NETWORK AND TOPOLOGICAL ANALYSIS OF SCHOLARLY META-DATA: A PLATFORM TO MODEL AND PREDICT COLLABORATION

by

Lance Novak

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Master of Science in Biomedical Engineering

Weldon School of Biomedical Engineering

West Lafayette, Indiana

August 2019

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Tamara Kinzer-Ursem, Co-Chair

Weldon School of Biomedical Engineering

Dr. Pete Pascuzzi, Co-Chair

Libraries and School of Information Studies

Dr. Jacqueline Linnes

Weldon School of Biomedical Engineering

Approved by:

Dr. George R. Wodicka

Head of the Graduate Program

Dedication

I would like to thank my advisors Dr. Tamara Kinzer-Ursem and Dr. Pete Pascuzzi for their guidance, and the opportunity given to me by the Information and Library Studies and Biomedical Engineering departments. I would like to thank my family, Peter, Renee, Tyler, Signe, and Keisha for their love and support. I would like to thank Hunter, Emilie, George, Barrett, Andy, and Madison for their unwavering friendship. I'm certain that each of your contributions have had a substantial impact on my ability, opportunity, and this work. I'm ever grateful to each of you.

ACKNOWLEDGMENTS

I would like to acknowledge Dr. Kinzer-Ursem for setting me on the path to this work. I would like to acknowledge Dr. Pascuzzi, who provided the proper tools to pursue this work in the fullest. I would like to acknowledge Lovepreet Singh, my predecessor who helped lay the foundation of this work.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

Author: Novak, Lance, C. MS
Institution: Purdue University
Degree Received: August 2019
Title: Network And Topological Analysis of Scholarly Meta-Data: A Platform To Model and Predict Collaboration
Committee Chair: Tamara Kinzer-Ursem; Pete Pascuzzi

The scale of the scholarly community complicates searches within scholarly databases, necessitating keywords to index the topics of any given work. As a result, an author's choice in keywords affects the visibility of each publication; making the sum of these choices a key representation of the author's academic profile. As such the underlying network of investigators are often viewed through the lens of their keyword networks. Current keyword networks connect publications only if they use the exact same keyword, meaning uncontrolled keyword choice prevents connections despite semantic similarity. Computational understanding of semantic similarity has already been achieved through the process of word embedding, which transforms words to numerical vectors with context-correlated values. The resulting vectors preserve semantic relations and can be analyzed mathematically. Here we develop a model that uses embedded keywords to construct a network which circumvents the limitations caused by uncontrolled vocabulary. The model pipeline begins with a set of faculty, the publications and keywords of which are retrieved by SCOPUS API. These keywords are processed and then embedded. This work develops a novel method of network construction that leverages the interdisciplinarity of each publication, resulting in a unique network construction for any given set of publications. Post-construction the network is visualized and analyzed with topological data analysis (TDA). TDA is used to calculate the connectivity and the holes within the network, referred to as the zero and first homology. These homologies inform how each author connects and where publication data is sparse. This platform has successfully modelled collaborations within the biomedical department at Purdue University and provides insight into potential future collaborations.

1. INTRODUCTION

The cumulative work of researchers, scientists, and their collaborative work drive invention and innovation. The network of these collaborators is extensive, unpredictable, and not rooted in any institution or discipline. As such, the ability to discern ongoing and potential work within a field is complex and under consistent investigation. These investigations have been shown to provide industrial and academic gain [1]. Early in scholarly communication and publishing there was a need to identify and measure work within a field [2]. These measurements, called bibliometrics, result from the analysis of scholarly meta-data of any given work. The meta-data of a work consists of its citations, keywords, subject area, and other associated characteristics that define its place in a discipline. Bibliometrics provide insight into the local impact and evaluation of an article within its discipline but fails to capture the global direction and collaborations associated with that discipline. As a result, scholarly analysis has recently investigated the application of network analysis to understand the global landscape of a discipline [3]. The evolution of scholarly analysis has evolved dynamically with technology, and understanding this evolution lays the foundation for the next step in scholarly network analysis.

1.1 Bibliometrics and networks

The industry of publishing is continuously growing, and with it the companies providing predictive bibliometrics have developed a market worth 10 billion USD in 2017 [4]. In academia, these same predictive metrics are being used to curate bibliographies for researchers and provide efficient search results to advance research. In both sectors there are many efforts to model the complex interactions of researchers. However, the importance of analyzing scholarly communication has not always been a priority and is an ever-evolving field.

The first recorded measure of scholarly works is a study in 1917, where Cole and Eales developed the first systematic review in history [2]. Specifically, their goal was to outline the history of comparative anatomy from the sixteenth century to 1860, and then apply statistical methods to determine the impact of said works [5]. Others point to the work of Campbell, who studied grouping of subjects in chemistry publications in 1896 [6]. The investigation of bibliometrics

continued to evolve [2], however with the digitization of publications there was a paradigm shift in accessibility. Electronic resources provided accessible means to scientific output, and with this accessibility came an overwhelming need to organize them systematically. Scholarly databases rely on publishers, keywords, authors, and titles to catalogue literature; they continuously update citations for each piece of literature, track users, and relate articles by user popularity. These standard search engine procedures facilitate the correlation of papers, and the generation of new and proprietary bibliometrics [7]. Opposed to these proprietary metrics are open source tools. These tools range from classically simple, such as citation networks, to computationally complex, such as machine learning applications [3], [8], [9].

Common between all open source tools is the need to retrieve data and acknowledging the source of data is necessary for their evaluation and verification. Most scholarly data mining sources have a formal application programming interface (API). An API is a specified request for a developer to access data, especially so that the data mining can be formally monitored and not slow the normal use of the provider's service [10]. There are many API systems available and some example sources include Elsevier, Clarivate, and PubMed. The efficiency and standardization of scholarly API requests enables the precise data acquisition for any and all meta-data types. These precise methods permit repeatable results, and the efficient retrieval of data.

Recent co-authorship, citation, and keyword trends have been investigated by homologous and heterogeneous networks [3], [9], [11], where a homologous network consists of only one data type and a heterogeneous network consists of several. For instance, Sun *et al* show explore how authors can be linked by authorship, common collaborators, common venues, common publishers, and by citation [3]. In contrast, a citation network only consists publication nodes where edges indicate which article cited which. The complexity of the heterogeneous network provides an intricate view into scholarly communication but increases the computational complexity of network analysis. Likewise, the homogeneous network provides a simpler view of scholarly communication through a single lens but doesn't explore all the features that result from scholarly communication. Both of these methods are being pursued in the current research landscape [3], [9], [11].

1.2 Network analysis

Publications are central to representing any scholar's work, and their role in scholarly networks permeates through every subtype of network; citation, keyword, co-authorship, publisher, venue, and institutional networks all rely on publications. It follows that when representing an author's academic profile, the most important factor is their body of publications. The current application of network analysis leverages the topology of the network to explore co-authorship and expert identification [3], [12]. These topics are explored by leveraging the connections formed by publishing. Each publication has a venue in which it's discussed, a topic it covers, authors who write it, and other publications that it cites. The exploration of these systems gives insight into the current use of publication meta-data, and how each can be used to depict scholarly communication through the lens of publications.

The use of network analysis to find co-authors uses a heterogeneous network that is composed of authors, publications, venues, and topics; it then applies machine learning to determine co-author relationshiops. Authors, publications, and venues are all retrieved from a scholarly database; conversely, the topics for each publication are determined by computationally extracting information from the title of the publication [3]. After the meta-data is curated and structured, the network is built by identifying how each publication connects to the authors, venues, and topics [3]. Next, Sun *et al* investigates the degree of separation between any two authors, otherwise known as the number edges between any two authors [3]. The paths that connect authors can include shared publications, venues, colleagues, and topics. These relationships are then fed to a supervised model to predict which authors are the most likely to be co-authors [3].

More recently this group used network analysis to identify experts within a given discipline; this analysis furthers the complexity of the heterogeneous network by introducing keywords. As before, the heterogeneous network consists of authors, publications, and venues; however, the "topics" of the previous publication has been replaced with an object called "terms" [12]. "Terms" refers to the set of keywords for each publication and connects publications based on a method called concept hierarchy. In short, this work identifies the hierarchy of keywords and uses this to connect publications of varying concepts. For instance, the discipline of Biomedical Engineering has a sub-discipline called Biomolecular Engineering, and normally it wouldn't be possible for a computer

to understand this relationship. However, the local training of a word embedding makes it possible for computers to understand the hierarchy of disciplines [12]. For reference, the definition and explanation of a word embedding can be found in the technical introduction. Related concepts are connected to one another and the publications are connected to their terms, venues, and authors [12]. The resulting network is analyzed with a random walk. This analysis traverses the network by a random path and tracks the citation, number of edges, and interactions of each publication within the network [12]. This data and the concept hierarchy are then used to rank the authors within a discipline or sub-discipline. This implementation of keyword analysis facilitates the linking of semantically similar disciplines and achieving a higher resolution of expert identification [12].

1.3 Landscape summary and gap analysis

The recorded evolution of bibliometrics and analysis of scholarly communication spans the last 100 years of scientific investigation [2]; this ever-expanding investigation has recently begun to predict collaborations and experts within a field using publication meta-data [3], [12]. First beginning with a simple statistical analysis of subject clusters in Chemistry, bibliometrics have become an integral part of the publishing system [6]. Digital articles and API systems have since increased the accessibility for these analyses and laid the foundation for network analysis. Recent work in network analysis has developed both homogenous and heterogeneous networks that model scholarly communication [3], [9], [11], [12]. These models detect the underlying phenomenon by leveraging the existing connections between publications and their citations, venues, authors, and keywords. Of these meta-data types, only keywords are limited by semantic connections and meaning. Sun *et al* doesn't address the use of keywords in their network, instead choosing to focus on topics that are generated from the title [3]. Conversely, the first investigation into connecting keywords based on semantics is accomplished by Gui *et al* [12]. Their solution of concept hierarchy is the latest step in understanding the use of keywords at the intersection of natural language and mathematics [12]. However, Gui *et al* only apply their concept hierarchy to disciplines, which precludes technique-, method-, and technology-based keywords from being considered in the analysis [12].

The current landscape of scholarly network analysis has set the stage for the development of a semantically defined keyword network; more specifically, this novel network connects authors through related disciplines, techniques, methods, and technologies. The topological investigation of topics by Sun *et al* is one of the few scholarly networks to explore the basic topology [3]. Likewise, Gui *et al* applies natural language processing to understand the semantic relationship of some terms within each publication's keyword set [12]. In this work we will capitalize on this opportunity and advance the understanding of keyword networks by analyzing the semantic relationship between authors' keywords. Our system is comprised of three basic components: the data retrieval system, the embedding of the keywords, and the mathematical analysis of the embedded set. The data retrieval system enables this work to apply to any set of faculty, which provides future scalability and investigation. After retrieval, the set of authors, their publications, and their keywords are formally structured, and the keywords are transformed to mathematical objects by a word embedding. Lastly, these mathematical objects are subjected to k-means clustering and topological data analysis, the details of which are all explored in the technical introduction. This system of three basic components can be seen below as a procedural pipeline in figure 1.

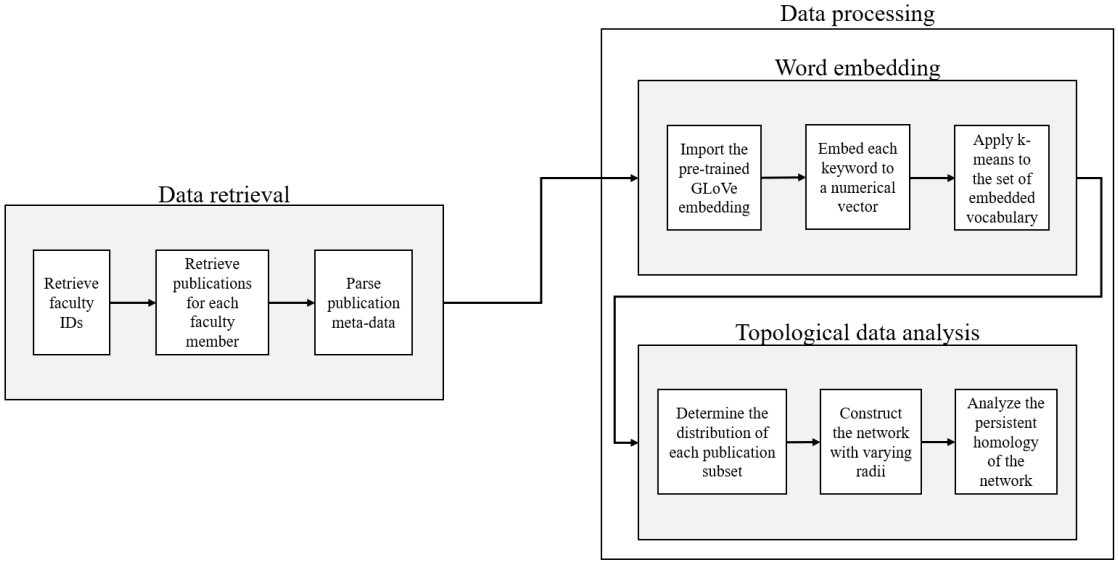

Figure 1 System components and their contents. This work leverages word embeddings and topological data analysis to make a novel contribution to keyword network analysis. This processing method facilitates the connection of keywords by semantic proximity and results with a topological output that can be analyzed with mathematical tools.

2. TECHNICAL INTRODUCTION

Keyword choice is at the discretion of the author in most fields; these uncontrolled vocabularies are prohibitive when linking any two authors because only identical words can form a connection. Some examples of uncontrolled terms are acronyms, plural and singular forms, typos, and synonyms. There is a clear need to combat uncontrolled vocabulary, and until recently it was only possible with supervised data cleaning and adding controlled terms after publishing [13]. Unsupervised language analysis has only recently become possible by machine learning, and more specifically the application of word embedding platforms [14].

2.1 Word embedding

Word embedding is the process of mapping words to high dimensional numeric vectors, such that these vectors carry semantic information. Word embedding platforms must apply machine learning algorithms to capture semantic information and require a large volume of training literature [14], [15]. This training can take two forms, the first is when embedded vectors are changed continuously while the machine reads [15]. The second is when the embedded vectors are only defined after machine reading is completed [14]. This study leverages Global Vectors for Word Representation (GLoVe), a Stanford University platform for word embedding which defines the vectors post-reading [14]. The GLoVe platform provides several pre-trained embeddings for use in common natural language processing (NLP) problems, these sets are trained on all of Wikipedia, the common crawl, and Twitter. The common crawl is itself a platform that collects webpage information for the purposes of providing datasets to data mining and NLP [14]. However, it doesn't filter any websites it pulls into its dataset or scrutinize the contents for accurate information. Our objective is to embed unique keywords that are applied in a scholarly context; both Twitter and the common crawl are unregulated and may result in detrimental correlations. Moving forward, Wikipedia is the only pre-trained embedding that is compatible with our work. Moreover, as an exploratory study of word embeddings on scholarly text, and with no such pre-trained embedding existing, we choose to employ the Wikipedia pre-trained 50-dimensional embedding [14].

Technology	Terrain
Computer	
Landscape	
Mountain	
Tablet	
Smart phone	
River	
Creek	

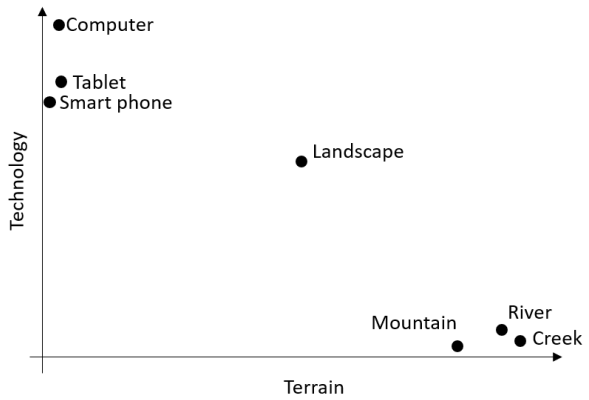

Figure 2. A word embedding takes a list of terms or phrases and converts them to numerical vectors whose position is determined by semantic relationships. Above the relationship to technology versus terrain is depicted graphically on two dimensions for a given vocabulary set, as more terms are added the need for more dimensions will increase to accommodate all semantic relationships.

Embedding a vocabulary transforms each word to a numerical vector, these vectors must be a high dimension to accurately convey every meaning a word can have. As described in Figure 2, there is a set of 2-dimensional vectors that can be represented as scatter plot in 2-dimensional space. By adding a third axis, we can easily imagine how the additional dimension would facilitate the accurate embedding of more words. Even more so, adding more axes allows us to accurately represent the complex semantic relations of language. For example, a third axis may reveal that "landscape" is closer to terrain than technology in the third dimension. Consequently, each additional axis brings an additional dimension to the vectors, so when referring to a 50-dimensional embedding, this translates to a vector of the form $<n_1,n_2,\ldots,n_{50}>$. Like the 2-dimensional vector in 2D space, a 50-dimensional vector is represented as a point in 50-dimensional space, and a set of vectors result in a point cloud.

Transforming the data to a set of 50-dimensional vectors is pivotal to the analysis of the dataset and prepares them to be analyzed mathematically. The abstract relationship of each word is now formalized in the 50-dimensional Euclidian space. As an instance of a Euclidean n-space it carries all the functions and properties of a Euclidean space. Natural functions of this space include the Euclidean distance, arc cosine, dot product, and many more [16]. Note that the provided vocabulary is responsible for generating the point cloud and this vocabulary can be any set of

keywords, thus the point cloud has no defined orientation. With these properties in mind, we can begin to identify what analytical tools and frameworks are candidates for our point cloud analysis.

2.2 Clustering: k-means

Since the data is unorientable, the natural first step is to detect any clusters within the point cloud; since each point corresponds to a keyword, their clustering may indicate disciplines or subgroups within the data. In our methods we will apply both k-means clustering, and k-means k optimization. The k-means clustering algorithm requires two inputs, the number of expected, k, clusters and the point cloud to be clustered [17]. Clustering begins by initializing k centroids randomly within the same space as the data. The data points nearest to any given centroid are designated to be part of that centroid's cluster. The true center of each cluster is then calculated, and the centroid of that cluster is translated to its cluster's true center. Once again, each data point's cluster group changes to that of the closest centroid. The true center of each new group is calculated, and again its centroid is moved. This process continuously repeats until a given iteration. The validity of the clustering is verified by cluster groups remaining the same between iterations [17]. This process can be seen in figure 3.

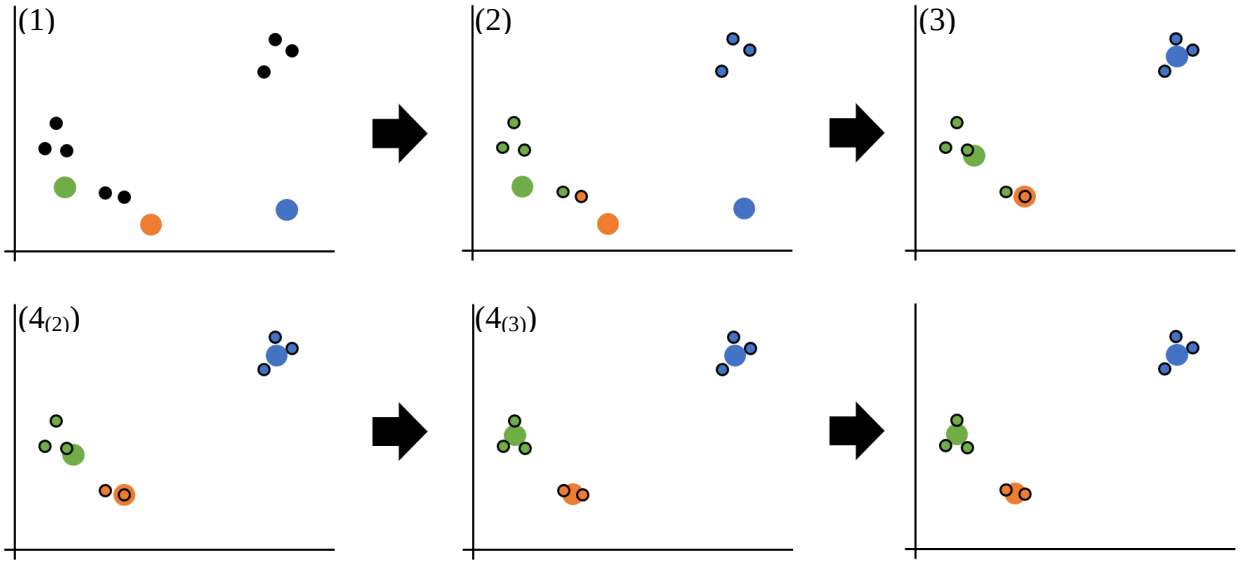

Figure 3 The process of k-means clustering where k = 3. (1) randomly position the three centroids on the graph. (2) group the points with the closest centroid by the Euclidian distance. (3) calculate the center of each group and reposition the centroid to those coordinates. (4) repeat steps (2) and (3) until the groups don't change between iterations.

In circumstances when k is not known, one must first determine the number of clusters algorithmically. This algorithm performs k-means clustering with a range of input k, and then calculates the sum of squared errors (SSE) of each k iteration [18]. Denote the number of data points as n, and say that k ranges from 1 to n. That is, the number of clusters will increase from 1 to n clusters. It can be assumed that n > 2 because clustering 1 or 2 points is trivial. It's also likely to be visually trivial for any relatively small n, however because it's mathematically possible we will consider all n > 2. When k = 1, the entire point cloud is a single cluster and the SSE is at its maximum. As k increases to n the SSE will approach to zero, and finally when k = n the SSE will be zero for each cluster. This is because each cluster is composed of only a single point, and thus the SSE is 0. According to literature, the accepted method for determining the optimal k is the elbow method [18]. This method defines a line between the first and last point of the system. The elbow point is then determined to be the observed point of maximum curvature in relation to this line [18]. An outline of this process can be seen below in figure 4.

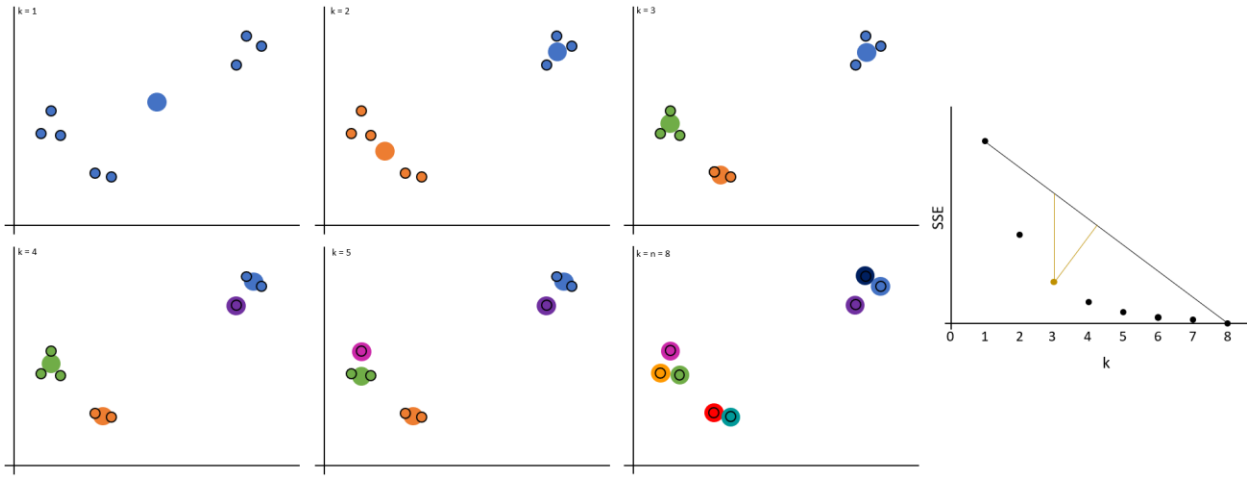

Figure 4 The kneedle algorithm for k-means group optimization. The number of centroids varies from 1 to n, where n is the number of nodes. The optimal k is selected by finding macimum point of curvature in SSE as a function of k, shown in gold.

2.3 Topological data anlaysis

The categorization of each word is useful for semantic clustering but limited in its ability to link authors by keywords. Recently, Temcinas et al showed the application of topological data analysis to be a useful tool for analyzing word embeddings [19].

Topological data analysis (TDA) is a relatively new mathematical tool that identifies topological features within data [20]. The classic example of a metric space is a Euclidean n-space [19]. Once converted, a simplicial complex is constructed on the data. A simplicial complex is a mathematical object that is made of points (0-simplex), edge (1-simplex), triangles (2-simplex), tetrahedrons (3-simplex), and higher dimensional shapes (n-simplex) [20]. The n-simplex notation refers to the minimal dimension of that shape, for example a point exists in the 0-dimension, an edge in the 1-dimension, and so on. When constructing a simplicial complex one can cutoff higher dimensional simplices, where the highest dimension is some number m. The resulting structure is then called an m-skeleton, for instance a 1-skeleton is a simplicial complex composed of only points (0-simplices) and edges (1-simplices) [20]. Moreover, simplicial structures are related to each other in terms of boundaries, where the boundary of an n-simplex is the (n–1)-simplices that come together to form that n-simplex. The boundary of a 0-simplex is simply empty [20]. The 0 to 3 simplices can be seen below in figure 5, as well as their relationships.

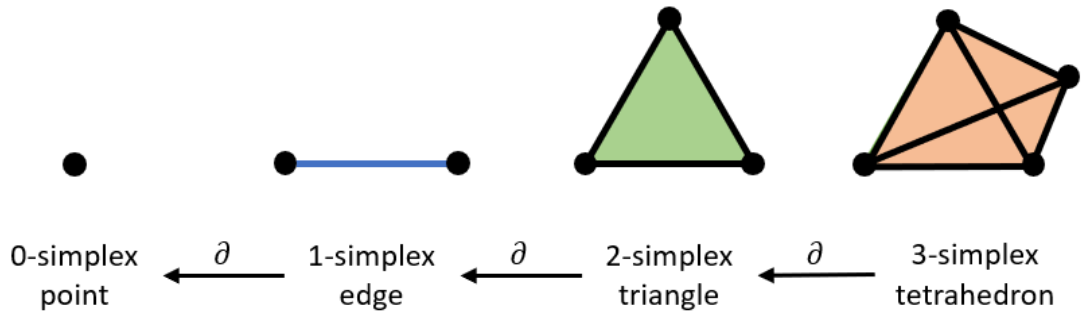

Figure 5 Each simplex for the first three dimensions, notice the boundary (∂) of a tetrahedron is 4 triangles, the boundary of a triangle is 3 edges, and the boundary of an edge are its two points.

The construction of a simplicial complex on a point cloud can be achieved through several means. The most common methods in the literature are the Vietoris-Rips (VR) complex and the Cech complex [21], [22]. Both employ hyperspheres to construct their complex and are only differentiated by how they define the existence of n-simplices when n > 1. The VR complex will automatically form the highest possible simplex when the edges are available, so if 3 edges meet to form a triangle, then the VR complex will automatically add a 2-simplex to the complex. In contrast, the Cech complex will only form an n-simplex if every hypersphere needed intersects one another. In practical terms this means 3 edges could meet to form a triangle, but they would

not necessarily contain a 2-simplex. The differences of these two complexes are highlighted in figure 6.

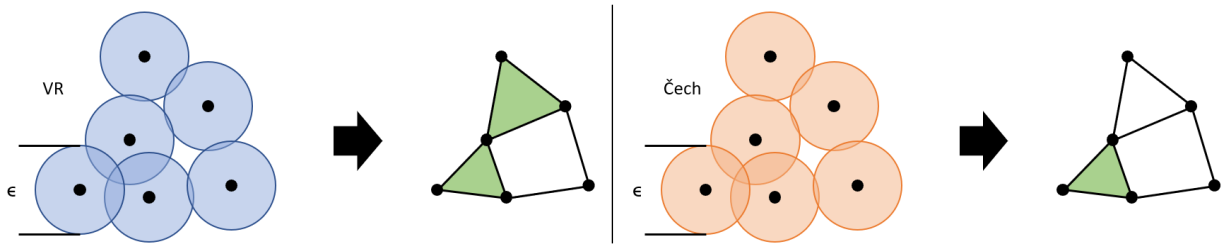

Figure 6 The distinction between the VR and Čech complex, a VR complex will always embed the highest dimensional feature it can on a set of edges; a Čech complex must have intersecting hyperspheres to form its highest dimensional feature.

Notice that the diameter of the hypersphere, epsilon (ϵ), is the decisive element of the complex structure. Small diameters lead to less simplices, and the simplices formed are of lesser dimensions. Whereas larger diameters lead to more simplices and higher dimensions. The selection of a single diameter limits the scope of topological evaluation and no universal ϵ calculation exists. The solution is studying the persistent homology of the complex. Like the k-means k selection, a persistent homology is defined by varying the diameter of the hyperspheres from 0 to an arbitrary cutoff [20]. The persistent homology seeks to understand the evolution of the simplicial complex structure and quantifies the 'persistence' of a topological feature. An n-simplex's birth is when it first appears in the complex, and its death is when it becomes the boundary of an (n+1)-simplex [20]. For instance, a hole of size could have a birth at $\epsilon = 63$, and a new edge at $\epsilon = 100$ forms an edge of a triangle. Since the edge has become a boundary of a 2-simplex, its death is $\epsilon = 100$. It can then be stated that the edge was persistent for 37 units. Any n-simplex can be persistent for 0 units, where the birth and death are equal; this means both the n-simplex and its (n+1) counterpart came into existence at the same time. A diagram that outlines persistent homology with a corresponding network can be seen below in figure 7, here we choose to depict the persistent homology with the barcode diagram.

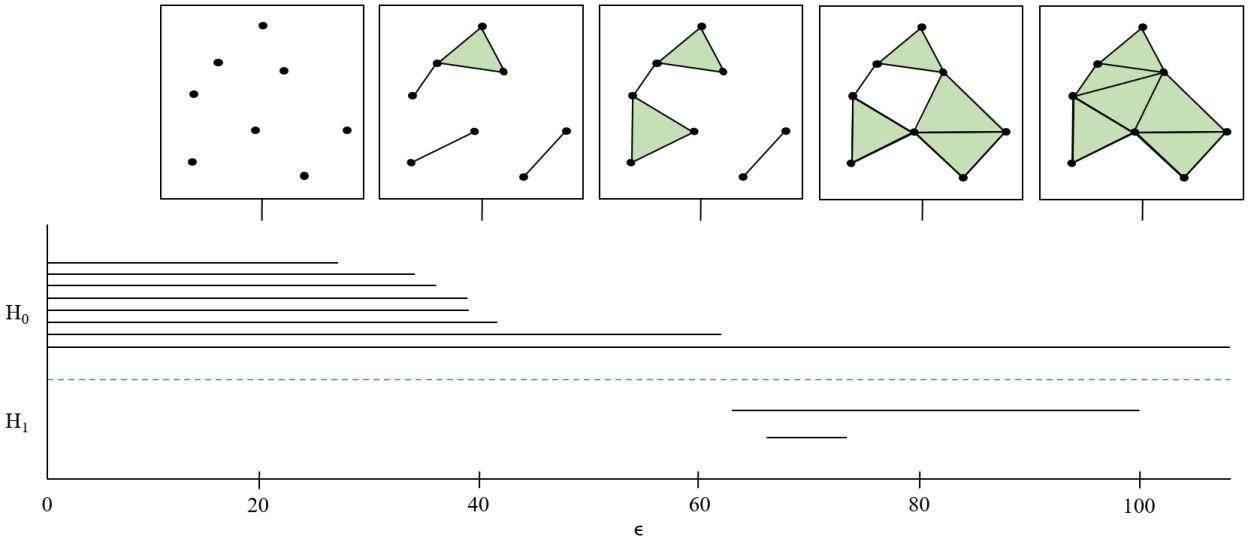

Figure 7 A barcode diagram of an example network. H_0 is the zero-dimensional homology and it reports the number of components for any given diameter by counting the bars vertically (e.g. $\epsilon = 20$ has 8 components, $\epsilon = 60$ has 2 components). H_1 is the first-dimensional homology and it reports the number of holes in the network by bars counting vertically (e.g. $\epsilon = 80$ has 1 hole).

The barcode diagram is useful for identifying persistent structures and trends visually. The persistence diagram depicts this exact information but in the form of a scatter plot. A point is plotted for each simplicial complex, where birth is the x-axis and death the y-axis. Any point that lies on the line y = x is a simplex that had 0 persistence, and it's not possible for any point to be less than y = x [20]. An example persistence diagram can be seen below in figure 8, the corresponding barcode from figure 7 is plotted to the left for convenience.

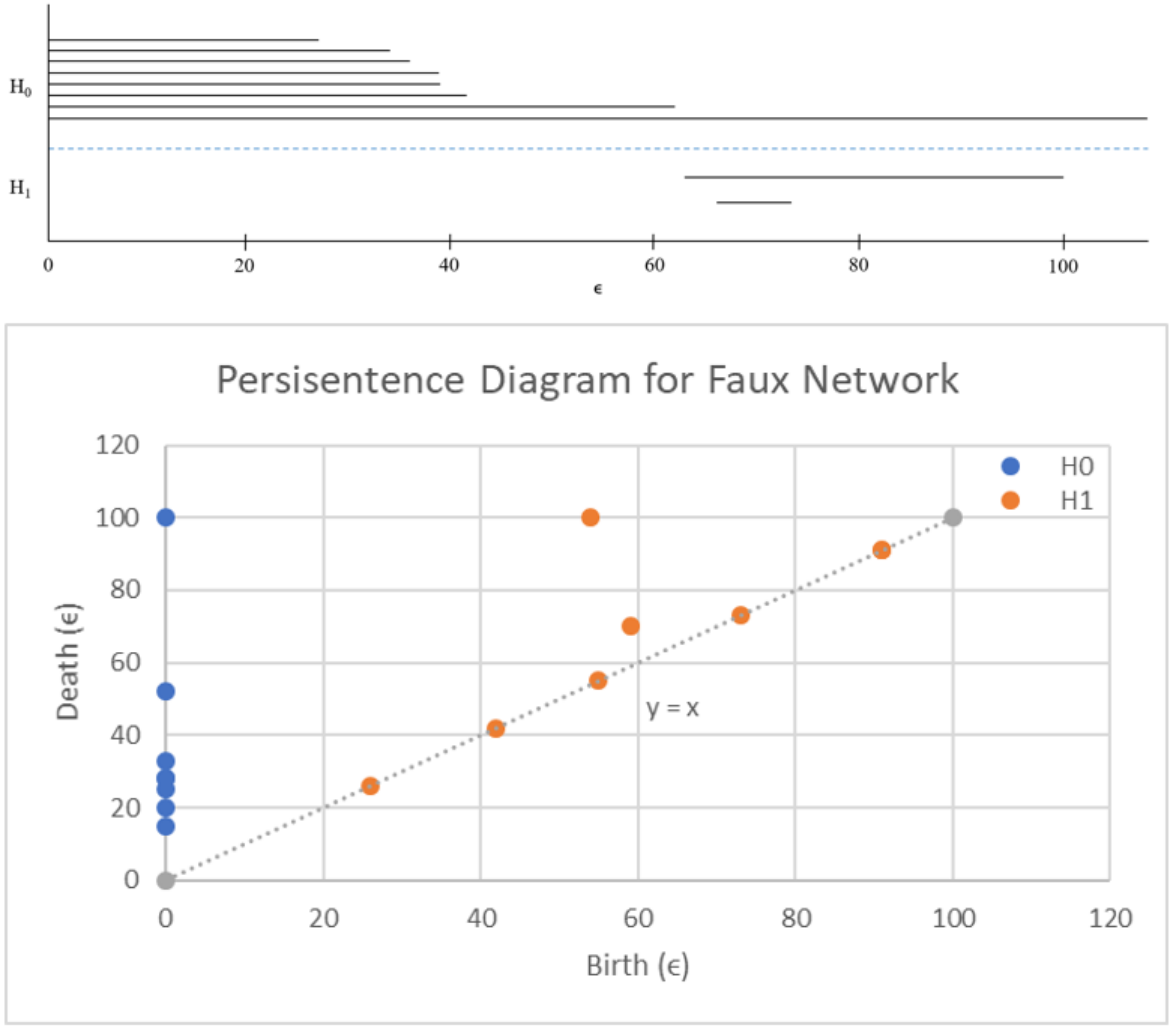

Figure 8 A persistence diagram is another method to visualize the lifespan of a network. Shown here is the barcode juxtaposed with its corresponding persistence diagram. The line y = x is displayed to show where simplices are born and die at the same time, this information isn't displayed on barcode diagrams.

In total, the persistence homology is used to identify meaningful structures within the data. Depending on the domain of the input, the result of TDA can vary [20]. Structures that tend to be interesting are stars, holes, and combined surfaces [20], [23]. A star is when a single node connects to many, and those it connected to only connect to it. A cycle is a series of nodes and edges, such that the ending and beginning node are the same, and any cycle greater than 3 is a hole. Lastly, a combined surface is when multiple triangles join to create a larger 2D object, as seen in the figure 7 complex. Both the barcode and the persistence diagram are designed to help identify these types of structures. TDA can also be used to cluster data, and functions similarly to hierarchical

clustering. Specifically, one can observe how the data clusters topologically according to the chosen distance function [20].

It's necessary to benchmark TDA against current methods for clustering scholarly works. As stated above, the most analogous system currently in use is hierarchical clustering [17]. Specifically, agglomerative clustering is the most like topological data analysis. This is when each data point begins in its own category and clusters with its nearest neighbor, which itself becomes a category. This process repeats until only one category remains. Hierarchical clustering of keywords can leverage the Jaccard distance, which does not employ word embedding [24]. The Jaccard distance determines the commonality between two sets. First, take the intersection of the two sets and find its length, for a publication's keywords this means find how many keywords the two have in common [24]. Next find the length of the union of the two sets. For publications this is the total number of keywords between the two, where no keyword is repeated. The Jaccard distance can then be defined as one minus the intersection length divided by the union length. Therefore, if two publications had 4 keywords in common and 6 keywords each, then the length of the intersection is 4 and the length of the union is 8 (6 + 6 – 4), so the Jaccard distance must be 0.50 (1 – 4 / 8). Note that if all the words are in common, then the Jaccard distance is 0 and if none are shared then its 1, but this limited interpretation predominantly links co-publishing faculty [24]. The ability to predict collaborations is predicated on identifying topics, methods, and technologies that combine the interests of unlinked faculty. Finally, the Jaccard distance is susceptible to the same pitfalls of the classic keyword network, where typos, acronyms, and uncontrolled vocabulary may result in missed connections [24]. Using NLP and TDA tools we seek to circumvent these issues and provide meaningful insight into how faculty collaborate, beginning with a test set of the Purdue BME faculty.

3. METHODS

3.1 Data retrieval

Similar to a keyword network, our network initialization requires a set of faculty and the institutional affiliation of those faculty. Informed selections from teams, groups, or departments are the most representative and unambiguous sets, and are considered the standard input. The lower bound for the number of faculty is one, meaning the evolution of a single faculty can be observed. There are two optional date inputs that provide a lower and upper bound for publication retrieval. If neither field is filled, then all available publications will be retrieved.

Our data is retrieved by SCOPUS API, a service provided by Elsevier [10]. Each author in the faculty set has their unique SCOPUS ID retrieved, where each ID is specific to the affiliated institution. Using this set of unique author ID's, we query the SCOPUS abstract API to retrieve each author's entire body of work accessible by SCOPUS. The results of this search return each publication's meta-data, this meta-data is formally labelled with title, keywords, citations, and publication date. Each faculty profile is constructed by compiling their publications, and each publication is then represented by its set of keywords. As addressed previously, lack of controlled vocabulary limits the efficacy of keyword networks. To circumvent the issue of uncontrolled vocabularies, we consider a natural language processing approach to network generation.

3.2 Data metric and filter

The first step in our natural language processing is to convert each word into high-dimensional vectors, a process that is called word embedding. Here we choose to leverage "global vectors for word representation" (GLoVe), a word embedding platform that provides pre-trained embeddings. Specifically, we selected the Wikipedia pre-trained embedding with 50-dimensions. This embedding covers a breadth of topics and contains over 400,000 words. The breadth of topics permits the user to input any faculty in any subject area, and the ability to process almost all keywords. The limitations of this word embedding will be addressed in the discussion.

The raw set of keywords from the data retrieval process are not yet ready to be embedded. Many authors use phrases, spaces, acronyms, hyphens, and other types of punctuation. These text decorators don't always occur in the GLoVe dictionary and would prevent words from being embedded. The raw vocabulary is parsed and embedded using the following algorithms. Note that before each iteration of parsing, we check if the word is in GLoVe. If the word is in GLoVe, it won't be parsed and instead embedded. This is done to ensure that the most complex version of the raw vocabulary is embedded.

a)

```
def keywordEmbedding(authors,glove):
    for author in authors:
        for publication in author['publications'].keys():
            embeddedWords = []
            for keyword in author['publications'][publication]:
                parsedWords = wordSetProcessing(keyword,glove)
                for parsedWord in parsedWords:
                    if len(parsedWord) > 3 and parsedWord in glove:
                        embeddedWords.append(parsedWord)

            author['publications'][publication] = {}
            author['publications'][publication]['keywords'] = embeddedWords

    return(authors)
```

b)

```
def wordSetProcessing(wordSet,glove):

    wordSet.removeAll('(')
    wordSet.removeAll(')')
    wordSet.replaceAll('  ',' ')

    parsers = [' ','-',',','/','–','.',':',';','?','!']

    wordSet = [wordSet]
    for parser in parsers:
        wordSet = wordParsing(wordSet,parser,glove)

    return(wordSet)
```

c)

```
def wordParsing(wordSet,parser,glove):

    parsedWords = []

    for word in wordSet:

        if word != '' and word in glove:
                parsedWords.append(word)
        elif word != '':
            for parsedWord in word.split(parser):
                parsedWords.append(parsedWord)

    return(parsedWords)
```

Figure 9 a) keywordEmbedding is the function that is responsible for calling the embedding workers, it iterates through each keyword within each publication by each author; it then checks and saves any parsed term that is longer than three letters and in the GLoVe dictionary. b) wordsetProcessing receives the raw keyword from keywordEmbedding, where it removes parenthesis and double spaces; then it iteratively calls word parsing for each pre-defined parser. c) wordParsing iterates through each word in wordSet, if the word can be embedded it is saved in that state, if it can't be embedded it is parsed with the current parser in wordsetProcessing.

After parsing each word and defining the embedded vocabulary, each word can now be represented as a unique 50-dimensional vector. Similar to Temcinas *et al.* [19], we will denote the vector for *word* as v_{word}. The complete set of these vectors is a point cloud in 50-dimensional space. This point cloud facilitates the computer's understanding of context by proximity; however, it also results in information loss. For instance, it is no longer clear how many times a keyword is used because it can only be embedded once. However, through clustering and data analysis we can impose structures on the point cloud and model the hierarchical structure of our data.

Before network construction, we perform k-means cluster group optimization and then k-means clustering. The k-means cluster optimization reveals approximately the number of groups that exist in the point cloud, and the clustering identifies what category each word is in. With each point in a cluster, this aids in the network coloring and visualization later in the platform. Next, we must construct the network to embed author centric features into the system.

3.3 Network construction

Here we seek to begin network construction, and as such we appeal to topological data analysis (TDA). Recall that TDA requires both a dataset and a distance function that can relate any two data points in the dataset. Given our dataset is a point cloud in 50-dimensional space, a Euclidean n-space, we have access to several different measures of distance [19]. Of the options available, we choose the angle produced by any two data vectors in our dataset, otherwise known as the arc cosine. It has been shown that arc cosine better identifies semantic relationships between two embedded words than any other distance available in Euclidean space [14], [15], [25]. The combination of 50-dimensional space and distance function result in the necessary metric space that can be leveraged by TDA [19]. Recall that TDA builds a simplicial complex, which not only contains points and edges but also faces, tetrahedrons, and higher dimensional simplexes. If we remove the faces, tetrahedrons, and higher dimensional simplexes from our construction, then we are left with only points and edges; this is called the 1-skeleton of our simplicial complex. Since the components of a 1-skeleton are identical to a network, we will refer to the 1-skeleton as a network for simplicity.

As outlined in the introduction, there are several ways to construct a simplicial complex, and by extension a network. As a basis for our work, we consider the Vietoris-Rips (VR) filtration which doesn't require oriented data and is computationally faster than the Cech filtration. As stated in the introduction, the VR complex constructs networks by expanding hyperspheres around each point, and when two intersect an edge is created. This method of construction, as well as its limitations, are outlined in figure 10.

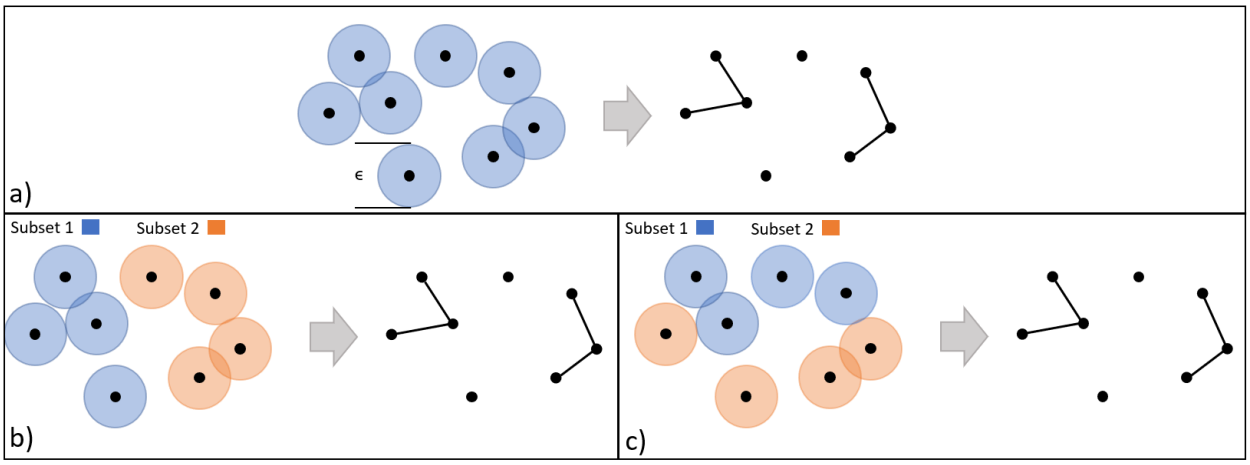

Figure 10 a) Canonical VR complex, as ϵ increases more connections will be formed. b) A Vietoris-Rips complex on the same point cloud, now with underlying subsets. c) Once again, the same point cloud, now formed by subsets that are distinctly different from (b). Notice that (b) and (c) have isomorphic topologies because the underlying subsets don't affect hypersphere expansion.

Since the VR filtration is unable to convey keyword subsets it will always construct identical networks for identical vocabularies. The underlying cause is twofold, and results from both word embeddings and the VR filtration. After training, GLoVe is like a dictionary, where each word corresponds to a unique vector in 50-dimensional space. Since the dictionary is pre-determined, then a given vocabulary will always be embedded to the same point cloud. The only methods to change this embedding are to retrain on a new dataset, or to fine tune the current embedding with supplementary texts. Both processes are computationally expensive and require the generation of new and unique training sets.

Thus, we choose to consider how the VR filtration constructs a network. The VR hyperspheres expand uniformly and without regard to subsets, each having diameter ϵ (figure 10). However, if

ϵ could be variably weighted based on subsets, then each network generated would be unique to the set of faculty who created the vocabulary.

Here we consider hyperspheres that expand at varying radii, where the radii is dependent on the distribution of the publication subspace. We define a publication subspace as the subset of a publication's embedded keywords. To calculate the distribution, we find the distance between every point in the publication subspace, then calculate the mean distance and standard deviation. An example of this process can be seen in figure 11. Once every publication distribution for every author is calculated, we can then define the new data structure that will facilitate network construction.

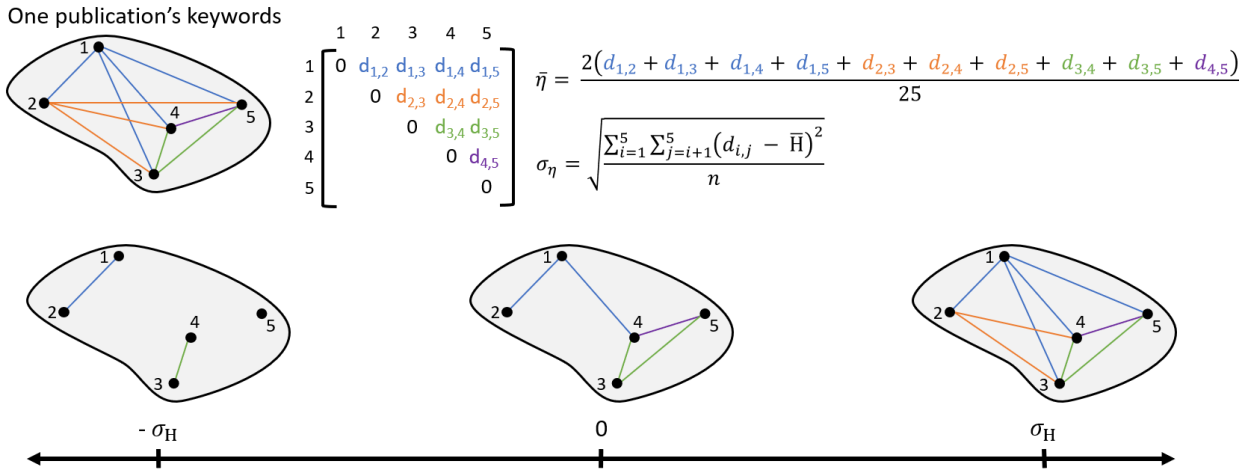

Figure 11 The distance between every point is calculated, then the mean and standard deviation are calculated. The subspace can now be constructed in terms of standard deviations from the mean. As shown above, filtering any edge larger than one deviation below the mean results in the leftmost network, which is simply the initial connections between the closest words. Following this, the central network filters an edge that is larger than zero deviations from the mean, resulting in a network that is just a single component and a 2-simplex (3, 4, 5). Lastly, we show a network that filters any edge larger than one standard deviation above the mean, this network is composed of long connections and two 3-simplices (1,2,3,4) and (1,3,4,5).

The resulting data structure is a list of authors who each have a set of publications and a corpus of keywords. Each publication has a set of keywords, a mean, and a standard deviation. The author's corpus is defined as the set of keywords generated by the author's publications. Each keyword has the mean and standard deviation of the publication that generates it. If a keyword is generated by

more than one publication, we only consider the maximum mean and standard deviation for that author's use of the word. Lastly, we define the distance vector for each word in the author's corpus. Notice, that the distance vector for any word can be redefined in terms of the author's mean and standard deviation for that word. An example of the data structure and distance rescaling is shown in figure 12.

Figure 12 A faux dataset for downstream examples, f is the transformation function that converts Euclidian distances to distributions from the mean. a) The vocabulary of the entire dataset, and their indices for the "distances" vector. b) An author profile designed to show that duplicate keywords are only redefined within the domain of an author (i.e. Dr. Pascuzzi's data is unaffected by Dr. Kinzer-Ursem's). c) An author profile designed to show how duplicate keywords used by the same author must select the most interdisciplinary paper (e.g. protein uses 9.1815 as its deviation).

When considering how all authors contribute to a network it's clear that subsequent edges between two points carry no topological meaning (i.e. once an edge is made by one author, any subsequent authors who make that same connection don't contribute to the topology). However, these

subsequent edges still factor into the post analysis for each author's individual networks and must be calculated. This can be depicted for the example data structure as concentric hyperspheres around each point, seen in figure 13. We do not consider loops in our analysis, which is an edge between a node and itself.

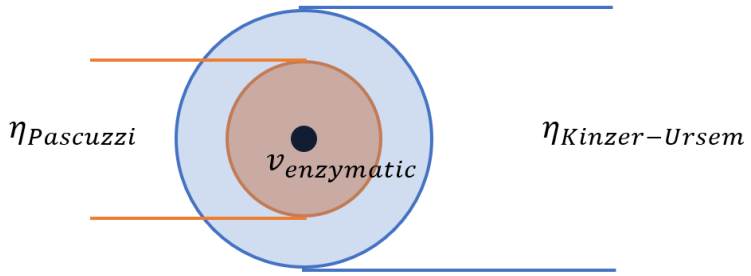

Figure 13 Concentric hyperspheres of Dr. Pascuzzi and Dr. Kinzer-Ursem for the embedded enzymatic point, notice that Dr. Pascuzzi's deviation for the word enzymatic (figure 12b) is smaller than Dr. Kinzer-Ursem's (figure 12c), thus resulting in the different radii for the same word.

After preparing the data structure and establishing the use of concentric hyperspheres, we can finally begin network construction. Notice that the transformed distances are now unique to their subsets, meaning the hyperspheres radii are distinctly different from the previous ϵ. As such, we now refer to the radius of each hypersphere as eta (η). Naturally, η is initialized at the smallest observed standard deviation where an edge occurs, and so we consider this minimal distance to correspond to η being zero. In other words, when η is zero there is only the minimal edge set in the network. By referring to the data structure in figure 12, we can observe the minimal edge to be between chemistry and engineering within Dr. Kinzer-Ursem's corpus. In contrast, the maximum η could be bounded by the global maximal distance deviation; however, an η this large results in tenuous connections and the computational complexity exponentially increases. Instead we define the upper bound as zero deviations from the mean. In other words, η is 100 when each radius of every hypersphere is, at most, the mean distance of its publication subspace. All deviation distances between these bounds are linearly scaled to a number between 0 and 100, each with 2 significant figures. This construction method is demonstrated in figure 14.

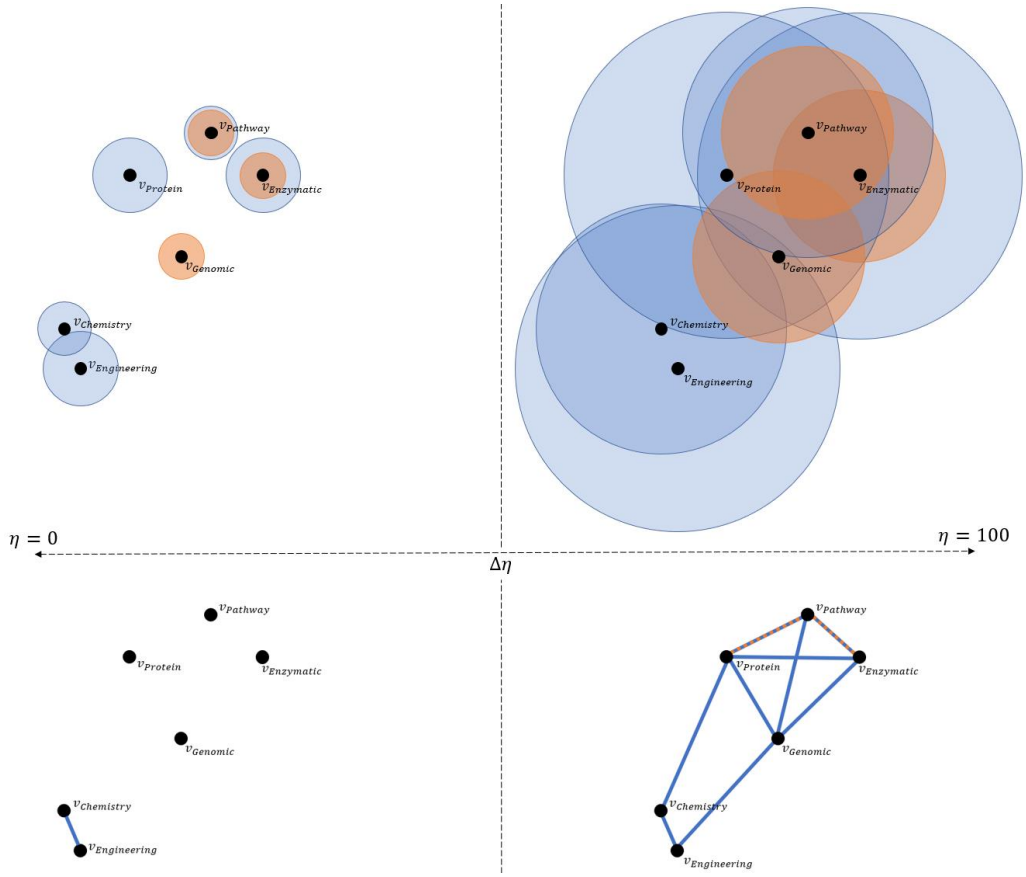

Figure 14 Continuing with the faux dataset, we now visualize the η method of network construction, Dr. Pascuzzi's edges are colored orange and Dr. Kinzer-Ursem's are blue. On the top left we visualize the network when η = 0 and there is only the minimal edge between $v_{engineering}$ and $v_{chemistry}$, and the bottom left depicts the 1-skeleton of that connection. On the top right we visualize the network when η = 100 and all hyperspheres have a radius equal to their subset mean, again the bottom right depicts the 1-skeleton of this final structure. Notice the nested edges where Dr. Pascuzzi and Dr. Kinzer-Ursem overlap.

This technique of construction determines how many deviations from the mean the radius, η, must be to form an edge. Since we know the radius at which every edge forms, we can bucket these edges into the corresponding η. The front-end bucketing of edges saves computational time for later analysis. For instance, consider the global minimal deviation for the example set, -2.9018, which corresponds to η equals 0. Now consider that 0, the upper bound for deviations, corresponds to η equals 100, we can interpolate that -1.4509 deviations from the mean corresponds to η equals 50. The full network dataset is represented in figure 15.

```
{
  "nodes": [
    {"id":"enzymatic","group":1,"publications":["84838498158","85019700323"]},
    {"id":"protein","group":1,"publications":["84838498158","85038283729"]},
    {"id":"engineering","group":0,"publications":["84838498158"]},
    {"id":"pathway","group":1,"publications":["85038283729","85019700323"]},
    {"id":"chemistry","group":0,"publications":["85038283729"]},
    {"id":"genomic","group":1,"publications":["85019700323"]}
  ],
  "authors": ["tamara kinzer-ursem","pete pascuzzi"],
  "data": {
    "0.0":[{"source": "engineering", "target": "chemistry", "value": 1, "author":"tamara kinzer-ursem"}],
    "9.05":[{"source": "enzymatic", "target": "pathway", "value": 1, "author":"tamara kinzer-ursem"}],
    "22.06":[{"source": "chemistry", "target": "engineering", "value": 1, "author":"tamara kinzer-ursem"}],
    "27.82":[{"source": "protein", "target": "pathway", "value": 1, "author":"tamara kinzer-ursem"}],
    "32.24":[{"source": "pathway", "target": "enzymatic", "value": 1, "author":"tamara kinzer-ursem"}],
    "51.84":[{"source": "enzymatic", "target": "protein", "value": 1, "author":"tamara kinzer-ursem"},
      {"source": "protein", "target": "enzymatic", "value": 1, "author":"tamara kinzer-ursem"}],
    "51.86":[{"source": "enzymatic", "target": "protein", "value": 1, "author":"tamara kinzer-ursem"},
      {"source": "protein", "target": "enzymatic", "value": 1, "author":"tamara kinzer-ursem"},
      {"source": "enzymatic", "target": "pathway", "value": 1, "author":"pete pascuzzi"},
      {"source": "pathway", "target": "enzymatic", "value": 1, "author":"pete pascuzzi"}],
    "51.88":[{"source": "enzymatic", "target": "pathway", "value": 1, "author":"pete pascuzzi"},
      {"source": "pathway", "target": "enzymatic", "value": 1, "author":"pete pascuzzi"}],
    "53.36":[{"source": "pathway", "target": "protein", "value": 1, "author":"tamara kinzer-ursem"}],
    "58.74":[{"source": "protein", "target": "genomic", "value": 1, "author":"tamara kinzer-ursem"}],
    "73.96":[{"source": "pathway", "target": "protein", "value": 1, "author":"pete pascuzzi"}],
    "75.9":[{"source": "enzymatic", "target": "genomic", "value": 1, "author":"tamara kinzer-ursem"}],
    "79.05":[{"source": "protein", "target": "chemistry", "value": 1, "author":"tamara kinzer-ursem"}],
    "94.97":[{"source": "pathway", "target": "genomic", "value": 1, "author":"tamara kinzer-ursem"}],
    "97.29":[{"source": "engineering", "target": "genomic", "value": 1, "author":"tamara kinzer-ursem"}]
  }
}
```


Figure 15 The data structure that results from our construction. Where “nodes” is the vocabulary set, and each node contains the publication(s) that generated it. Authors is the set of authors we are interested in visualizing the connections between. Data contains the η buckets and each bucket corresponds to a set of edges that appears at that radius, where each edge has equal weight and the author who facilitated the connection.

With the final data structure complete, we can now turn our attention to network analysis. In the next section we will consider network visualization, curating publications from networks, embedding higher dimensional topological features, and using those features to predict collaborations.

3.4 Output

3.4.1 Visualization tool

In this section we seek to analyze and dynamically visualize the constructed network. We exclusively employ JavaScript with data visualization packages to facilitate network visualization and web-app development. The packages and their roles in the visualization are outlined here.

1. D3: dynamic data representation in JavaScript for web app visualization [26]
2. Three: enable the creation of unique geometries and scenes for animations [27]
3. 3d-force-graph: a package that enables 3D force graph visualization; dependent on D3 and Three [28]

To initialize a visualization, the 3d-force-graph engine requires a set of nodes and a set of edges. By default, this visualization is a static representation of the nodes and edges. The default settings allow the user to zoom, pan, and rotate the 3D environment. Other default features include automatic node and edge coloring, node and edge size, and node identity on mouse hover.

In our visualization we retain these features. The nodes are colored per their k-means grouping, which gives a general sense of their proximity; the edges are colored to indicate which author generated them. In addition, we developed interactive systems to facilitate network exploration. Our visualization allows the user to filter edges that appear after a certain η, we refer to this tool as the η slider. By default, our visualization removes any node that doesn't have an edge. The initialization of the graph is set to an η equal to 15, an arbitrary number that facilitates efficient rendering. The ability to filter edges and nodes is not native to 3d-force-graph and is a development of this project. We extend the ability to filter to authors as well, allowing the user to remove authors by checkbox. An instance of the visualization can be seen below in figure 16, where the η slider is set to 40.

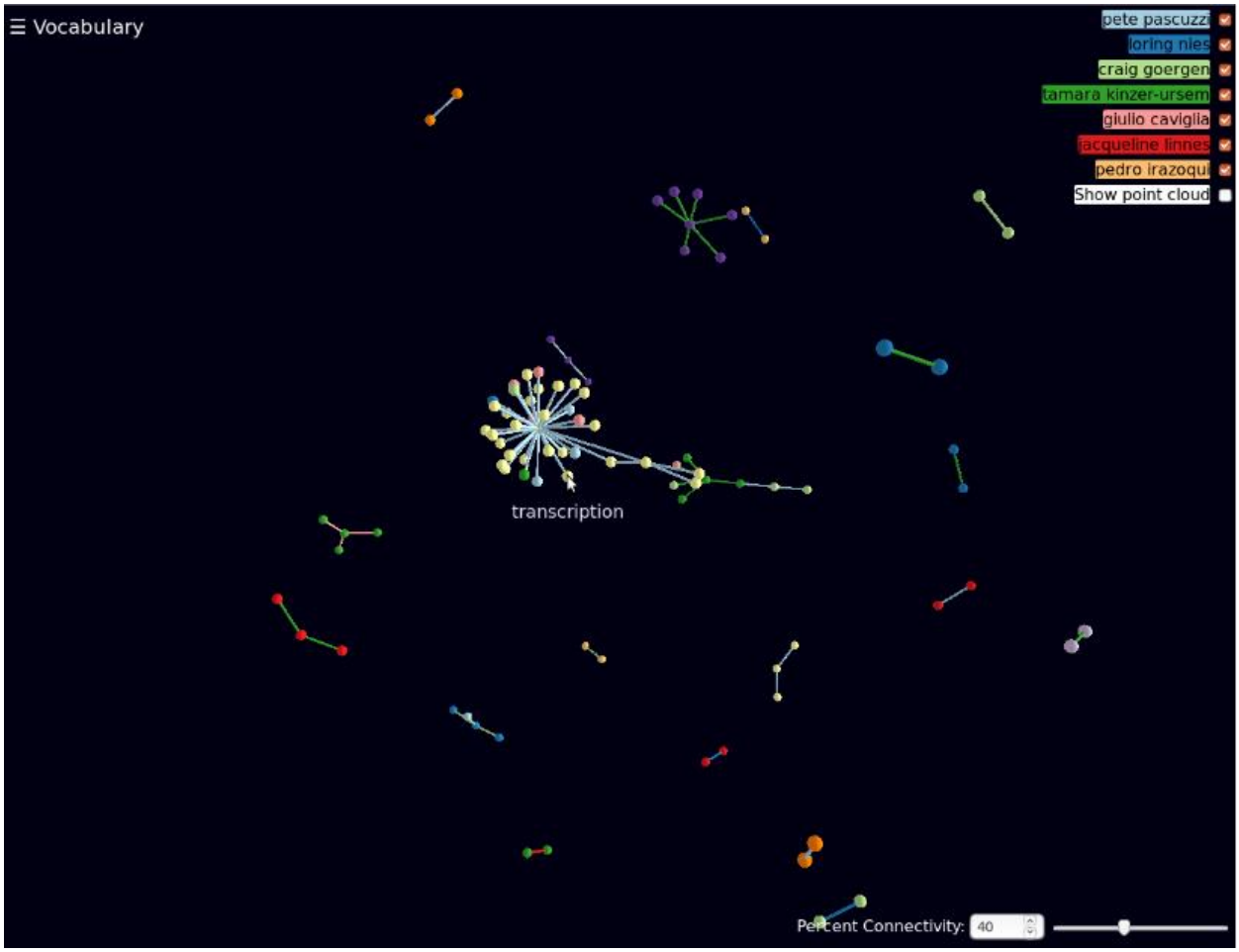

Figure 16 An instance of the 3D 1-skeleton visualization. The bottom right corner contains the η slider, allowing full control over network connectivity. The top right corner contains a list of checkboxes that allows the permutation of authors and the display of the point cloud. Lastly, the top left corner contains a side menu where the user can save a custom vocabulary.

As shown above, the visualization environment allows the user to customize the upper bound of η, the faculty who are displayed in the network, a unique vocabulary, and if the point cloud is displayed. The use of faculty selection, vocabulary generation, and point cloud display can be seen below in figure 17.

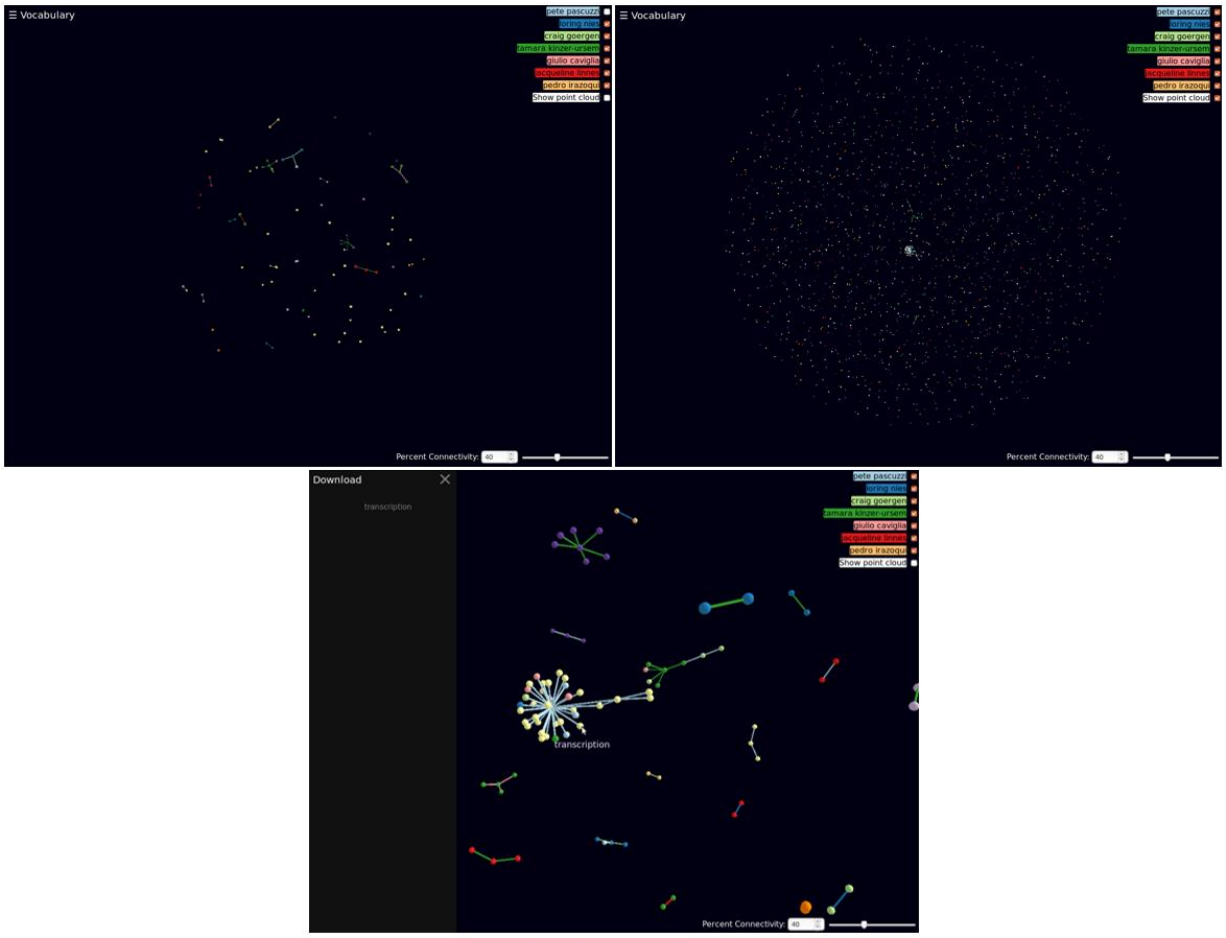

Figure 17 a) A visualization of when an author is removed from the figure, allowing the nodes to remain gives insight into the deleted author's impact. b) A visualization of the entire point cloud, which helps the user understand the volume of connections still not made. c) An instance of adding a word to the custom vocabulary window, in the top left corner there is a download button for downstream analysis

Each feature furthers the exploration of the network, allowing the user to view any author permutation from their chosen η filtration. Following the connectivity of the network, the user can identify keywords and save them to their custom vocabulary bank. This qualitative network analysis and exploration is limited by the user's ability to traverse complex networks. For example, figure 17 analyzes how seven Purdue faculty connect with all the keywords used by the biomedical engineering department. This example network of 1,715 words quickly grows with η and any interesting connections become indiscernible to the human eye. As a result, we developed several computational methods to identify the connections that result in useful topologies. The first of which is the vocabulary download option seen in figure 17c.

3.4.2 Curating publications

Once an interesting topological feature has been identified in the network, one can select nodes that generated it and save the corresponding keywords to the custom vocabulary bank; ultimately, this results in a curated stack of publications. Specifically, the option downloads all publications use at least one keyword in the custom vocabulary. Resulting in a set of uniquely filtered publications, which can't be obtained by existing keyword search platforms for three distinct reasons. First, the set of publications is generated only by the faculty within the system, resulting in a narrowed scope of people within the institution. Second, each publication only needs one of its keywords saved to the custom vocabulary to be pulled, resulting in a wide spectrum of publications. Lastly, the set of nodes selected is a result of topological data analysis, the varying scope and features of which provide unique connections between publications. One such topological feature is a star, a network component where all but one nodes are exclusively connected to a central node. A star can be seen below in figure 18 and was generated using the augmented BME faculty test set. Once each term in the star is saved to the vocabulary list, we can calculate the Jaccard distance for each publication in the downloaded set. Recall from the introduction that the Jaccard distance calculates the number of shared terms divided by the number of all terms between any two publications. We calculate the Jaccard distance between all publications in the custom bibliography, which results in a Jaccard matrix. This matrix is processed to display the hierarchical clustering, allowing us to visually observe the similarity of each publication from our keyword selection, the clustering output will be discussed in more detail in the results.

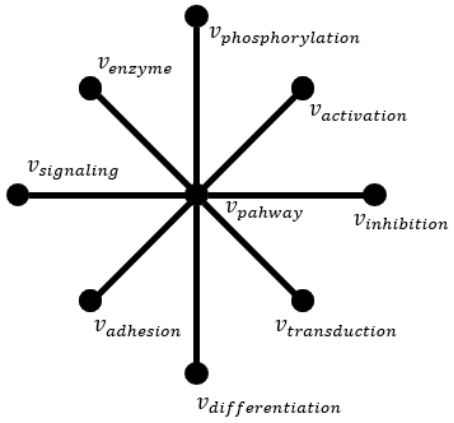

Figure 18 An example of a star from the BME test set shown in figures 14 and 15. Each keyword in the star is associated with a list of publications.

Here we choose to explore the star as a topological feature due to its significance within our system. In figure 18 and in our network, the central node of the star connects to eight different nodes. These orbital nodes exclusively connect to the central node, which is interpreted as all nodes being related only through that keyword. For example, in figure 18 all the orbital keywords are connected to "pathway" but not each other, this indicates that "pathway" is related to a diverse group of subjects that aren't as semantically similar. This cluster of nodes shows the grouping of distinctly different subjects about one central idea.

After clustering, the custom set of publications is converted to a MEDLINE format, which we confirmed to be compatible with bibliography software [29]. This bibliography is curated to group publications together based on the Jaccard distance, and these groups are then ordered from greatest to least by average citation. So, a group of 4 publications, where the average the number of citations is 10 will be placed above a group of 8 with average citation of 5.

3.4.3 Topological Toolkit

The first step in classifying topological structures is the ability to efficiently identify them. Similar to the visualization output described previously, the input to our topological toolkit will be the network data structure, e.g. the structure seen in figure 15. Most topological analysis toolkits are for the theoretical study of topological data analysis by mathematicians, and those that are for the use of the public are proprietary. Here we create a suite of topological analysis tools for the explicit use with our data structure, these tools are able to calculate the zero-dimensional homology and the first dimensional homology of our network (i.e. connectivity and 2D-faces).

The 0D homology is simply how many components are in the network at a given time. For example, two nodes that are not connected are two separate components; if an edge is put between these nodes, then there is only one component. Therefore, when we initialize the network on n nodes (where n is the size of the vocabulary), and at η equals zero, then there are n-1 components because only one edge exists at η equals zero. As η is increased, each edge is added incrementally to the graph, and if the edge connects two separate components, then they are concatenated to one component. Since each η corresponds to a list of edges, each edge is added one-by-one in the order

of that list. If the new edge connects two nodes within the same component, then this doesn't affect the 0D-homology.

The purpose of the 1D homology is to identify cycles within the network. A cycle is only possible if the newly added edge connects two nodes that already had one pre-existing path between them. In other words, the 1D homology isn't affected by edges that connect two separate components. Therefore, if the two nodes exist in the same component, the new edge results in at least one new cycle. More precisely, the number of pre-existing paths between two nodes, call this p, is exactly the number of cycles after adding the new edge, therefore the number of cycles is p. The process flow for 0D and 1D homology calculations can be seen in figure 19.

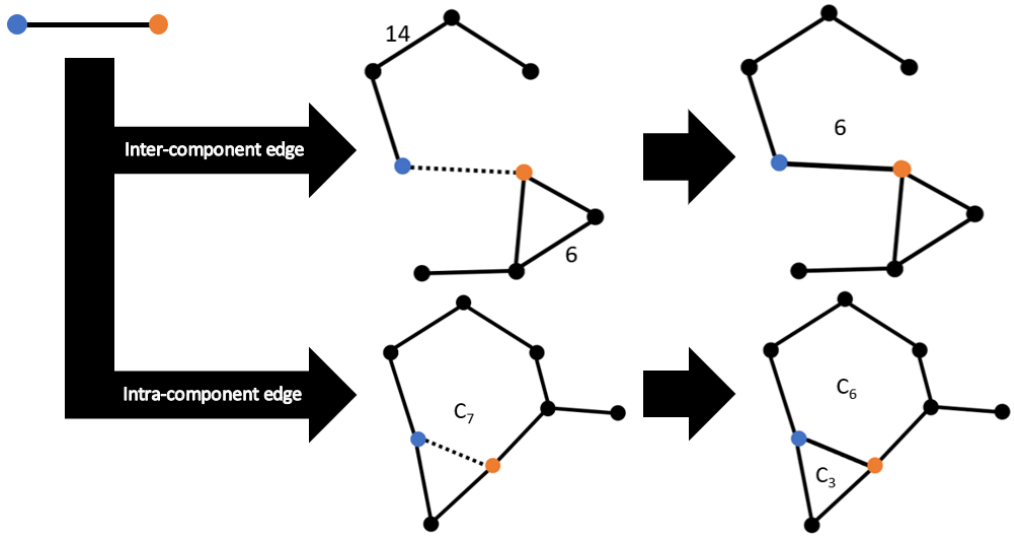

Figure 19 When a new edge is added to the network it undergoes a test to determine if it will join two components, or if it will join two nodes in the same component. Notice the two components are labeled 14 and 6, the label of the resulting component will always take the "younger" component's label. Notice the intra-component edge results in the creation of a new cycle, and a reduction of the existing cycle.

While the inner mechanics of 0D homology calculations are adequately outlined in figure 19, the 1D homology requires more intricate processes. Clearly intra-component connections result in cycles; however, simply counting these connections leaves the cycle size and elements unknown. Before we investigate the process, the following notation and functions in table I are critical to describing our method.

Table 1 The mathematical notation that to be used in figure 20.

e_j	The new edge being added to our network
S	The component that contains e_j
u	First node of edge e_j
v	Second node of edge e_j
$\boldsymbol{P}$	The set of paths between u and v, excluding e_j
xPy	The path from x to y
p_{min}	The path with minimal length in $\boldsymbol{P}$
N(x)	Neighbor function, returns the next layer of neighbors for a given set of vertices
$x \cap y$	Intersection function, returns the intersection of the sets x and y
$p \cup e_j$	The union of a path p with edge e_j
$x \circ y$	Difference function, remove $x \cap y$ from x and y and return the two distinct sets

Assume that u and v are within the same component, else they can't form a cycle but only a new edge between two separate components. Also note that u and v have no edge between them, else no new cycle would be created by their joining. Therefore, e_j must be the first edge between u and v, and there must be at least one path that connects u and v. Next, we seek to identify a minimal path between u and v. The minimal path between u and v is found by recursively calling the neighbor function from both the u and v direction. The first instance in which the two outputs intersect is the set of minimal paths. We require only the first minimal path in the set, call this path p_{min}. Therefore, the addition of edge e_j results in the newest minimal cycle of $p_{min} \cup e_j$, call this cycle C. Every cycle that contains the path p_{min} is reduced by taking the disjoint union with cycle C. In other words, by taking the disjoint union of C and all cycles that contains p_{min} we replace p_{min} with e_j, thus reducing the cycles. The process of minimal cycle detection and disjoint cycle reduction is shown below in figure 20.

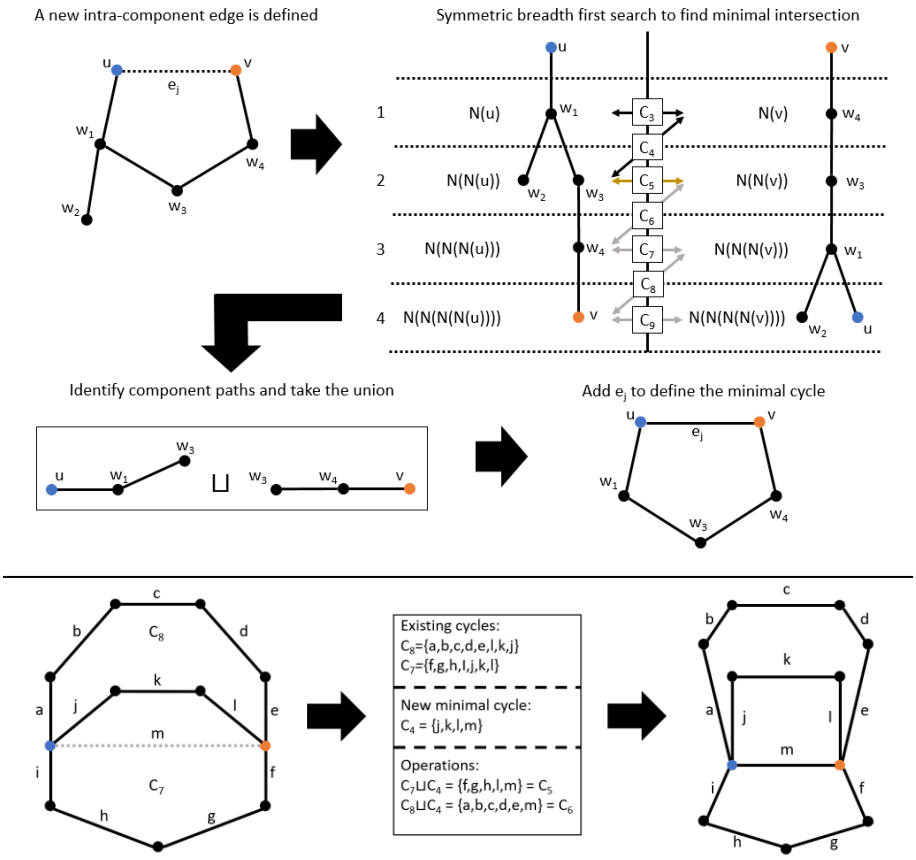

Figure 20 The detection of all cycles begins with an intra-component edge. Then a breadth-first-search (BFS) algorithm is implemented symmetrically from both ends of the new edge. The first instance in which the two BFS trees intersect (e.g. the gold arrow) is midpoint between them, call it m. Then p_{min} is the union of the paths uPm and mPv; adding the edge e_j to this shortest path creates a new minimal cycle. The minimal cycle is then used to reduce every other pre-existing cycle by taking the disjoint union of all cycles that contain path p_{min}.

Now that our process of 0D and 1D homology computation is understood, we can leverage this information to observe connectivity, gaps, and author growth as a function of hypersphere radii. This information is further expanded on in our exploration of 0D and 1D homology application.

3.4.4 Homology applications on networks

Lastly, we investigate the 0D and 1D homology with barcode and persistent diagrams. We use a barcode diagram to visualize the 0D homology, where we can track how network connectivity increases as a function of hypersphere radius. Normally, a barcode diagram has a horizontal line spanning from the birth radius to the death radius for each component. However, we know that every node is born at radius zero, therefore the only relevant information is component death.

Therefore, we instead choose to plot a point at the death radius. The visual justification for this choice can be seen below in figure 21.

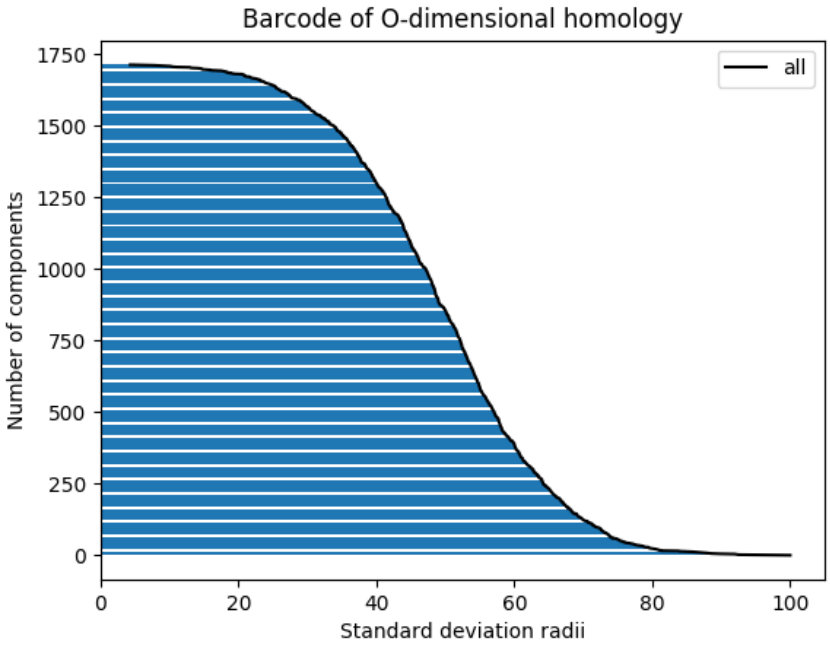

Figure 21 All nodes are born at $\eta = 0$, therefore the bars of the barcode are unnecessary because the same information is shown by plotting the point of death for that component, shown by the line in black.

The barcode diagram above shows how all the authors contribute to building a network on the point cloud of the embedded vocabulary. However, since we used concentric hyperspheres on an author-by-author basis, we can visualize how quickly individual authors connect to each node on the network when they are alone. An example of this diagram can be seen below in figure 22, the analysis of these figures will be discussed in the results section.

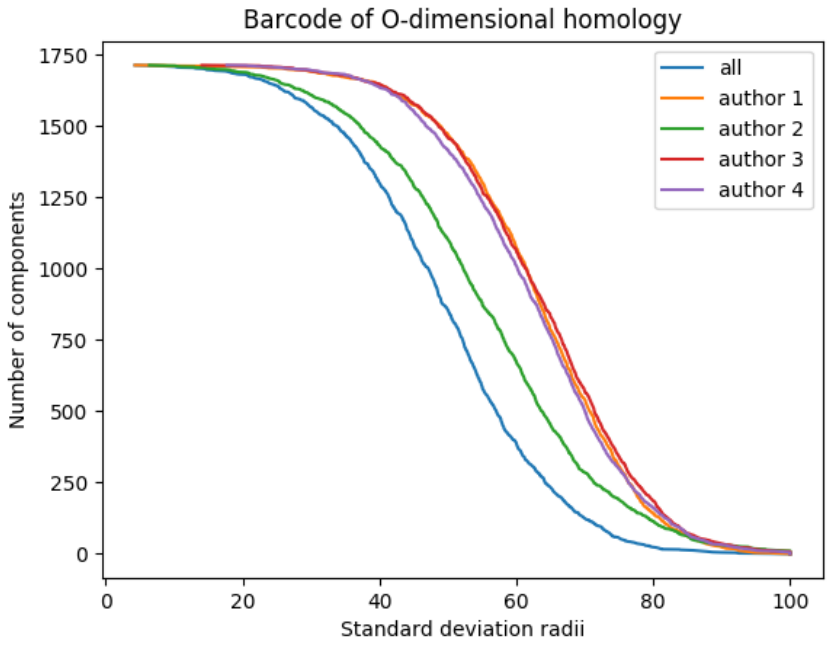

Figure 22 By plotting only the points of death for each component, we can visualize multiple network permutations at once. Shown above is the collective H_0 for authors 1 through 4, as well as how each individual author would construct their H_0 if they were alone.

The 0D homology is better visualized by the barcode because of the uniform birth of its components. Meanwhile, the 1D homology has varying births and deaths. Visual inspection of a 1D homology barcode diagram is possible when there are approximately 10^1–10^2 nodes, however our system requires analysis of networks on 10^2–10^3 nodes, call the number of nodes N. Since maximum number of cycles is $\binom{N}{3}$ the complexity of the 1D homology increases exponentially with N, which means the number of elements on the y-axis of the barcode also increase exponentially. Therefore, we instead consider the persistence diagram, which visualizes a homology on axes which are independent of N. An example persistence diagram can be seen below in figure 23 for our test dataset.

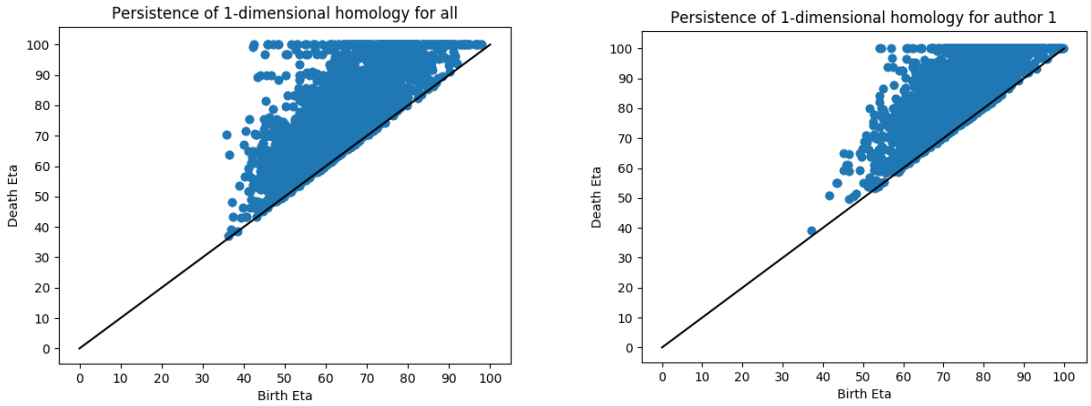

Figure 23 This is the H_1 output for the collective authors (1-4) and for author 1 plotted on a persistence diagram. The density of the data is prohibitive for analysis, and we will explore a density plot of these diagrams in the results.

Lastly, we explore the 0D homology with a weighted scoring. Until now we have analyzed the network from just an edge perspective, here we reorient and consider the interaction of nodes and edges. For notation, we will be using the term investigator to refer to a faculty member of interest, and the remaining faculty will be called members. The investigator and all members have their distinct corpuses, and the investigator has a dictionary of all the members and their scores, initialized at zero. Here we seek to rank the faculty set as collaborators of the investigator by determining how early and frequently the investigator forms edges with other members' keywords. By iterating through the list of filtered edges we can see at what η an edge is formed between two words, and what investigator is responsible for the edge. The source node of this edge must belong to the investigator, while the target node belongs to at least one author. Should the target word only belong to the investigator, then we do nothing. However, if the target word belongs to other authors, then we add $100 - \eta$ to the score of the investigator-collaborator score defined earlier. Thereby ranking how the investigator connects with each member of the faculty set. Moreover, we track how each word contributes to the final scores between each investigator and collaborator, allowing the user to probe into what keywords resulted in the ranking. For the sake of consistency, each collaborator score is rescaled between 1 and 0 by dividing by the maximum score for the give investigator-collaborator relationship. The scoring process is shown below in figure 24.

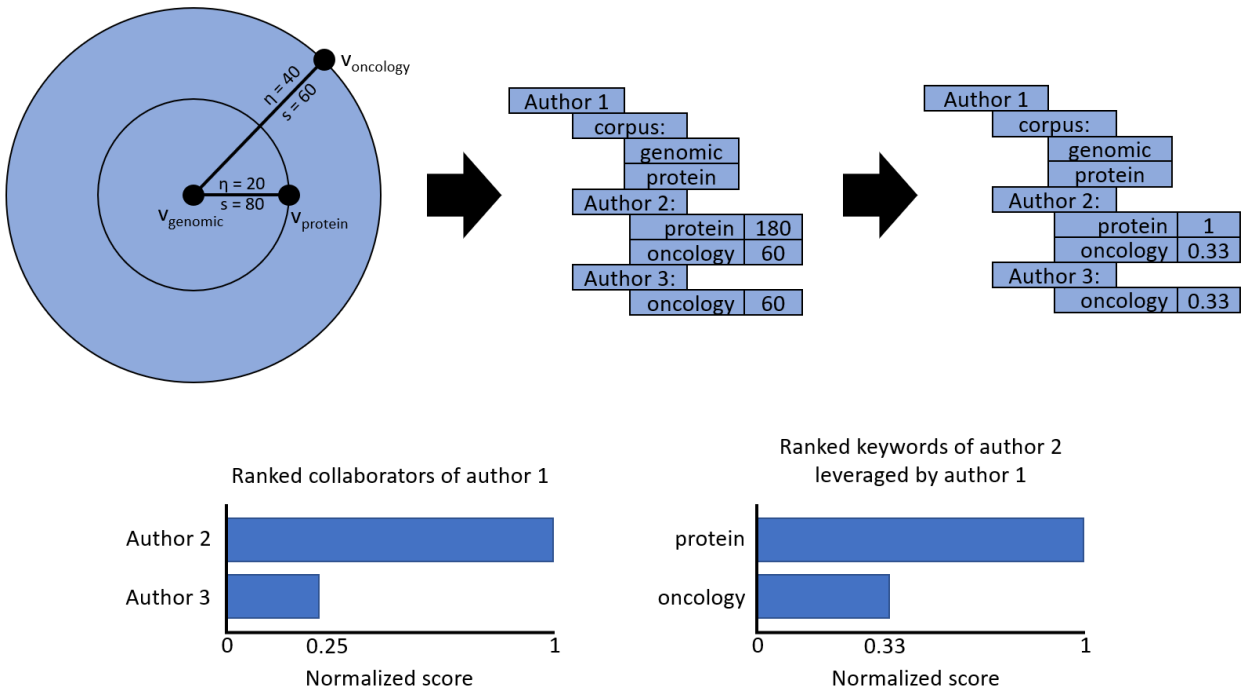

Figure 24 Author 1 has a corpus with words "genomic" and "protein", author 1 also has a list of the other faculty in the system and their respective corpuses. As connections are formed between author 1's nodes and the others, we sum the scores of those connections. For authors who use the exact same word, we give them score that connection as 100. For example, $v_{protein}$ is used by both author 1 and author 2, so they immediately share a score of 100. Then, $v_{genomic}$ forms an edge with $v_{protein}$ at $\eta = 20$, which results in an additional 80 points for author 1's scoring of author 2 (but not vice versa), for a total score of 180. This process continues until $\eta = 100$. After which, the scores are rescaled between 1 and 0 by dividing by the maximum. We are then able to plot a ranking of potential collaborators for author 1. In addition, we can also plot a rankings of the keywords used by another author as they pertain to author 1.

3.5 Results and discussion

To demonstrate the application of TDA results in an accurate collaboration model we have selected a test dataset that is verifiable at the Purdue campus. This dataset is the most recently published Weldon School of Biomedical Engineering (BME) faculty list [30]. In addition to the BME faculty, we imported Pete Pascuzzi, Loran Nies, and Giulio Caviglia to investigate the impact of external authors. Recall that the SCOPUS API data retrieval is limited to faculty affiliated with Purdue, so faculty who have yet to publish with Purdue will be removed from the list. After data retrieval, the publications of each faculty member were processed with the methods described in section 2. When constructing the network, we only allow a small set of faculty to form edges. This subset is

known as the selected faculty set and allows us to understand the keyword network from their collaborative perspective. The investigation of these select faculty on the BME network allows us to understand how each member is related in the greater context of BME. For terminology, the BME point cloud is the keywords of all the BME faculty, and not that of just the selected faculty. The output of this processing was also shown in the methods, but the exact interpretation and relevance will be elaborated here.

The first output to interpret is the custom keyword vocabulary from the network visualization tool. The custom vocabulary is the most subjective output of our system because it relies on user input. Since we seek to verify the system, we only consider the ideal use of the tool. Recall that figure 18 contained a star, the structure where a single middle node exclusively connects to satellite nodes. Stars have been shown to be important points of intersection for data clusters [23]. Once each word in the star from figure 18 is saved, we retrieve all publications that use any word in the selected set. We consider the resulting set of publications as a topological cluster of articles. To benchmark this technique will investigate the topological cluster with classic clustering techniques. In figure 25 we show the agglomerative clustering of this set with respect to the Jaccard distance.

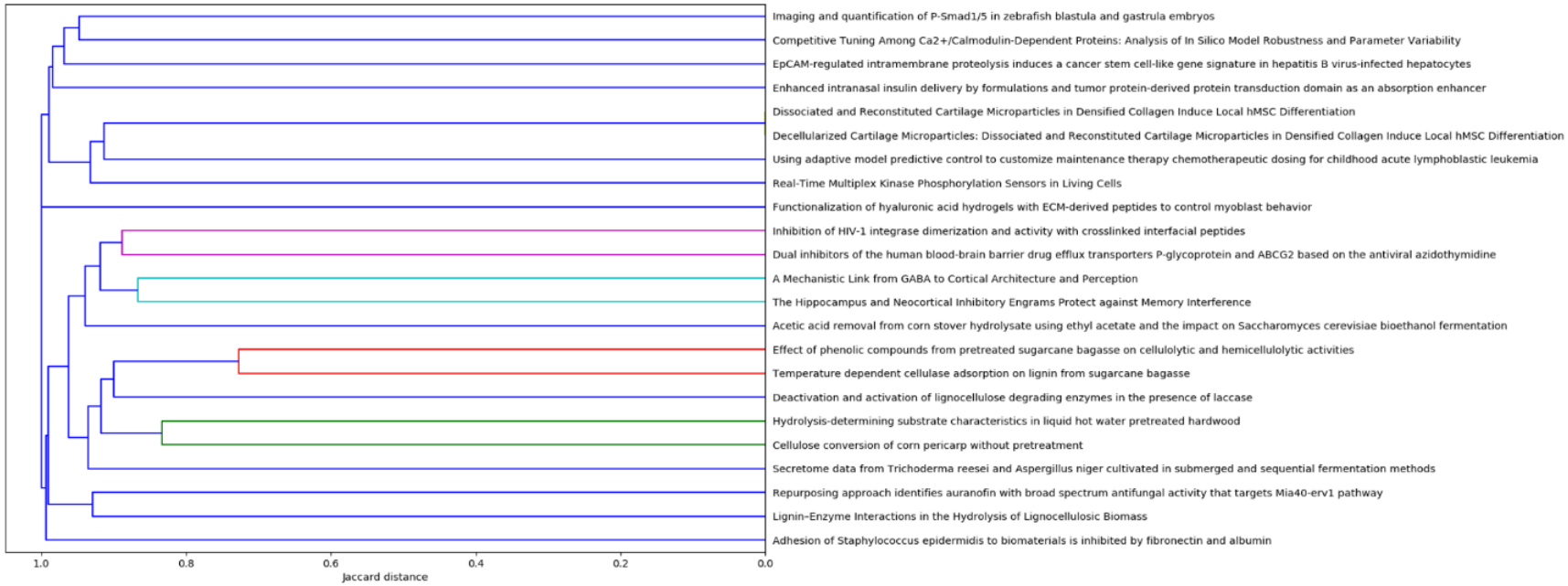

Figure 25 Agglomerative clustering of the publications who generated the star in figure 18. Blue indicates clustering above 0.9, and non-blue indicate clustering before 0.9. This figure is composed of the 23 publications that contained at least one keyword from figure 18.

The limitations of hierarchical clustering are clearly depicted above. There are 23 publications shown above, 12 of which are works of a single author's lab. This author, Dr. Ladisch, has a total of 21 publications affiliated with Purdue University at the time of this data collection. Meaning, the agglomerative clustering above shows no distinct clusters for over half his body of work at Purdue University. Naturally, this is a result of the Jaccard distance only considering words that are exactly the same, whereas our technique considers semantically similar words. This result is the first indication that meaningful data is embedded in the topology of embedded keyword networks. This single example of TDA capturing patterns, which were indiscernible to classic clustering, is justification for the further investigation of the topology. As such, we can move the discussion from subjective vocabulary selection to more concrete forms of topological analysis. The zero-dimensional homology (H0) is the measure of network connectivity over the step-wise increase of hypersphere radius η. Analysis of H0 is completed by visual inspection of the barcode. Figure 22 was displayed in the results to show the necessity of the data as a line, that same data is the output of the BME test dataset. The authors displayed on this figure are the selected faculty. In other words, we are observing how the selected authors fit into the BME point cloud.

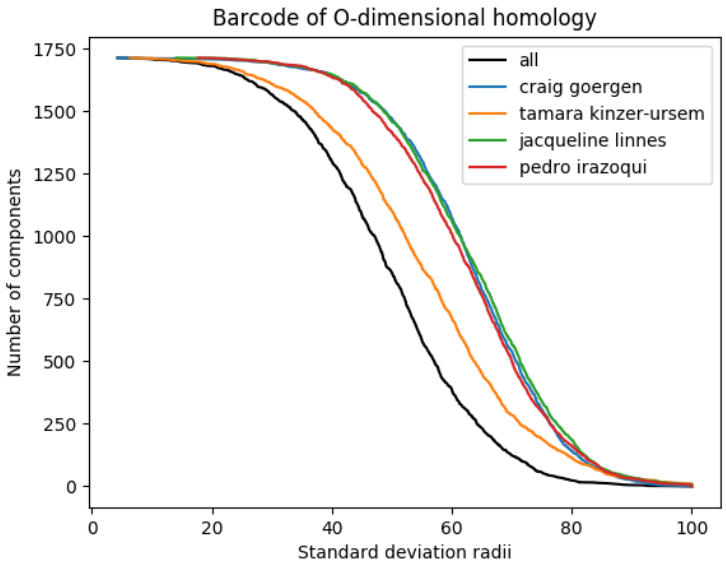

Figure 26 A simplified barcode diagram that allows the evaluation of each individual network construction, as well as the collective network between the four faculty. The rapid convergence to zero indicates interdisciplinary work.

With H0 we observe how quickly the connectivity of the network converges to a single component. Not only can we observe the collective contributions to network construction, but also how individual authors construct their unique network on the BME point cloud. The differences and similarities between the sigmoidal trends are clear, and the network that connects every component at the smallest radius is unsurprisingly the collective network. In the context of individual networks, the degree of connectivity as a function of η clearly indicates how the author fits within the greater context of the BME dataset.

For example, we notice that Dr. Kinzer-Ursem's individual network converges to a single component before any other author's network. We know Dr. Kinzer-Ursem develops tools that apply across an array of biomedical research topics and publishes in many sub-disciplines of BME. As described in the methods, interdisciplinary work allows an author form connections earlier because the larger deviation increases hypersphere expansion. Also, authors who publish in multiple sub-disciplines of BME are better able to seed the point cloud because they span more of the whole dataset.

Conversely, Dr. Georgen, Dr. Linnes, and Dr. Irazoqui have more focused subjects and applications to their respective fields. As explained above, having a focused research interest in a single sub-discipline limits the reach and span of nodes for that author. The more focused topics result in smaller deviations for the hypersphere expansion, and the single sub-discipline of the author limits their entry point to form edges. The result is that more focused authors within the dataset converge to a single component at the slowest radius increment, and some authors don't ever reach a single component in the given η boundaries.

With the analysis of H0 complete, we turn our discussion to first-dimensional homology (H1). Recall that H1 investigates the holes within a network. Similar to the barcode discussed for H0, the analysis of the persistence diagram is visual inspection. Figure 23 shows the raw H1 persistence diagrams of two networks on the BME point cloud, a) the collective network, and b) Dr. Pascuzzi's network. Notice, that the density of points is prohibitive for visual analysis of the diagram. As such, we depict the persistence diagram as a 2D Histogram (figure 27), a density plot of the data that depicts the concentration of gaps for any given lifespan.

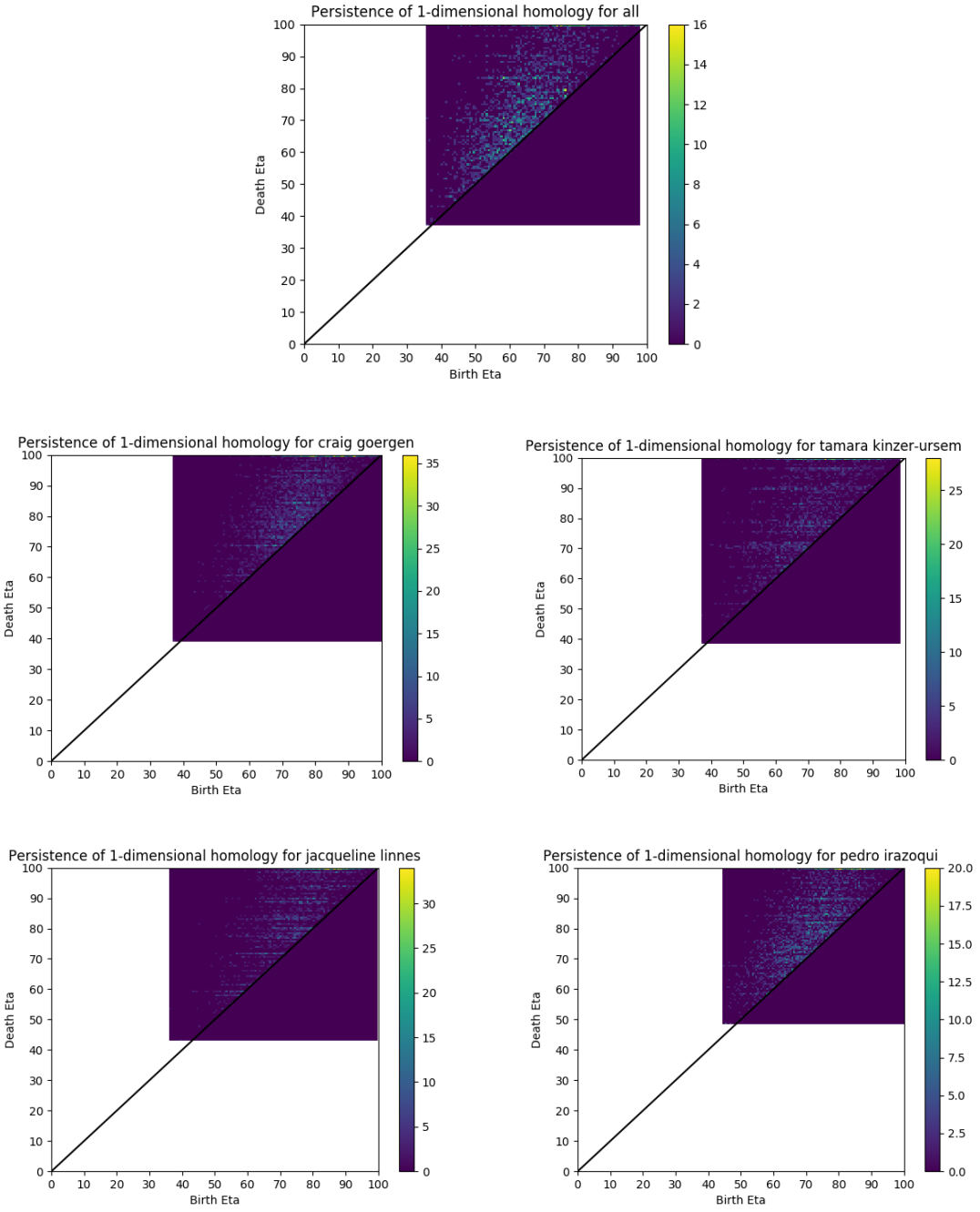

Figure 27 H1 for the collective set of authors, and the individual permutations. The density of each homology is indicated by the intensity bar, which varies in scale. a) The collective network has more holes, but each are less persistent. b) Each individual author has a unique H1 profile; Dr. Irazoqui and Dr. Goergen both trend toward a dense center; conversely, Dr. Kinzer-Ursem and Dr. Linnes distribute the density across a broader spectrum of persistence.

The investigation of H1 is in its preliminary stages due to the lack of statistical methods for analysis of persistence data [20]. H1 analysis on 1715 input points is computationally taxing and a non-deterministic polynomial time hard problem (np-hard). We accomplish local H1 construction of our system within the order of 10^4 seconds (2.7 hours) and return the points responsible for each hole. Knowing the point composition of each hole permits further investigation into the underlying cause of network gaps. However, due to the volume of holes, efficient gap inspection couldn't be implemented. More tools must be developed to identify and analyze H1 results, such analysis is outside the scope of this work.

While the computational filtering and extrapolation of H1 data is outside the scope of this project, the analysis of H0 is simpler and implemented for the modelling of collaboration. The conceptual process of this algorithm is outlined in figure 24, and here we explore the results of this analysis on the BME test dataset. Some example outputs of computational H0 analysis are shown below for varying faculty. The following set of figures that demonstrate this point are between pages 47 and 52.

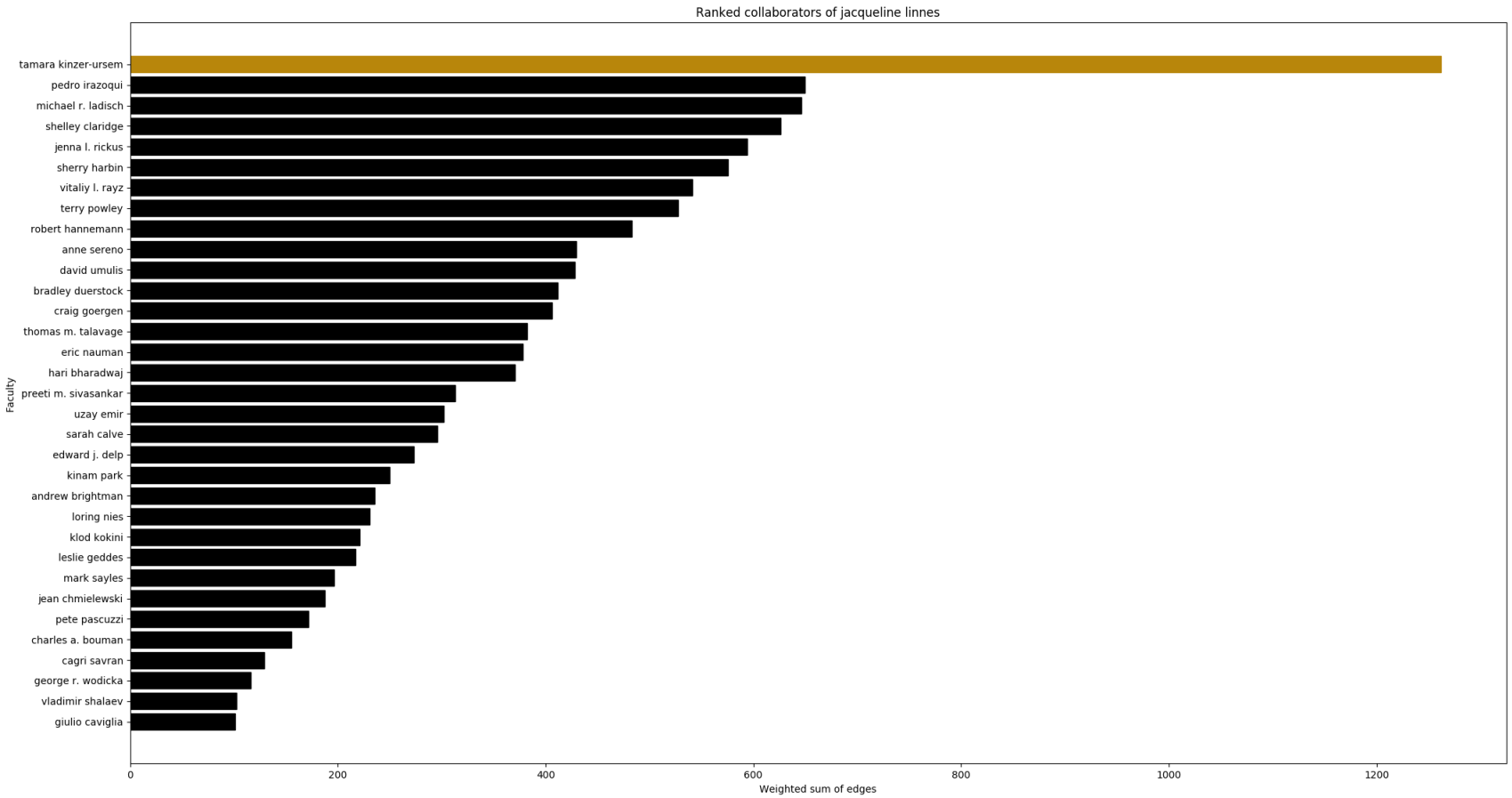

Figure 28 The weighted sum of connections for a faculty member results in a ranked list of faculty. Dr. Linnes' ranking of faculty results from her connections with all other faculty in the BME test dataset. The known ground truth of the BME department confirms this ordering to be accurate, and known co-authorships are denoted with a gold bar. Notice Dr. Caviglia, a math professor, is the least likely to work with Dr. Linnes. Dr. Pascuzzi (ranked 28) and Dr. Nies (ranked 23) are not in the BME department, however, they publish on bioinformatics and water treatment, which is complementary with the work of Dr. Linnes, and justifies their position above other BME faculty.

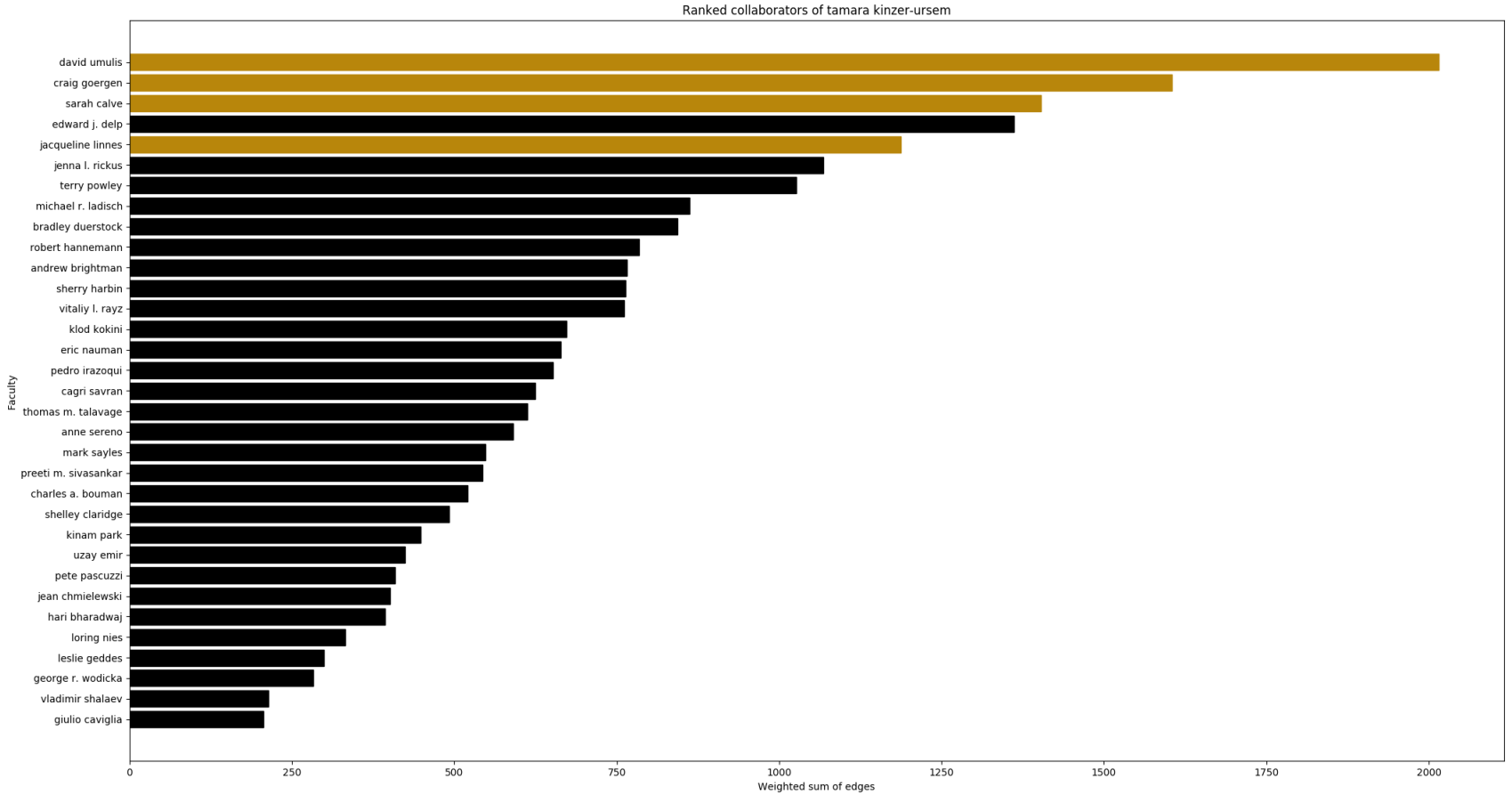

Figure 29 Dr. Kinzer-Ursem's faculty ranking uniquely shows how interdisciplinary work can be visualized by TDA. Once agaoin known co-authorships are denoted in gold. Notice that Dr. Delp is not a known co-author of Dr. Kinzer-Ursem, and their use of keywords indicates a possible overlap in interests. The potential collaboration between them is feasible given Dr. Delps computational work and Dr. Kinzer-Ursem's study of computational modeling. This example highlights the possibility for spurring intra-departmental connections that have yet to form.

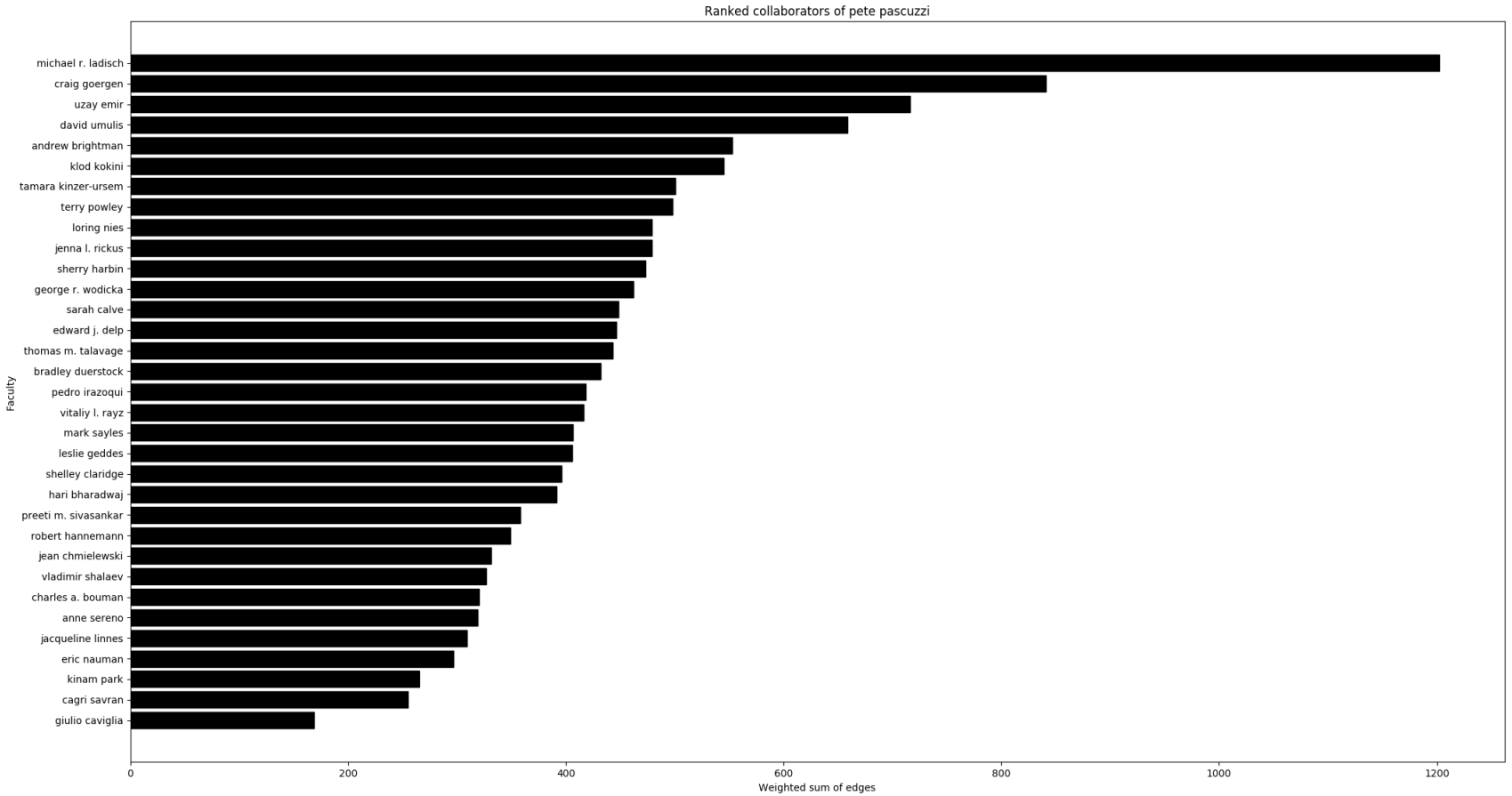

Figure 30 The point cloud formed is predominately BME faculty, only Dr. Pascuzzi, Dr. Caviglia, and Dr. Nies are non-BME contributors to the embedded vocabulary. As such, Dr. Pascuzzi has no known co-authorships with this set of faculty, and this ranking is an indication of where he would fit into the department. Those who ranked highest with Dr. Pascuzzi work at the interface of computation and biology. Those who ranked lowest are Dr. Caviglia of the math department, and Cagri Savran, a joint-appointed BME professor whose main appointment is mechanical engineering. In total, this use-case serves to coordinate the evaluation of new faculty and model their impact on the perspective department.

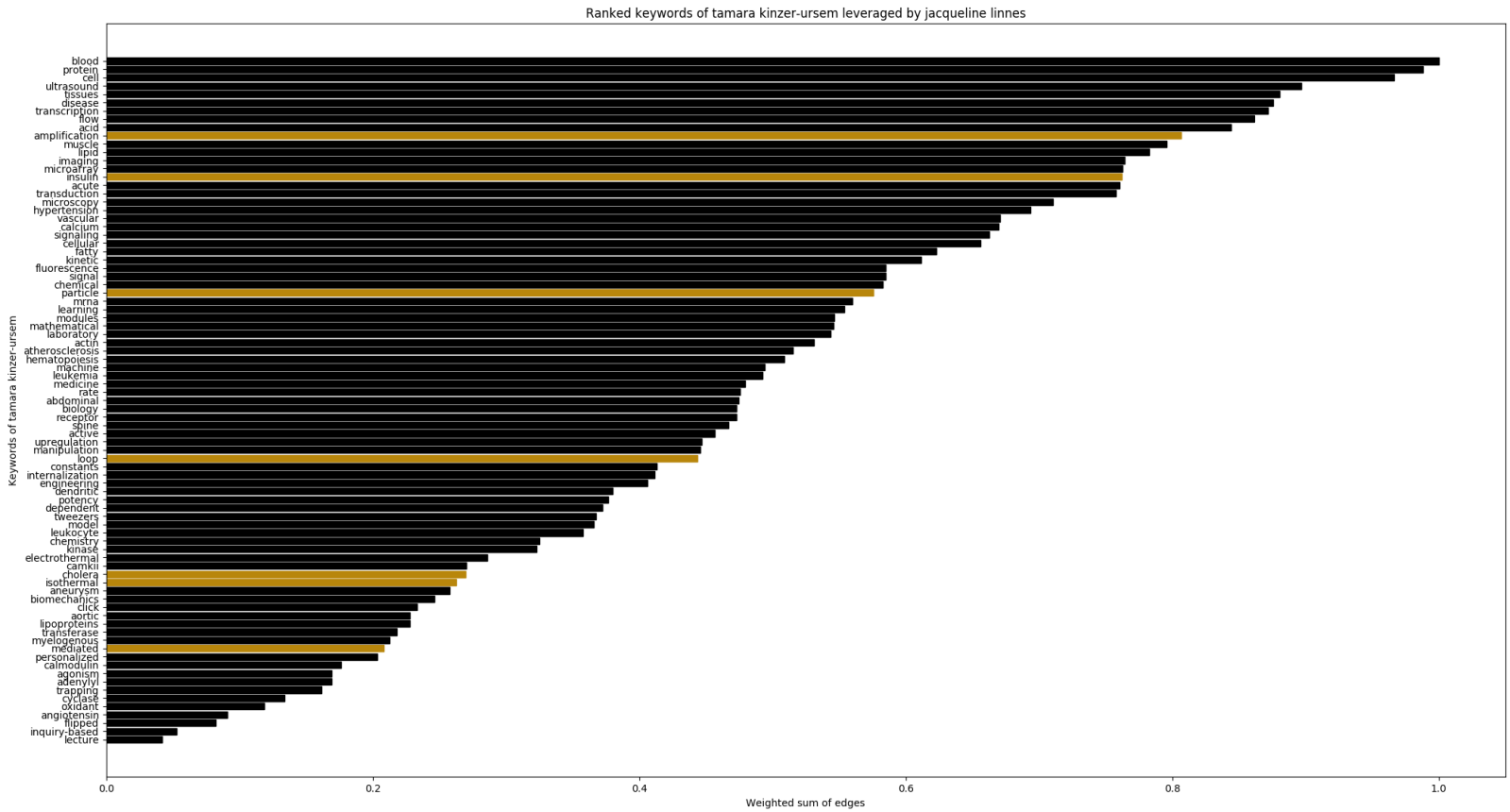

Figure 31 The ranking of faculty serves to distinguish overlapping interests and gaps in collaboration. Naturally, the further probing of these interactions yields a keyword ranking. Here we show the keyword corpus of Dr. Kinzer-Ursem ranked through the lens of Dr. Linnes' work, and denote keywords from co-authored publications with gold bars. Their recent collaborations in blood born, bacterial, and water born pathogen DNA detection is apparent within the highest scoring group of keywords. Likewise, Dr. Kinzer-Ursem's work with calmodulin and protein systems (e.g. adenylyl cyclase, agonism, angiotensin) are ranked low, indicating the divergence of interests between the two faculty.

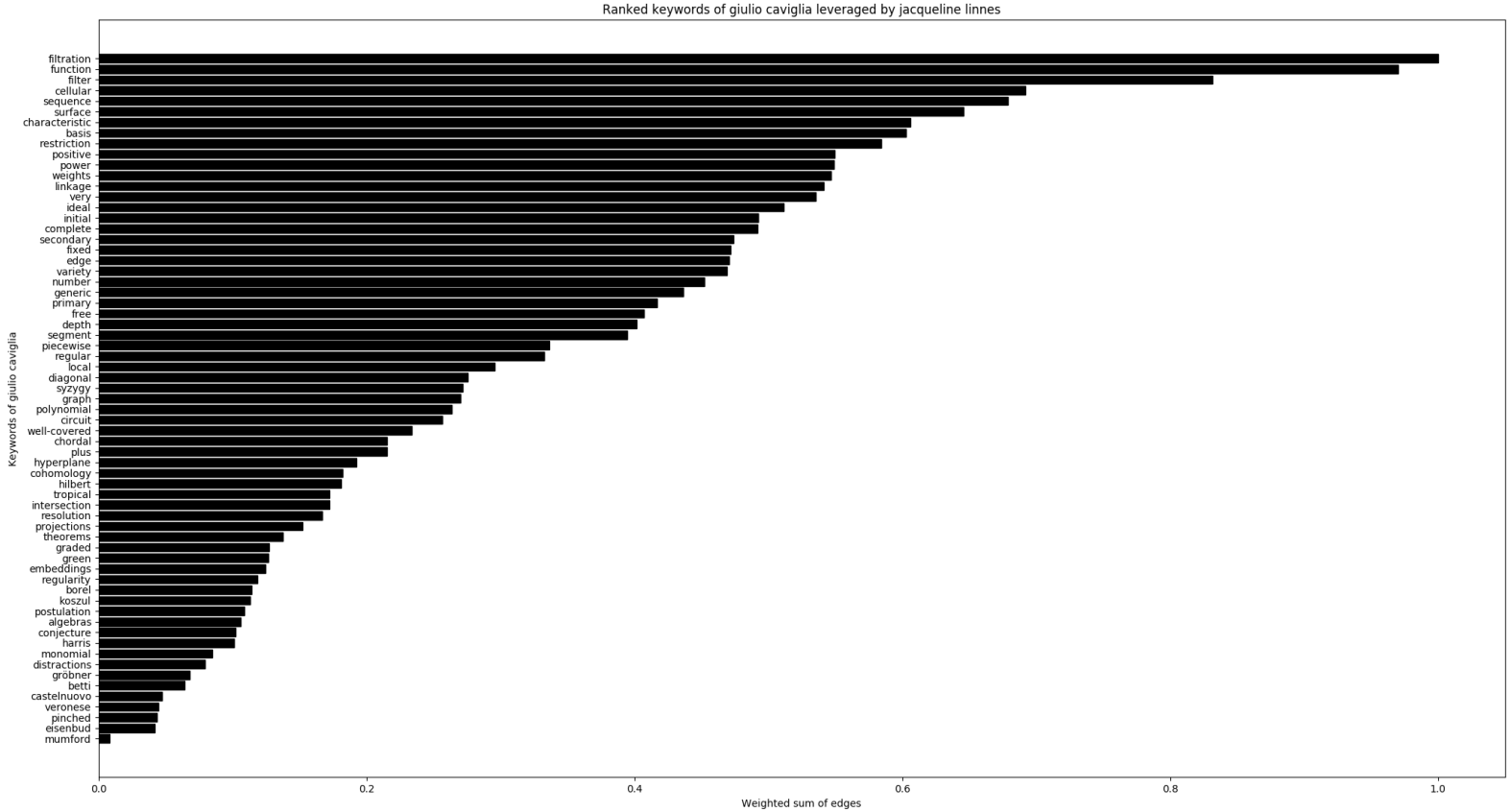

Figure 32 Dr. Caviglia is the lowest ranking faculty for Dr. Linnes in figure 28, and since they aren't co-authors there is no possibility for gold bars within the figure. Probing further into this interaction, we observe that Dr. Linnes ranks Dr. Caviglia's keywords according to her BME expertise, an artifact of our algorithm design. This calls attention to the first limitation of our work, where ambiguous terms can have different meanings across disciplines. For instance, filtration can refer to a simplicial complex (as it does in this article), or to the filtration of objects and data. Similarly, Dr. Caviglia uses cellular in the context of high-dimensional geometries, whereas Dr. Linnes applies it in the traditional meaning. However, Dr. Caviglia's ranking makes it clear that our system is robust to noise, and not affected by keyword homonyms.

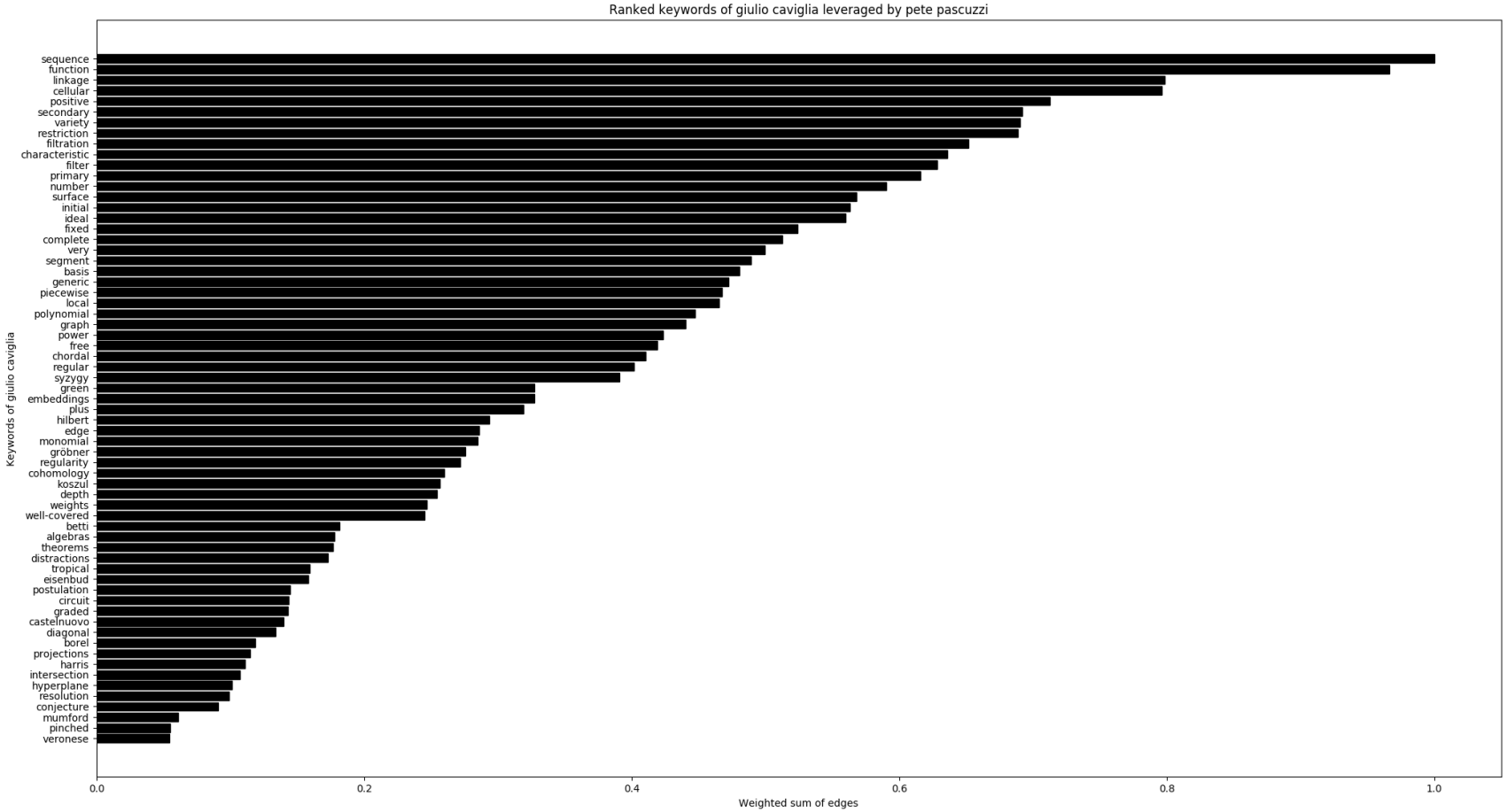

Figure 33 Here we observe Dr. Caviglia's vocabulary through the lens of Dr. Pascuzzi, neither of which are BME faculty. Once again, the ambiguous use of cellular is observed. In contrast, the potential computational relationship is made clear by the terms sequence, function, and linkage. In total, Dr. Caviglia's niche position as a pure mathematician in a biomedical context is noted in his continually low ranking. Showing that we can not only detect readily made collaborations, but also which collaborations would require intermediate players or extensive dissemination.

4. CONCLUSION

This work develops a novel application of TDA on highly structured keyword vocabularies and establishes a robust method of scholarly pattern identification. We submit that this method circumvents the limitations of classical keyword networks and Jaccard clustering through the implementation of word embedding tools. The embedding of our highly structured vocabulary to numerical vectors facilitates the identification of semantically similar keywords, thus quantifying the manifold of uncontrolled vocabulary. The resulting point cloud is equipped to undergo TDA, however, the data hierarchy is no longer intact due to the embedding process. This loss lead to the development of a novel simplicial complex that imposes and preserves subset hierarchy within the point cloud. The construction method is rescaled to reflect the distribution of the subsets, in practice this is the keyword set for a given publication.

Following this construction, the zero and first homologies are calculated. The lack of TDA toolkits in the python environment required the development of custom H0 and H1 toolkits (figure 19, figure 20). The validity of these toolkits was confirmed by network inspection using our 3D visualization application. This same visualization application is used to create custom vocabularies for the investigation of topological structures. A proof of concept showed that these structures can identify underlying patterns, and it was shown that the agglomerative-Jaccard technique could not (figure 25). Lastly, we explored H0 by computationally scoring weighted connections. The results of this scoring yield an ordered list of suggested faculty who share interests with the selected author. The relationship between this investigator and their colleagues is further investigated by scoring the potential collaborator's keywords through the investigator's perspective. This allows an in-depth view into how any two faculty may interact, what subjects they share an interest in, and how interdisciplinarity impacts the ranking of faculty.

This work has several limitations that constrain its use and will need to be addressed in the future. First, the computational complexity of network construction and homology analysis are not efficient and require several hours to fully compute. This limitation is complicated further by the first homology having a dense distribution of points which limits visual analysis of persistent holes. Second, the selection of vocabulary in the web application allows a user to potentially ignore the

topological structures and randomly select a custom vocabulary. This limits the efficacy of the network construction and must be resolved for the accurate investigation of the topology. Lastly, the word embedding itself limits the network construction. Ambiguous words and shallow understanding of complex subjects prevent the embedding from capturing all semantic relationships between keywords. Each of these limitations have the potential to be resolved with future work.

4.1 Future technical development

In future work, this project could develop more post-analyses of the homologies. Future analysis of the zero homology could identify the half max point of the sigmoidal curve and use this as an classify the interdisciplinarity of an author within the field. This more formalized analysis would show that interdisciplinary faculty have a lower half-max η than that of their more focused counterparts. Following the post-analysis of H0, the strengthening of analysis through integrating H1 data could lead to more accurate models and better identification of gaps within the research. This work would require the procedural filtering and cleaning of H1 data. In addition, the calculation of H1 is the most computationally expensive process in our system. Future work could seek to parallelize or optimize the hole detection algorithm. Once the H1 calculation is streamlined, the H2 calculation may be feasible to implement. The introduction of H2 would allow us to observe the 3D structures generated by the network construction.

There is much work to be done regarding the visualization, user interface, and web application development; ultimately the identification of topological structures within the visualization will allow users to download interesting bibliographies. The visualization process is currently in development and not ready for laymen use. Further visualization and web-app improvements include the development of a front-end interface, the ability to select dates for retrieved publications, and the ability to filter publications by citation counts. Following these improvements, there's a need to procedurally select structures in the visualization to mitigate the misuse of the application. Currently, a user could select nodes at random and still generate a curated bibliography. This misuse would negate the topological clustering of data and result in a meaningless insight into the dataset. In the future, work must be done to identify topological structures and classify

their meaning for a keyword network. After which, these classified structures can be used to autonomously find unknown clusters of publications within the network.

Future work could investigate fine-tuned word embeddings, otherwise known as the local training of a word embedding that identifies specific meanings and contexts within a field. The use of the pre-trained Wikipedia embedding may not contain the depth necessary to capture all academic material. One could create a scholarly trained embedding, which would provide further insight into the semantic relationships between academic vocabulary or even train a new language for non-English publications. Future work might look to post-training fine tuning, where a pre-trained embedding is further specified to a certain class or field for a more comprehensive embedding. Lastly, the mining of text could yield new keywords that improve the embedding accuracy and identifies keywords that may have been neglected by the author. The culmination of these changes would facilitate the downstream identification of sub-disciplines, specific technologies, and give context to ambiguous words.

Finally, future work could explore the evolution of the network with respect to time, which would permit the investigation of research trends, departmental decisions, and the roster of faculty. Specifically, this could be accomplished by incrementally increasing the window for publication retrieval. Each iteration the set of publications and keywords would undergo embedding and TDA. The resulting homologies would then be compared across the varying years of publications to highlight the differences and trends within the faculty set. The culmination of these improvements could result in a valuable tool that applies across all scholarly work.

4.2 Future applications of this work

The practical applications of this research began with a focus on predicting collaboration and facilitating the development of new fields. Over the course of development, other applications for this work have become clear. One such application is the prediction of new faculty interactions within an established department. The research landscape of a department can shift with the introduction of a single faculty and drive new collaborations and innovation. The ability to predict and identify these paradigm shifts could lead to academic progress and an institutional advantage.

Next, we consider the impact of this work on systematic reviews. A systematic review is a comprehensive literature search for a particular field or study. Like keyword networks, systematic reviews are hindered by uncontrolled vocabulary. Therefore, the visualized network of semantically close keywords may streamline the process identifying search terms. Even more so, the topological clustering of the publications may allow for rapid discovery of related topics and keywords. With this simplification, the crafting of intricate search queries may become a high throughput process and increase the efficiency of systematic review.

Within the same domain, this work could improve the identification of medical developments and publications within a given field. Current medical practice requires the fast and accurate search of cutting-edge work to improve patient outcomes [31]. By leveraging the semantic relationship between keywords, we could craft medical search queries that specifically target certain topics within health sciences. The efficient and comprehensive nature of such queries could allow the physician to identify the latest developments within that field. The saved time and resources would benefit the patient with a timely response and permit the allocation of resources to more patients.

REFERENCES

[1] M. E. Falagas, E. I. Pitsouni, G. A. Malietzis, and G. Pappas, "Comparison of PubMed, Scopus, Web of Science, and Google Scholar: strengths and weaknesses," *FASEB J.*, vol. 22, no. 2, pp. 338–342, Sep. 2007.

[2] F. Narin, "Bibliometrics," *Annu. Rev. Inf. Sci. Technol.*, pp. 35–58, 1977.

[3] Y. Sun, R. Barber, M. Gupta, C. C. Aggarwal, and J. Han, "Co-author Relationship Prediction in Heterogeneous Bibliographic Networks," in *2011 International Conference on Advances in Social Networks Analysis and Mining*, 2011, pp. 121–128.

[4] R. Johnson, A. Watkinson, and M. Mabe, "The STM Report An overview of scientific and scholarly publishing." International Association of Scientific, Technical, and Medical Publishers, Oct-2018.

[5] F. J. COLE and N. B. EALES, "THE HISTORY OF COMPARATIVE ANATOMY: PART I.—A STATISTICAL ANALYSIS OF THE LITERATURE," *Sci. Prog. 1916-1919*, vol. 11, no. 44, pp. 578–596, 1917.

[6] W. Hood and C. Wilson, "The Literature of Bibliometrics, Scientometrics, and Informetrics," *Scientometrics*, vol. 52, no. 2, pp. 291–314, Oct. 2001.

[7] É. Archambault, D. Campbell, Y. Gingras, and V. Larivière, "Comparing bibliometric statistics obtained from the Web of Science and Scopus," *J. Am. Soc. Inf. Sci. Technol.*, vol. 60, no. 7, pp. 1320–1326, 2009.

[8] S. M. Gerrish and D. M. Blei, *A Language-based Approach to Measuring Scholarly Impact*. .

[9] A. Duvvuru, S. Kamarthi, and S. Sultornsanee, "Undercovering research trends: Network analysis of keywords in scholarly articles," in *2012 Ninth International Conference on Computer Science and Software Engineering (JCSSE)*, 2012, pp. 265–270.

[10] "Elsevier Developer Portal." [Online]. Available: https://dev.elsevier.com/tecdoc_ir_cris_vivo.html. [Accessed: 15-Jul-2019].

[11] L. Dietz, S. Bickel, and T. Scheffer, "Unsupervised Prediction of Citation Influences," in *Proceedings of the 24th International Conference on Machine Learning*, New York, NY, USA, 2007, pp. 233–240.

[12] H. Gui, Q. Zhu, L. Liu, A. Zhang, and J. Han, "Expert Finding in Heterogeneous Bibliographic Networks with Locally-trained Embeddings," *ArXiv180303370 Cs*, Mar. 2018.

[13] H. J. Lowe and G. O. Barnett, “Understanding and Using the Medical Subject Headings (MeSH) Vocabulary to Perform Literature Searches,” *JAMA*, vol. 271, no. 14, pp. 1103–1108, Apr. 1994.

[14] J. Pennington, R. Socher, and C. Manning, “Glove: Global Vectors for Word Representation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014, pp. 1532–1543.

[15] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed Representations of Words and Phrases and their Compositionality,” in *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 3111–3119.

[16] T. Temčinas, “Topological Tools in Data Analysis,” Masters in Mathematics and Philosophy, University of Oxford, Oxford, England, 2018.

[17] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data Clustering: A Review,” *ACM Comput Surv*, vol. 31, no. 3, pp. 264–323, Sep. 1999.

[18] V. Satopaa, J. Albrecht, D. Irwin, and B. Raghavan, “Finding a ‘Kneedle’ in a Haystack: Detecting Knee Points in System Behavior,” in *2011 31st International Conference on Distributed Computing Systems Workshops*, Minneapolis, MN, USA, 2011, pp. 166–171.

[19] T. Temčinas, “Local Homology of Word Embeddings,” *ArXiv181010136 Cs Math*, Oct. 2018.

[20] F. Chazal and B. Michel, “An introduction to Topological Data Analysis: fundamental and practical aspects for data scientists,” *ArXiv171004019 Cs Math Stat*, Oct. 2017.

[21] A. Zomorodian, “Fast construction of the Vietoris-Rips complex,” *Comput. Graph.*, vol. 34, no. 3, pp. 263–271, Jun. 2010.

[22] S. Dantchev and I. Ivrissimtzis, “Efficient construction of the Čech complex,” *Comput. Graph.*, vol. 36, no. 6, pp. 708–713, Oct. 2012.

[23] B. Gao, T.-Y. Liu, X. Zheng, Q.-S. Cheng, and W.-Y. Ma, “Consistent Bipartite Graph Co-partitioning for Star-structured High-order Heterogeneous Data Co-clustering,” in *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, New York, NY, USA, 2005, pp. 41–50.

[24] S. Niwattanakul, J. Singthongchai, E. Naenudorn, and S. Wanapu, “Using of Jaccard Coefficient for Keywords Similarity,” *Hong Kong*, p. 6, 2013.

[25] O. Levy and Y. Goldberg, “Linguistic Regularities in Sparse and Explicit Word Representations,” in *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, Ann Arbor, Michigan, 2014, pp. 171–180.

[26] M. Bostock, V. Ogievetsky, and J. Heer, “D3 Data-Driven Documents,” *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 12, pp. 2301–2309, Dec. 2011.

[27] R. Cabello, *JavaScript 3D library. Contribute to mrdoob/three.js development by creating an account on GitHub*. Three, 2019.

[28] V. Asturiano, *3D force-directed graph component using ThreeJS/WebGL: vasturiano/3d-force-graph*. 2019.

[29] “MEDLINE®/PubMed® Journal Article Citation Format,” *MEDLINE®/PubMed® Journal Article Citation Format*, 08-Aug-2018. [Online]. Available: https://www.nlm.nih.gov/bsd/policy/cit_format.html. [Accessed: 15-Jul-2019].

[30] “Faculty - Alphabetical by Campus - Biomedical Engineering - Purdue University,” *Weldon School of Biomedical Engineering*. [Online]. Available: https://engineering.purdue.edu/BME/People/Faculty. [Accessed: 15-Jul-2019].

[31] H. C. H. Coumou and F. J. Meijman, “How do primary care physicians seek answers to clinical questions? A literature review,” *J. Med. Libr. Assoc.*, vol. 94, no. 1, pp. 55–60, Jan. 2006.