



A Quality Metric for Visualization of Clusters in Graphs

Amyra Meidiana¹(✉), Seok-Hee Hong¹, Peter Eades¹, and Daniel Keim²

¹ University of Sydney, Sydney, Australia
amei2916@uni.sydney.edu.au, {seokhee.hong,peter.eades}@sydney.edu.au
² University of Konstanz, Konstanz, Germany
keim@uni-konstanz.de

Abstract. Traditionally, graph quality metrics focus on readability, but recent studies show the need for metrics which are more specific to the discovery of patterns in graphs. Cluster analysis is a popular task within graph analysis, yet there is no metric yet explicitly quantifying how well a drawing of a graph represents its cluster structure.

We define a clustering quality metric measuring how well a node-link drawing of a graph represents the clusters contained in the graph. Experiments with deforming graph drawings verify that our metric effectively captures variations in the visual cluster quality of graph drawings. We then use our metric to examine how well different graph drawing algorithms visualize cluster structures in various graphs; the results confirm that some algorithms which have been specifically designed to show cluster structures perform better than other algorithms.

1 Introduction

Clustering is an important task in graph analysis. Visualization can be a useful tool in this task, where a good drawing of a network should be able to highlight important group structures within the network and allow a user to accurately answer group-level analytical tasks. To this end, a number of graph layout algorithms specifically focused on faithfully depicting clusters within a graph have been introduced.

The quality of a drawing of a graph is often measured using *aesthetic criteria* which rate the readability of the visualization, such as the number of edge crossings or symmetry. However, these measures become less significant when working with large graphs (e.g. [19]). More recent work considers quality metrics more extensible to large graphs, such as *shape-based metrics* which compare the original topology of a graph to one derived from the positioning of vertices in its drawing [9]. Newly introduced is also the concept of more specific quality metrics concerned with the discovery of specific patterns with visualizations [5]. Although general quality metrics are still necessary, these more specific metrics are useful when developing visualizations geared for a more specific

This work is supported by ARC DP grant.

purpose - for example, clustered graph visualizations which can be used to support various classes of group-level tasks [34].

Despite a longstanding recognition of cluster discovery as one important goal in graph visualization and the definition of quality metrics that regard the depiction or discovery of specific structures, there is yet to be defined a metric that explicitly quantifies how well a visualization represents the underlying clustering structure of the graph. We therefore introduce a *clustering quality metric* which scores a drawing of a graph based on how well the clustering structure of the graph is displayed within it. We present the following contributions:

1. We define the *clustering quality metric*, a new metric to measure the visual cluster quality of node-link graph drawings. In our framework, we compare the ground truth clustering provided for the vertices a graph to the geometric clustering derived from the graph’s drawing, and the similarity of both clusterings denotes the quality of the visualization of clusters within the drawing.
2. We validate the metric through deformation experiments of graph drawings. Results of the experiment confirm that as the graphs are distorted resulting in the clusters to become visually less distinct from each other in the drawings, the scores computed using our metric decrease.
3. We compare various graph drawing algorithms using our metric to discover which methods perform better in visualizing cluster structures. We compare drawing algorithms of different types, including layouts that have been designed specifically to emphasize clusters. Our experiments confirm that these layouts perform better than others not explicitly geared towards cluster visualization, especially for real world graphs.

2 Related Work

2.1 Graph Drawing Quality Metrics

Aesthetics have been described as one criterion to be achieved by graph drawing algorithms [3]. The concept of aesthetics is concerned with the readability of graphs and include standards such as the minimization of edge crossing and bends, and minimization of drawing area used. A number of studies have verified the correlation of such aesthetic metrics with the ability of users to execute tasks on the graph (e.g. [17,30,31]). However, these studies tend to focus on smaller graphs, and newer studies (e.g. [19]) have discovered that the effects of these aesthetic criteria are not as apparent in larger graphs.

Shape-based metrics [9] attempt to address this limitation by computing a shape graph based on the drawing of a graph, where two vertices are connected with an edge if they are “close” to each other, and comparing it to the topology of the original graph - a good drawing is expected to have a shape graph similar to its actual topology. For recent work on visualization quality metrics, Behrisch et al. [5] provides a survey covering various visualization techniques, including but not limited to node-link drawings, and notes that measuring the effectiveness of node-link drawings in supporting analytical tasks is an open research question.

2.2 Clustering Comparison Metrics

Clustering refers to the division of a set of items into *clusters*, where items in the same cluster are more similar to each other than to items in a different cluster [1]. Despite the seemingly simple definition, the notions of “similarity” and what constitutes a “cluster” differ between contexts, leading to the birth of various clustering algorithms and thus multiple ways to cluster the same set [11]. To compare two clusterings C and C' of the same set, a number of metrics exist:

- *Rand Index (RI)* measures the similarity of C and C' based on the number of pairs of elements classified into the same group in both C and C' and the number of pairs of elements classified into different groups in both C and C' [32]. *Adjusted Rand Index (ARI)* [18] is a version corrected for chance.
- *Mutual Information (MI)*, when applied to two random variables, measures how much information of one can be gathered from the other, and is also applicable to comparisons between two clusterings C and C' [7]. *Normalized Mutual Information (NMI)* [36] is a normalized version, while *Adjusted Mutual Information (AMI)* [38] is a version adjusted for chance.
- *Fowlkes-Mallows Index (FMI)* compares a clustering C' to a target clustering C using the number of true positives, false positives, and false negatives [12].
- *Homogeneity (HOM)* and *completeness (CMP)* have been described as desirable outcomes of a cluster assignment C' compared to a target clustering C , where homogeneity measures to what extent each cluster in C' only contains members of the same cluster in C , and completeness refers to the extent that all members of a cluster in C are assigned to the same cluster in C' [33].

2.3 Graph Drawing Algorithms

In this section, we briefly describe a number of types of algorithms used to compute graph layouts:

- *Force-directed* layouts model a graph as a system where repulsive forces exist between all pairs of vertices and neighboring vertices attract each other [13].
- *Multi-level* layouts improve the time efficiency of force-directed layouts through steps of *coarsening* the graph into a smaller graph such as through clustering, applying the layout on the smaller graph, and using it as an initial layout to draw the less coarse graph until a layout for the original is computed [15].
- *Multi-dimensional scaling (MDS)* methods are based on dimension reduction techniques that aim to display high-dimensional data in fewer dimensions while preserving the distances between the data points [37].
- *Stress-based* layouts utilize the stress function found in the MDS literature. These methods compute a layout by minimizing an adapted stress function that considers the geometric and theoretical distances between vertices [14].
- *Spectral* methods compute the layout of a graph using the eigenvectors of matrices related to the graph, such as adjacency or Laplacian matrices [20].

3 Clustering Metric for Graph Visualization

We propose a new task-specific metric for graph visualization, the *clustering quality metric*, for measuring how well a drawing of a graph represents its underlying clustering structure. We compute the similarity between a ground truth clustering of a graph’s vertices to a geometric clustering derived from its drawing and compute the clustering quality using the similarity of the two clusterings. Figure 1 summarizes the framework used for our proposed metric.

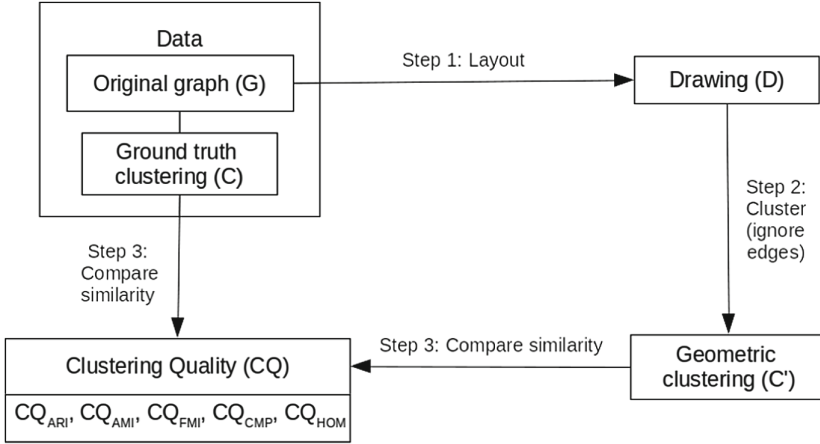


Fig. 1. The framework for the clustering quality metric. The framework takes as input a graph G with a predefined ground truth clustering C . A drawing D is produced by applying a layout algorithm to G , from which a geometric clustering C' of the vertices is computed. Computing the similarity of C and C' produces the clustering quality CQ score, which can be done using a variety of clustering comparison metrics.

Let $G = (V, E)$ be a graph and $C = \{C_i, i = 1 \dots k\}$ be the ground truth clustering of V , the vertex set of G . Although in some applications a vertex may belong to multiple clusters, in this study, we focus on non-overlapping clusters as a starting point in developing the metric.

Step 1: We apply a layout algorithm to G to obtain a graph drawing D , which provides geometric positions for each node in G . A node-link drawing of a graph with no additional visual variables implicitly denotes groupings of vertices through the proximity of vertices to each other and a user is more likely to perceive two vertices drawn close together as belonging to the same group rather than two vertices drawn further apart.

Step 2: We compute a geometric clustering $C' = \{C'_i, i = 1 \dots k\}$ purely based on the geometric positions of vertices in D . Any geometric clustering algorithm can be used, but in this work, we use *k-means clustering*, which partitions a set into k subsets that minimize the within-class variance [25]. We use *k-means clustering* as it is a widely used method applicable to geometric clustering with

existing fast and efficient heuristic approximations and because for our experiments, we know the number of ground truth clusters.

Step 3: Using C' , we compute the clustering quality of D by computing the similarity of C with C' to produce a clustering quality score CQ . Any clustering comparison metrics can be used with our framework, however we use the following metrics discussed in Sect. 2.2: Adjusted Rand Index (CQ_{ARI}), Adjusted Mutual Information (CQ_{AMI}), Fowlkes-Mallows Index (CQ_{FMI}), Homogeneity (CQ_{HOM}), and Completeness (CQ_{CMP}). These metrics have been established for measuring a clustering’s quality when a target ground truth is available. In the cases of CQ_{ARI} and CQ_{AMI} , they were taken over other variants of RI and MI as they are adjusted for chance. All these metrics produce a score of 1 for perfect clustering, while independent clusterings attain values close to 0.

4 Validation Experiments

4.1 Experiment Design

To validate our metric, we designed deformation experiments for graph drawings. We start with a drawing of a graph that displays its clusters such that the number of visible clusters and their respective sizes accurately represent the ground truth clusters and the clusters are well-separated from each other with no overlap.

We then progressively deform the drawing. In each experiment, we performed 10 steps of deformation, where in each step, the coordinates of each vertex from the previous step are perturbed by a small value in the range $[0, \delta]$, with δ being in the range of 0.05-0.1 multiplied by the drawing area. We compute the clustering quality score and compare the scores across all steps of the deformation.

Based on the clustering comparison metrics, we expect our approach to produce scores in the range of $[0, 1]$ where a higher value denotes a closer similarity between the geometrical clustering C' derived from the drawing D and the ground truth clustering C . Therefore, we formulate the following hypothesis in order to validate our metric:

Hypothesis 1: The clustering quality metric scores will decrease as the graph drawings are deformed.

To create the initial layout, we used the Backbone layout from Visone [4] as this layout produced drawings scoring 1 or nearly 1 on our metric for our datasets. The exception is that we used `sfdp` from Graphviz [10] for *cv-many-verydense-mid* and *gnm-many-mid-verysparse*, where `sfdp` produces drawings with higher clustering quality metric scores than backbone. We used cluster comparison metrics implementations from scikit-learn [29].

Each dataset for our validation experiment is created by first creating a small graph. Each vertex is replaced with a larger graph of a specified internal density - each will become a cluster of the dataset. Then, each edge is replaced with inter-cluster edges with a specified external density. Table 1 shows the dataset details. $|c|$ stands for the number of clusters and $avg(cd)$ denotes the average

internal density of the clusters, as opposed to the global density denoted in the previous column.

Each graph is named in the format $[name] - [no.ofclusters] - [internaldensity] - [externaldensity]$, where we vary the parameters to increase generality. The prefixes denote the structure used to generate the clustered graph - c stands for a complete graph, b denotes a bipartite graph, s denotes a star graph, t denotes a tree, p denotes a path, rn denotes an r -regular graph, cv is a complete graph with variable cluster sizes, and gnm denotes a $G_{n,m}$ random graph.

Table 1. Validation datasets

Name	$ V $	$ E $	$ c $	Density	$avg(cd)$
$c - few - verydense - mid$	439	9552	9	0.0497	0.399
$s - few - verydense - dense$	2051	256108	10	0.122	0.400
$t - few - dense - mid$	2082	164180	15	0.0379	0.400
$c - few - dense - mid$	898	31516	9	0.0391	0.349
$p - few - verydense - verysparse$	3002	230055	15	0.0511	0.759
$c - mid - verydense - mid$	815	18674	15	0.0563	0.797
$r3 - mid - dense - verysparse$	1773	53103	20	0.0338	0.670
$cv - many - verydense - mid$	2000	54749	30	0.0274	0.788
$r3 - many - verydense - sparse$	3045	124727	30	0.0269	0.801
$gnm - many - mid - verysparse$	2685	26098	30	0.00724	0.214

4.2 Results

Figure 2 displays one deformation experiment example, where vertices are colored based on their combinatorial cluster membership. In step 0 (Fig. 2 (a)), vertices of the same cluster are positioned close to each other, there is minimal overlap between each cluster and the layout produces CQ scores of 1. As the

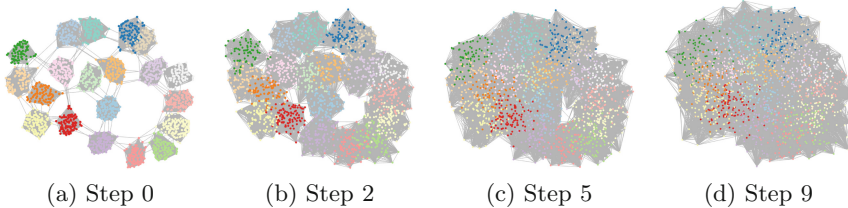


Fig. 2. Deformation experiment for $r3 - mid - dense - verysparse$, drawn using Backbone layout, showing how each subsequent step further deforms the clusters in the drawing.

positions are perturbed, vertices of the same cluster grow further apart. The clusters also continue mixing with each other, until vertices are no less likely to be placed closer to members of other clusters than vertices in its own cluster.

Figure 3 shows the clustering metric scores for each deformation step, with the scores averaged for all datasets in Table 1. We expect to see the CQ scores decreasing after each deformation step, which is indeed what the figure shows, confirming Hypothesis 1 for a wide variety of clustered graphs.

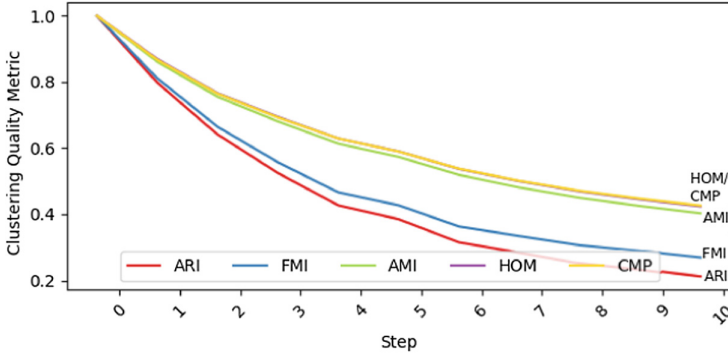


Fig. 3. Average of clustering quality scores for all validation experiments. The decreasing trend for all clustering comparison metrics show that our metric successfully captures the deteriorating visual clustering quality and validates Hypothesis 1. We also see that CQ_{AMI} and CQ_{FMI} are more sensitive to changes in the visual cluster quality, from the steeper curves. Also note that CQ_{HOM} and CQ_{CMP} produce highly similar results such that their curves overlap.

4.3 Discussion and Summary

Figure 3 shows that the plots of the clustering quality metric scores produce a downward slope. This validates our metric and the usage of all selected clustering comparison metrics with our framework. It can also be seen that the scores of our metric deteriorate at different rates when different clustering comparison metrics are used: CQ_{ARI} deteriorates at the fastest rate, followed closely by CQ_{FMI} . CQ_{HOM} and CQ_{CMP} obtains very similar scores with their curves overlapping, while CQ_{AMI} degrades at a slightly faster rate. Therefore, we conclude that CQ_{ARI} and CQ_{FMI} are more sensitive to changes in clustering visualisation quality than the other metrics.

In summary, the validation experiments have shown that our metric reflects the visual clustering quality of drawings of clustered graphs. Furthermore, from the different rates of change of the clustering quality scores when different clustering comparison metrics are used, we conclude that CQ_{ARI} and CQ_{FMI} are better at capturing changes in visual cluster quality and are recommended for use with our framework.

5 Layout Comparison Experiments

5.1 Experiment Design

After the validation experiments have shown that our metric effectively measures visual cluster quality, we compare the performance of a number of graph drawing algorithms against our metric. We selected layouts of different types:

- Force-directed: *Fruchterman-Reingold (FR)* [13] and *Organic* from yfiles [39].
- Multi-level: *FM3* [15] and *sfdp* [10, 16].
- MDS: *Metric MDS* based on classical scaling [37] and *Pivot MDS* [6].
- Stress-based: *Stress Majorization* [14] and *Sparse Stress Minimization* [28].
- Spectral: spectral layout with graph laplacian.

We also selected a few layouts which purport to focus on the discovery of clusters or important community structures in a graph to test their claims:

- *LinLog* [26] modifies the force-directed model to emphasize clusters.
- *Backbone* [27] utilizes triadic or quadratic Simmelian backbones to extract important community structures from “hairball” graphs.
- *tsNET* [22] is based on t-distributed Stochastic Neighbor Embedding (t-SNE), a dimensionality reduction technique [24], and aims to preserve point neighborhoods.

Based on the selection of algorithms, we formulate the following hypothesis:

Hypothesis 2: LinLog, backbone, and tsNET will score higher on our metric than other selected layouts in visualizing clusters in graphs.

We used implementations provided from Tulip [8] (FR, FM3, Pivot MDS, Stress Majorization, LinLog), visone [4] (Backbone, Metric MDS, Sparse Stress Minimization, Spectral), yEd [39] (Organic), Graphviz [10] (sfdp), and Krueger’s implementation of tsNET [21]. We re-used some datasets from the validation experiments and created some new ones, listed in Table 2. We also selected real

Table 2. Additional layout comparison datasets




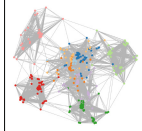


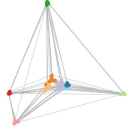

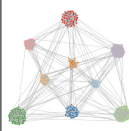
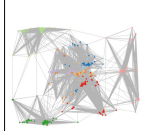

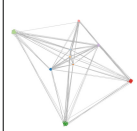
Name	$ V $	$ E $	$ c $	Density	avg(cd)
<i>b – many – dense – sparse</i>	1797	49210	30	0.0305	0.560
<i>cv – mid – verydense – mid</i>	939	21798	20	0.0495	0.162
<i>s – mid – mid – sparse</i>	2116	24175	20	0.0108	0.216
<i>w – many – mid – verysparse</i>	2485	68844	25	0.0223	0.554
<i>r4 – many – verydense – verysparse</i>	3045	124727	30	0.0269	0.801
<i>revije – 90</i>	124	1334	14	0.127	0.377
<i>SS – Butterfly – 0 – 85</i>	832	13009	10	0.0376	0.258
<i>email – Eu – core – lcc</i>	986	16687	34	0.0344	0.490

world graph datasets with existing vertex categorization, which are listed under the double line in Table 2. The datasets were taken from Pajek [2] and Stanford Network Analysis Project’s (SNAP) repository [23, 40].

5.2 Results

Tables 3 and 4 show layout comparison examples, with colours representing ground truth clusters, with CQ scores displayed in Figs. 4 and 5 respectively. LinLog, tsNET, and Backbone score higher than other layouts for both datasets, supporting Hypothesis 2. In Table 3 and Fig. 4, where the number of clusters are small, other layouts such as sfdp, FR, FM3, and spectral also score close to 1. Meanwhile, in the example in Table 4 and Fig. 5 displaying a real world graph with a larger number of clusters, LinLog, tsNET, and backbone’s performances more clearly surpass the other layouts.

Table 3. Layout comparison for $c - few - verydense - mid$

FR	Organic	Stress Maj.	Metric MDS	Backbone	FM3
					
Spectral	S. Stress Min.	tsNET	Pivot MDS	sfdp	LinLog
					

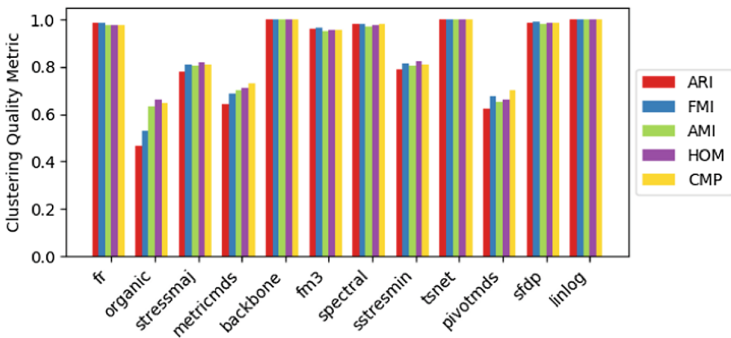


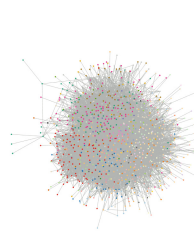
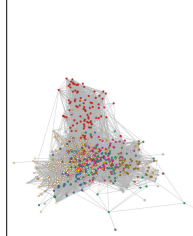


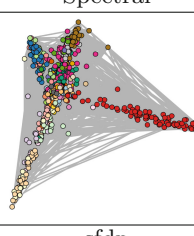

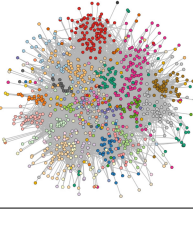
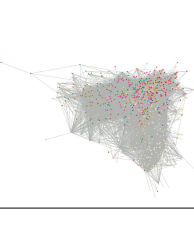
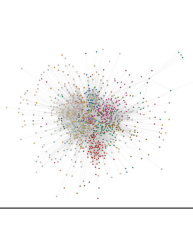
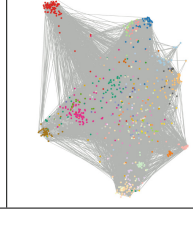


Fig. 4. Clustering quality metrics for $c - few - verydense - mid$. LinLog, tsNET, and Backbone produces scores of 1 on our metrics, in line with Hypothesis 2. For this dataset, sfdp, FR, FM3, and spectral also score highly, close to 1.

Table 4. Layout comparison for *email – Eu – core – lcc*

FR	Organic	Stress Maj.	Metric MDS
			
Backbone	FM3	Spectral	S. Stress Min.
			
tsNET	Pivot MDS	sfdp	LinLog
			

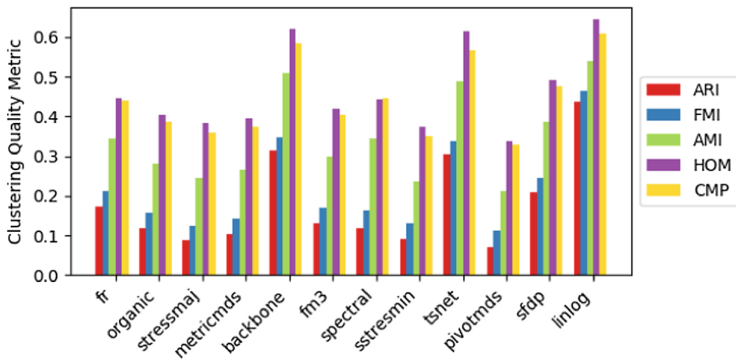
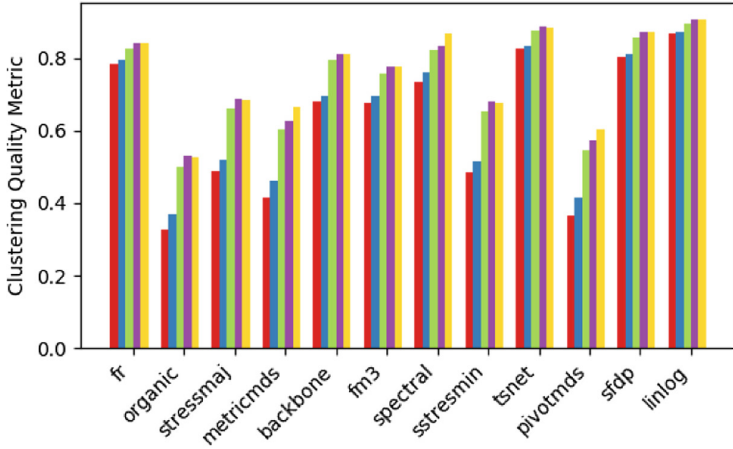
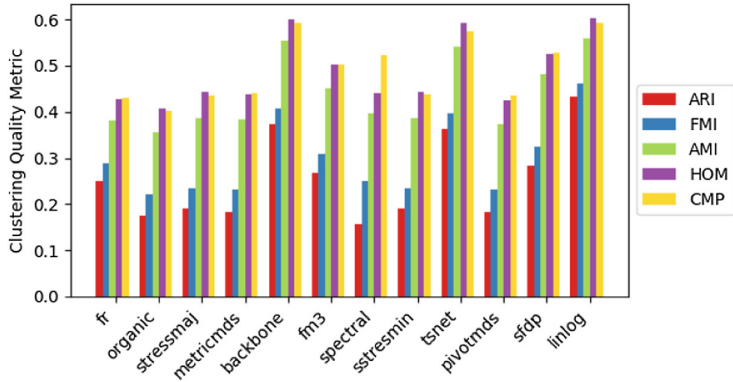


Fig. 5. Clustering quality metrics for *email – Eu – core – lcc*. LinLog, backbone, and tsNET clearly outperform other layouts, as expected from Hypothesis 2. Among non-cluster-focused layouts, sfdp produces the highest scores.



(a) Average for all datasets



(b) Average for real world datasets

Fig. 6. Clustering quality metrics averaged per layout for all layout comparison datasets (a) and for real world datasets only (b). In (a), we see that tsNET and LinLog produce the highest scores, validating Hypothesis 2 for the two layouts. Meanwhile in (b), we see that on real world datasets, LinLog, tsNET, and Backbone outperforms other layout algorithms in accordance to Hypothesis 2.

Figure 6(a) shows the scores averaged across all layout comparison datasets and Fig. 6(b) show the scores averaged across real world datasets. Averaged across all datasets, LinLog scores the highest, with tsNET close behind, confirming Hypothesis 2 for these two layouts. Backbone scores well on many graphs, but sometimes deteriorates in quality when the number of clusters becomes larger compared to the total size of the graph, causing it to score lower than tsNET and LinLog on average (see Fig. 6(a)). Even so, it still outperforms the other algorithms on real world datasets as seen in Fig. 6(b), which supports Hypothesis 2 for Backbone on real world graphs.

In the case of synthetic datasets, sfdp also tends to perform well, as seen in the overall averaged clustering quality metric scores in Fig. 6(a). LinLog, backbone, and tsNET still outperforms it with real world datasets as seen from Fig. 6(b), however, in line with Hypothesis 2.

5.3 Discussion and Summary

Our experiments verify that LinLog and tsNET attains the highest average scores on our metrics across all comparison datasets and Backbone attains equally high average scores on real world datasets.

A point of note is that LinLog often has issues with excessive node overlaps, especially when the internal cluster density is high - this can be seen in Table 3, where the nodes of each cluster are positioned very close together such that they almost appear as only one node, and to a lesser extent in Table 4 where the red cluster is packed quite closely together. Backbone does not have this problem on any tested graphs. Thus, we can conclude that Backbone also has its advantages for practical applications of clustered graph visualization.

In summary, our experiments have confirmed Hypothesis 2 for LinLog and tsNET, which consistently obtained the highest scores across all datasets, while for Backbone it is more supported on real world structures.

6 Conclusion and Future Work

We have introduced a new graph drawing quality metric for the visualization of clusters in graph. Deformation experiments has shown the effectiveness of the metric in measuring how well a drawing of a graph depicts the clusters in the graph. We have also compared graph drawings produced by layouts emphasizing cluster structures to non-cluster-focused layouts and validated the claims of these cluster-focused layouts especially on real world structures.

A direction for future work is to refine the metric by combining it with readability metrics, such as to address node overlaps, and further validating it with human evaluation. Other geometric clustering algorithms besides k -means can also be tested, including fuzzy clustering algorithms that accomodate overlaps between clusters, and concepts of visual cluster separations for scatterplots [35] can also be considered.

References

1. Aldenderfer, M.S., Blashfield, R.: Cluster Analysis. Beverly Hills: Sage Publications, Thousand Oaks (1984)
2. Batagelj, V., Mrvar, A.: Pajek data sets (2003). <http://pajek.imfm.si/doku.php?id=data:index>
3. Battista, G.D., Eades, P., Tamassia, R., Tollis, I.G.: Graph Drawing: Algorithms for the Visualization of Graphs. Prentice Hall PTR, Upper Saddle River (1998)

4. Baur, M., Benkert, M., Brandes, U., Cornelsen, S., Gaertler, M., Köpf, B., Lerner, J., Wagner, D.: Visone Software for Visual Social Network Analysis. In: Mutzel, P., Jünger, M., Leipert, S. (eds.) GD 2001. LNCS, vol. 2265, pp. 463–464. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45848-4_47
5. Behrisch, M., Blumenschein, M., Kim, N.W., Shao, L., El-Assady, M., Fuchs, J., Seebacher, D., Diehl, A., Brandes, U., Pfister, H., Schreck, T., Weiskopf, D., Keim, D.A.: Quality metrics for information visualization. In: Computer Graphics Forum, vol. 37, pp. 625–662. Wiley Online Library (2018)
6. Brandes, U., Pich, C.: Eigensolver methods for progressive multidimensional scaling of large data. In: Kaufmann, M., Wagner, D. (eds.) GD 2006. LNCS, vol. 4372, pp. 42–53. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-70904-6_6
7. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley-Interscience, New York (1991)
8. David, A.: Tulip. In: Mutzel, P., Jünger, M., Leipert, S. (eds.) GD 2001. LNCS, vol. 2265, pp. 435–437. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45848-4_34
9. Eades, P., Hong, S.H., Nguyen, A., Klein, K.: Shape-based quality metrics for large graph visualization. *J. Graph Algorithms Appl.* **21**(1), 29–53 (2017)
10. Ellson, J., Gansner, E., Koutsofios, L., North, S.C., Woodhull, G.: Graphviz—open source graph drawing tools. In: Mutzel, P., Jünger, M., Leipert, S. (eds.) GD 2001. LNCS, vol. 2265, pp. 483–484. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45848-4_57
11. Estivill-Castro, V.: Why so many clustering algorithms: a position paper. *SIGKDD Explor. Newsl.* **4**(1), 65–75 (2002). <https://doi.org/10.1145/568574.568575>
12. Fowlkes, E.B., Mallows, C.L.: A method for comparing two hierarchical clusterings. *J. Am. Stat. Assoc.* **78**(383), 553–569 (1983). <https://doi.org/10.1080/01621459.1983.10478008>
13. Fruchterman, T.M.J., Reingold, E.M.: Graph drawing by force-directed placement. *Softw.: Practice Exp.* **21**(11), 1129–1164 (1991). <https://doi.org/10.1002/spe.4380211102>
14. Gansner, E.R., Koren, Y., North, S.: Graph drawing by stress majorization. In: Pach, J. (ed.) GD 2004. LNCS, vol. 3383, pp. 239–250. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-31843-9_25
15. Hachul, S., Jünger, M.: Drawing large graphs with a potential-field-based multilevel algorithm. In: Pach, J. (ed.) GD 2004. LNCS, vol. 3383, pp. 285–295. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-31843-9_29
16. Hu, Y.: Efficient, high-quality force-directed graph drawing. *Math. J.* **10**(1), 37–71 (2005)
17. Huang, W., Hong, S.H., Eades, P.: Effects of crossing angles. In: 2008 IEEE Pacific Visualization Symposium, pp. 41–46. IEEE (2008)
18. Hubert, L., Arabie, P.: Comparing partitions. *J. Classif.* **2**(1), 193–218 (1985). <https://doi.org/10.1007/BF01908075>
19. Kobourov, S.G., Pupyrev, S., Saket, B.: Are crossings important for drawing large graphs? In: Duncan, C., Symvonis, A. (eds.) GD 2014. LNCS, vol. 8871, pp. 234–245. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45803-7_20
20. Koren, Y.: Drawing graphs by eigenvectors: theory and practice. *Comput. Math. Appl.* **49**(11–12), 1867–1888 (2005). <https://doi.org/10.1016/j.camwa.2004.08.015>
21. Kruiger, J.F.: tsnet (2017). <https://github.com/HanKruiger/tsNET/>
22. Kruiger, J.F., Rauber, P.E., Martins, R.M., Kerren, A., Kobourov, S., Telea, A.C.: Graph layouts by t-SNE. *Comput. Graph. Forum* **36**(3), 283–294 (2017). <https://doi.org/10.1111/cgf.13187>

23. Leskovec, J., Krevl, A.: SNAP Datasets: Stanford large network dataset collection, June 2014. <http://snap.stanford.edu/data>
24. Maaten, L.V.D., Hinton, G.: Visualizing data using t-SNE. *J. Mach. Learn. Res.* **9**(Nov), 2579–2605 (2008)
25. MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 281–297. University of California Press (1967)
26. Noack, A.: An energy model for visual graph clustering. In: Liotta, G. (ed.) *GD 2003*. LNCS, vol. 2912, pp. 425–436. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24595-7_40
27. Nocaj, A., Ortmann, M., Brandes, U.: Untangling the hairballs of multi-centered, small-world online social media networks. *J. Graph Algorithms Appl.* **19**(2), 595–618 (2015). <https://doi.org/10.7155/jgaa.00370>
28. Ortmann, M., Klimenta, M., Brandes, U.: A sparse stress model. In: Hu, Y., Nöllenburg, M. (eds.) *GD 2016*. LNCS, vol. 9801, pp. 18–32. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-50106-2_2
29. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: machine learning in python. *J. Mach. Learn. Res.* **12**(Oct), 2825–2830 (2011)
30. Purchase, H.: Which aesthetic has the greatest effect on human understanding? In: DiBattista, G. (ed.) *GD 1997*. LNCS, vol. 1353, pp. 248–261. Springer, Heidelberg (1997). https://doi.org/10.1007/3-540-63938-1_67
31. Purchase, H.C., Cohen, R.F., James, M.: Validating graph drawing aesthetics. In: Brandenburg, F.J. (ed.) *GD 1995*. LNCS, vol. 1027, pp. 435–446. Springer, Heidelberg (1996). <https://doi.org/10.1007/BFb0021827>
32. Rand, W.M.: Objective criteria for the evaluation of clustering methods. *J. Am. Stat. Assoc.* **66**(336), 846–850 (1971). <https://doi.org/10.1080/01621459.1971.10482356>
33. Rosenberg, A., Hirschberg, J.: V-measure: a conditional entropy-based external cluster evaluation measure. In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp. 410–420 (2007)
34. Saket, B., Simonetto, P., Kobourov, S.: Group-level graph visualization taxonomy. *CoRR abs/1403.7421* (2014)
35. Sedlmair, M., Tatu, A., Munzner, T., Tory, M.: A taxonomy of visual cluster separation factors. *Comput. Graph. Forum* **31**(3pt4), 1335–1344 (2012). <https://doi.org/10.1111/j.1467-8659.2012.03125.x>
36. Strehl, A., Ghosh, J.: Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.* **3**(Dec), 583–617 (2002)
37. Torgerson, W.S.: Multidimensional scaling: I. Theory and method. *Psychometrika* **17**(4), 401–419 (1952). <https://doi.org/10.1007/BF02288916>
38. Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. *J. Mach. Learn. Res.* **11**(Oct), 2837–2854 (2010)
39. Wiese, R., Eiglsperger, M., Kaufmann, M.: yfiles - visualization and automatic layout of graphs. In: Jünger, M., Mutzel, P. (eds.) *Graph Drawing Software*. Mathematics and Visualization, pp. 173–191. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-642-18638-7_8
40. Zitnik, M., Sosič, R., Maheshwari, S., Leskovec, J.: BioSNAP Datasets: Stanford biomedical network dataset collection, August 2018. <http://snap.stanford.edu/biodata>