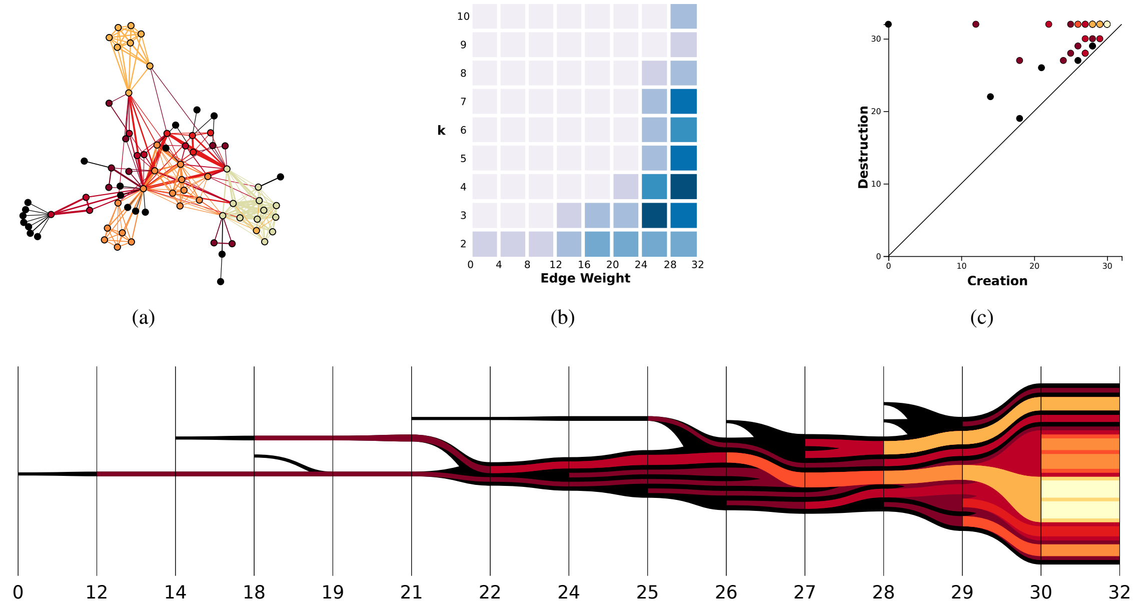
Clique Community Persistence: A Topological Visual Analysis Approach for Complex Networks

Bastian Rieck, *Member, IEEE*, Ulderico Fugacci, *Member, IEEE*,
Jonas Lukasczyk, *Student Member, IEEE*, and Heike Leitte, *Member, IEEE*



(a) (b) (c)

(d)

Fig. 1. All components of our proposed approach, shown for the "Les Misérables" co-occurrence network, which we analyze in Section 4.1. If the size of the graph permits it, we show a force-directed graph layout of the network (a), where each vertex is colored according to the maximum degree of their associated clique community. A 2D histogram (b) of the maximum number of individual clique communities for all edge weights and all clique degrees helps in finding relevant edge weight thresholds. The persistence diagram (c) gives an overview of all clique communities and their merging behavior. The nested graph (d) shows how individual clique communities merge when the edge weight of the network increases. Furthermore, it permits tracking the evolution of a single community.

Abstract—Complex networks require effective tools and visualizations for their analysis and comparison. Clique communities have been recognized as a powerful concept for describing cohesive structures in networks. We propose an approach that extends the computation of clique communities by considering persistent homology, a topological paradigm originally introduced to characterize and compare the global structure of shapes. Our persistence-based algorithm is able to detect clique communities and to keep track of their evolution according to different edge weight thresholds. We use this information to define comparison metrics and a new centrality measure, both reflecting the relevance of the clique communities inherent to the network. Moreover, we propose an interactive visualization tool based on nested graphs that is capable of compactly representing the evolving relationships between communities for different thresholds and clique degrees. We demonstrate the effectiveness of our approach on various network types.

Index Terms—Persistent homology, topological persistence, cliques, complex networks, visual analysis.

1 Introduction

Complex network analysis [35, 42, 47] is an active research topic with applications in multiple fields of interest, such as sociology, physics, electrical engineering, biology, and economics. Generally, complex networks are used to represent different kinds of systems that consist of individuals interacting with each other. A *local* analysis often focuses on the connections of a single node and its local relevance. Centrality measures such as betweenness or closeness help identify key nodes. A study of structural properties of the *entire* network, by contrast, concentrates on groups of nodes and their connections. The connectivity of a network can be measured using a large variety of attributes and descriptors such as density, cohesion, diameter, and small-worldness. For this kind of analysis, it is necessary to study *communities* or clusters [16]. Although a concrete definition depends on the application context, a community is usually considered to be a highly-connected group of nodes of the network. All of these concepts augment the description of the local and the global structure of a network. Moreover, they can be used to compare different networks. However, despite the effectiveness of these concepts for evaluating similarities between networks, a tool

• *Bastian Rieck, Ulderico Fugacci, Jonas Lukasczyk, and Heike Leitte are with TU Kaiserslautern. E-mail:{rieck, fugacci, lukasczyk, leitte}@cs.uni-kl.de.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

for systematically comparing the global and the community structure of two networks is still missing in the literature.

A common approach for identifying communities in networks employs the detection of *clique communities*, e.g., via the clique percolation method [37]. As we formally describe later in Section 3.1, a k-clique community of a network consists of a collection of densely interconnected groups of k individuals. Although these communities are generally hard to calculate for high values of k, their use for identifying the global structure of (edge-weighted) networks leads to several advantages: (i) Different from other clustering techniques, clique percolation permits the existence of overlapping communities. This is consistent with real-world networks, which often contain overlaps between different communities, so that an actual partition would result in an unrealistic decomposition. (ii) A community-centered view permits a simplified assessment of the relevance of individual nodes based on the number of different communities they belong to. (iii) Unlike other techniques (e.g., clustering) that strongly depend on parameters set by the user, the decomposition in k-clique communities is parameter-free: there are only finitely many cliques and finding a k-clique automatically entails finding k cliques of degree $k-1$ because cliques are nested. In most of the data sets presented in Section 4, we are thus able to fully enumerate the community structure. Thanks to all these desirable properties, the use of clique communities has been recognized as a relevant and effective tool in a large number of application domains [16]. However, some issues affect clique percolation. In most applications, the clique degree k and the edge weight threshold w with which to perform the network decomposition are not properly set up but just established according to somewhat arbitrary criteria. Furthermore, the literature is lacking (i) effective visualization tools able to manage different values for k and w in a single view (ii) a theoretical framework for comparing networks according to their clique community structure.

The main contribution of this paper faces these issues by developing a tool for detecting and tracking the evolution of the clique communities of a network as both k and w vary. This has been made possible by extending the notion of *persistent homology*, a topology-based tool aimed at the characterization and the comparison of the global structure of discretized topological spaces [12], to the framework of clique communities of a network. This paper addresses a threefold task: (i) to *compute* clique communities for different values of k and w through a persistence-based algorithm, (ii) to *visualize* through an interactive tool the clique communities of a network and their evolution, (iii) to *analyze* the retrieved information using centrality and comparison measures. Specifically, we propose a new algorithm for clique community detection based on persistent homology that is able to retrieve and track communities for *any* value of k and w. We define new descriptors for comparing global structures of weighted networks. Furthermore, we develop an interactive visualization tool by combining several persistence-based feature detectors with the notion of *nested graphs* [34]. Figure 1 depicts an instance of this tool for the well-known character co-occurrence network of the historical novel “Les Misérables” by Victor Hugo; please refer to Section 4.1 for more details. Finally, we exploit the connection with persistent homology in order to define a new centrality measure for the individuals of the network based on the persistence of clique communities.

2 Related work

This section surveys related work in clique community detection, visualization techniques for graphs, and persistent homology.

2.1 Detection of clique communities

The notion of clique communities has proven to be an effective tool for analyzing a network with respect to its cohesive substructures. The detection of such communities has led to fruitful applications in numerous scientific areas such as biology [25,26], economics [22], and social network analysis [19,30,36,45]. Several methods are known for computing these communities. The first approach is due to Palla et al. [37], whose algorithm exploits the Bron–Kerbosch algorithm [3] in order to enumerate all *maximal* cliques in the network. Next, a clique-clique overlap matrix is built and used to easily retrieve clique communities for any value of k. This approach constitutes the *de facto* standard in clique community detection. Based on this method, the free software `CFinder` has been released [1]. Various improvements to this approach have been proposed in the literature [5,20,21,29,39].

2.2 Visual analysis of networks

Analyzing graphs and networks requires a complex interplay of visualization algorithms and filtering/aggregation techniques [46]. The two most common visualization techniques are (i) the matrix representation, in which the adjacency matrix of the network is displayed while any edge attributes are encoded as the entries of the matrix, (ii) the node–link diagram, in which nodes and links are shown in a planar diagram while their positions are calculated using a variety of layout algorithms. For generic undirected networks, node–link diagrams with force-directed layout algorithms are shown to outperform other layout strategies [46]. The scalability of these direct visualizations is not always guaranteed even for static networks, which we consider in this paper. Often, filtering or aggregation techniques are required [24]. In this paper, we focus on *topological*, i.e, structural, properties of a graph. Such properties are used in graph layout algorithms [2] or to guide user interactions [18]. Our approach here is different in that we represent our features in a more abstract manner. At the same time, this level of abstraction permits us to compare networks based on their structural similarity. In contrast to the few existing methods for this purpose, such as `ManyNets` [17], our tool uses a single property of the network—the connectivity of its clique communities—but is able to compare them both visually and numerically (using a well-defined distance measure).

2.3 Persistent homology in network analysis

Persistent homology, the main paradigm that we employ in this paper, is not a tool that was originally developed for complex network analysis. Nonetheless, it started being frequently adopted for analyzing the topological structure of a network. Specifically, applications involve networks of various types, including *collaborative* [7,48], *social* [27,38,40], *sensor* [11], *brain* [32,33], and *random* [23] networks. In all of these works, however, the analysis merely takes into account the evolution of the connected components and cycles of a graph—to the best of our knowledge, our work is the first to combine clique analysis and persistent homology. Note that while persistent homology can be used to retrieve higher-dimensional topological information, it is not yet fully clear how to meaningfully interpret the resulting homology classes. Despite this apparent limitation, persistent homology has proven to be an effective tool to compare and discriminate the general structure of complex networks or multivariate data.

3 Methods

In this section, we define required terms, give a brief overview of topological data analysis, and describe our algorithms.

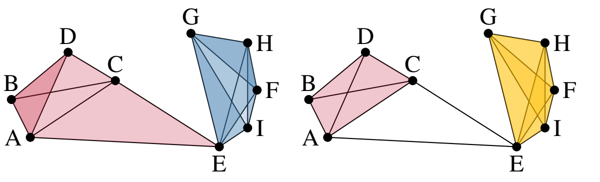
3.1 Complex networks and clique communities

Complex network analysis concerns the study of systems representing connections between distinct elements or actors [35,42,47]. Networks have become a useful tool for representing systems from a wide variety of research fields. Commonly, they are modeled as a graph $\mathcal{G} = (V,E)$, with a set of vertices V and a set of edges $E \subseteq V \times V$, whose elements describe the relationships between vertices. In many applications, vertices and edges contain additional attributes, e.g., labels and weights.

An integral part of network analysis involves the detection—and subsequent analysis—of cohesive substructures of a complex network. As previously outlined, the notions of *k-cliques* and *k-clique communities* have proven very successful for this purpose.

Definition 1 (**k-clique**) *We call a subset of cardinality k of V, whose induced graph is the complete graph on k vertices, a k-clique. For example, three vertices form a* 3*-clique if each one of them is connected to the remaining two.*

A clique thus describes a subset of vertices that are more closely connected than their neighbors. We first define an adjacency relation between cliques.



(a) 3-cliques (b) 4-cliques

Fig. 2. 3-cliques and 4-cliques in an undirected, unweighted graph. Cliques are drawn semi-opaquely in order to show their overlap. Cliques that are in the same community have the same color.

Definition 2 (**k-clique adjacency**) *Two k-cliques σ and σ' are* adjacent *if their intersection is a $(k-1)$-clique, i.e., if they share $k-1$ vertices. For example, in order for two triangles (3-cliques) to be considered connected, they must share a common edge (2-clique).*

The adjacency relation permits a natural extension by considering sequences of connected k-cliques.

Definition 3 (**k-clique connectivity**) *We call two k-cliques σ and σ'* connected *if there exists a sequence of k-cliques of $\mathscr{G}$ such that any two consecutive k-cliques are* adjacent *as defined above.*

Finally, we extend this connectivity relation in order to be able to describe cliques that are "maximally connected". This is akin to connected components in a graph.

Definition 4 (**k-clique community**) *A k-clique community of $\mathscr{G}$ is a maximal union of k-cliques that are pairwise connected.*

Figure 2 illustrates the notions of cliques and clique communities for an undirected, unweighted graph. We can see that vertices $\{A,B,C,D,E\}$, for example, are part of the same clique community: all their 3-cliques (triangles) are connected by a 2-clique, i.e., an edge. The remaining vertices form a 3-clique community on their own because there is no shared edge between the two communities. Note that the vertices $\{E,F,G,H,I\}$ form a 5-clique; we do not visualize it because it overlaps with some of the 4-cliques. In general, given a graph $\mathscr{G}=(V,E)$ and a fixed value k, the k-clique communities do not partition the vertices of $\mathscr{G}$. For example, vertex E in Figure 2 belongs to multiple 3-clique communities and to multiple 4-clique communities. Note that as a consequence of the adjacency definition, $(k+1)$-clique communities are nested in exactly one k-clique community.

3.2 0-dimensional persistent homology

Persistent homology [12] is a fundamental tool in topological data analysis that permits a multi-scale description of shapes. Roughly speaking, given a shape Σ, its kth homology group reveals the presence of k-dimensional holes in the shape. These holes permit an intuitive description in lower dimensions: for $k=0$, they correspond to the connected components of Σ, for $k=1$ to its tunnels, and for $k=2$ to its voids or cavities. This process may be generalized to higher dimensions as well. Persistent homology is an extension of the concept of (simplicial) homology. It describes the changes in homology that occur to an object that evolves with respect to a parameter, such as a scale. Both homology and persistent homology have been defined within a generic framework of simplicial complexes and permit the retrieval of k-dimensional holes for any value of k up to the dimension of the analyzed shape. However, in this work, we focus on the 0-dimensional persistent homology of graphs, for two reasons: First, it is still unclear how to interpret high-dimensional holes, whereas connected components afford an intuitive description. Second, 0-dimensional persistent homology affords a highly-efficient computation and is still sufficiently expressive for our purposes.

We now give a brief mathematical explanation of persistent homology for weighted graphs. This requires a graph $\mathscr{G}=(V,E)$ with n vertices and a weight function $\mathrm{w}\colon V\to\mathbb{R}$ defined on its vertices. $\mathrm{w}(\cdot)$ only assumes finitely many values, which we bring into non-decreasing order, i.e., $\mathrm{w}_1\leq\mathrm{w}_2\leq\cdots\leq\mathrm{w}_n$. A *filtration* of $\mathscr{G}$ is defined as a growing sequence of graphs,

$$\emptyset\subseteq\mathscr{G}_0\subseteq\mathscr{G}_1\subseteq\cdots\subseteq\mathscr{G}_{n-1}\subseteq\mathscr{G}_n=\mathscr{G}, \tag{1}$$

where $\mathscr{G}_i=(V_i,E_i)$ is the graph consisting of the vertices in V and edges in E with weight less than or equal to w_i, i.e.,

$$V_i:=\{v\in V\mid \mathrm{w}(v)\leq\mathrm{w}_i\}, \tag{2}$$

and

$$E_i:=\big\{e=\{u,v\}\in E\mid \mathrm{w}(e):=\max\big(\mathrm{w}(u),\mathrm{w}(v)\big)\leq\mathrm{w}_i\big\}. \tag{3}$$

For $c,d\in\{1,\dots n\}$ such that $c\leq d$, the *(c,d)-persistent zero-dimensional homology group* of $\mathscr{G}$ is defined as the image of the inclusion map of the connected components of $\mathscr{G}_c$ into $\mathscr{G}_d$.

The previous description follows a simple intuition: as we traverse the individual graphs of the filtration, new components can be created or existing connected components can merge. At such a merge, we consider the component with the larger weight (i.e., the one that appeared *later* in the filtration) to be merged into the component with the lower weight. We denote such a merge by a *persistence pair* $(c,d)\in\mathbb{R}^2$, where c denotes the weight at which a connected component was created and d denotes the weight at which it was destroyed. Connected components that are never destroyed thus have a persistence pair of the form (c,∞). The value $\operatorname{pers}(c,d):=|d-c|$ is called the *persistence* of (c,d). It denotes the "lifespan" of the connected component. A high persistence is usually considered to indicate that a connected component (or, equivalently, a topological feature in higher dimensions) is relevant [13].

Persistence pairs are commonly visualized as points in the plane, forming the *persistence diagram $\mathscr{D}$*. Persistence diagrams are a popular way of obtaining a topological summary of data: they have well-known stability properties [8, 9], meaning that the points in a persistence diagram are a continuous function of the weight function on the input data. All points in the persistence diagram are situated above the diagonal of the first quadrant, and the lifespan of a point (c,d) is indicated by its distance to the diagonal with respect to the L_∞-norm. Persistence pairs of the form (c,∞) are usually drawn in the upper part of the first quadrant.

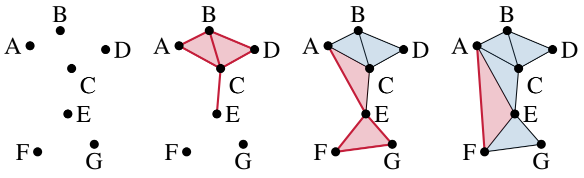
We now briefly discuss how to calculate a 0-dimensional persistence diagram from a graph $\mathscr{G}$ because it is a central part of our method. Using a Union–Find data structure [10, pp. 561–568], we keep track of how connected components change during the filtration: we traverse vertices and edges in ascending order of their weight, letting vertices precede edges if their weight coincides. Whenever we process an edge e with associated weight w_e, it potentially merges two connected components with corresponding weights w_1, w_2. Without loss of generality, we assume that $\mathrm{w}_1\leq\mathrm{w}_2$. We refer to the connected component belonging to w_1 as the "older" connected component. Following the *elder rule* [12, p. 150] in computational topology, we merge the "younger" component into the "older" connected component. We summarize each of these merges by the tuple $(\mathrm{w}_2,\mathrm{w}_e)$, indicating that a connected component was created at weight w_2 and destroyed at weight w_e. Together, these tuples form the 0-dimensional persistence diagram of $\mathscr{G}$. Algorithm 1 gives a pseudo-code description of this procedure. It is highly efficient and has a complexity of $\mathscr{O}(n\alpha^{-1}(n))$, where n denotes the number of edges of $\mathscr{G}$ and $\alpha^{-1}(\cdot)$ is the extremely slow-growing inverse of the Ackermann function.

3.3 Persistent homology for clique communities

In this section, we extend the previously-described algorithm to clique communities. We assume that we are given a graph $\mathscr{G}=(V,E)$ and a weight function $\mathrm{w}\colon V\to\mathbb{R}$ defined on its vertices. Similarly, we also permit $\mathrm{w}(\cdot)$ to be defined on the *edges* of $\mathscr{G}$ only, and set $\mathrm{w}(v):=0$ for every vertex. We then extract all cliques of $\mathscr{G}$, using one of the numerous algorithms available to obtain them; see, e.g., Fortunato [16]

Algorithm 1 0-dimensional persistent homology calculation

```
Require: A weighted graph $\mathscr{G}$
UF ← $\emptyset$                                   ▷ Initialize an empty Union–Find structure
$\mathscr{D} \leftarrow \emptyset$                                    ▷ Initialize an empty persistence diagram
for every edge $(u,v) \in \mathscr{G}$ in ascending order of its weight do
    $c \leftarrow$ UF.Find($u$)
    $c' \leftarrow$ UF.Find($v$)
    if $\mathrm{w}(c) < \mathrm{w}(c')$ then          ▷ $c$ is the older component; merge $c'$ into it
        UF.Union($c'$, $c$)
        $\mathscr{D} \leftarrow \mathscr{D} \cup (\mathrm{w}(c'), \mathrm{w}(u,v))$
    else                                       ▷ $c'$ is the older component; merge $c$ into it
        UF.Union($c$, $c'$)
        $\mathscr{D} \leftarrow \mathscr{D} \cup (\mathrm{w}(c), \mathrm{w}(u,v))$
    end if
end for
return $\mathscr{D}$
```



(a) $\varepsilon = 0.0$ (b) $\varepsilon = 1.0$ (c) $\varepsilon = 2.0$ (d) $\varepsilon = 3.0$

Fig. 3. An illustration of persistent 3-cliques. For $\varepsilon = 1.0$, a new 3-clique community is being created. It is not connected to the 3-clique community that appears for $\varepsilon = 2.0$. For $\varepsilon = 3.0$, the second 3-clique community merges into the first one.

for a survey. A crucial aspect of the detection process is the extension of the weight function $\mathrm{w}(\cdot)$ to an arbitrary clique σ,

$$\mathrm{w}(\sigma) := \max_{\tau \subseteq \sigma} \mathrm{w}(\tau), \tag{4}$$

i.e., the maximum weight of its subsets. Once the clique communities have been detected, we can extract the *k-clique connectivity* graph $\mathscr{G}^k = (V^k, E^k)$. This graph has a vertex for every k-clique of $\mathscr{G}$. Its edges are defined by

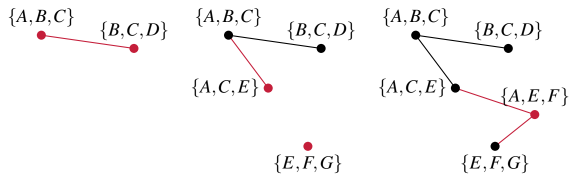
$$E^k := \left\{ \{\sigma, \sigma'\} \in V^k \times V^k \mid \sigma \text{ and } \sigma' \text{ are adjacent} \right\}, \tag{5}$$

i.e., $\mathscr{G}^k$ has an edge between two k-cliques σ, σ' if and only if they intersect in a $(k-1)$-clique. This is equivalent to saying that σ and σ' share $k-1$ vertices, hence σ and σ' are adjacent in the sense of Definition 3. A similar set of graphs is also used by the *clique percolation method* [37], which is the de facto standard in clique analysis. We again extend the weight function $\mathrm{w}(\cdot)$ to edges of $\mathscr{G}_k$ by setting

$$\mathrm{w}(\sigma, \sigma') := \max\left(\mathrm{w}(\sigma), \mathrm{w}(\sigma')\right). \tag{6}$$

So far, we offered another description of clique connectivity analysis in terms of clique connectivity graphs. The crucial difference to other methods is that this setup permits us to calculate the 0-dimensional persistent homology of $\mathscr{G}^k$ using Algorithm 1. We thus obtain a persistence diagram that describes the "evolution" of k-clique communities. This diagram enables the analysis of changes in clique community connectivity with respect to a weight parameter. In particular, we are able to detect all merges between clique communities, whereas previous approaches are only capable of depicting clique communities at a single "snapshot" of the graph. As a consequence, our method is capable of detecting and summarizing clique community behavior at a much more granular level.

Figure 3 demonstrates the advantages of our approach on a simple graph. We insert edges according to their increasing weights (letting ε refer to the current weight threshold in the filtration) and mark those edges in red. For $\varepsilon = 1.0$, two new 3-cliques $\{A,B,C\}$ and $\{B,C,D\}$ are being created. Since they are connected by a 2-clique, i.e., the shared edge $\{B,C\}$, they form a 3-clique community. For $\varepsilon = 2.0$, a new 3-clique $\{E,F,G\}$ appears. It is not connected to the other clique community, so it forms a new community. Moreover, the first community continues to grow. It now contains the 3-clique $\{A,C,E\}$, as well. Last, at $\varepsilon = 3.0$, the addition of the edge $\{A,F\}$ creates a new 3-clique $\{A,E,F\}$, which finally connects the two clique communities. Following the elder rule in persistent homology [12, p. 150], we consider the clique community $\{E,F,G\}$ to be destroyed by the merge, while the other community persists. Figure 4 depicts the evolution of $\mathscr{G}^3$, the 3-clique connectivity graph, for this example. Traditional clique community analysis methods fail to detect the creation and destruction of the clique community because they do not take the "evolution" of the graph into account—they will only detect a single 3-clique community in the graph. The persistence diagram, on the other hand, contains the persistence pairs $(2,3)$ (indicating that a merge between the communities happened) and $(1,\infty)$.



(a) $\varepsilon = 1.0$ (b) $\varepsilon = 2.0$ (c) $\varepsilon = 3.0$

Fig. 4. Evolution of the 3-clique connectivity graph. New vertices and edges are marked red. The persistent homology of this graph encodes their merging behavior.

Note that the detected cliques depend on the weight function defined on the data. In this paper, we assume that weights are defined by the application. Persistence diagrams are known to be stable with respect to perturbations of weights [8, 9]. Users need to ensure that the weights of different networks are "compatible", e.g., by normalizing them.

3.4 Persistence indicator functions

A common task in complex network analysis involves detecting suitable weight thresholds for extraction, comparison, and analysis [6, 15]. Usually, various indices such as the clustering coefficient are then evaluated on each subgraph, leading to a response curve that is used as a fingerprint. In our setting, we can achieve similar results by analyzing the persistence diagrams. We associate a summarizing function, the *persistence indicator function*, to a persistence diagram $\mathscr{D}$. It is defined as

$$\begin{aligned}\mathbb{1}_{\mathscr{D}}\colon \mathbb{R} &\longrightarrow \mathbb{N}\\ \varepsilon &\longmapsto \operatorname{card}\left\{(c,d) \in \mathscr{D} \mid \varepsilon \in (c,d)\right\}\end{aligned} \tag{7}$$

and measures the number of connected components (or topological features) that are "active" for a given value of the threshold parameter ε. In general, $\mathbb{1}_{\mathscr{D}}$ is *not* injective, meaning different persistence diagrams may be assigned the same persistence indicator function. Nonetheless, it remains a useful summarizing function because it permits fast dissimilarity calculations: $\mathbb{1}_{\mathscr{D}}$ is a step function, hence its integral is a piecewise linear function. As a consequence, we can use an L_p distance to quantify the dissimilarity between persistence indicator functions, i.e.,

$$\operatorname{dist}(\mathbb{1}_{\mathscr{D}_1}, \mathbb{1}_{\mathscr{D}_2}) := \left(\int_{\mathbb{R}} |\mathbb{1}_{\mathscr{D}_1}(x) - \mathbb{1}_{\mathscr{D}_2}(x)|^p \mathrm{d}x \right)^{\frac{1}{p}}, \tag{8}$$

where usually $p = 2$. The L_p distance can be calculated much more easily than the bottleneck distance [8] or the Wasserstein distance [9]. Another advantage of $\mathbb{1}_{\mathscr{D}}$ is that it permits the calculation of a *mean persistence indicator functions* for ensembles of weighted networks. Figure 5 shows the persistence indicator function for a simple example.

Since we have a persistence indicator function for every value of k, but we are only interested in the amount of activity—measured in the form of active topological features, i.e, clique communities, at a given threshold—we also require a condensed glyph: to this end,

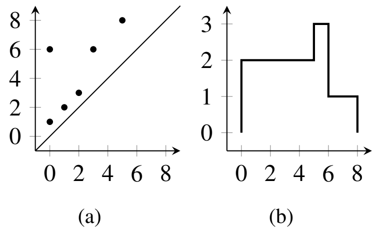


Fig. 5. A persistence diagram (a) and its corresponding persistence indicator function (b).

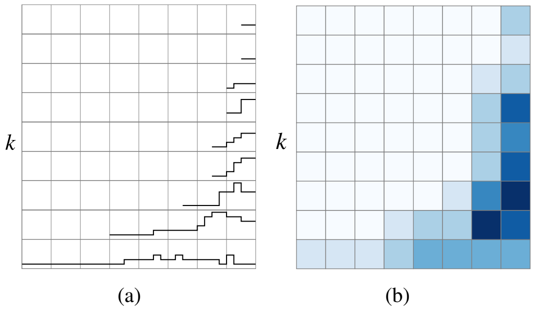


Fig. 6. Persistence indicator functions (a) and their histogram (b) for the "Les Misérables" co-occurrence network.

BC	CC	EC	CCC
Valjean	Valjean	Gavroche	Valjean
Myriel	Marius	Valjean	Gavroche
Gavroche	Javert	Enjolras	Fantine
Marius	Thénardier	Marius	Marius
Fantine	Gavroche	Bossuet	Enjolras

Table 1. The five most central characters for the "Les Misérables" co-occurrence network, ranked by different centrality measures (betweenness centrality, closeness centrality, eigenvector centrality, and our clique community centrality).

Data set	Cliques	Time
Les Misérables	2922	0.09 s
Shakespeare (Antony & Cleopatra)	53399	1.66 s
Brain	126526	4.38 s
Collaboration network (1999)	1052701	52.86 s
Collaboration network (2003)	2965703	185.32 s
Collaboration network (2005)	6530308	537.16 s

Table 2. Processing times (including centrality calculations) for several of the analyzed networks.

we first discretize the domain of the functions into uniformly-spaced bins. For each k and each bin, we now calculate $\max \mathbb{1}_{\mathscr{D}}$, i.e., the *maximum* amount of active topological features in the bin. This results in a "band" in which the color indicates the maximum value of $\mathbb{1}_{\mathscr{D}}$. By stacking these histograms, we obtain a visual summary of the clique community activity of a weighted network. Figure 6 demonstrates this for a set of persistence indicator functions of the "Les Misérables" co-occurrence network. The glyph is also included in our interactive tool (see Figure 1 or the accompanying video) in order to provide an overview of interesting thresholds. The glyph shows that most of the activity concentrates on very high thresholds, i.e., $\varepsilon \in [24, 32]$, in the network. Only the 2-cliques, depicted in the lowest band of the glyph, exhibit a non-zero number of clique communities for lower thresholds.

3.5 Clique community centrality

Another important issue in the analysis of complex networks is the assessment of the (relative) importance of a given node. For this purpose, multiple centrality measures are known in the literature. In the context of our work, the *cross-clique connectivity* [14] is highly relevant. The cross-clique connectivity of a vertex v is the number of cliques the vertex is a part of. We can extend this definition to clique communities and define the *clique community centrality* of a vertex v to be

$$\Gamma_{\mathrm{c}}(v) := \sum_{v \in C} \operatorname{pers}(C), \tag{9}$$

where C refers to all clique communities the vertex is a part of. By measuring the persistence of every clique community, we are automatically taking into account the relevance of a particular vertex: vertices that participate in many clique communities of low persistence will be assigned a lower centrality value than vertices that participate in few clique communities of high persistence.

To demonstrate the utility of our measure, we briefly compare it with existing centrality measures on the "Les Misérables" co-occurrence network, which we shall analyze in more detail in Section 4.1. Using different centrality measures, we extracted the five most central nodes of the network. Table 1 shows the results. Since the utility of a centrality measure is application-dependent [31], there is no clear "best" measure. All measures are capable of detecting the main character, *Valjean*, for example. The short ranking serves to elucidate some properties of our measure, though: (i) Nodes with a high degree will not automatically be considered to be more important: *Myriel* (a character with a high degree whose connections do not form cliques) hence does not appear in the list. (ii) Since centrality is calculated over all k-clique communities for all k, our measure is capable of assigning a larger importance to "key players" of a community, i.e., nodes that occur in multiple clique communities for different values of k. Thus, *Enjolras*, the leader of a group of revolutionary students, is assigned a slightly larger clique community centrality value than other members (e.g., Bossuet) of the revolutionary group (whose other members are not shown in the table). In subsequent sections, we will use the centrality values in order to analyze changes in network structure.

3.6 Implementation & technical details

Our implementation uses C++ and is made publicly available (among other algorithms) within `Aleph`[1], an open-source library for persistent homology. We first calculate all cliques up to a maximum threshold for k using an incremental algorithm [49]. The result is a *simplicial complex*, i.e., a generalization of a graph whose elements, called simplices, correspond to the retrieved cliques. We extract the k-clique connectivity graph by traversing all simplices of the complex and store their co-faces in a map, from which we finally extract all edges of the graph. This extraction step is currently performed independently for every k and not yet heavily optimized. Table 2 shows the performance of our algorithm on a desktop machine (Intel Core i7-6700K, 64 GiB RAM). The calculation time includes centrality measure calculations, making them faster than traditional centrality measures such as *betweenness centrality*, whose calculation alone takes several hours for the collaboration networks.

4 Case Studies

In the following, we exemplify the use of clique community persistence by analyzing several networks. Our method is highly generic and may be applied to any weighted network, provided the notion of cliques is useful for the particular application scenario.

4.1 "Les Misérables" co-occurrence network

This network describes co-occurrences between characters in Victor Hugo's novel "Les Misérables". The edge weights correspond to the number of co-occurrences between two characters. Consequently, we have to invert the weights because we consider edge weights to correspond to *proximity*. The network is very small, comprising 77 nodes

[1] https://github.com/Submanifold/Aleph

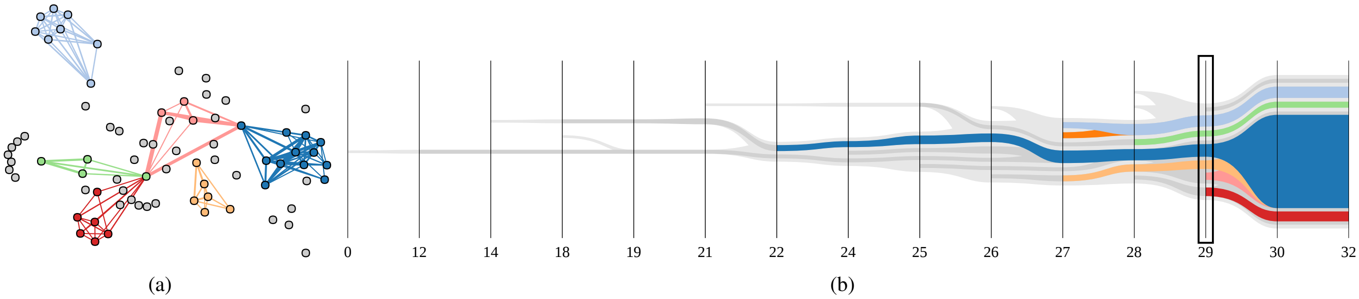


Fig. 7. A force-directed graph layout (a) of the “Les Misérables” co-occurence network for edge weight 29 in which 4-clique communities are represented through different colors. The nested graph visualization (b) showing the evolution of the 4-clique communities of the network according to all edges weight thresholds.

and 254 edges, but we can use it to illustrate the properties of our method. It contains numerous cliques up to $k = 10$. Despite its size, the data set is well-known in the network analysis community, representing a small but significant benchmark for demonstrating how our approach addresses relevant challenges in this field. In the introduction, we already discussed the relevance of clique communities: they decompose a network into cohesive communities while still permitting overlaps between individual groups. The definition of a clique community mainly depends on two parameters: (i) the degree of a community (ii) the weight of the edges of the network. To the best of our knowledge, there is no other approach for analyzing the clique community structure of a network that also takes into account these parameters. In the following, we show how considering both of them can significantly enhance the analysis of a network.

In order to depict the evolution of clique communities retrieved by our method, we use the *nested graphs* paradigm by Lukasczyk et al. [34]. Their visualization is originally used to track merges between superlevel set components in scientific data sets, which satisfy a nesting relationship. The same holds for k-cliques and their communities. A $(k+1)$-clique community may only contain exactly one k-clique community. We may thus represent the different clique degrees as *levels* in the graph, while the x-axis represents different edge weight thresholds. The edges of the nested graph illustrate the evolution of k-clique communities. In the following sections (as well as in the accompanying video), we detail how clique community detection and nested graphs support users in their analysis.

Figure 1(a) shows the nested graph for the complete data set. Different colors correspond to different clique degrees, with higher degrees nested in lower degrees. For small edge weight values, the network consists of a single, small, 2-clique community. By increasing the weight threshold, new 2-clique communities start to appear and merge. At the same time, colors turn into brighter shades, thereby revealing that the connectivity between nodes become stronger. As expected, communities merge over time, while it is not possible that communities split for increasing edge weight thresholds. For the highest edge weight value, the presence of a single gray structure in the background reflects the fact that the network is edge-connected. In contrast to a standard connectivity analysis, the use of clique communities reveals more information. By only considering connected components (gray edges in the graph’s background), we cannot reveal the network structure; focusing on brighter colors, on the other hand, reveals the presence of various communities. For this specific network, this reflects the fact that even if all the characters are part of the same story, there exist several distinct subplots.

In combination with several interaction mechanisms of the proposed visualization tool, our method permits exploring different edge weights and degrees. This enables users to discover information that is not available by just focusing on a single one of these parameters. For instance, analyzing the network for the largest edge weight results in a clear community structure for $k = 4$. Leaving k fixed and moving “horizontally” in the nested graph, it is possible to track the evolution of a selected community. Thanks to this approach, moving from threshold 32 to 29, we find that the big dark blue community at 32 consists of three different sub-communities (see Figure 7(a) for the corresponding sub-graphs and Figure 7(b) for the nested graph). Moreover, these six communities turn out to be highly relevant for the structure of this network, as each of them corresponds to a significant group of characters: the members of the revolutionary association called *Les Amis de l’ABC* (dark blue), the circle of friends of the young *Fantine* (light blue), the members of the *Patron-Minette* crime gang (orange), the social circle of Bishop *Myriel* (green), the participants at *Champmathieu*’s trial (red), and the family of *Marius* (light red). Intuitively, varying the edge weight threshold helps extract the “core” of a community, while changing the degree permits analyzing the same social circle according to different granularity levels by revealing the sub-communities it consists of.

The histogram and persistence diagram support the analysis of inexperienced users while providing further guidelines and confirmations to experts. Specifically, the histogram of the persistence indicator functions (Figure 1(b)) permits detecting relevant threshold values for the analysis. For instance, the communities discussed above were obtained for edge weight 29 and degree 4, as the corresponding cell in the histogram is highly relevant. This is quite effective, especially when dealing with more complex networks. Furthermore, the overall information provided by the histogram makes it useful for an initial comparison between different networks. Similarly, the persistence diagram (Figure 1(c)) helps identify interesting communities as they correspond to points far from the diagonal. Here, some of those points correspond, e.g., to the community of the main characters of the novel, or of the core members of the *Les Amis de l’ABC*.

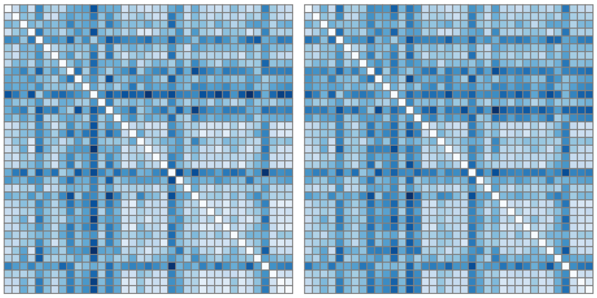
We can also observe limitations of our current visualization: currently, we cannot handle the disappearance of edges (which is also not modeled in the data, though). Consequently, some of the clique communities persist and are never merged even though they bear no more importance to the plot.

4.2 Shakespearean network analysis

To demonstrate that our method aids in comparative analysis of networks, we used publicly available co-occurrence networks of 37 Shakespearean plays. The networks contain the characters of a play as the nodes, while edges signify that two characters appear in the same scene of a play. The edge weights are set according to the amount of speech uttered by two characters in the same scene [40]. We transform the edge weights as discussed above and extract all clique communities up to $k = 16$ (higher-order cliques do not occur). Following Shakespeare’s *First Folio*, we classify all plays as either *Comedy* (e.g., “The Tempest”), *Tragedy* (e.g., “Hamlet”), or *History* (e.g., “Henry V”).

4.2.1 Comparing distance measures

We first use the networks to demonstrate the structural stability of the persistence indicator functions or, more precisely, their discretized variants. To this end, we calculate the Wasserstein distance between all



(a) Wasserstein distance (b) L_2 distance

Fig. 8. Distances matrices for network dissimilarity measures. The Wasserstein distance (left) is a well-established topological dissimilarity measure. Our histogram distance (right) produces virtually identical results at a fraction of the calculation complexity.

persistence diagrams of all plays, i.e.,

$$W_p(a,b) = \left(\sum_k \inf_{\eta_k \colon \mathscr{D}_{a,k} \to \mathscr{D}_{b,k}} \sum_{x \in \mathscr{D}_{a,k}} \|x - \eta_k(x)\|_\infty^p \right)^{\frac{1}{p}}, \tag{10}$$

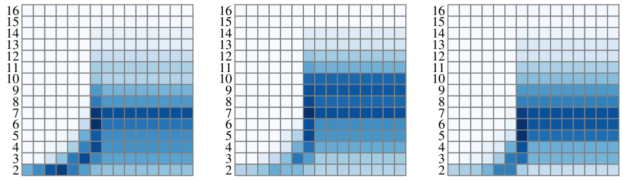
where a and b refer to the individual persistence diagrams of each network, and k ranges over the clique community degrees. We also compute the L_2 distance between the discretized persistence indicator functions (with 15 uniformly-spaced bins). This is akin to calculating a distance between histograms. Figure 8 depicts the resulting distance matrices. The patterns shown in both matrices are virtually identical. A numerical analysis shows that the matrices are highly-correlated with Pearson's correlation coefficient $R^2 \approx 0.96$. This means distances measured by the Wasserstein distance and distances measured by the L_2 distance are related by a linear transformation. Other numerical experiments (please refer to the supplementary materials for more details) confirm the relationship between the two measures. We may thus be confident that the L_2 distance suffices for capturing structural dissimilarities between networks.

4.2.2 Structural differences between groups of plays

For our first analysis, we focus on structural differences between groups of plays. We want to check whether the community structures typically found in comedies differs from, say, tragedies. In order to simplify the subsequent comparison, we assume that the weights are scaled from $[0,1]$. We now calculate all clique community persistence diagrams and convert them to their persistence indicator function. Following this, we calculate the mean persistence indicator function for every value of k. Last, we convert these functions into 2D histograms with $n = 15$ bins. This yields a mean 2D histogram that displays the amount of clique community activity for every threshold and every k. Figure 9 depicts the results. At first glance, the 2D histograms appear to be very similar: all histograms display an elongated structure indicating that more communities are merged at higher thresholds. A closer inspection shows that the mean 2D histogram for comedies is different. It has a larger amount of clique community activity for small values of k than either tragedies or histories. Furthermore, there is less activity for large values of k, indicating that the number of connected characters tends to be smaller on average. The activity for smaller values of k is caused by a higher number of subplots, or even "plays with a play", which are often a feature of Shakespeare's comedies.

4.2.3 Structural differences between all plays

As a second step, we demonstrate how our method permits a comparison of structural differences between *all* networks. In previous work [40], the authors used an embedding based on the Wasserstein distance between the persistence diagrams corresponding to a play. Only zero-dimensional and one-dimensional persistent homology was taken into account, though, whereas our method includes higher-dimensional connectivity information in the form of cliques.



(a) Comedies (b) Tragedies (c) Histories

Fig. 9. Mean histograms for Shakespeare's plays, grouped according to their categorization.

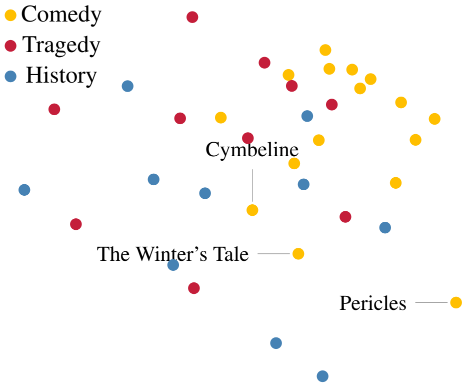


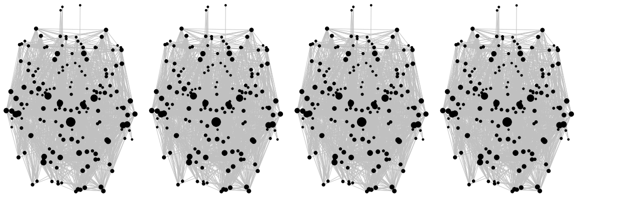
Fig. 10. An embedding of Shakespeare's plays according to the histogram distance. We do not show all labels due to layout reasons.

Figure 10 depicts an embedding based on the L_2 distance, which we calculated using multidimensional scaling. Colors show the category of a play. We can see that most comedies (yellow) are structurally similar, so they form a cluster. Histories and tragedies, on the other hand, exhibit no cluster structure because they are structurally too different. Similar observations were made in previous work [40]. The interesting comedies are those that are *remote* from the cluster center because their structure is somewhat atypical. Here, we have marked three plays that are typically considered to be problematic by scholars: in PERICLES, for example, Shakespeare was only a co-author of the play, which may be the reason why its line-up of characters is so different from other comedies. The other two highlighted comedies are also special: both CYMBELINE and THE WINTER'S TALE feature a larger number of clique communities with high persistence values.

In summary, we showed how our method can be used to obtain embeddings of the structural similarity between different networks. Our histogram-based distance measure is fast and easy to calculate, while maintaining important information.

4.3 Brain networks

The connectivity of the human brain—usually referred to as the *human connectome*—is a fundamental object of study in neurobiology research [44]. Of particular interest to researchers is the identification of changes in brain connectivity when certain areas are removed: how is the transfer of signals impaired by this change? This question is highly relevant for improving our understanding of diseases such as depression [28] or multiple sclerosis [43]. Usually, differences in networks are measured using graph-theoretic measures [4] or persistent homology based on connected components [32, 33]. Previous work [41] already showed that brain activity exhibits a community structure, whose analysis sheds light on neurological concepts such as brain function. Our method is the first approach that permits a multiscale analysis and comparison based on these community structures. It is therefore an extension or generalization of approaches based on connected components.



(a) All fibers (b) Variant 1 (c) Variant 3 (d) Variant 5

Fig. 11. The high inter-connectivity of the brain network makes it hard to see differences between the original network with all fibers and several variants in which numerous fibers have been removed.

Variant	Density	Diam. (weighted)	Avg. degree (weighted)
0	0.125	4 (60.0)	21.21 (2300.3)
1	0.124	4 (60.0)	21.06 (2296.0)
2	0.124	4 (60.0)	21.13 (2295.2)
3	0.124	4 (60.0)	21.16 (2282.0)
4	0.124	4 (60.0)	21.15 (2279.3)
5	0.125	4 (60.0)	21.19 (2264.0)
6	0.125	4 (60.0)	21.19 (2264.0)
7	0.124	4 (60.0)	21.16 (2279.6)
8	0.125	4 (60.0)	21.20 (2257.5)

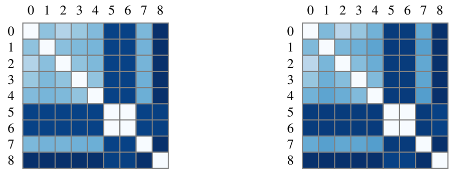
Table 3. Common graph measures are incapable of detecting salient differences between the individual brain networks.

In the following, we want to briefly analyze multiple variants of a brain connectivity network. The network consists of *fibers* (spatial curves) that connect areas in the brain. Fibers with the same target are collapsed to an edge whose weight is set to the number of fibers it contains. Every area is represented as a node with a set of associated coordinates, making the network easy to depict. Next to an unmodified network (variant 0), there are also variants in which different edges have been removed at random in order to simulate changes in connectivity. Figure 11 demonstrates that these changes are so slight that they do not show up in direct graph visualizations.

The clique community structure changes, however. To quantify this, we calculate all clique communities up to $k = 13$. From the resulting clique community persistence diagrams, we get the persistence indicator functions so that we are able to once again compare the common Wasserstein distance as well as our persistence indicator function distance. Figure 12 depicts dissimilarity matrices for the two different measures. Every entry in the matrix corresponds to a brain network. We can see that all modified variants $(1 \ldots 9)$ are unable to maintain the community structure to some extent, as indicated by the blue colors in the matrix. Networks 5, 6, and 8 are particularly dissimilar from the original data, which is not apparent in the direct graph visualization. Neurologists could now analyze these networks and assess to what extent brain function was impeded due to the removed edges. Confirming the previous case study, we find that the distance matrices are highly-correlated with $R^2 \approx 0.99$, hence both distance measures are nearly identical. Moreover, common graph measures such as density, diameter, and average weighted degree remain almost unchanged for all network variants (Table 3). Only our community-based method is capable of quantifying differences between the networks—over all weight thresholds and all clique degrees.

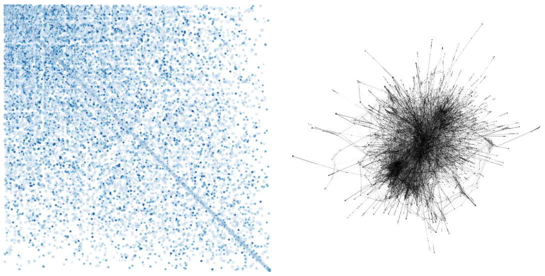
4.4 Condensed matter collaborations

Our method can also be used to detect changes in network structure. As a demonstration, we used three different data sets that describe co-authorship between scientists of the "Condensed Matter" category of the "arXiv" e-print repository. The edge weights in these networks are a function of the number of collaborations between authors. We invert them in order to treat them as distances. Furthermore, we normalize them to $[0,1]$ in order to simplify comparisons. The data was compiled at three snapshots in time: 1999, 2003, and 2005; the network from 1999 is included in the 2003 network, for example. As pointed out by Figure 13, the network sizes (1999: 16,726 nodes, 47,594 edges; 2003: 31,1163 nodes, 120,029 edges; 2005: 40,421 nodes, 175,692 edges) pose challenges both for visualizing and analyzing the data. Our current implementation is able to calculate all cliques for all values of k only for the 1999 network—for the 2003 and 2005 networks, the size of the clique connectivity graph quickly exceeds the main memory of a desktop machine with 64 GiB RAM. We are able to obtain clique communities up to $k = 6$ so we can analyze at least a part of the structure. Plotting the persistence diagrams (Figure 14) demonstrates that their structure is virtually identical for higher-order clique communities. For $k = 2$, however, merges tend to happen at higher thresholds for the 1999 data than for the other networks. The total number of these persistence pairs is negligible, though, so there is almost no variation in creation values, destruction values, and persistence values between the networks, which impedes structural comparisons.



(a) Wasserstein distance (b) L$_2$ distance

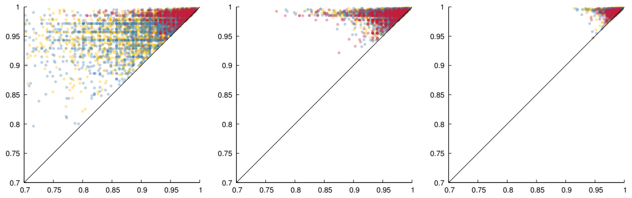
Fig. 12. Dissimilarity matrices for all variants of the brain network. Both measures show that all cuts in the network destroy connectivity to some extent. At least three cuts result in a markedly different network structure.



(a) Node–link diagram (b) Force-directed layout

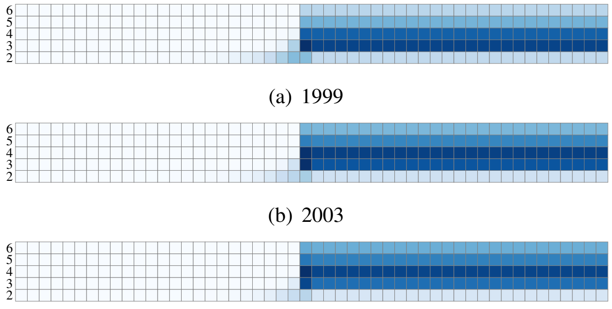
Fig. 13. The number of vertices and edges in the "condensed matter" networks makes visualizing and comparing them very challenging.

Calculating standard distance measures such as the bottleneck distance or the Wasserstein distance between the persistence diagrams is unfeasible, as each persistence diagrams contains tens of thousands of points and the distance computations do not scale well. Furthermore, the missing variation in persistence pairs—except for a negligible number of points—will result in meaningless distances. The situation is similar when we rely on the persistence indicator functions, as shown in Figure 15. In order to obtain information about structural differences, we thus use the clique community centrality values. Figure 16, left,



(a) $k = 2$ (b) $k = 3$ (c) $k = 4$

Fig. 14. Clique community persistence diagrams for the "Condensed matter" co-authorship networks for 1999, 2003, and 2005. Higher-order clique communities exhibit the same characteristics in all networks.



(a) 1999

(b) 2003

(c) 2005

Fig. 15. Histograms for the "Condensed matter" co-authorship network. The number of active clique communities hardly differs between the three networks, making it very hard to distinguish them.

shows histograms and kernel density estimates of the clique community centrality values. Here, the 1999 data is markedly different from the two other time steps: it has fewer centrality values (which is to be expected as its size is smaller) and they are distributed differently. Centrality values between 5 and 10 are more equally distributed than in the other years. The mean centrality increases from 6.55 in 1999 to 7.44 in 2003, and falls back to 7.25 in 2005. This is caused by an influx of nodes with lower centrality values. These correspond to researchers that are not (yet) well-connected. The mean centrality value of new researchers, i.e., nodes do not occur in the data set for a previous year, is 4.01 for the 2003 data, while it is 5.345 for the 2005 data. The connectivity of newcomers to the network thus increases. We also find that the centrality of known nodes, i.e., nodes that are available for all three time steps, *stop* increasing—the mean increase from 1999 to 2003 is 1.41, while the mean increase from 2003 to 2005 is only 0.16. The community structure appears to be saturated after a certain point.

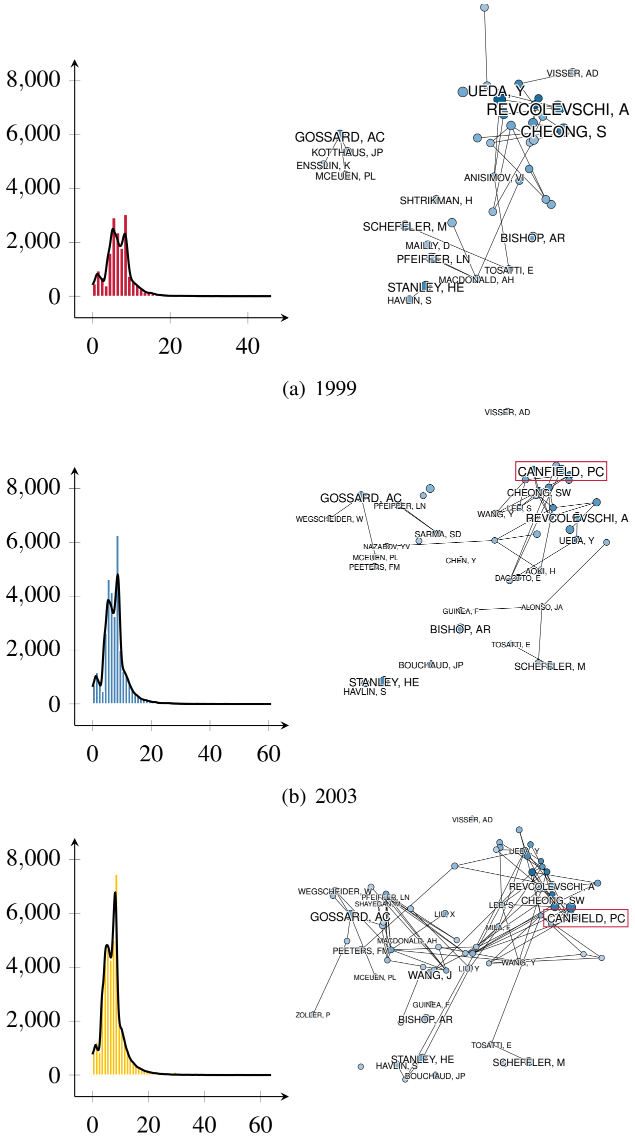
The centrality values can also be used to filter away all but the most central of all nodes. The results of this are shown in Figure 16, right. Nodes are colored according to their degree, while their size corresponds to their clique community centrality. The graph is thus reduced to the "key players" of the network and we can see how they change over time. For instance, we observe that *Paul C. Canfield*, whose research group was founded in 1992, improved his collaboration network from 1999 to 2003, thus starting to appear as a key player in the 2003 data. It is also interesting to see that even the key players become more interconnected over time. The filtered graph for the 2005 data is almost a single connected component, while earlier years do not contain *direct* connections between the key players.

In summary, this example demonstrates the utility of clique community centrality: its distribution can be analyzed to draw assumptions about the connectivity of a network. Moreover, it can act as simple filter—and thereby permit the analysis of graphs for which traditional visualization techniques are not readily applicable.

5 Conclusion

We developed an extension of persistent homology to the analysis of clique communities in weighted networks. In contrast to earlier methods, our algorithm is capable of analyzing the connectivity relations for *all* clique degrees and *all* weight thresholds simultaneously. We also presented different visualizations for showing information about clique communities and demonstrated their utility on various data sets.

For future work, we envision using our method to compress a graph, e.g., by removing edges that are irrelevant for the clique community structure. Furthermore, we want to augment our method so that it can handle time-varying networks in which connections between individual nodes are permitted to *disappear* at certain thresholds. This requires changes to the underlying model for clique community persistence, though. We also want to consider the nesting relationships between k-clique communities and $(k+1)$-clique communities. So far, our method performs the analysis for a single k only, but it is possible that a given $(k+1)$-clique community completely "absorbs" a k-clique community. An extension to this case would support the understanding of how communities merge.



(a) 1999

(b) 2003

(c) 2005

Fig. 16. Histograms (left, with density estimates) of the clique community centrality values for different time steps of the condensed matter collaboration networks. Non-central nodes have been removed in order to make the force-directed graph layout (right) less cluttered.

Acknowledgments

We thank Jan Hering (Center for Machine Perception, Czech Technical University) for helpful discussions and for providing us with the annotated brain networks. We also thank the reviewers for their detailed and considerate comments that helped us improve the paper.

References

[1] B. Adamcsek, G. Palla, I. J. Farkas, I. Derényi, and T. Vicsek. `CFinder`: locating cliques and overlapping modules in biological networks. *Bioinformatics*, 22(8):1021–1023, 2006.

[2] D. Archambault, T. Munzner, and D. Auber. TopoLayout: Multilevel graph layout by topological features. *IEEE TVCG*, 13(2):305–317, 2007.

[3] C. Bron and J. Kerbosch. Algorithm 457: Finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575–577, 1973.

[4] E. Bullmore and O. Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, 10(3):186198, 2009.

[5] F. Cai, Z. Kang, F. Zhicun, H. Lansheng, and C. Jing. K-clique community detection based on Union–Find. In *International Conference on Computer, Information and Telecommunication Systems (CITS)*, pp. 1–5, 2014.

[6] D. S. Callaway, M. E. J. Newman, S. H. Strogatz, and D. J. Watts. Network robustness and fragility: Percolation on random graphs. *Physical Review Letters*, 85(25):5468–5471, 2000.

[7] C. Carstens and K. Horadam. Persistent homology of collaboration networks. *Mathematical Problems in Engineering*, 2013:815035:1–815035:7, 2013.

[8] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1):103–120, 2007.

[9] D. Cohen-Steiner, H. Edelsbrunner, J. Harer, and Y. Mileyko. Lipschitz functions have L_p-stable persistence. *Foundations of Computational Mathematics*, 10(2):127–139, 2010.

[10] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT press, 2009.

[11] V. De Silva and R. Ghrist. Homological sensor networks. *Notices of the American Mathematical Society*, 54(1):10–17, 2007.

[12] H. Edelsbrunner and J. Harer. *Computational topology: An introduction*. American Mathematical Society, 2010.

[13] H. Edelsbrunner, D. Letscher, and A. J. Zomorodian. Topological persistence and simplification. *Discrete Comput. Geom.*, 28(4):511–533, 2002.

[14] M. R. Faghani and U. T. Nguyen. A study of XSS worm propagation and detection mechanisms in online social networks. *IEEE Transactions on Information Forensics and Security*, 8(11):1815–1826, 2013.

[15] I. Farkas, D. Ábel, G. Palla, and T. Vicsek. Weighted network modules. *New Journal of Physics*, 9(6):180, 2007.

[16] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3–5):75–174, 2010.

[17] M. Freire, C. Plaisant, B. Shneiderman, and J. Golbeck. ManyNets: An interface for multiple network analysis and visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 213–222. ACM, New York, NY, 2010.

[18] E. R. Gansner, Y. Koren, and S. C. North. Topological fisheye views for visualizing large graphs. *IEEE TVCG*, 11(4):457–468, 2005.

[19] M. C. González, H. J. Herrmann, J. Kertész, and T. Vicsek. Community structure and ethnic preferences in school friendship networks. *Physica A: Statistical mechanics and its applications*, 379(1):307–316, 2007.

[20] E. Gregori, L. Lenzini, and S. Mainardi. Parallel *k*-clique community detection on large-scale networks. *IEEE TVCG*, 24(8):1651–1660, 2013.

[21] F. Hao, G. Min, Z. Pei, D.-S. Park, and L. T. Yang. *k*-clique community detection in social networks based on formal concept analysis. *IEEE Systems Journal*, 11(1):250–259, 2015.

[22] T. Heimo, J. Saramäki, J.-P. Onnela, and K. Kaski. Spectral and network methods in the analysis of correlation matrices of stock returns. *Physica A: Statistical Mechanics and its Applications*, 383(1):147–151, 2007.

[23] D. Horak, S. Maletić, and M. Rajković. Persistent homology of complex networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(03):P03034:1–P03034:24, 2009.

[24] Y. Jia, J. Hoberock, M. Garland, and J. Hart. On the visualization of social and other scale-free networks. *IEEE TVCG*, 14(6):1285–1292, 2008.

[25] P. F. Jonsson and P. A. Bates. Global topological features of cancer proteins in the human interactome. *Bioinformatics*, 22(18):2291–2297, 2006.

[26] P. F. Jonsson, T. Cavanna, D. Zicha, and P. A. Bates. Cluster analysis of networks generated through homology: automatic identification of important protein communities involved in cancer metastasis. *BMC Bioinformatics*, 7(1):2, 2006.

[27] K. F. Kee, L. Sparks, D. C. Struppa, and M. Mannucci. Social groups, social media, and higher dimensional social structures: a simplicial model of social aggregation for computational communication research. *Communication Quarterly*, 61(1):35–58, 2013.

[28] M. S. Korgaonkar, A. Fornito, L. M. Williams, and S. M. Grieve. Abnormal structural networks characterize major depressive disorder: A connectome analysis. *Biological Psychiatry*, 76(7):567574, 2014.

[29] J. M. Kumpula, M. Kivelä, K. Kaski, and J. Saramäki. Sequential algorithm for fast clique percolation. *Physical Review E*, 78(2):026109, 2008.

[30] J. M. Kumpula, J.-P. Onnela, J. Saramäki, K. Kaski, and J. Kertész. Emergence of communities in weighted networks. *Physical Review Letters*, 99(22):228701, 2007.

[31] A. Landherr, B. Friedl, and J. Heidemann. A critical review of centrality measures in social networks. *Business & Information Systems Engineering*, 2(6):371–385, 2010.

[32] H. Lee, M. K. Chung, H. Kang, B.-N. Kim, and D. S. Lee. Discriminative persistent homology of brain networks. In *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 841–844, 2011.

[33] H. Lee, H. Kang, M. K. Chung, B. N. Kim, and D. S. Lee. Persistent brain network homology from the perspective of dendrogram. *IEEE TMI*, 31(12):22672277, 2012.

[34] J. Lukasczyk, G. Weber, R. Maciejewski, C. Garth, and H. Leitte. Nested tracking graphs. *Computer Graphics Forum*, 36(3):12–22, 2017.

[35] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003.

[36] G. Palla, A.-L. Barabási, and T. Vicsek. Quantifying social group evolution. *Nature*, 446(7136):664–667, 2007.

[37] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.

[38] F. I. R. Fellegara, U. Fugacci and L. De Floriani. Analysis of geolocalized social networks based on simplicial complexes. In *9th ACM SIGSPATIAL International Workshop on Location-Based Social Networks (LSBN)*. ACM, 2016.

[39] F. Reid, A. McDaid, and N. Hurley. Percolation computation in complex networks. In *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 274–281, 2012.

[40] B. Rieck and H. Leitte. *Shall I compare thee to a network?*—Visualizing the topological structure of Shakespeare's plays. In *Workshop on Visualization for the Digital Humanities at IEEE VIS*. Baltimore, MD, 2016.

[41] A. J. Schwarz, A. Gozzi, and A. Bifone. Community structure and modularity in networks of correlated brain activity. *Magnetic Resonance Imaging*, 26(7):914–920, 2008.

[42] J. Scott. *Social Network Analysis*. SAGE Publications Ltd., 3rd ed., 2012.

[43] N. Shu, Y. Liu, K. Li, Y. Duan, J. Wang, C. Yu, H. Dong, J. Ye, and Y. He. Diffusion tensor tractography reveals disrupted topological efficiency in white matter structural networks in multiple sclerosis. *Cerebral Cortex*, 21(11):2565–2577, 2011.

[44] O. Sporns, G. Tononi, and R. Kötter. The human connectome: A structural description of the human brain. *PLOS Computational Biology*, 1(4), 2005.

[45] R. Toivonen, J.-P. Onnela, J. Saramäki, J. Hyvönen, and K. Kaski. A model for social networks. *Physica A: Statistical Mechanics and its Applications*, 371(2):851–860, 2006.

[46] T. von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J. van Wijk, J.-D. Fekete, and D. Fellner. Visual analysis of large graphs: State-of-the-art and future research challenges. *Computer Graphics Forum*, 30(6):1719–1749, 2011.

[47] S. Wasserman and K. Faust. *Social network analysis: Methods and Applications*. Cambridge University Press, 1994.

[48] A. C. Wilkerson, T. J. Moore, A. Swami, and H. Krim. Simplifying the homology of networks via strong collapses. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5258–5262, 2013.

[49] A. Zomorodian. Fast construction of the Vietoris–Rips complex. *Computers & Graphics*, 34(3):263–271, 2010.