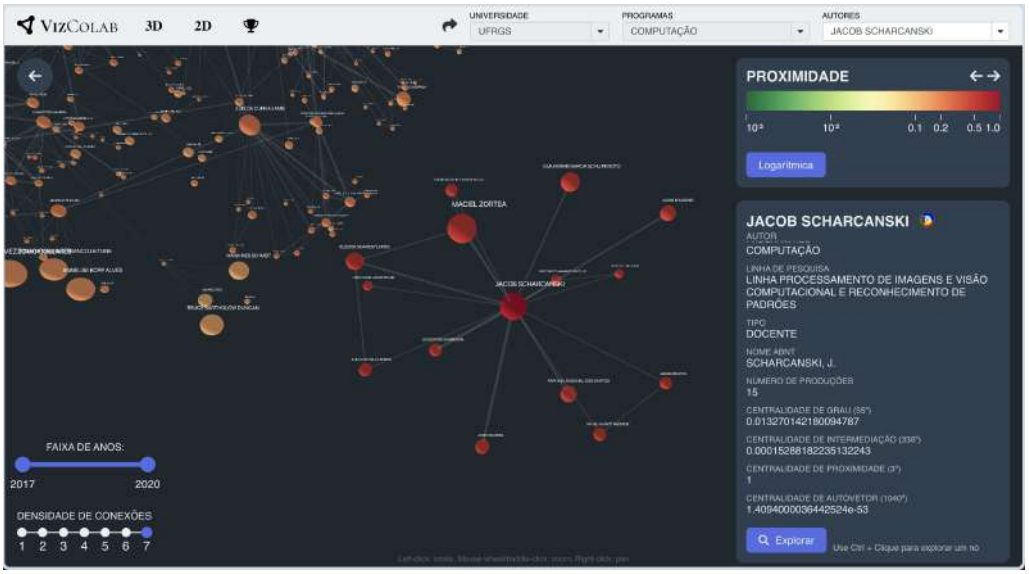


Figure 6.3 – The node and his students are diconnected from the greater computer science network of UFRGS if we take into consideration only the top 7 connections of every author.

the cluster reveals that asides from the central node, no other node in it corresponds to a UFRGS researcher, with the large majority of them being students.

6.3 Identifying unique and common traits of authors and students networks

If we go into the graphs for individual authors they are rendered with greatly disparate centralities to others in most cases, as expected, because those are graphs whose premise is that nodes connected with the author. One example can be seen on the graph for Juliano Araujo Wickboldt in figure 6.4. A thing of note, however, is that such an effect does not seem to happen with students, which is shown on the same figure for the graph of Gustavo Herminio de Araujo, a student. Their graphs show much more evenly distributed centralities for the nodes. That's a reflection of the fact that they have small amounts of articles published and that those articles are usually in co-authorship with the same group of researchers, that are themselves very connected to each other. It can happen, however, that the graph of a researcher shows other nodes with very similar centralities, that is the case to a lesser degree with Cristiano Bonato Both and Lisandro Zambenedetti Granville in Juliano's own graph too. That can be attributed to the fact that Lisandro, who is one of the most prolific authors from UFRGS according to our data, was Juliano's thesis advisor and one of his main collaborators early on. Cristiano was the coordinator of a project where Juliano acted as a postdoctoral researcher. To a greater degree, that effect is seen in figure 6.5. That seems to indicate both authors publish disproportionately often together. VizColab makes publication patterns such as those evident. They can be

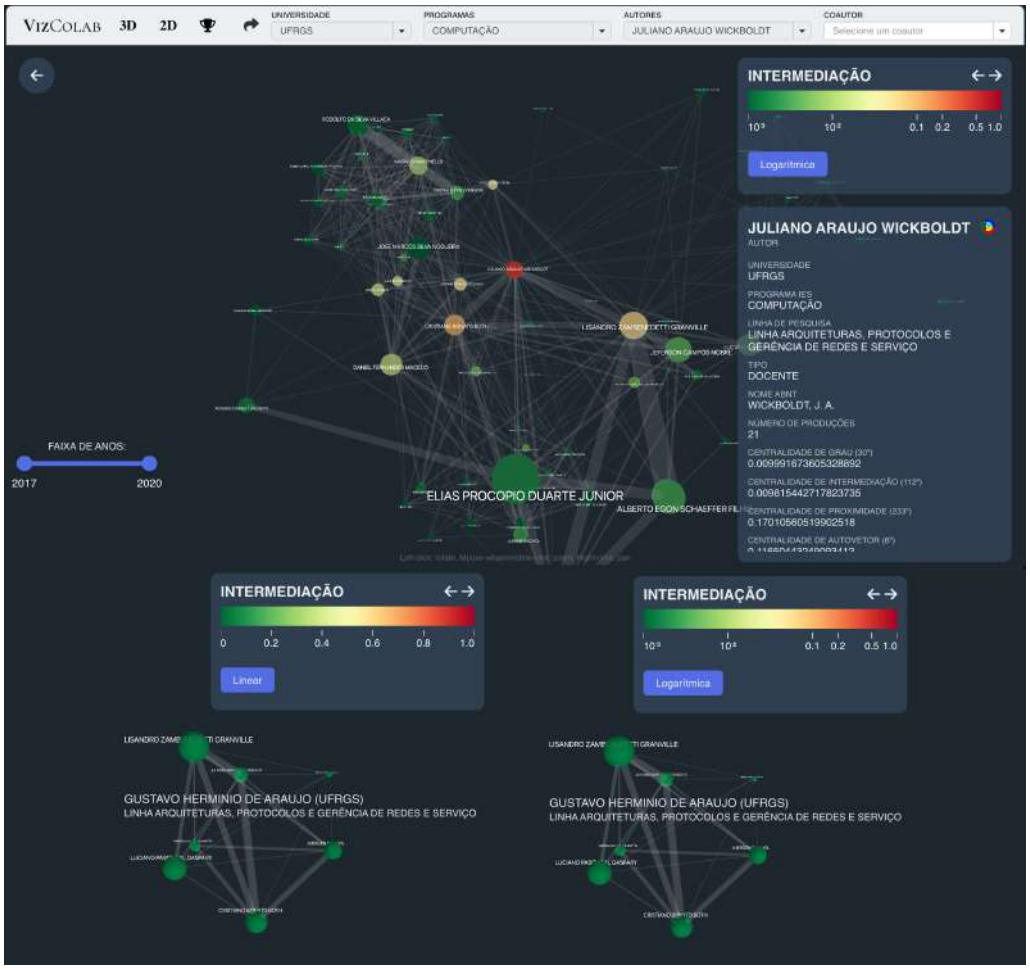


Figure 6.4 – A researcher's own collaboration network usually shows them with relatively high centrality. That is contrasted with student graphs which are more homogeneous in that regard.

Figure 6.5 – Two researchers that publish very often together.

of great interest to officials in governmental agencies that oversee educational institutions and seek to understand the evolution of research within the country, for example.

6.4 Quantifying the relative participation of researchers in different programs

In the data-set, the intellectual productions of nodes aren't differentiated according to the context of what postgraduate program they were published in. Moreover, researchers with large corpuses tend to participate in more than one program. Thus, that data point in isolation is not sufficient to fully grasp the participation of a researcher in any one specific program. We've found however, that combining the centrality metrics with the production counts provides a good approximation of the relative participation of a researcher in different programs. Figure 6.6 shows a researcher that specializes in microelectronics and participates in both the more general postgraduate program in Computing and the more focused Microelectronics program. He contributes significantly to both programs, but is much more central to the Microelectronics one. This can be explained by his known field of expertise.

6.5 Identifying outstanding data for each year

Combining the time filtering feature with the table or ranking view, it's possible to survey the top researchers for each year according to the metrics made available. Doing so reveals that the metrics don't always correlate. Some researchers may hold important

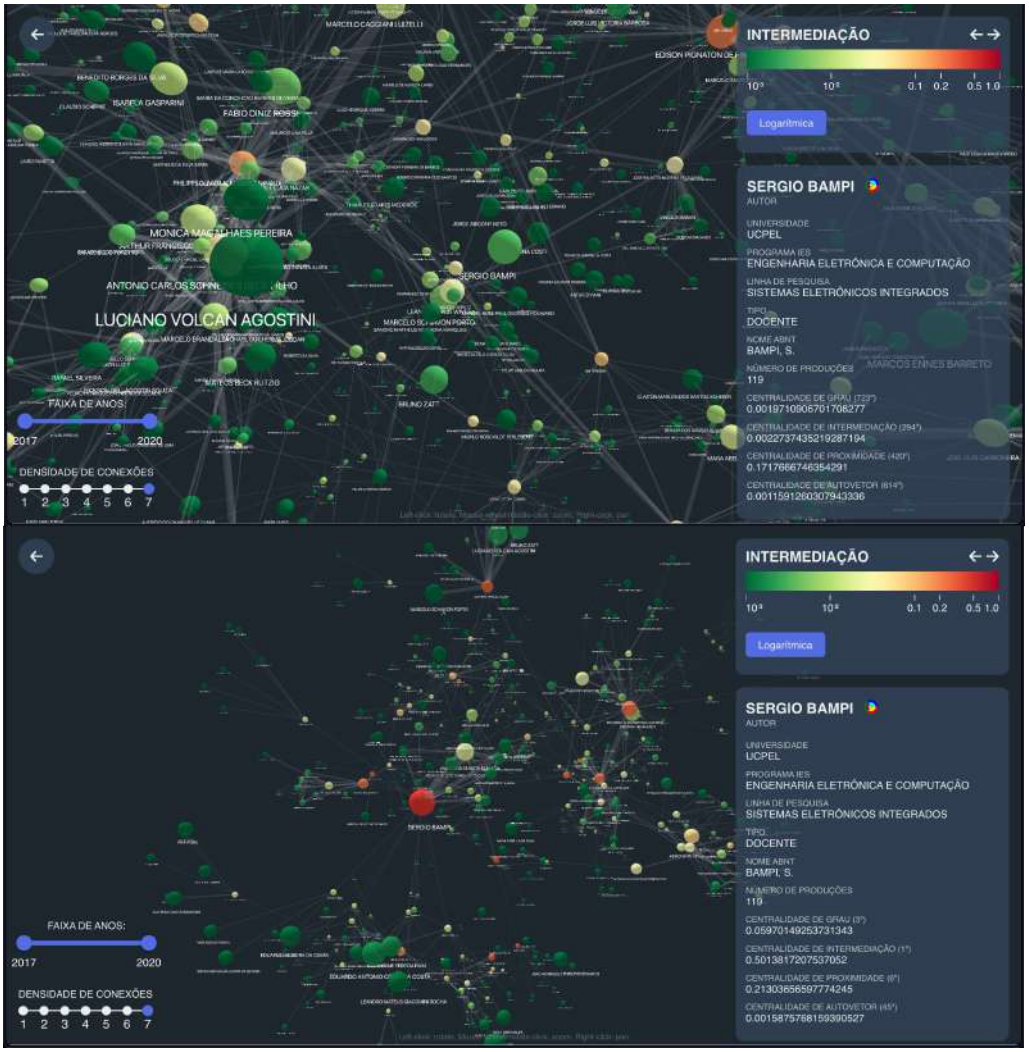


Figure 6.6 – Researchers that participate in many programs can have their specialties highlighted by the centrality metrics. They're relative participation in each program may be quantatively compared considering such metrics.

#	Autor	Produções	Centralidade de I...	Centralidade de Grau	Centralidade de Prox...
1	LISANDRO ZAMBENEDETTI GRANVILLE	27	0.24166223687255994	0.06657608695652174	0.23210214318285455
2	LUCIANO PASCHOAL GASPARY	15	0.15704583592319202	0.03940217391304348	0.21854873336195793
3	LUCIANA PORCHER NEDEL	6	0.14547876709402263	0.01358695652173913	0.16932801064537592
4	CARLA MARIA DAL SASSO FREITAS	6	0.13913934865836822	0.014945652173913044	0.20665854648802273
5	EDISON PIGNATON DE FREITAS	18	0.11346534249317013	0.03940217391304348	0.18289615522817104

#	Autor	Produções	Centralidade de I...	Centralidade de Grau	Centralidade de Prox...
1	LISANDRO ZAMBENEDETTI GRANVILLE	23	0.15689323683084433	0.050824175824175824	0.19990850869167429
2	LUCIANO PASCHOAL GASPARY	7	0.14314935465270312	0.02197802197802198	0.2092911877394636
3	ANA LUCIA CETERTICH BAZZAN	10	0.11071579727013015	0.016483516483516484	0.17721005677210055
4	LEANDRO AVILA DE AVILA	1	0.1002466292899666	0.005494505494505495	0.1958762886597938
5	PHILIPPE OLIVIER ALEXANDRE NAVAUX	17	0.09703296715470593	0.024725274725274724	0.17742590336987413

#	Autor	Produções	Centralidade de I...	Centralidade de Grau	Centralidade de Prox...
1	LISANDRO ZAMBENEDETTI GRANVILLE	23	0.16924661249544837	0.04417670682730924	0.1864202490960225
2	GABRIEL LUCA NAZAR	8	0.1259252868148115	0.012048192771084338	0.19888555507929703
3	JUERGEN ROCHOL	4	0.11481285460508482	0.013386880856760375	0.16854340719215402
4	ALBERTO EGON SCHAEFFER FILHO	16	0.10761232357059437	0.03748326639892905	0.18664521319388577
5	CARLA MARIA DAL SASSO FREITAS	7	0.1049436314698713	0.025435073627844713	0.14463840399002495

#	Autor	Produções	Centralidade de I...	Centralidade de Grau	Centralidade de Prox...
1	PHILIPPE OLIVIER ALEXANDRE NAVAUX	23	0.22294797927506974	0.03734439834024896	0.20389507154213038
2	LUCIANO PASCHOAL GASPARY	8	0.18118248427897848	0.017980636237897647	0.20668815471394036
3	LISANDRO ZAMBENEDETTI GRANVILLE	19	0.16429004455265772	0.05255878284923928	0.17893268224625045
4	EDISON PIGNATON DE FREITAS	23	0.15992174980374224	0.040110650069156296	0.14652956298200515
5	DALVAN JAIR GRIEBLER	14	0.15854173285430367	0.008298755186721992	0.19953325554259044

Figure 6.7 – The top five researchers by betweenness centrality for each year, from top to bottom 2017 to 2020.

positions with few publications, an extreme case of this is shown in figure 6.7, where a researcher occupies one of the most central positions in the network having published a single paper in the year of 2018.

7 FINAL CONSIDERATIONS AND FUTURE WORK

This article presented the enhancements made to the original VizColab, a graph visualization tool designed to explore co-authorship networks of scientific articles from all universities and postgraduate programs in Brazil. These enhancements focused on improving the scalability, usability, and interactive capabilities of the tool. In particular, we have presented the implementation of a Kubernetes cluster to facilitate the tool's deployment and scaling, and we have described new features such as time slice filtering, adapted InterRing visualization, two-dimensional view, node search and focus, sharing visualizations, a table view, and coloring nodes by centrality measures.

The time slice filtering feature provides users with the ability to explore the network within specific time periods, enabling the examination of temporal trends in co-authorship networks. The adapted InterRing visualization in VizColab provides a comprehensive view of an author's co-authorship history over time. This adaptation focuses on the quantity and persistence of collaborations rather than inferring the relative contributions of co-authors. A unique strategy to color the arcs, described in 5.3.1, ensures clarity and consistency in visualizing co-authors. The addition of two-dimensional and tabular views provide users with alternative visualization methods that are less resource-intensive, more easily reproduced in documents and interacted with.

We have also implemented a search and focus feature to facilitate finding specific nodes within the network, and a sharing feature that allows users to share their visualizations, including their specific settings and node positions, via a unique URL. The coloring of nodes by centrality measures allows users to quickly identify key actors within the network based on different centrality metrics.

The enhancements presented in this article significantly improve the user experience and expand the utility of the VizColab tool for exploring co-authorship networks. The tool can be useful for researchers, academic institutions, and funding agencies seeking to understand research collaborations and patterns within the academic community in Brazil.

For future work, we suggest several possible directions:

- **Implement Advanced Filtering Options:** Providing users with more advanced filtering options, such as selecting nodes based on specific attributes or centrality metrics, would allow for more targeted exploration of the network.
- **Implement Additional Centrality Measures:** While we have implemented color-

ing by degree, closeness, betweenness, and eigenvector centralities, there are other centrality measures, such as harmonic centrality or Katz centrality, that could provide further insights into the network. Now that the groundwork has been done, integrating new centralities is straightforward.

- **Incorporate Community Detection Algorithms:** Community detection algorithms could be used to identify and visualize clusters of researchers within the network, highlighting natural research communities or collaborations.
- **Improve Camera Rotation Recreation:** As noted in section 5.6.4, the current sharing feature does not recreate camera rotations. Finding a way to record and replay camera rotations would enhance the accuracy of shared visualizations.
- **Explore Relationship Details:** Both (HUANG; HUANG, 2006) and (KUROSAWA; TAKAMA, 2012) provide more detailed information about the articles considered in the composition of the networks displayed in the tools they developed. Whether it's a list of articles in the case of (HUANG; HUANG, 2006), or a list of keywords used in the case of (KUROSAWA; TAKAMA, 2012). Considering the need for a deeper understanding of the works carried out together, allowing the *exploration* of edges could be interesting. For lower connection densities(2), the feature of performing a *ctrl click* on an edge to reveal a panel with the list of productions that compose it could be made available.
- **Mixing Data from Multiple Universities:** Enhance the ability to combine data from more than one university in the author graph level, or offer more opportunities to combine data in different ways in the graphs.

As scientific research becomes increasingly collaborative, tools such as VizColab that facilitate the exploration and understanding of co-authorship networks will continue to be of great value to the research community.

REFERENCES

ABBASI, A.; ALTMANN, J.; HOSSAIN, L. Identifying the effects of co-authorship networks on the performance of scholars: A correlation and regression analysis of performance measures and social network analysis measures. **Journal of Informetrics**, Elsevier, v. 5, n. 4, p. 594–607, 2011.

ABBASI, A.; ALTMANN, J.; HWANG, J. Evaluating scholars based on their academic collaboration activities: Two indices, the rc-index and the cc-index, for quantifying collaboration activities of researchers and scientific communities. **Scientometrics**, v. 83, p. 1–13, 04 2010.

ABBASI, A.; HOSSAIN, L.; LEYDESDORFF, L. Betweenness centrality as a driver of preferential attachment in the evolution of research collaboration networks. **Journal of Informetrics**, v. 6, n. 3, p. 403–412, 2012. ISSN 1751-1577. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S175115771200003X>.

BONACICH, P. Power and centrality: A family of measures. **American journal of sociology**, University of Chicago Press, v. 92, n. 5, p. 1170–1182, 1987.

CAPES. **CAPES Dados Abertos**. 2023. <https://dadosabertos.capes.gov.br/>. Acessado em: 16/03/2023.

FISCHER, E. S. **VizColab: Visualização de uma rede de colaboração acadêmica brasileira de larga escala gerada a partir de dados da CAPES**. Porto Alegre: [s.n.], 2022. Monografia de Bacharelado – Universidade Federal do Rio Grande do Sul, Instituto de Informática. Orientador: Juliano Araújo Wickboldt.

HUANG, T.-H.; HUANG, M. L. Analysis and visualization of co-authorship networks for understanding academic collaboration and knowledge domain of individual researchers. In: **International Conference on Computer Graphics, Imaging and Visualisation (CGIV'06)**. [S.l.: s.n.], 2006. p. 18–23.

KUROSAWA, T.; TAKAMA, Y. Co-authorship networks visualization system for supporting survey of researchers' future activities. **Journal of Emerging Technologies in Web Intelligence**, v. 4, 02 2012.

LI-CHUN, Y. et al. Connection and stratification in research collaboration: An analysis of the collnet network. **Inf. Process. Manage.**, Pergamon Press, Inc., USA, v. 42, n. 6, p. 1599–1613, dec 2006. ISSN 0306-4573. Available from Internet: <https://doi.org/10.1016/j.ipm.2006.03.021>.

MILOJEVIć, S. Modes of collaboration in modern science: Beyond power laws and preferential attachment. **Journal of the American Society for Information Science and Technology**, v. 61, n. 7, p. 1410–1423, 2010. Available from Internet: <https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.21331>.

UTZERATH, C.; FERNáNDEZ, G. Shaping science for increasing interdependence and specialization. **Trends in Neurosciences**, v. 40, n. 3, p. 121–124, 2017. ISSN 0166-2236. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S0166223616301928>.

WANG, W.; WU, Y.; PAN, Y. An investigation of collaborations between top chinese universities: a new quantitative approach. **Scientometrics**, Springer, v. 98, n. 2, p. 1535–1545, 2014.

WOOLGAR, S. The identification and definition of scientific collectivities. In: ____. [S.l.]: De Gruyter Mouton, 2012. p. 233–246. ISBN 978-90-279-7743-4.

YAN, E.; DING, Y. Applying centrality measures to impact analysis: A coauthorship network analysis. **Journal of the American Society for Information Science and Technology**, v. 60, n. 10, p. 2107–2118, 2009. Available from Internet: <https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.21128>.