DEPARTMENT'S DEEP STRUCTURE

FACULTY MENTOR: YULIY BARYSHNIKOV
GRAD STUDENT MENTOR: SUJEET BHALERAO
DEV PATEL, PEIYAN WU, YIZHI ZHANG, ZHEYU ZHANG

ABSTRACT. The goal of this project was to make a Python program that accepts as input a list of mathematicians, say from a department at any institution, and detects intrinsic research clusters (i.e. groups of people working in close areas, publishing in the same journals etc.) within the department.

CONTENTS

1. INTRODUCTION

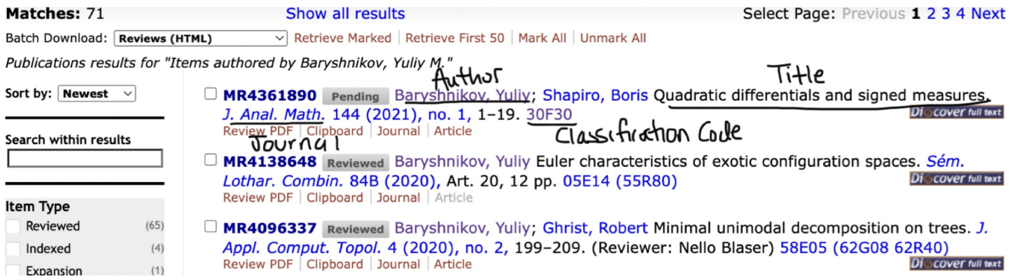
What is the structure of the math department? The question is not as simple as it seems: the research group composition changes fast, and their impact on the global scale might be smaller or larger than it seems. The goal of this project is to detect research clusters in our math department. There are two parts in achieving this goal: data collection and clustering. For data collection, we extracted information for each paper for each faculty member such as references, citations, and classifications from MathSciNet. From this data, we generated matrices that were used for clustering algorithms. This clustering is based on data scraped from the online database MathSciNet using the Python programming language. The data we collected allowed us to define proximity measures (that is, a notion of how close faculty members are), of which we applied hierarchical clustering algorithms to gain insights into the research structure of our math department. Hierarchical clustering works by initially treating each observation as a separate cluster. Then, it repeatedly identifies two clusters that are closest together and merges the two most similar clusters. This iterative process continues until all the clusters are merged together. Finally, we aggregated and visualized the results of clustering using phylogenetic trees (dendrograms).

This report is structured as follows. Section 2 describes the process of data collection and lists all datasets generated from MathSciNet. This includes an example of a code snippet, and a sample submatrix of a full distance matrix for one of the datasets. Section 3 illustrates the theory behind the clustering algorithms that were used, and includes a discussion of the necessary preprocessing of data. The method for combining results from different datasets into a unified result as a single consensus tree is also described. Section 4 reports the research clusters that were found in the math department and the final result is stated in the form of a dendrogram. Section 5 talks about the future directions of this project.

2. DATA COLLECTION

We used python packages Selenium and BeautifulSoup to extract data from MathSciNet. Selenium helps automate browser actions such as clicks, while BeautifulSoup helps to find HTML elements in a given page. An example of a search result on MathSciNet for publications from a faculty member is given below. Each piece of information such as author name or classification code is associated with an HTML element that is used to extract the information. HTML elements are a set of tags and attributes that provide definitions to different parts of the web document. They tell the web browser how to display the different pieces of information.

Date: May 13, 2022.



Matches: 71 Show all results Select Page: Previous **1** 2 3 4 Next

Batch Download: Reviews (HTML) Retrieve Marked | Retrieve First 50 | Mark All | Unmark All

Publications results for "Items authored by Baryshnikov, Yuliy M."

Sort by: Newest

Search within results

Item Type

□ Reviewed (65)

□ Indexed (4)

□ Expansion (1)

□ **MR4361890** Pending Baryshnikov, Yuliy; Shapiro, Boris Quadratic differentials and signed measures. *J. Anal. Math.* 144 (2021), no. 1, 1–19. 30F30
Review PDF | Clipboard | Journal | Article

□ **MR4138648** Reviewed Baryshnikov, Yuliy Euler characteristics of exotic configuration spaces. *Sém. Lothar. Combin.* 84B (2020), Art. 20, 12 pp. 05E14 (55R80)
Review PDF | Clipboard | Journal | Article

□ **MR4096337** Reviewed Baryshnikov, Yuliy; Ghrist, Robert Minimal unimodal decomposition on trees. *J. Appl. Comput. Topol.* 4 (2020), no. 2, 199–209. (Reviewer: Nello Blaser) 58E05 (62G08 62R40)
Review PDF | Clipboard | Journal | Article

We extracted faculty names, references, citations, MathSciNet classification codes, and journal names for each publication of each faculty member in the department. Next, we generated similarity matrices (number of common journals, citations, references, etc). Finally, we also created matrices for SVD-based clustering method with respect to classification codes, references, and citations.

The code snippet below is used to generate the references matrix describing the number of common references each pair of faculty have. First, we enter names of professors and starting year in the search bar. We then click on the first research paper for the author using the class name property. We use time.sleep(0.4) between each action to suspend execution of the code for 0.4 seconds in order to prevent an overload of requests in a small amount of time.

```
for professor in list_of_profs:
    time.sleep(0.4)
    driver.find_element_by_css_selector("input[type='radio'][value='pubyear']").click()
    time.sleep(0.4)
    select = Select(driver.find_element_by_id('yrop'))
    time.sleep(0.4)
    select.select_by_visible_text('>')
    time.sleep(0.4)
    driver.find_element_by_id("yearValue").send_keys("2010")
    time.sleep(0.4)
    driver.find_element_by_name("s4").send_keys(professor)
    time.sleep(0.4)
    driver.find_element_by_name("Submit").click()
    time.sleep(0.4)
    driver.find_element_by_class_name("mrnum").click()
```

Second, we find the HTML element for references and gather all references for all papers for one author. Each reference is listed as an 'MR' or 'ar' code.

```
    listreferences = []
    time.sleep(0.2)
    if (check_exists_by_class_name()):
        soup = BeautifulSoup(driver.page_source, "html.parser")
        spans = soup.find_all('a', href = True)
        for word in spans:
            if (word.get_text()[0:2] == "MR" or word.get_text()[0:2] == "ar"):
                listreferences.append(word.get_text())
```

We keep on adding these references until there is no next paper for the author.

```
    while (True):
        time.sleep(0.2)
        try:
            driver.find_element_by_partial_link_text("Next").click()
            time.sleep(0.2)
            if (check_exists_by_class_name()):
                time.sleep(0.4)
                soup = BeautifulSoup(driver.page_source, "html.parser")
                time.sleep(0.2)
                spans = soup.find_all('a', href = True)
                time.sleep(0.2)
                for word in spans:
                    if (word.get_text()[0:2] == "MR" or word.get_text()[0:2] == "ar"):
                        listreferences.append(word.get_text())
                time.sleep(0.4)
        except:
            break
```

Finally, we store the data in a dictionary with key professor and value of all of their references as MR codes.

```
profdict[professor] = listreferences
time.sleep(0.4)
```

We return back to the home page, so we can repeat the process for the next professor.

```
driver.find_element_by_link_text("Home").click()
time.sleep(0.4)
```

2.1. **List of datasets generated.** The following data sets were collected. A description of each is listed below. The first six matrices are distance matrices where the rows and columns are faculty names and each entry represents the strength of the relationship between each pair of faculty. The last two matrices are used for SVD clustering methods where the rows are faculty names while the columns are codes associated with citations, references, and classifications.

Collaboration Distance. "Collaboration distance" is obtained from a tool on MathSciNet. This distance represents minimumly how many papers away two authors are. For example, if A coauthored a paper with B, but A worked with C, C worked with D, and D worked with B, then the collaboration distance between A and B would be 3 (A-C-D-B). Similarly, collaboration distance between A and C is 1, between A and D is 2.

Number of Joint Publication. This data represents how many paper each pair of faculties co-authored. This dataset is symmetrical.

Number of Shared Citation. This data represents how many times each pair of faculties are cited in the same paper. For example, if there are 10 papers that cited both faculty A and B, then entry (A, B) of this dataset would be 10. This dataset is symmetrical.

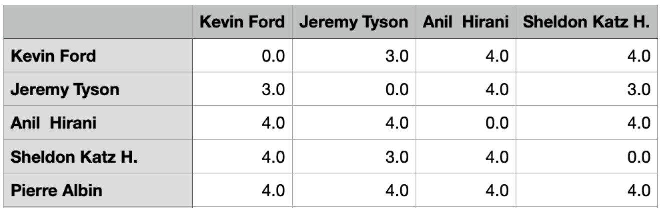
Number of Directed Citation. This data represents how many times one faculty cited another faculty in all his/her publications. For example, if A cited B 10 times in all of A’s publications, then the entry (A, B) would be 10. This dataset is NOT symmetrical.

Number of Common Journal. This data represents how many papers each pair of faculties published in same journals. For example, if A published 3 papers in journal X and 4 papers in journal Y, while B published 2 papers in journal X and 5 papers in journal Y, then entry (A, B) of this dataset would be 3+2+4+5 = 14. This dataset is symmetrical.

Number of Common Reference. This data represents how many papers each pair of faculties both refered to in their publishings. For example, if A refered to paper x, y, z in her papers, and B refered to paper w, x, y in his papers, then entry (A, B) of this dataset would be 2. This dataset is symmetrical.

Citation Matrix. This data represents how many time for any author has citation(s) for one specific paper. For example, if A’s paper has been citated by a paper called P. Then in the matrix, cell in the row A and column P has value of 1. This dataset is NOT symmetrical.

Reference Matrix. This data represents how many time for any author has reference(s) for one specific paper. For example, if A’s paper has been referenced by a paper called P. Then in the matrix, cell in the row A and column P has value of 1. This dataset is NOT symmetrical.



	Kevin Ford	Jeremy Tyson	Anil Hirani	Sheldon Katz H.
Kevin Ford	0.0	3.0	4.0	4.0
Jeremy Tyson	3.0	0.0	4.0	3.0
Anil Hirani	4.0	4.0	0.0	4.0
Sheldon Katz H.	4.0	3.0	4.0	0.0
Pierre Albin	4.0	4.0	4.0	4.0

FIGURE 1. This is an example sub-matrix for collaboration distances. Each entry represents the number of papers between each pair of faculty.

3. CLUSTERING

3.1. **Data Pre-processing.** Even without any processing, the raw data we extracted from MathSciNet naturally describes the similarity between mathematicians with respect to research areas. Take the number of shared citation as an example, the more two mathematicians are cited together, the more likely they work in the same or closely related areas. Thus, a reasonable distance measure for how closely mathematicians work together can simply be:

$$d(a,b) := \frac{1}{2^{s_{a,b}}} \tag{1}$$

Here, $s_{a,b}$ is the similarity measures between mathematicians a, b represented by the data we extracted, and $d(a,b)$ is the distance that can be used for clustering.

However, a problem with this simple approach is that different research areas can have vastly different research output. For example, in 2021, the number of reviewed publications on MathSciNet in spectral theory and eigenvalue problems is 422, while the number in random dynamical systems is only 102. As a result, our data can skew toward mathematicians in areas of high research output. That is, our simple approach will likely assign closer distances to mathematicians working in high output areas than those who work in lower output areas.

To resolve this issue, we performed normalization on the data we collected to prevent biases in clustering. For each type of metric data we collected, we normalized the data with respect to the total number of corresponding metric of each mathematician. Take the number of shared citation as an example, we define the normalized similarity measure $\hat{s}_{a,b}$ as following:

$$\hat{s}_{a,b} = \frac{s_{a,b}}{\sqrt{a_{\text{total}} \cdot b_{\text{total}}}} \tag{2}$$

Here, $s_{a,b}$ is the number of shared citation between mathematicians a, b, and $a_{\text{total}}, b_{\text{total}}$ are the total number of citations a, b have respectively.

After normalization on similarity measures, distances can be calculated using equation (1).

3.2. **Clustering Algorithm.** We would like to perform clustering techniques on our data to uncover the intrinsic research structure within our math department. However, such structure is likely to exhibit a multiscale characteristic (Figure 2). For example, a group of mathematicians working on partial differential equations can form a valid cluster, but if we zoom in, there may be a finer cluster that works on representations of solutions, another on first-order equations, another on higher-order equation, and so on.

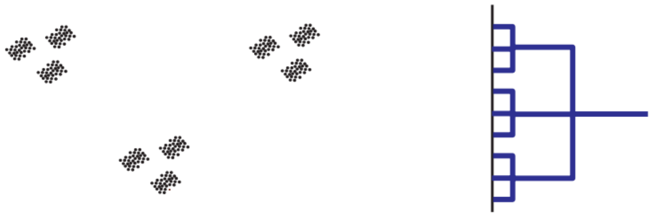


FIGURE 2. Data with multiscale structures [2]

This makes standard clustering methods such as K-Mean unsuitable for our task, as they only return a fixed partition of data. Thus, the main clustering method we used on analyzing our data is agglomerative hierarchical clustering. Unlike standard clustering algorithms, this method returns a hierarchical decomposition of input data [2]. Roughly speaking, this is a method of clustering in which each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy. The precise algorithm for merging clusters depends on the linkage criterion one uses, which is the functions that define distances between clusters. The linkage criteria we used is unweighted average linkage, also know as UPGMA:

Given distance d. For clusters A and B,

$$D(A,B) = \frac{1}{|A| \cdot |B|} \sum_{a \in A} \sum_{b \in B} d(a,b) \tag{3}$$

Given a linkage criteria $D(A,B)$ that defines the distance between any pair of clusters (A,B), the clustering algorithm is then described below:

Algorithm:

(1) Initialize all data points as clusters of one element.
(2) Compute distance $D(A,B)$ between all clusters.
(3) Find the shortest distance r between any pair of clusters.
(4) For each pair of cluster (A,B) such that $D(A,B) = r$, merge them into a single cluster.
(5) Save the current state labeled with r.
(6) Repeat step 2-5 until every data point is merged into a single cluster.

This algorithm produces an hierarchy of clusters, which can be represented as a tree (or dendrogram) of clusters.

3.3. **Tree Consensus.** After performing hierarchical clustering on multiple sources of data, we would like to determine the cluster structure that is supported by multiple different sources. That is, we would like to combine the result trees obtained from multiple sources of data into one representative tree.

To accomplish this, we deployed the majority rule consensus tree method. A majority rule tree contains exactly those clusters or splits that appear in more than half of the input trees [1]. A more detailed example of majority consensus tree can be found in [1]. This is implemented using Bio.Phylo[3].

3.4. **SVD-based Clustering.** We would also like to experiment with SVD-based clustering methods. Specifically, we want to apply singular value decomposition to the citation and reference matrices, and extract the most dominant principle components or features, then perform K-mean clustering on these features [4]. This can give us an insight into the department's structure at some specific scale.

4. RESULTS

We performed clustering using the aforementioned hierarchical clustering method on all the data we collected. All result trees can be found in Appendix A. From these results, we found that joint publication data is very sparse and only produces a few clusters, as it is rare for even mathematicians in the same areas to always work on the same papers. Common journal data gives a reasonable but not very accurate result, as many major journals cover a very broad range of topics. After comparing the rest of results with known research groups within our department, we determined that the clustering results from citations and references data are the most accurate.

We then combined the results from normalized joint citation and common references data using majority consensus tree method (Figure 3). From this result, we see that most current research groups within our department are well identified. A few individuals are misplaced by our algorithm, but mainly due to reasons that are impossible for the algorithm to resolve, such as one changing area of research recently, or one only recently became an UIUC faculty.

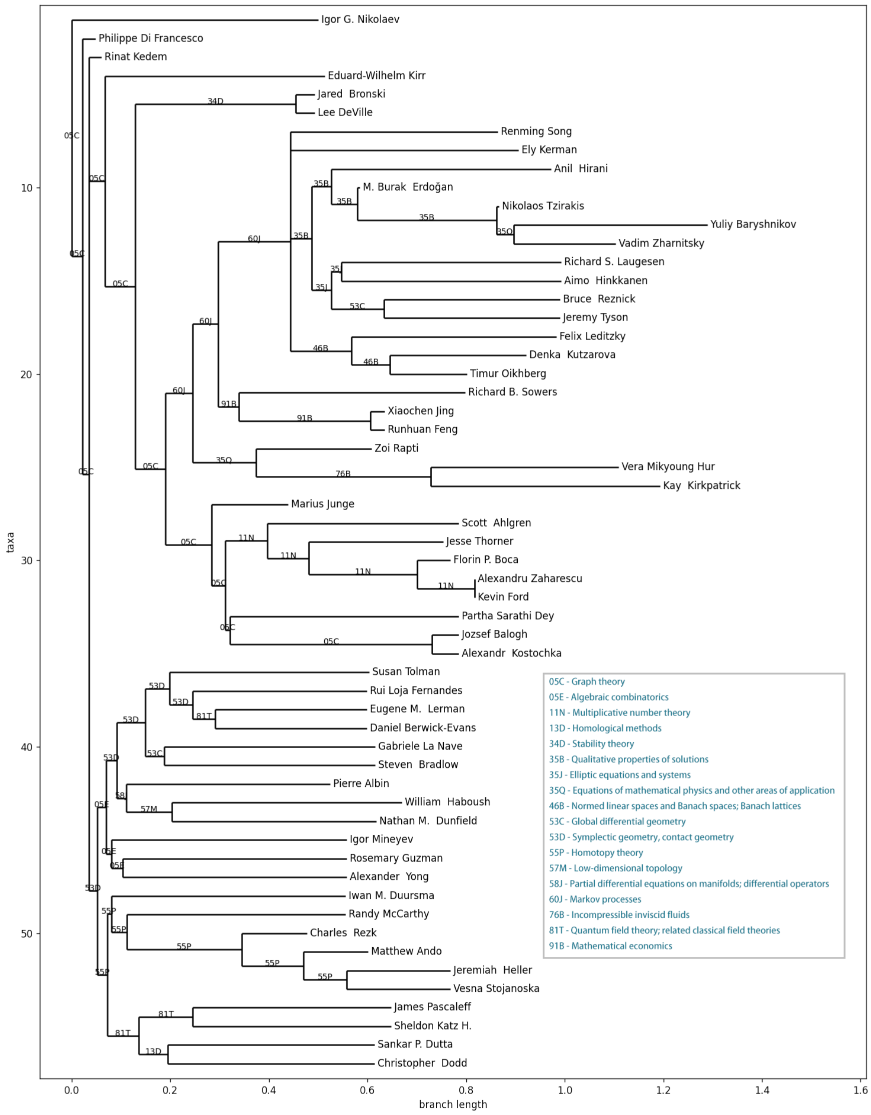


FIGURE 3. Consensus tree from hierarchical clustering results of joint citation and common references data. The branch labels are MathSciNet classification codes (MSC) of the research areas that the corresponding sub-clusters published most in.

5. CONCLUSION AND FUTURE WORK

In this project, we created a program that successfully detected intrinsic research clusters within Department of Mathematics at University of Illinois at Urbana-Champaign. Moreover, since our final program doesn't require any human labeling or selection, it can automatically produce similar results for any other U.S. institutions in MathSciNet's database. This provides a very efficient way for administrations of institutions and prospective graduate students to discover and visualize the research landscape within their own department of mathematics. However, there are still a lot needs to be improved. We would like to make better data visualization for our result, and incorporate the MathSciNet classification into the visualization in a more elegant way. We would also like to extend this project to other fields such as physics and computer science research. Furthermore, we want to integrate the data collection and cluster analysis parts into a single package for a better user experience. Additionally, we can integrate other online databases such as zbMath (formerly Zentralblatt MATH) as additional data sources. Finally, we want to explore SVD-based clustering methods to obtain finer classifications on our data.

References

[1] David Bryant. "A classification of consensus methods for phylogenetics". In: Apr. 2003, pp. 163–183. ISBN: 9780821831977. DOI: 10.1090/dimacs/061/11.

[2] Gunnar Carlsson and Facundo Mémoli. "Characterization, Stability and Convergence of Hierarchical Clustering Methods." In: *Journal of Machine Learning Research* 11 (Apr. 2010), pp. 1425–1470.

[3] Peter JA Cock et al. "Biopython: freely available Python tools for computational molecular biology and bioinformatics". In: *Bioinformatics* 25.11 (2009), pp. 1422–1423.

[4] Chris Ding and Xiaofeng He. "K-Means Clustering Via Principal Component Analysis". In: *Proceedings, Twenty-First International Conference on Machine Learning, ICML 2004* 1 (Sept. 2004). DOI: 10.1145/1015330.1015408.

6. Appendix: Result Dendrograms

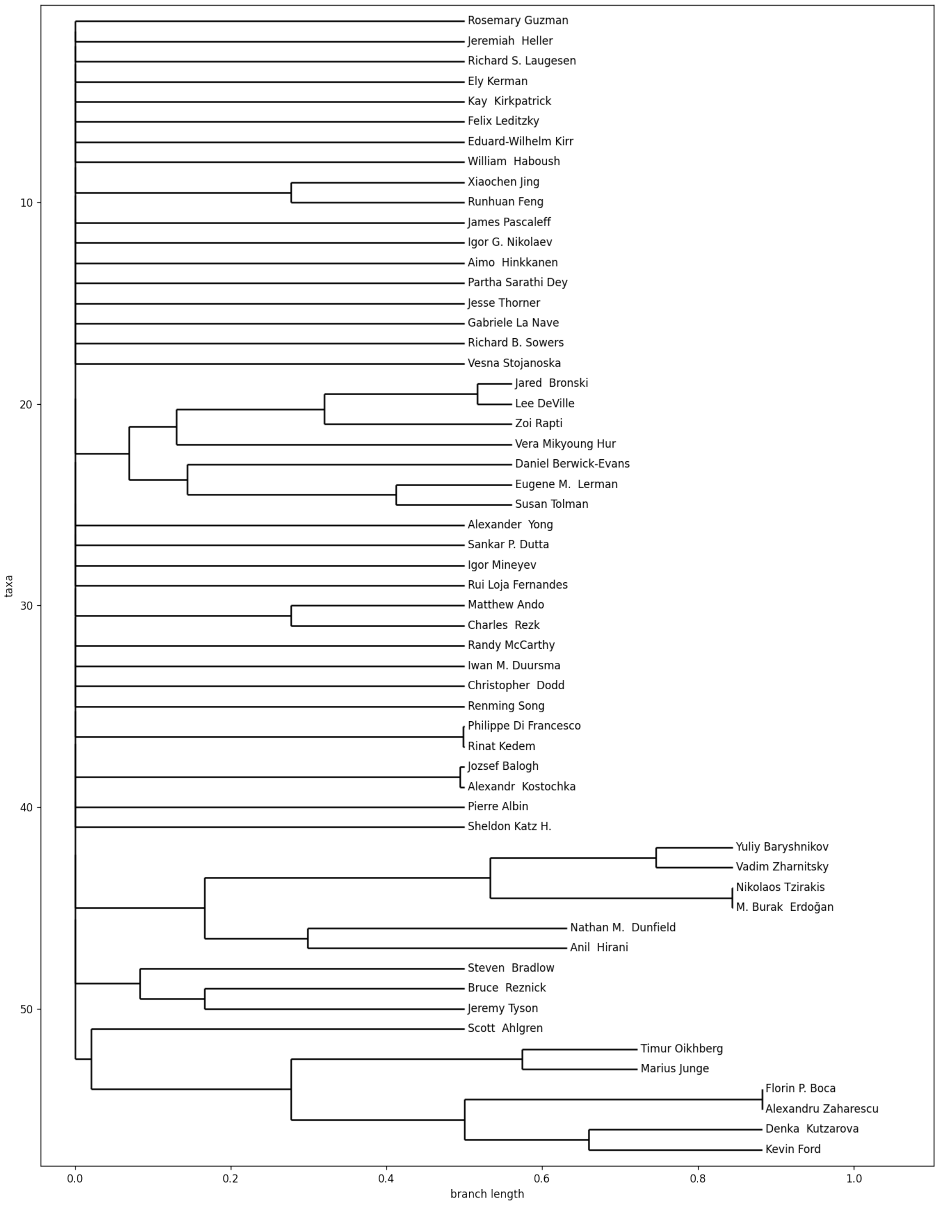


Figure 4. Hierarchical clustering result from joint publication data

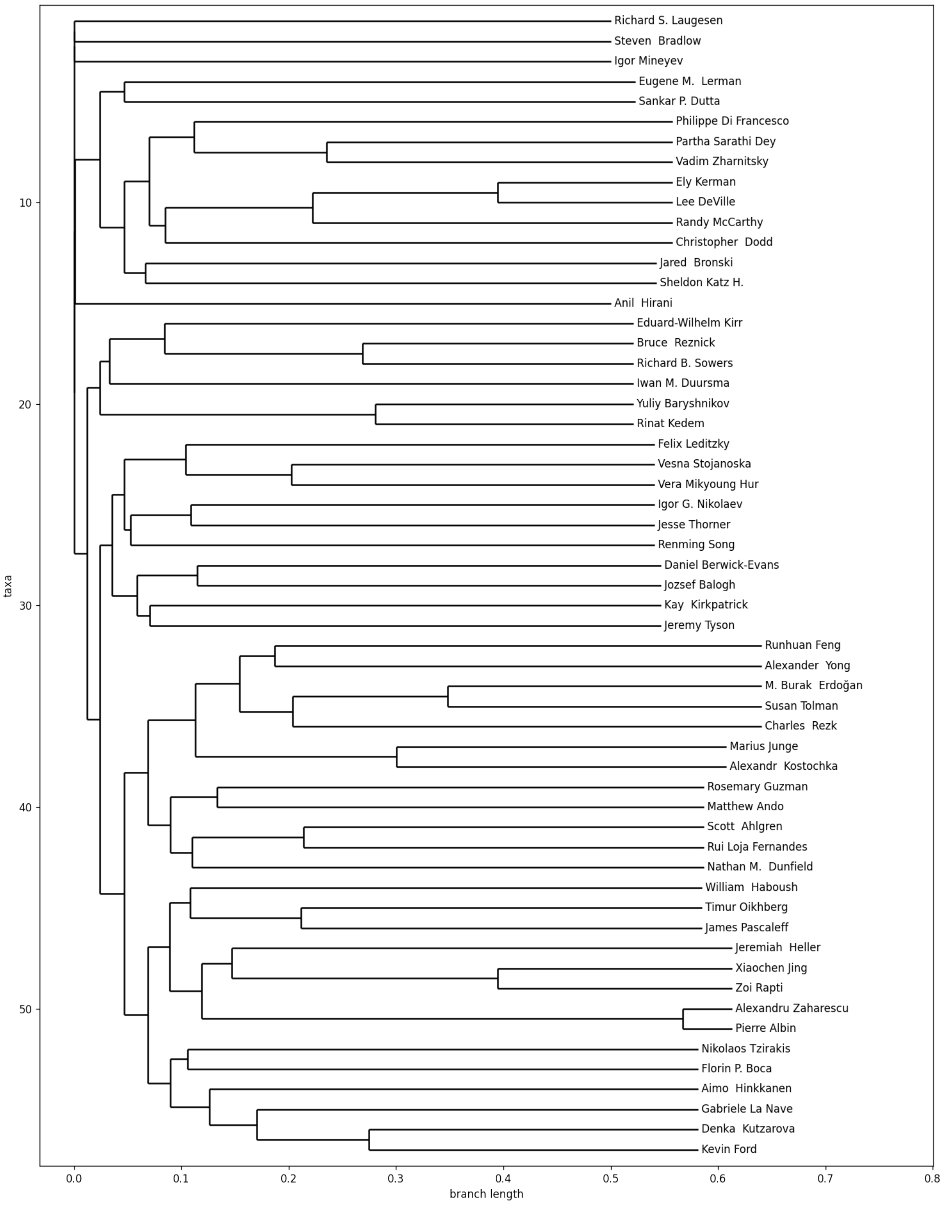


FIGURE 5. Hierarchical clustering result from direct citation data

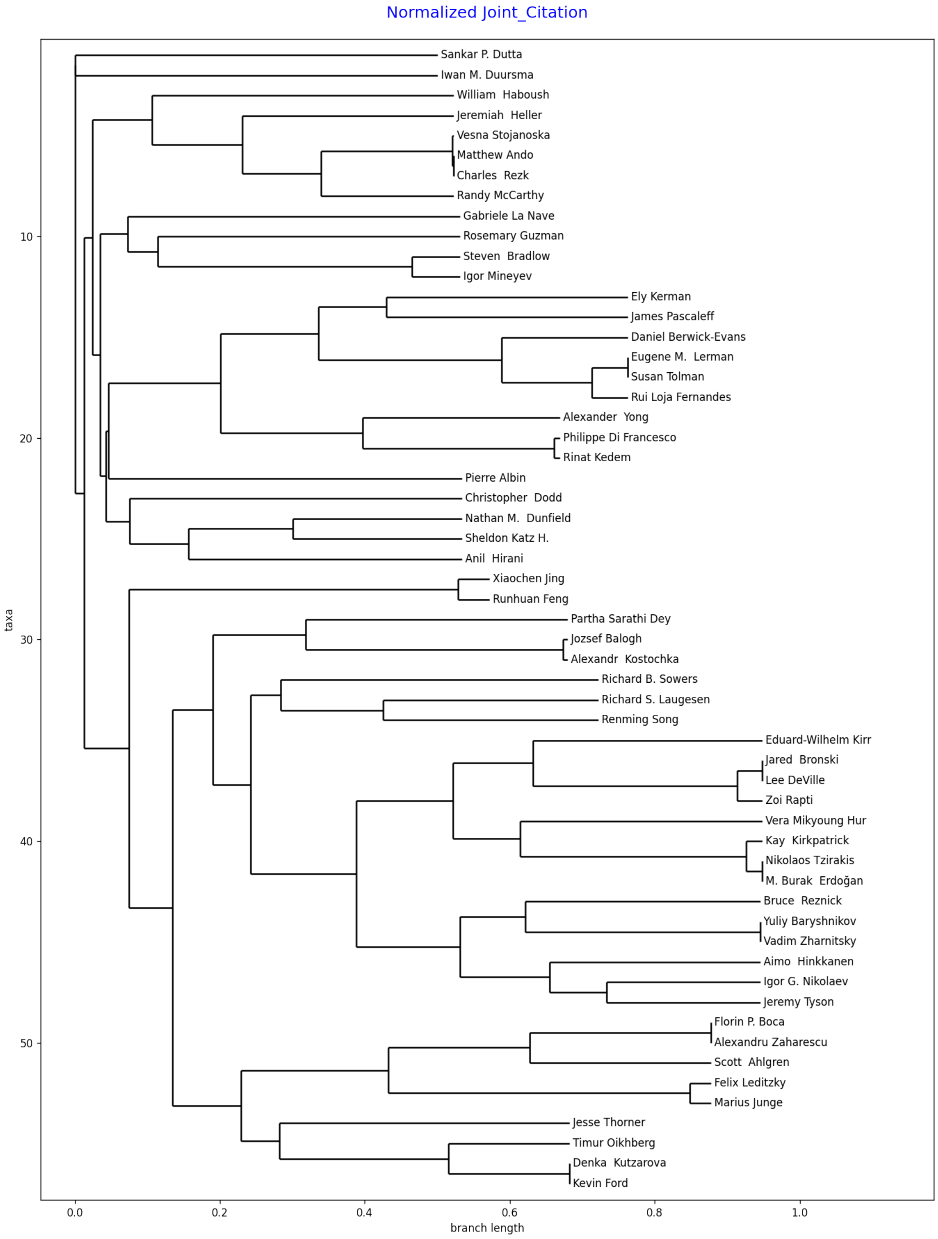


FIGURE 6. Hierarchical clustering result from shared citation data

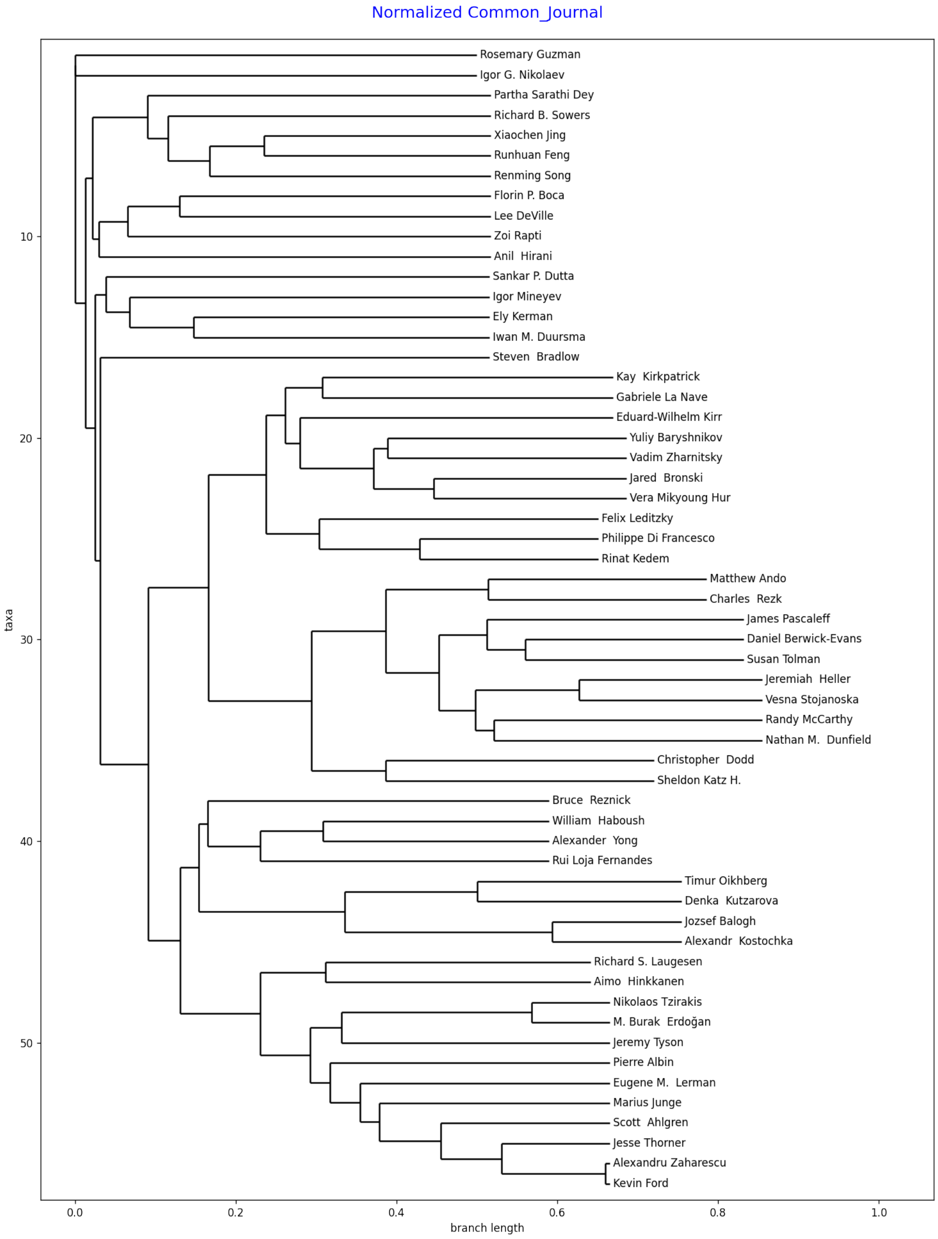


FIGURE 7. Hierarchical clustering result from common journal data

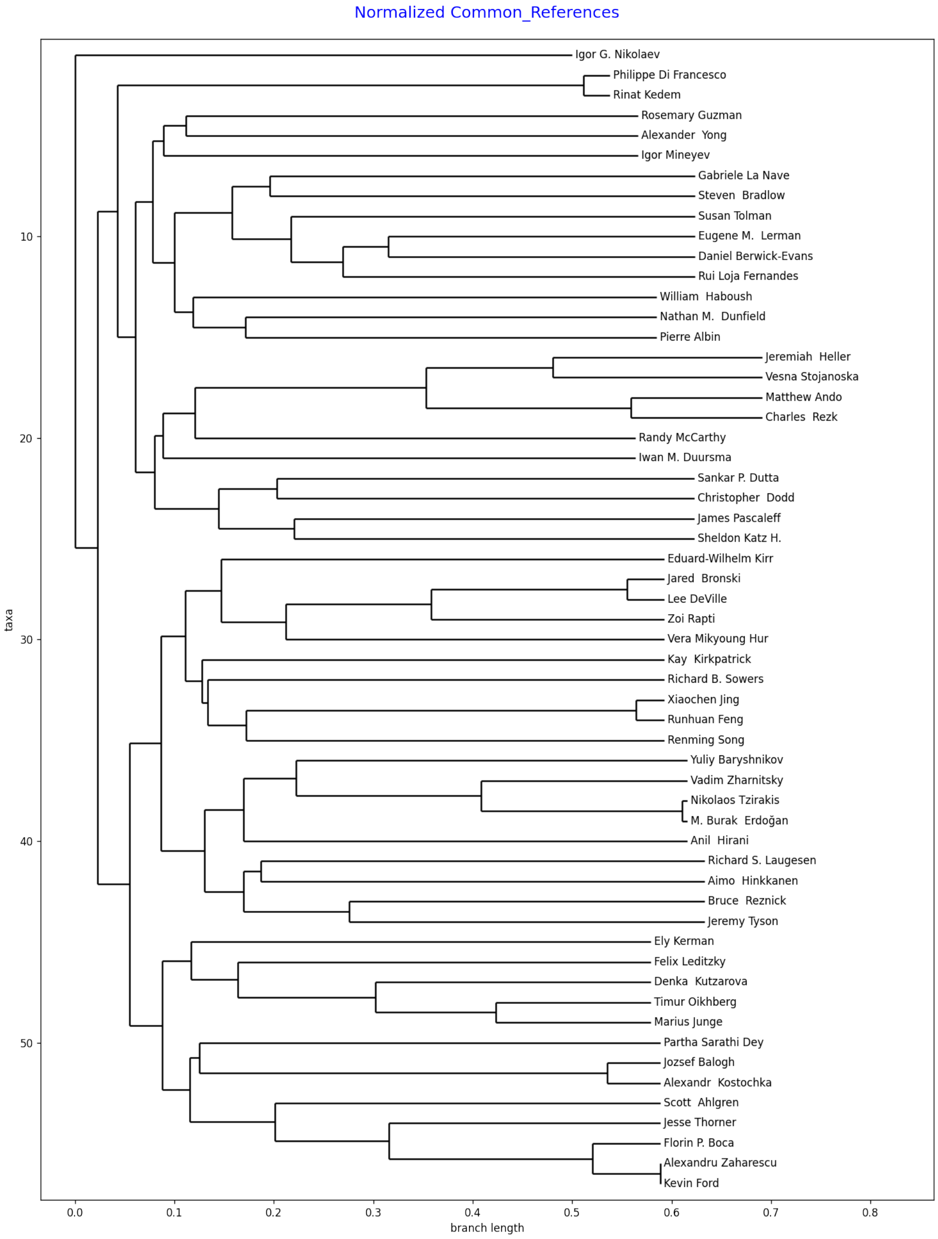


FIGURE 8. Hierarchical clustering result from common reference data

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN