

# Sub-optimal Recall in Visual Cluster Retrieval: When Clusters Look Like Bridges

Mathieu Jacomy<sup>1,\*</sup>, Matilde Ficozzi<sup>1</sup> and Anders K. Munk<sup>2</sup>

<sup>1</sup>Aalborg Universitet, Denmark

<sup>2</sup>Danmarks Tekniske Universitet, Denmark

## Abstract

Force-directed node placement algorithms, a popular technique to visualise networks, are known to optimize “cluster separability”: when sets of densely connected nodes get represented as well-separated groups of dots. Using these techniques leads us to conceive networks as sets of clusters connected by bridges. This is also how we tend to think of the “community structure” model embedded in clustering techniques like modularity maximization. Yet this mental model has flaws. We specifically address the notion that clusters (“communities”) necessarily look like groups of dots, through the mediation of a node placement algorithm. Although often true, we provide a reproducible counterexample: topological clusters that look like bridges. First, we present an empirical case that we encountered in a real world situation, while mapping the academic landscape of AI and algorithms. Second, we show how to generate a network of arbitrary size where a cluster looks like a bridge. In conclusion, we open a discussion about layout algorithms as a visual mediation of a network’s community structure. We contend that when it comes to the accuracy of retrieving clusters visually, node placement algorithms have an imperfect recall despite an excellent precision.

## Keywords

human-centered computing, graph drawing, network visualization, community detection, visual cluster retrieval, visual network analysis

## 1. Introduction

The purpose of drawing a large graph (a.k.a. making a network map) is generally to mediate its topological features. One aims to produce visual patterns providing insights about its community structure [22]. The expected visual pattern mainly consists of visually separated aggregates of dots, that one interprets as clusters or communities, i.e. densely connected sets of nodes. In practice, one rarely sees a cluster if it does not exist in the topology (although we do not assess that statement here). And one tends to assume that conversely, if a cluster exists in the graph topology, it is necessarily visible in the layout. In this paper, we argue that this assumption can be wrong in common situations. We find that a common topological cluster pattern could often get mediated differently from the compact aggregate of dots one generally expects, for instance as a bridge (Figure 1). In short, some clusters can be dense enough to be consistently picked up by common clustering techniques, yet not dense enough to be displayed

---

CHR 2024: Computational Humanities Research Conference, December 4–6, 2024, Aarhus, Denmark

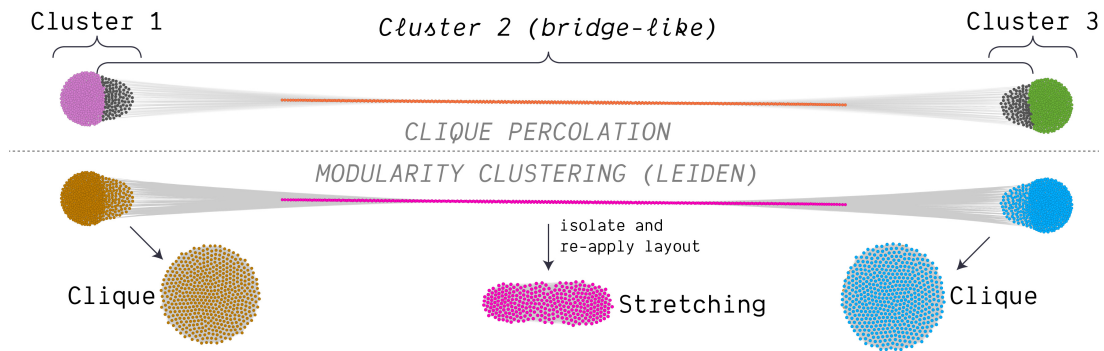
\*Corresponding author.

✉ mathieu.jacomy@gmail.com (M. Jacomy); matildefic@ikl.aau.dk (M. Ficozzi); ankm@dtu.dk (A. K. Munk)

🆔 0000-0002-6417-6895 (M. Jacomy); 0009-0001-8660-5734 (M. Ficozzi); 0000-0002-5542-3065 (A. K. Munk)

© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).





**Figure 1:** A network with a bridge-like cluster generated by our method. Node colors represent clusters detected by clique percolation (top) and modularity maximization by the Leiden method (bottom). The clusters on the side are denser (cliques) than the bridge-like cluster (a “stretching”, see section 3).

as visual aggregates by common layout techniques.

In section 2 we report our observations from a real-world situation where we find such clusters in a co-occurrence network about AI and algorithms in science. In section 3 we draw on those observation to build a procedure to generate clusters that look like bridges. We conclude by framing this issue as an accuracy problem for the task of retrieving clusters visually, more specifically as a *recall* issue.

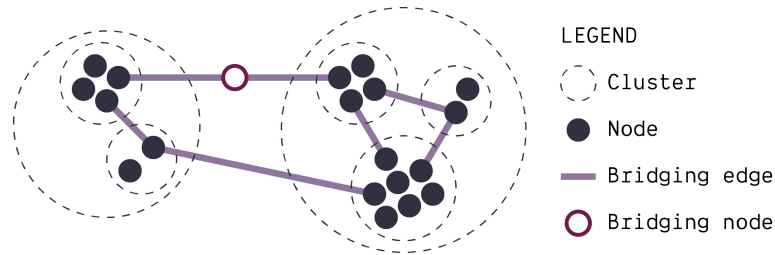
### 1.1. Related work

If community detection techniques like modularity maximization [3, 12, 21] have become a staple of visual network analysis, it is notably because the retrieved communities closely match the visual clusters produced by force-driven algorithms. This result was voiced by Noack as “modularity clustering is a force-directed layout” [14].

Noack formalized graph drawing as an optimization problem about the “separation of communities” [14]. In this mental model, graphs are imagined as communities connected by bridges. On a practical level, using force-directed layouts and community detection algorithms to analyze networks reinforces the notion that a network’s community structure solely consists of clusters connected by bridges (Figure 2). Indeed, the visualizations produced in popular tools like Cytoscape [19] Gephi [1] or NodeXL [20] often feature separated groups of nodes (seen as clusters) connected by edges or chains of edges (bridges). In this context, it is most reasonable to assume that a node is either part of a cluster, of a bridge, or neither.

In this implicit mental model, the macro-structure is nested and graph-like: clusters play the role of nodes, and bridges that of edges. Yet as it turns out, relevant patterns in graphs are not limited to the cluster/bridge dichotomy. We generally consider bridges as made of nodes [10, 18], sometimes made of edges [7], but rarely as a mix of both; yet such structures can effectively be an important part of the community structure. Intuitively, very dense clusters may be connected by less dense clusters.

The notion of community structure has multiple definitions [16], meanings [6], computational commitments [5, 17] and visual representations [2, 22]. In the next section, we report on



**Figure 2:** Implicit mental model of modular graphs

a real-world case where the community structure has richer patterns than the cluster/bridge dichotomy. It shows that the nuance and complexity of a network’s structure can be lost in the translation enacted by the layout, notably when some “communities” get represented as visually non-compact shapes.

## 2. Case: a co-occurrence network of keywords from articles about AI and algorithms

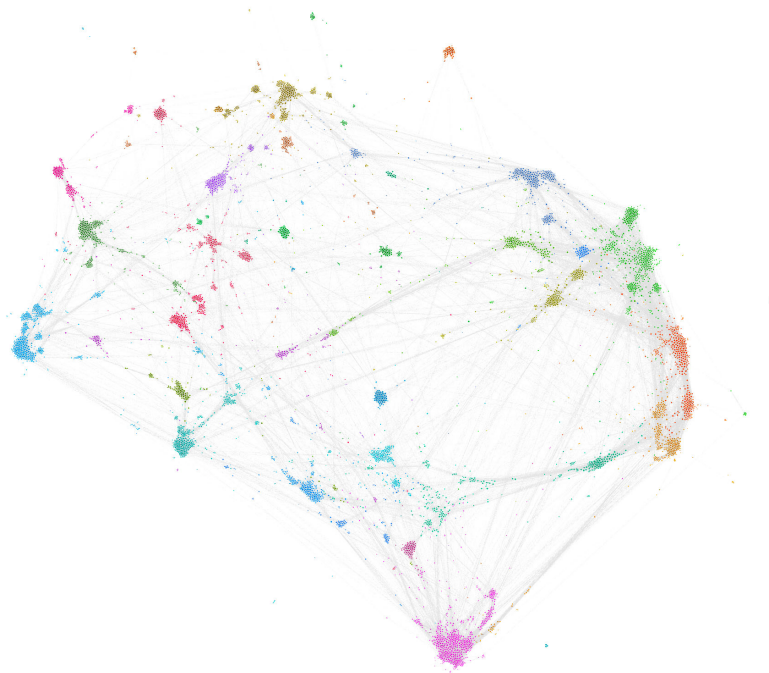
**Description of the case:** We generated this network by extracting expressions co-occurring in the abstract or title of academic articles mentioning AI, algorithms, or machine learning. Our motivation was to map what algorithms were doing in science. We present our full methodology and our statistical analysis of the corpus in another paper [11].

This network’s 7,562 nodes represent the most frequent expressions in our corpus, such as “encrypted”, “image denoising”, or “CNNs”. Each of the 85,215 edges represents a co-occurrence of connected expressions with a sufficiently high *pointwise mutual information* (PMI) score [4]. Through clique percolation [15] with  $k = 7$  we uncovered 166 clusters corresponding to various topics or semantic fields. We rendered the graph as a network map using the layout algorithm LinLog [13] using the Force Atlas 2 implementation [9]. In order to annotate the network map and use it as a discussion elicitation device with our project partners, we underwent the following task: for each cluster, sample the most representative documents, read them, and produce a short summary of the purpose and agency of AI and algorithms in those publications [11].

The network obtained has a blatant community structure, and at first glance, seems to consist of clusters connected by bridges (Figure 3).

**Issue at stake:** Our annotation task demanded us to explain *visible* clusters, as rendered by the layout algorithm, while we had computed *topological* clusters through clique percolation. To our surprise, we faced many mismatches: some computed clusters did not look like groups of dots in the network map, or conversely, some clearly visible clusters were not captured by the clique percolation process.

One may think that when there is a mismatch between computed and visualized clusters, the computation should always take precedence over the visual, because it is more true to the data



**Figure 3:** Screenshot of the network in Gephi [1].

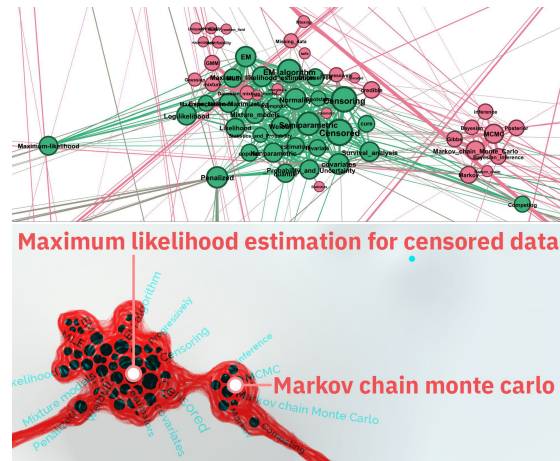
(then the mismatch would not be an issue). But that perspective fails to account for the purpose of annotation, which is to provide a context for the visible patterns. One cannot annotate a pattern that is not visible, even if it exists in the data; and conversely, one *should* annotate the artifacts of the method, i.e. anti-patterns that are visible but do not exist in the data (precisely to mark them as artifacts).

As soon as a visualization is used, even for exploratory data analysis, accounting for what it makes visible becomes a methodological necessity.

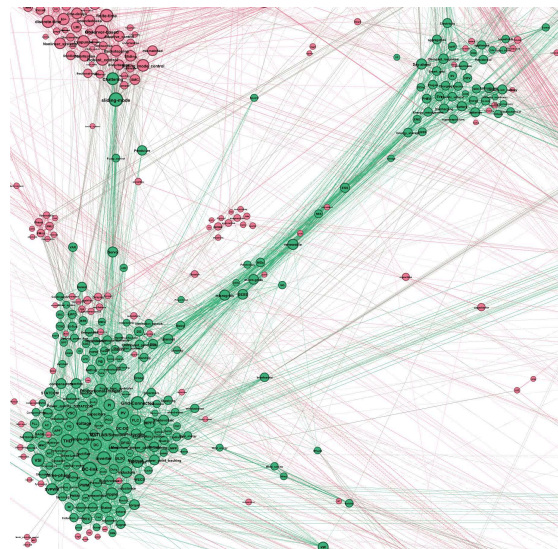
**Sub-issue 1: granularity level.** To retrieve clusters we used clique percolation as we did not need to annotate the entire network (only dense areas). We picked a clique size of  $k = 7$  because it was low enough to capture the smallest relevant clusters. In many cases, this approach worked: the cluster retrieved looked like what one expects (Figure 4).

However, this approach had a significant drawback:  $k$  was too low to break the bigger and denser clusters down to subclusters (Figure 5). In the most extreme case, one of the retrieved clusters contained the 31% of the nodes (the entire semantic field related to health and medicine).

Large computed clusters were an issue because they contained multiple smaller visual clusters. However, this issue was easy to address by subdividing them until we reached the desired granularity. We chose the modularity maximization by the Leiden method [21] for the convenience of its resolution setting.



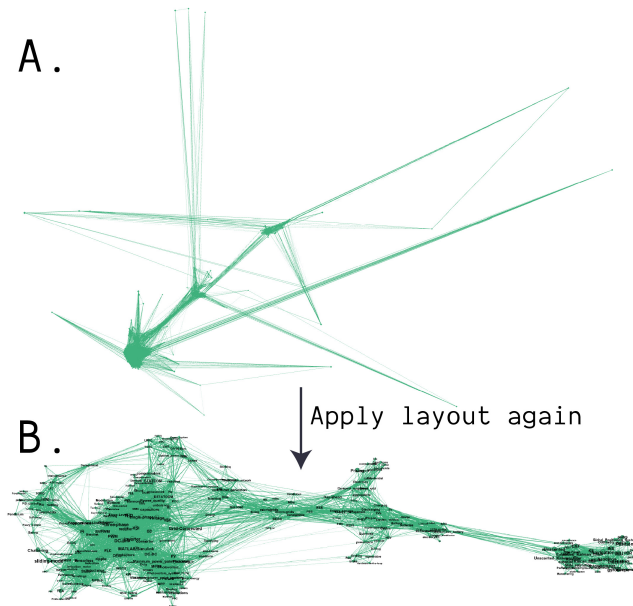
**Figure 4:** A well-behaved cluster. This cluster from clique percolation looks like a well-separated group of dots. Top: the cluster as shown in Gephi (in green). Bottom: how we annotated it in the final map.



**Figure 5:** A recalcitrant cluster (“renewable energy storage”, partial screenshot).

**Sub-issue 2: clusters not looking like clusters.** Some clusters were not only too big, but also contained non-compact structures, such as bridges. We could not solve that issue, and had to live with the trouble. For instance, the recalcitrant cluster of Figure 5 seemed to contain a mix of clusters and bridges, which appears more clearly if we isolate it from the rest (Figure 6-A). It is not compact, but extremely spiky. Yet from the method, its topology satisfies the same criterion as the well-behaved cluster of Figure 4: each node has at least 6 fully interconnected neighbors in the cluster.

The spikiness is partly due to how this cluster is interconnected with the rest of the map,



**Figure 6:** The recalcitrant cluster from Figure 5 (“renewable energy storage”) isolated and in full view. A: The cluster with the map’s original layout but without the other nodes. B: After the layout is applied again.

and partly due to its internal structure. We can observe it by applying the same layout to the isolated network (Figure 6-B): although the layout is less spiky and more compact, it remains very elongated. Even when we repeat this process recursively, many clusters keep resisting and refuse to become compact unless we tear them down to confetti (Figure 7).

We had to accept that some clusters are dense enough to be captured by community detection (both clique percolation and modularity clustering) yet not dense enough to look like compact shapes on the map. Many of those clusters looked like bridges, stretching between compact clusters. We ultimately decided to annotate the compact clusters and the bridge clusters the same way, the only difference being that compact clusters got attached to a landmark point while the title of bridge clusters followed the contour of the connection (Figure 8).

**Takeaways from the case.** By splitting initial clusters to the desired granularity, we settled on 235 sub-clusters to annotate. Of those, 58 were of the bridge kind (25%): at least in this case, bridge-looking clusters are not a marginal phenomenon.

We also observed that those bridges were straightforward to explain. The main bridge in our recalcitrant cluster was about *renewable energy storage*, and it connected the topics of *renewable energy* and *power distribution* with those of *smart grids* and *lithium-ion batteries*, which makes a lot of sense. Other examples included *scanning for cancer* bridging *scanners* with *cancer research*; or *language* bridging *text analysis* with *speech recognition*; etc.



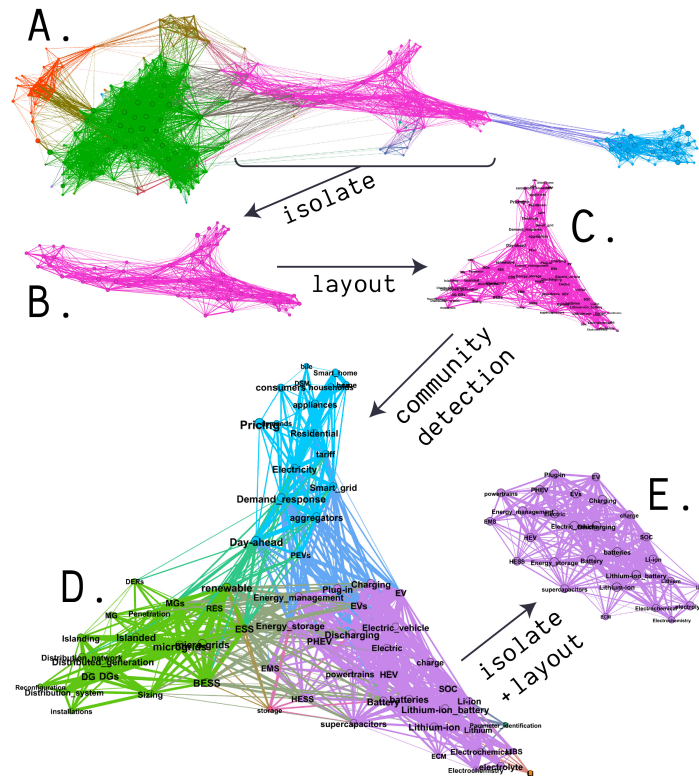


Figure 7: Recalcitrant cluster from Figure 5 broken down recursively: it never gets visually compact.

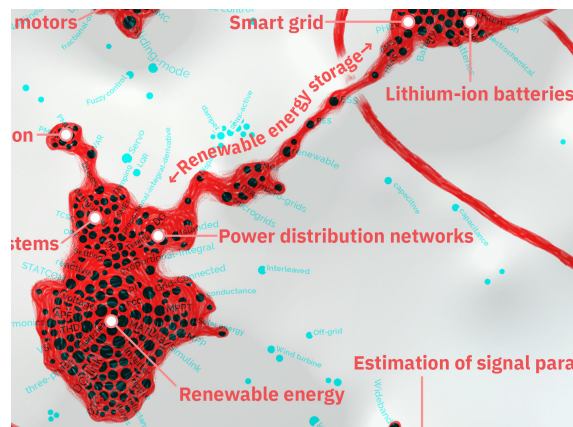


Figure 8: Recalcitrant cluster in the final map (under the label “renewable energy storage”).

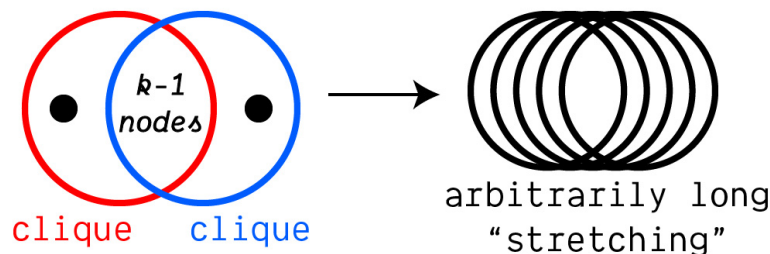
### 3. Generating clusters that look like bridges

Drawing on these observations, we devised a method to generate clusters that look like bridges. An example is shown in Figure 1.

**Criteria satisfied:** The method works for any clique size  $k$  set in advance. It generates an arbitrarily large graph such that clique percolation using said  $k$  will find exactly 3 clusters, one of which will look like a bridge (it get represented as a visually non-compact set of dots spread out between the two other clusters).

Our tests show that modularity clustering by the Louvain and Leiden methods will also find 3 clusters, although it depends on the resolution used (we cannot guarantee the result as those algorithms are not deterministic). However, the result for clique percolation is granted by design of the method.

**Method:** Generate two cliques of a size significantly greater than  $k$ . Generate a “stretching” [8] by stacking as many cliques of size  $k$  as desired, as shown in Figure 9. Merge  $k - 2$  nodes from the first clique with as many of the first stacked clique of the stretching; similarly merge  $k - 2$  nodes of the second clique with the *last* stacked clique of the stretching. More details in the reference implementation (Python).



**Figure 9:** Stretching: stacked cliques.

In the example of Figure 1, we used  $k = 100$ ; the cliques had 500 nodes and the stretching stacked 400 cliques.

**Why it works:** Each clique will be retrieved by clique percolation because its size is greater than  $k$ . The stretching will be retrieved because it satisfies clique percolation by construction. However, the merge of  $k - 2$  creates a bottleneck that prevents clique percolation from joining the stretching with either clique, because it is smaller than the minimum overlap of  $k - 1$  nodes required.

The stretching will look like a bridge for the same two reasons we mentioned in our observations. First, it is less compact than a clique by design; in fact, it is just compact enough to satisfy the criterion of clique percolation at the decided level  $k$ . Second, it gets pulled in two opposite directions by the cliques, whose mutual repulsion is dominating the balance of forces in the layout.



## 4. Bridge-like clusters as an accuracy problem for visual cluster retrieval

The situation can be reformulated as an accuracy problem for the task of retrieving clusters visually, which is the purpose of force-directed layout algorithms as formalized by their authors (for instance [13, 9]; see also [22]). The clusters we see but cannot retrieve by computational means are the false positives, responsible for the task's *precision*, which we assume as generally good (or at least, it is not the focus of this paper). Conversely, the clusters that do not look like a compact group of dots and remain undetected are the *false negatives* responsible for the *recall* (Figure 10).

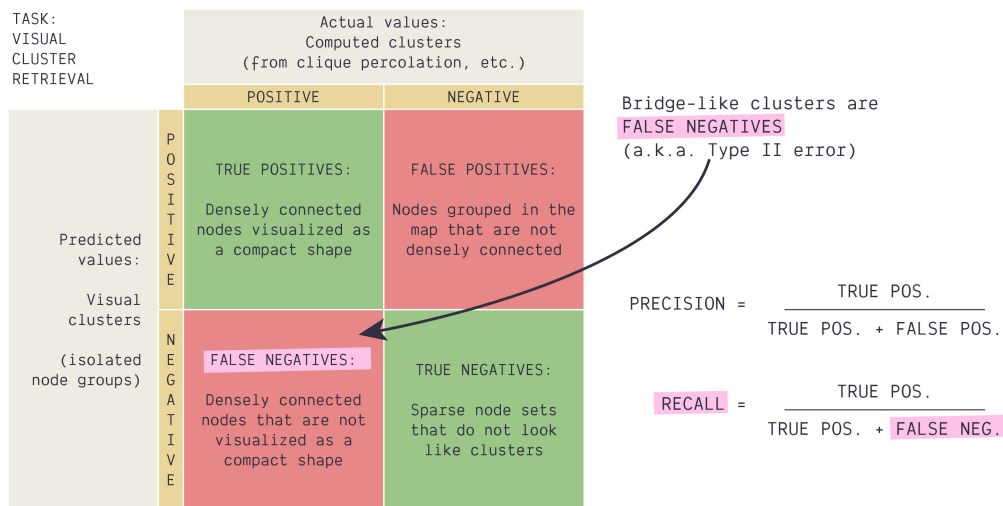


Figure 10: Accuracy for visual cluster retrieval: bridge-like clusters create a recall problem.

## 5. Conclusion: visual cluster retrieval has a sub-optimal recall

We presented an empirical case where clusters retrieved from modularity clustering did not always look like the compact groups of dots we usually expect. Building on our observations, we built a method to generate clusters that look like bridges: satisfying the criterion of clique percolation for an arbitrary clique size, yet looking like bridges stretched between compact clusters when visualized by a force-directed layout. We do not provide evidence about the pervasiveness of these visually non-compact clusters, but our generation method shows that the conditions for their emergence are commonplace in the context of large graphs with a community structure.

The existence of bridge-like clusters creates a recall issue for the task of retrieving clusters visually from a force-directed layout, even when the precision is good: unbeknownst to the visualization's reader, some clusters may remain unseen. Being aware of that possibility is an important insight for the researchers and experts using network maps to explore relational data.

## References

- [1] M. Bastian, S. Heymann, and M. Jacomy. “Gephi: an open source software for exploring and manipulating networks”. In: *Proceedings of the international AAAI conference on web and social media*. Vol. 3. 2009, pp. 361–362.
- [2] C. Bennett, J. Ryall, L. Spalteholz, and A. Gooch. “The aesthetics of graph visualization.” In: *CAe*. 2007, pp. 57–64.
- [3] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. “Fast unfolding of communities in large networks”. In: *Journal of statistical mechanics: theory and experiment* 2008.10 (2008), P10008.
- [4] G. Bouma. “Normalized (pointwise) mutual information in collocation extraction”. In: *Proceedings of GSCL 30* (2009), pp. 31–40.
- [5] S. Fortunato. “Community detection in graphs”. In: *Physics reports* 486.3-5 (2010), pp. 75–174.
- [6] L. C. Freeman. “The sociological concept of” group”: An empirical test of two models”. In: *American journal of sociology* 98.1 (1992), pp. 152–166.
- [7] L. HAO, X. Li, X. Liu, and W. Wang. “BBTA: Detecting communities incrementally from dynamic networks based on tracking of backbones and bridges”. In: *Applied Intelligence* 53 (2022). DOI: 10.1007/s10489-022-03418-2.
- [8] M. Jacomy. “Situating Visual Network Analysis”. PhD thesis. Aalborg University, 2021. DOI: 10.54337/aau435977255.
- [9] M. Jacomy, T. Venturini, S. Heymann, and M. Bastian. “ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software”. In: *PloS one* 9.6 (2014), e98679.
- [10] N. Meghanathan. “Neighborhood-based bridge node centrality tuple for complex network analysis”. In: *Applied Network Science* 6 (2021). URL: <https://api.semanticscholar.org/CorpusID:235691166>.
- [11] A. K. Munk, M. Jacomy, M. Ficozzi, and T. E. Jensen. “Beyond artificial intelligence controversies: What are algorithms doing in the scientific literature?” In: *Big Data & Society* 11.3 (2024), p. 20539517241255107. DOI: 10.1177/20539517241255107.
- [12] M. E. Newman and M. Girvan. “Finding and evaluating community structure in networks”. In: *Physical review E* 69.2 (2004), p. 026113.
- [13] A. Noack. “Energy models for graph clustering.” In: *J. Graph Algorithms Appl.* 11.2 (2007), pp. 453–480.
- [14] A. Noack. “Modularity clustering is force-directed layout”. In: *Physical review E* 79.2 (2009), p. 026102.
- [15] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. “Uncovering the overlapping community structure of complex networks in nature and society”. In: *nature* 435.7043 (2005), pp. 814–818.

- [16] L. Peel, D. B. Larremore, and A. Clauset. “The ground truth about metadata and community detection in networks”. In: *Science Advances* 3.5 (2017), e1602548. DOI: 10.1126/sciadv.1602548.
- [17] T. P. Peixoto. *Descriptive vs. inferential community detection in networks: Pitfalls, myths and half-truths*. Cambridge University Press, 2023.
- [18] A. N. Rabchevskiy, V. S. Zayakin, and E. A. Rabchevskiy. “A Method for Identifying Bridges in Online Social Networks”. In: *Recent Trends in Analysis of Images, Social Networks and Texts - 10th International Conference, AIST 2021, Tbilisi, Georgia, December 16-18, 2021, Revised Supplementary Proceedings*. Ed. by E. Burnaev, D. I. Ignatov, S. I. 0002, M. Y. Khachay, O. Koltsova, A. Kutuzov, S. O. Kuznetsov, N. V. Loukachevitch, A. Napoli, A. Panchenko, P. M. Pardalos, J. Saramäki, A. V. Savchenko, E. Tsymbalov, and E. Tutubalina. Vol. 1573. Communications in Computer and Information Science. Springer, 2021, pp. 166–175. DOI: 10.1007/978-3-031-15168-2\_14.
- [19] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker. “Cytoscape: a software environment for integrated models of biomolecular interaction networks”. In: *Genome research* 13.11 (2003), pp. 2498–2504.
- [20] M. Smith, N. Milic-Frayling, B. Shneiderman, E. Mendes Rodrigues, J. Leskovec, and C. Dunne. *NodeXL: a free and open network overview, discovery and exploration add-in for Excel 2007/2010*. 2010.
- [21] V. A. Traag, L. Waltman, and N. J. Van Eck. “From Louvain to Leiden: guaranteeing well-connected communities”. In: *Scientific reports* 9.1 (2019), p. 5233.
- [22] T. Venturini, M. Jacomy, and P. Jensen. “What do we see when we look at networks: Visual network analysis, relational ambiguity, and force-directed layouts”. In: *Big Data & Society* 8.1 (2021), p. 20539517211018488.