A Density-Based Statistical Analysis of Graph Clustering Algorithm Performance

Pierre Miasnikof *[1], Alexander Y. Shestopaloff[2], Anthony J. Bonner[3], Yuri Lawryshyn[1], and Panos M. Pardalos[4,5]

[1]University of Toronto, Department of Chemical Engineering & Applied Chemistry, 200 College St., Toronto, ON, M5S 3E5, Canada
[2]The Alan Turing Institute, 96 Euston Road, London, NW1 2DB, United Kingdom
[1]University of Toronto, Department. of Computer Science, 40 St. George Street, Toronto, ON, M5S 2E4, Canada
[4]University of Florida, Department of Industrial and Systems Engineering, P.O. Box 116595, Gainesville, FL, 32611-6595, USA
[5]HSE University, Laboratory of Algorithms and Technologies for Networks Analysis, 136 Rodionova St., Nizhny Novgorod, 603093, Russian Federation

Abstract

We introduce graph-clustering quality measures based on comparisons of global, intra- and inter-cluster densities, an accompanying statistical significance test and a step-by-step routine for clustering quality assessment. Our work is centered on the idea that well clustered graphs will display a mean intra-cluster density that is higher than global density and mean inter-cluster density. We do not rely on any generative model for the null model graph. Our measures are shown to meet the axioms of a good clustering quality function. They have an intuitive graph-theoretic interpretation, a formal statistical interpretation and can be tested for significance. Empirical tests also show they are more responsive to graph structure, less likely to breakdown during numerical implementation and less sensitive to uncertainty in connectivity than the commonly used measures.

[*]corresponding author: p.miasnikof@mail.utoronto.ca

1 Introduction

While there are many graph clustering [1] algorithms in the literature (e.g., [38, 12, 13, 45, 3, 37]), measuring their performance, assessing the quality of the clusters they identify, remains an open problem [28, 27, 31, 2, 36, 50, 26, 21, 9, 11, 4, 49, 24, 40]. In fact, it's been stated in a very recent publication that *"in the literature, there is no universally accepted metric for evaluating the performance of community detection algorithms"* [40].

Graph clustering is the process of assigning common labels to vertices that are considered similar, vertices that should belong to a common set (cluster). It is a form of unsupervised learning, where one typically cannot count on labeled data to assess results. For example, Reichardt and Bornholdt correctly assert that "*(...) running a clustering algorithm over a set of randomly generated data points will always produce clusters which, however, have little meaning*" [44] . For this reason, our only quality measure is a thorough examination of the graph's and resulting clusters' connectivity patterns.

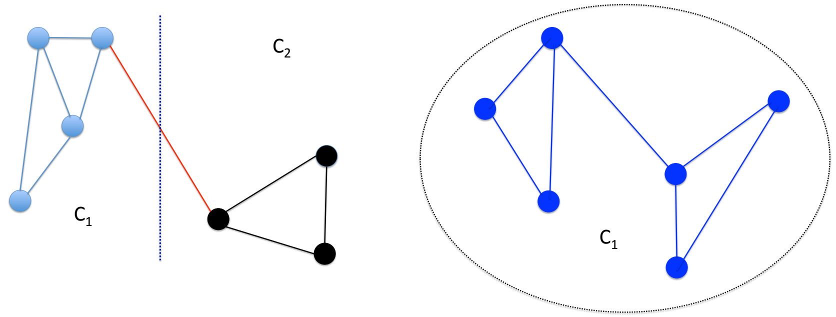
In this article, we present new (algorithm-independent) clustering performance measures to assess the strength of the clustering returned by any algorithm. Our measures can also provide comparisons of several clustering algorithms on a given graph. Through this presentation, we also offer our own definition of clustering quality and techniques to test its statistical significance.

Our measures are based on comparisons of global, intra- and inter-cluster densities. Our null model does not rely on any generative model for the graph, unlike modularity which uses the configuration model as a null. Also, unlike the context-dependent approach presented by Creusefond et al. [9], we propose a general purpose quality measure based on graph and subgraph density, well known graph characteristics, and formal statistical testing.

We begin with a review of two of the most common clustering performance measures, modularity and conductance. We empirically demonstrate how these measures may be drowned out by graph structure and lack sensitivity to it. We also demonstrate how our test of clustering quality based on our two statistical measures of graph structure is more robust, easier to interpret and consistent with the axioms of good clustering.

We restrict our attention to undirected unweighted and weighted graphs with positive edge weights. The graphs we consider also have no self-loops

[1] Note on vocabulary: Although there are subtle differences between the concepts of graph clustering and community detection, we use the two interchangeably.



(a) Well Clustered Graph

(b) Improperly Clustered Graph

Figure 1: Examples of Good and Bad Clustering

or multiple edges. It is important to emphasize that we are not trying to identify clusters and their constituent vertices, in this article. The work in this article focuses exclusively on assessing the quality of the clusters identified by a clustering algorithm. For example, we want an objective measure that allows us to conclude the algorithm that clustered the graph in Figure 1a performed well, while the algorithm that clustered the graph in Figure 1b performed poorly and did not partition the graph adequately.

2 Commonly Used Performance Measures, Quality Functions

The term "clustering quality function" of Van Laarhoven and Marchiori [26] is often used to designate clustering performance measures. These authors use the term "quality function" to describe a function that takes in a graph G and a set of node clusters $C = \{c_1, c_2, \ldots\}$ and returns a real number, the quality measure. All the measures discussed in this article fit this description.

In this section, we describe the two most frequently used clustering quality functions, modularity and conductance. These functions are not only used to assess clustering quality but are also often used as objective functions in optimization-based clustering techniques (e.g., [5, 46]).

2.1 Modularity

Modularity (Q) is by far the most popular measure of clustering performance [38, 8, 6, 14, 31, 23, 41, 42]. Originally introduced by Newman and Girvan in

2004 [38], it has since been extensively used both as a performance measure and objective function to be maximized (e.g., [38, 10, 3, 37]). In this section, we present modularity (Q) as shown in Clauset et al. [8]. Modularity is defined as

$$\begin{aligned} Q &= \sum_{i=1}^{k} \left(\underbrace{e_{ii} - a_i^2}_{q_i} \right) \\ \text{where,} & \\ e_{ii} &= \frac{1}{2m} \sum_{v,w} A_{v,w}\, \delta(c_v, i)\delta(c_w, i) \\ a_i &= \frac{1}{2m} \sum_{v} A_{v,\cdot} \vec{1}\, \delta(c_v, i)\,. \end{aligned}$$

Here, $m = |E|$ is the total number of edges in the graph, k is the number of clusters, $A_{v,w}$ is the element at the intersection of the v-th row and w-th column of the adjacency matrix, $A_{v,\cdot}$ is the entire v-th row of the adjacency matrix, $\vec{1}$ is a vector of ones of appropriate dimension, $\delta(x, y)$ is the Kroenecker delta function and c_v is the cluster in which vertex v is clustered by the algorithm. Expanding our expression above using the graph's adjacency matrix, we get

$$Q = \sum_{i=1}^{k} \left[\underbrace{\frac{1}{2m} \sum_{v,w} A_{v,w}\, \delta(c_v, i)\delta(c_w, i)}_{e_{ii}} - \underbrace{\frac{1}{4m^2} \left(\sum_{v} A_{v,\cdot} \vec{1}\, \delta(c_v, i) \right)^2}_{a_i^2} \right]. \quad (1)$$

Modularity always lies on the interval $[-\frac{1}{2}, 1]$ [6]. Values greater than 0.3 typically indicate a significant clustering [8].

In closing, it should be noted that modularity's biggest weakness is that it suffers from resolution limit. This weakness is well documented in the literature. For example, Fortunato and Bathélemy [15] devote an entire article to this topic. These authors describe how any clustering quality function that is defined as a sum of qualities of individual clusters, where terms from smaller clusters are dominated by terms from larger clusters, suffers from resolution limit. Because the smaller clusters' contribution to the sum is dominated by the larger clusters', the final result is also dominated and does not always reflect structure accurately. Indeed, in Equation (1) we see how larger clusters dominate the outer summation. Modularity also suffers from many other degeneracies, as described by Good et al. [17] among others. Additionally, Fortunato showed the difficulty of conducting statistical tests on modularity, due to the challenge of identifying its true distribution [14]. Finally, we note that Van Laarhoven and Marchiori have demonstrated modularity fails to meet some of the axioms of a good clustering quality function, namely locality and monotonicity [26].

2.2 Conductance

Conductance (ϕ, Φ) is another popular clustering performance measure [32, 31, 50, 46, 9]. It is also used by some authors as an objective to be minimized (e.g., [22, 46]). In this article, we use the definition presented by Spielman and Teng [46]. At the individual cluster level, conductance is defined as

$$\phi(c_i) = \frac{\partial(c_i)}{\min\left(d(c_i), d(V \setminus c_i)\right)} .$$

While at the graph level, it is defined as

$$\Phi(G) = \min_{c_i} \phi(c_i) .$$

Here, $\partial(c_i)$ is the number of edges joining vertices in cluster c_i to vertices outside c_i, $d(c_i)$ is the sum of vertex degrees within c_i and $d(V \setminus c_i)$ the sum of vertex degrees on the graph, outside c_i. A low conductance indicates strongly connected clusters.

3 A Definition of Good Clustering and New Clustering Performance Measures

We now present our statistical performance measures, the Kappas. Before we begin, we introduce our own definition of good clustering. The Kappas are rooted in this definition. Our performance measures are based on a comparison of global, inter- and intra-cluster densities. We argue that the difference between these quantities is a measure of clustering strength.

3.1 A Definition of Clustering Quality

In accordance with every other definition of a good clustering, we expect that an efficient clustering algorithm will label vertices such that intra-cluster connectivity is greater than global and inter-cluster connectivity (e.g., [14, 41, 42]) (if the graph does indeed have a clustered structure). We expect that nodes with the same label will have more common connections to each other than to ones with different labels. Alternately, we expect mostly uniform connectivity between vertices in the case of improperly clustered graphs. In summary, we expect that clustered vertices will form dense subgraphs within a sparser graph, when they are properly labeled.

In step with these expectations, to gain a macroscopic view of the entire graph, we set up our average case benchmarks. We posit that a good clustering will group vertices so they form clusters whose vertices are more strongly connected to each other than to vertices belonging to different clusters, on average. To gauge the strength of clustering, we rely on graph theory and statistics. We compare mean intra-cluster density, the graph's overall density and mean inter-cluster density. These quantities are defined below, in Section 3.2.

We postulate that a good clustering must meet two necessary conditions. Together, these necessary conditions form a sufficient condition for a clustering to be of good quality, on average. First, a good clustering should be characterized by a mean intra-cluster density that is higher than the graph's global density. Global density should, in turn, also be higher than mean inter-cluster density. Meeting this set of inequalities is the first necessary condition.

The second necessary condition, also an inequality, is that mean intra-cluster density must be significantly greater than mean-inter cluster density. This last inequality automatically holds numerically, whenever the first necessary condition is met. However, this second necessary condition is a condition on statistical significance. We want the difference between these two mean densities, a quantity we call γ, to be non-trivial, statistically significant.

The idea of using mean "intra-connectivity" and mean "inter-connectivity" to measure clustering strength was initially presented by Mancoridis et al. [33], although their definitions were not based on simple strict graph density. These authors' goal was to find optimal partitions of software system components and their model was based on directed graphs with self-loops. Later, Kannan et al. [22] used global inter-cluster density as an indicator of poor clustering, to be minimized along with conductance. Fortunato [14] then introduced the idea of using inter- and intra-cluster densities and comparing them to global density as a measure of the clustering strength of single clusters. We extend these ideas to gain a macroscopic view of the entire graph's cluster labeling, using the standard definition of density.

While they are inspired by Mancoridis et al. and by Fortunato's follow-up on those ideas, our measures are tailored to undirected graphs and provide a macroscopic view. Unlike the measures proposed by Mancoridis et al., they correspond to standard definitions for simple graphs and can be generalized to weighted graphs as well. Unlike the measures proposed by Fortunato, they offer a graph level, not cluster level, picture.

In the case of unweighted graphs, our measures are bounded within the interval $[0, 1]$. In the case of weighted graphs, they become proportional to edge weight. High values denote densely connected graphs, clusters or cluster pairs and vice-versa. Also, because we take means over the entire graph, our measures provide a graph-wide picture, have statistical meaning and can easily be subjected to hypothesis testing. For these reasons, our measures are also more meaningful and more solidly grounded in graph and statistical theories than either conductance or modularity.

The well established and widely used measures of clustering strength, modularity and conductance, measure intra-cluster connectivity strength. Instead, we measure the strength of intra- and inter-cluster connectivity relative to each other and to the overall graph's connectivity. In doing so, we tailor our definition of and conclusions on the quality of clustering to the specific graph structure being analyzed. For example, in a moderately densely connected graph we expect clusters to be even more strongly connected. Meanwhile, moderately strong inter-cluster connections can be consistent with a good partition, as long as it remains weaker than intra-cluster connectivity. Conversely, in a sparsely connected graph, moderately strong inter-cluster connectivity is a symptom of a poor clustering.

Our claim regarding inter- and intra-cluster densities is empirically validated by the simulations of our null hypothesis, shown in Section 4.1. Through those simulations, we empirically confirm our hypothesis that in a random clustering of a graph, arguably the poorest of non-trivial clusterings, the difference between mean intra-cluster and mean inter-cluster densities is approximately zero.

3.2 The Kappas, Measures of Density

We define Kappa (K) as the graph's overall connectivity measure, mean Kappa intra-cluster ($\bar{K}_{\text{intra}}$) as the measure of intra-cluster connectivity and mean Kappa inter-cluster ($\bar{K}_{\text{inter}}$) as the measure of inter-cluster connectivity. These quantities are the graph's global density, mean intra-cluster density and mean inter-cluster density, respectively. Here, we extend Fortunato's idea of examining inter- and intra-cluster density to determine the strength of a clustering [14], but with a focus on average-case subgraphs, not individual clusters. As mentioned in the previous section, we expect that under a good clustering the inequalities $\bar{K}_{\text{inter}} < K < \bar{K}_{\text{intra}}$ will hold. These inequalities are similar to those formulated by Fortunato [14], but they are formulated at a graph-wide scale, on the basis of graph-wide average-cases. Our model also allows these inequalities to be formulated as statistical hypothesis tests, as will be shown later.

Below, we present the formulation for our clustering measures, for an unweighted undirected graph, but they easily generalize to weighted graphs as well. For weighted graphs, when edge weights are proportional to connection strength, our measures are computed by replacing the cardinality of edge sets (edge counts) with the sums of the corresponding edge weights (all edge weights or intra-/inter-cluster weights). However, it should be noted that densities in the weighted case are no longer contained within the interval $[0, 1]$, although they remain non-negative (since we only consider positive edge weights).

In our formulation, we use the following variables. The set of all clusters is $C = \{c_1, \ldots, c_\ell\}$, with $|C| = \ell$, the total number of vertices in the graph is $|V| = N$, the total number of vertices in cluster i is n_i, the set of all edges on the graph is $E = \{e_1, \ldots, e_m\}$, where $|E| = m$. Finally, E_{ij} is the set of edges connecting a vertex in cluster i to a vertex in cluster j, and $|E_{ij}| = m_{ij}$. As a special case, note that E_{ii} is the set of edges within cluster i, and m_{ii} is the number of edges connecting vertices within cluster i.

Using standard definitions, we take the ratio of the edge counts (or sum of weights) over the maximum possible number of edges given the number of vertices. For mean intra- and inter-cluster connectivity, we compute the ratio for each cluster or pair of clusters and take their mean as a graph-wide measure.

We compute the graph's connections ratio, global density, as

$$K = \frac{|E|}{0.5 \times N(N-1)} \ .$$

This quantity is the ratio of the total number of edges over the number of edges in a complete graph with the same number of vertices. In the case of an unweighted graph, the closer K is to 1, the closer the graph is to being a complete graph. Conversely, the closer K is to 0, the closer the graph is to being a set of disconnected vertices.

For a graph clustered into ℓ clusters, we also define the mean intra-cluster connections ratio, mean intra cluster density, as

$$\bar{K}_{\text{intra}} = \frac{1}{\ell}\sum_{i=1}^{\ell}\kappa_i = \frac{1}{\ell}\sum_{i=1}^{\ell}\frac{|E_{ii}|}{0.5 \times n_i(n_i - 1)}$$

where,

$$\kappa_i = \begin{cases} \frac{|E_{ii}|}{0.5 \times n_i(n_i-1)} & \text{if } n_i \geq 2 \\ 0 & \text{otherwise} \end{cases} .$$

The mean intra-cluster connections ratio is the mean ratio of the number of edges within each cluster over the maximum number of edges that could possibly connect the vertices within each cluster. Each term in the summation, denoted as κ_i, represents each cluster's internal density, the density of the induced subgraph formed by its vertices and the edges connecting them. It is a measure of how closely each cluster is to being a clique. In the unweighted case, each κ_i always lies on the interval $[0, 1]$, with a value of 0 indicating a cluster is just a set of disconnected vertices and a value of 1 indicating that a cluster is a clique. At the aggregate level, $\bar{K}_{\text{intra}}$ is the sample mean of the individual terms κ_i and also lies in the interval $[0, 1]$. Values close to 0 indicate poorly connected clusters on average, while values closer to 1 indicate densely connected clusters on average.

Finally, we define the mean inter-cluster connections ratio, mean inter-cluster density, as

$$\begin{aligned}
\bar{K}_{\text{inter}} &= \frac{1}{0.5 \times \ell(\ell-1)}\sum_{i=1}^{\ell}\sum_{j=i+1}^{\ell}\kappa_{ij} \\
&= \frac{1}{0.5 \times \ell(\ell-1)}\sum_{i=1}^{\ell}\sum_{j=i+1}^{\ell}\frac{|E_{ij}|}{0.5 \times ((n_i + n_j)(n_i + n_j - 1) - n_i(n_i - 1) - n_j(n_j - 1))} \\
&= \frac{1}{0.5 \times \ell(\ell-1)}\sum_{i=1}^{\ell}\sum_{j=i+1}^{\ell}\frac{|E_{ij}|}{n_i \times n_j} .
\end{aligned}$$

The mean inter-cluster connections ratio is the mean ratio of the number of edges joining vertices in a pair of clusters (c_i, c_j) where c_i and c_j are distinct clusters, over the total number of edges that could possibly connect each pair of vertices across the cluster pair (c_i, c_j). Each term in the double summation is the density of the induced bipartite graph formed by the vertices in each cluster pair, when we ignore the edges that join vertices within each cluster and only consider edges between vertices of either clusters of the pair. It is a measure of how close two clusters i and j are from a biclique, when considering only edges that have endpoints in either cluster. Here again, in the unweighted case these terms also lie in the interval $[0, 1]$. A value of 0 indicates no connection between a pair of clusters and a value of 1 indicates the pair of clusters forms a biclique, when we ignore the intra-cluster edges. At the aggregate level, $\bar{K}_{\text{inter}}$ is the sample mean of the individual κ_{ij} and also lies in the interval $[0, 1]$. Values close to 0 indicate low inter-cluster connections, on average, a desirable feature indicating strong cluster partitions. On the other hand, values closer to 1 indicate improperly partitioned clusters, on average.

3.2.1 Limitations:

Our inter- and intra-cluster measures are means and are therefore sensitive to outliers. For example, a few very dense clusters in an otherwise poorly clustered graph or very sparse clusters in an otherwise well clustered graph may render intra-cluster density uninformative. Mean inter-cluster density can also be affected in a similar way by outliers. A small number of densely inter-connected cluster pairs in an otherwise well clustered graph or a small number of sparsely inter-connected ones in an otherwise poorly clustered graph may also render mean inter-cluster density uninformative.

Our inter- and intra-cluster measures are also unweighted means, which implies they are unaffected by cluster or cluster pair sizes. This immunity to cluster sizes is a deliberate design feature. It makes our measures immune to resolution-limit, a common shortcoming of clustering quality functions. A more detailed discussion of this phenomenon can be found in Section 5.

Nevertheless, precautions should be taken with conclusions in cases where cluster or cluster pair sizes are very heterogenous. For example, an algorithm may cluster a few small clusters very well and lump the remaining vertices into one or a few large sparsely connected clusters, which would result in a high mean intra-cluster density, even if the graph is arguably poorly clustered.

Fortunately, in most cases, we expect that an inaccurately inflated mean intra-cluster density will also result in a high mean inter-cluster density. Likewise, we expect that an inaccurately low mean inter-cluster density will also result in a low mean intra-cluster density. However, as with any other analysis based on means, the sample should also be carefully examined. In this specific case, particular attention should be paid to individual cluster and cluster pair sizes and to the individual κ_i, κ_{ij} being averaged.

It should also be mentioned that in cases where the connectivity patterns

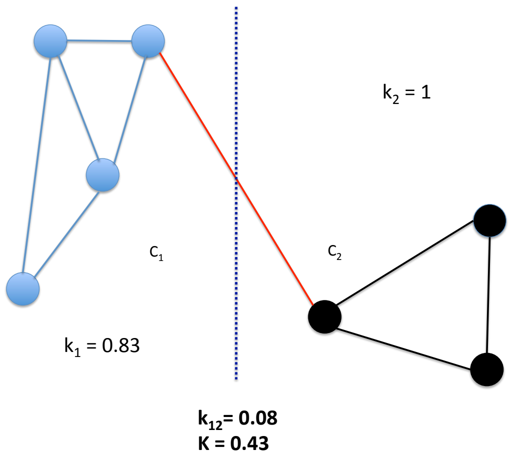


Figure 2: Well-Clustered Graph

of the clusters are very noisy (high variance), the median of the inter-/intra-cluster densities can be used in lieu of the mean. This substitution can produce more robust measures of inter- and intra-cluster connectivity. Unfortunately, it also makes interpretation and significance testing less obvious.

3.2.2 Illustrative Example:

In the previous section, we define a well clustered graph as one where the inequalities $\bar{K}_{\text{inter}} < K < \bar{K}_{\text{intra}}$ hold. We illustrate this definition using Figure 2, which contains what is arguably a well labelled (clustered) graph with two clusters. If we compute the mean inter- and intra-cluster densities and compare them to the graph's global density, we see the inequalities described in Section 3.2 hold:

$$\begin{aligned}
\bar{K}_{\text{intra}} &= \frac{1}{2}(1 + 0.83) = 0.92 \\
\bar{K}_{\text{inter}} &= \left(\frac{1}{0.5 \times 2 \times 1}\right)\frac{1}{4 \times 3} = 0.08 \\
K &= \frac{9}{0.5 \times 7 \times 6} = 0.43 \\
&\Rightarrow \bar{K}_{\text{inter}} < K < \bar{K}_{\text{intra}}
\end{aligned}$$

3.3 Statistical Interpretation of the Kappas and Clustering Quality

One of the appealing features of our Kappas lies in their statistical definition. The statistical definition provides a means to formally interpret them, assess their significance and compare their differences. Such comparisons and tests are problematic with many currently used quality measures. For example, in the case of modularity, these problems were identified by both Fortunato [14] and Traag et al. [47].

In the unweighted graph case, densities can be modeled and interpreted as Bernoulli probabilities. Under this Bernoulli model, the number of trials is the maximum number of possible edges, while the number of successes is the actual number of edges. The empirical estimate of the unconditional probability of connection between any two vertices on the graph is given by the graph's global density, K. Meanwhile, the conditional probability of a connection, given both nodes are in cluster i, is given by its internal density, κ_i. The conditional probability of connection, given one node is in cluster i and the other in cluster j, is given by their inter-cluster density, κ_{ij}. In each case, the quantities K, κ_i and κ_{ij} are the ratios of the observed number of edges over the maximum possible number of edges.

The cluster and cluster pair conditional probabilities of connection can also be generalized to obtain graph level values. This generalization provides probabilities of connection for all pairs of vertices within clusters or across cluster pairs. To obtain graph-wide estimates of these conditional probabilities, we take their means over all clusters or cluster pairs. Their means, $\bar{K}_{\text{inter}}$ and $\bar{K}_{\text{intra}}$ correspond to the graph level Bernoulli probabilities of two vertices sharing an edge, within clusters or across cluster pairs.

In the weighted case, these empirical estimates of connection probability become the mean edge weights. The quantities K, $\bar{K}_{\text{intra}}$ and $\bar{K}_{\text{inter}}$ are the empirical estimate of edge weights connecting two nodes anywhere on the graph, within or between clusters.

At this point, it must be mentioned that in estimating edge probabilities or weight means, our model assumes all clusters are equally likely. We do not weight our estimates by cluster or cluster pair size deliberately, to avoid resolution-limit. As mentioned earlier, a more detailed discussion of resolution-limit is provided in Section 5.

By exploiting the statistical interpretation of our clustering measures, we can offer a statistical description of good clustering. We posit that a good clustering will partition the graph such that the probability there exists an edge (e_{ij}) between two arbitrary nodes i and j is lower than the probability a connection exists if these nodes are in the same cluster (i.e., if $c_i = c_j$) and higher than when they belong to different clusters (i.e., $c_i \neq c_j$). Mathematically, we expect the following inequalities to hold in cases where the clustering returned by an algorithm is of good quality ($\hat{P}$ denotes the empirical estimate of the probablities):

$$\hat{P}\left[e_{ij}|c_i \neq c_j\right] < \hat{P}[e_{ij}] < \hat{P}[e_{ij}|c_i = c_j]\,.$$

In the case of a weighted graph, these empirical probability estimates become empirical estimates of the expected values of edge weights between arbitrary vertices, vertices within and vertices in different clusters. Consequently, we expect the following inequalities to hold ($\hat{E}$ denotes the empirical estimate of the expected values):

$$\hat{E}[e_{ij}|c_i \neq c_j] < \hat{E}[e_{ij}] < \hat{E}[e_{ij}|c_i = c_j]\,.$$

To better illustrate the intuition behind them, we use our statistical interpretation to examine our inequalities in the context of the following three base-case examples: an algorithm which lumps all vertices into one single cluster, an algorithm which assigns each vertex to its own cluster and an algorithm which randomly assigns vertices to k clusters. Together, these examples form the set of the worst possible degenerate clusterings.

3.3.1 All Vertices Into One Single Cluster

Of all three poor clustering base-cases, this case is the only one where intra-cluster density is greater than inter-cluster density. Because there is only one cluster, we have the following equalities and inequality:

$$\begin{aligned}
&\bar{K}_{\text{inter}} = P(e_{ij}|c_i \neq c_j) = 0 \\
&\bar{K}_{\text{intra}} = P(e_{ij}|c_i = c_j) = P(e_{ij}) = K \\
&\Rightarrow \bar{K}_{\text{intra}} \not> K > \bar{K}_{\text{inter}}\,.
\end{aligned}$$

In this case, inter-cluster density is equal to zero, by definition. As a result, intra-cluster density is automatically greater than inter-cluster density ($\bar{K}_{\text{intra}} > \bar{K}_{\text{inter}}$), if there is at least one edge. Nevertheless, intra-cluster density is not higher than the global density ($\bar{K}_{\text{intra}} \not> K$). Our inequalities test does indeed detect the degenerate clustering, since the first necessary condition is not met.

Generally, lumping all vertices into one cluster is a symptom of a poor clustering algorithm. However, there are instances where a graph may not be clusterable. It may not be formed by clusters, sets of dense subgraphs contained within a wider sparser graph. In such cases, assigning all its vertices to the same cluster is arguably the best possible clustering. For example, the best clustering for the vertices of a complete graph is to assign the same label to all of them.

The single-cluster clustering is a limiting case for our characterization of good-clustering, because it may inaccurately classify proper clusterings as being of bad quality. Fortunately, however, it is a rare occurrence. Nevertheless, the single-cluster clustering must be treated as a special case and clustering quality must be assessed by other means. In particular, the graph's global density, which is equal to intra-cluster density, in this case, remains a valid measure of clustering quality. Global density provides a measure to trivially judge the quality of clustering in the single-cluster case. Arguably, a graph whose vertices are all clustered into one cluster is well clustered only if its global density is very high. It is poorly clustered otherwise.

3.3.2 Single-vertex Clusters

A clustering where each vertex is assigned to its own cluster is another base-case degeneracy (in cases where the graph has at least one edge). While it can trivially be identified by a simple examination of the clustering algorithm's labeling, our first necessary condition does not hold. The poor-quality clustering is correctly identified and the following relations hold:

$$\bar{K}_{\text{intra}} = 0 < K = \bar{K}_{\text{inter}} \,.$$

3.3.3 Random Cluster Assignment

This degenerate clustering is not easily detectable by a simple examination of clustering results, unlike the previous two instances. In fact, when the number of vertices is large, this degeneracy is impossible to identify trivially. Fortunately, however, this is a case where our first necessary condition fails. The poor clustering is identified by these equalities:

$$\begin{aligned} &\bar{K}_{\text{inter}} = P(e_{ij}|c_i \neq c_j) = P(e_{ij}) = K \\ &\bar{K}_{\text{intra}} = P(e_{ij}|c_i = c_j) = P(e_{ij}) = K \\ &\Rightarrow \bar{K}_{\text{intra}} = K = \bar{K}_{\text{inter}} \,. \end{aligned}$$

Because it is not possible to trivially identify this degeneracy, we consider it as the canonical example of a bad clustering. For this reason, we use the random cluster labeling as our null model and as a null hypothesis for our statistical tests, which are described in the next section.

3.4 Link to K-means

K-means is one of the most widely used clustering algorithms. It applies to situations where covariates are quantitative and the Euclidian distance separating them is known [19]. Squared Euclidian distances are then used as dissimilarity measures between covariates. K-means is an iterative algorithm that groups covariates into clusters such that intra-cluster dissimilarity (distance) is minimized. Resulting clusters are formed by grouping covariates according to their similarity.

In the case of graphs, we typically do not have the benefit of inter vertex distances. We use density of connections as a measure of similarity. Under our definition of clustering quality, clusters are formed by vertices that are more similar to each other than to vertices in other clusters. Arguably, our definition corresponds to the rationale behind the K-means algorithm. An equivalent average-case relationship between intra cluster, inter cluster and global distances between covariates is expected to hold in instances of successful K-means clusterings. In such cases, it is naturally expected that the mean distances between covariates within clusters will be smaller than between covariates belonging to different clusters. In the case of a successful K-means clustering, we expect the following inequalities to hold:

$$\bar{D}_{\text{intra}} < \bar{D} < \bar{D}_{\text{inter}} \,.$$

In these inequalities, $\bar{D}_{\text{intra}}, \bar{D}, \bar{D}_{\text{inter}}$ denote the mean Euclidian distances between covariates within the same cluster, between any two covariates regardless of their cluster and between covariates belonging to different clusters. We note that in the case of K-means the inequalities are reversed, since the quantities being compared are dissimilarities.

3.5 A Complete Clustering Quality Assessment Routine

Our quality assessment routine consists of two steps, the verification of each of our two necessary conditions whose combined fulfillment forms a sufficient (average case) condition. In order to conclude on the quality of a clustering, we must verify that our first set of inequalities holds. In the event they do, we must also ensure they hold at a statistically significant level.

The first step consists of the computation of global, inter- and intra-cluster densities and their comparison through our first necessary condition inequalities test. In this first step, we verify that the inequalities $\bar{K}_{\text{inter}} < K < \bar{K}_{\text{intra}}$ hold. We verify that mean intra-cluster density is higher than global density and that global density is also higher than mean inter-cluster density. These inequalities form the foundations of our statistical (average case) definition of good clustering.

Once it has been established that our first set of necessary conditions have been met, we test the statistical significance of the inequality $\bar{K}_{\text{intra}} > \bar{K}_{\text{inter}} \Leftrightarrow \gamma = \bar{K}_{\text{intra}} - \bar{K}_{\text{inter}} > 0$, sample size permitting. This second step consists of a statistical significance test of the difference between intra- and inter-cluster densities, which we call γ. A detailed description of statistical significance tests is provided in the next section.

- **Clustering Quality Assessment Routine:**
    - Use clustering algorithm labels to compute the Kappas
    - Numerically verify that the inequalities $\bar{K}_{\text{inter}} < K < \bar{K}_{\text{intra}}$ hold (first necessary condition)
    - If they don't and if the number of clusters is greater than one, conclude the algorithm has poorly clustered the graph
    - If they don't and if the number of clusters is one, use global density to assess clustering quality
    - If they do hold and if the number of clusters is sufficiently large, perform statistical test to verify significance (second necessary condition) of the difference

      $$\gamma = \bar{K}_{\text{intra}} - \bar{K}_{\text{inter}}$$

    - If testing more than one algorithm that meet all benchmarks above, compare p-values to find best algorithm

In closing, we note that in many cases a simple examination of the inequalities in the first necessary condition, along with some domain expertise may be sufficient to draw conclusions on clustering quality.

4 Hypothesis Testing

As mentioned earlier, because our measures of clustering are also graph statistics, we can push our analysis further and verify our second necessary condition, which is described in Section 3. Through this test, we ensure the difference between intra- and inter-cluster densities is statistically significant. Of course, we could also test the significance of the difference between intra-cluster and global densities and between global density and intra-cluster density. However, these multiple tests would be redundant and would complicate the testing process. After all, according to all definitions of good clustering, the benchmark is that intra-cluster connectivity (density) be greater than inter-cluster connectivity (density).

The stochasticity which makes hypothesis testing possible stems from the fact each clustering is one sample of an unobserved distribution of vertex labelings. Each graph's clustering, as returned by one particular clustering algorithm, can be understood as being one sample drawn from this unobserved distribution of all possible clusterings into the same number of clusters as those identified by the algorithm (or set by external parameter) for the graph under study. The quantities, $\bar{K}_{\text{intra}}$ and $\bar{K}_{\text{inter}}$ are the corresponding sample means of intra- and inter-cluster densities.

To formally confirm statistical significance, we use a modified version of the standard Student's t-test, which is described in Section 4.2. It must be emphasized that because our performance measures are sample estimates of a mean, we do not face the problem of assigning them a distribution. This clear statistical definition is in contrast to the difficulty of assigning a distribution to modularity. Such difficulty renders formal statistical tests of its significance non-informative, as highlighted by Fortunato in 2010 [14]. Our Kappas are assumed to be distributed about their true value according to a Gaussian distribution, on the basis of the Central Limit Theorem, when sample size (number of clusters) is sufficiently large. Our computational experiments also reveal they remain Gaussian even for smaller sample sizes.

Here, we streamline our statistical test. In our previous article [35], we conducted two separate tests. We formulated two null hypotheses, $\bar{K}_{\text{intra}} = K$ and $\bar{K}_{\text{inter}} = K$, to avoid the effects of a possible correlation between $\bar{K}_{\text{intra}}$ and $\bar{K}_{\text{inter}}$ (K is a graph constant, not the result of a clustering). However, since our ultimate goal is to formally compare intra- and inter-cluster densities, we adapt the standard t-test to overcome any possible correlation and allow for a direct comparison of these graph statistics. This new test verifies that a clustering meets the second necessary condition for being classified as a good clustering. If it does meet this condition, we expect the inequalities $\bar{K}_{\text{intra}} > \bar{K}_{\text{inter}}$ to hold at a reasonable significance level (e.g., $\alpha = 0.95$).

Our test can also be used when comparing two or more algorithms' performances on a given graph. In such cases, in order to conclude algorithm 'a' is better than algorithms 'b', 'c',(. . .), we should observe better (smaller) p-values, $p_a < p_b < p_c < (\ldots)$. Although this procedure is not a formal statistical test, it is a valid and easily applicable heuristic.

Finally, let us note that our statistical definition also allows for uncertainty in the connectivity data, another open problem which was identified by Holder et al. in 2016 [20]. Unlike modularity and conductance, our measures are defined as statistical measurements with associated standard errors, not deterministic quantities.

4.1 Null Hypothesis

Under the null hypothesis, the algorithm is assumed to offer a random assignment of nodes to clusters. When nodes are randomly assigned cluster labels, we expect no significant difference between $\bar{K}_{\text{intra}}$ and $\bar{K}_{\text{inter}}$, as described in Section 3.3.

Here, we note that our null hypothesis does not rely on any generative model for the graph, unlike modularity which uses the configuration model as a null model. In fact, in our approach, the graph is not random, but is fixed. Instead, under the null hypothesis, the clusters are random.

Our significance test is an assessment of the statistical significance of the quality of a clustering returned by an algorithm. We argue that in the case of a good clustering, the gap between intra- and inter-cluster densities should be statistically significant (test 1). We also use the p-values of this test to heuristically compare the quality of the clusterings of a specific graph returned by two or more algorithms, as mentioned earlier (test 2).

Test 1

$H_0: \quad \bar{K}_{\text{intra}} = \bar{K}_{\text{inter}} \Leftrightarrow \gamma = \bar{K}_{\text{intra}} - \bar{K}_{\text{inter}} = 0$

$H_a: \quad \bar{K}_{\text{intra}} > \bar{K}_{\text{inter}} \Leftrightarrow \gamma = \bar{K}_{\text{intra}} - \bar{K}_{\text{inter}} > 0$

Test 2 (heuristic, two algorithm comparistion)

$H_0: \quad \bar{K}^{(1)}_{\text{intra}} = \bar{K}^{(1)}_{\text{inter}} = \quad \bar{K}^{(2)}_{\text{intra}} = \bar{K}^{(2)}_{\text{inter}} \Leftrightarrow \gamma_1 = \gamma_2 = 0$

$H_a: \quad \gamma_1 > \gamma_2$

4.2 Test Statistics

To test the hypotheses in the previous section, we compute the following test statistics, one for assessing the significance of γ (stat test 1) and one for comparing the quality of the clusterings as labeled by two or more clustering algorithms (stat test 2).

- Stat Test 1:

$$t = \frac{\gamma}{s.e.}$$

- Stat Test 2 (heuristic):
  - Compute t_a and t_b, the significance tests (test 1, here above) for algorithms 'a' and 'b' on a given graph
  - Obtain the respective p-values, p_a and p_b
  - If $p_a - p_b > 0$, conclude algorithm 'b' returned a better, more statistically significant, clustering
  - Note this is a heuristic decision tool, not a formal statistical test

The modification we make to the t-test lies in the computation of the standard error ($s.e.$) and in the degrees of freedom of the t statistic. In the classic t-test, the t statistic for Test 1 would be computed as follows.

$$\begin{aligned}
t &= \frac{\gamma}{s_\gamma} \\
s_\gamma &= \sqrt{\frac{s^2_{\text{intra}}}{|C|} + \frac{s^2_{\text{inter}}}{0.5 \times |C| \times (|C|-1)}} \\
s^2_{\text{intra}} &= \frac{1}{|C|-1} \sum_{i=1}^{|C|} \left(\kappa_i - \bar{K}_{\text{intra}}\right)^2 \\
s^2_{\text{inter}} &= \frac{1}{0.5 \times |C| \times (|C|-1) - 1} \sum_{i=1}^{|C|} \sum_{j=i+1}^{|C|} \left(\kappa_{ij} - \bar{K}_{\text{inter}}\right)^2
\end{aligned}$$

Instead, we use Monte-Carlo simulation to compute the t-statistic directly. The steps in this computation are described below:

1. Randomly label nodes as belonging to one of the $|C|$ clusters identified by the algorithm (null hypothesis)
2. Compute $\gamma = \bar{K}_{\text{intra}} - \bar{K}_{\text{inter}}$, under the null hypothesis
3. Repeat $r > 30$ times and compute the variance $Var(\gamma)$
4. Use $s.e. = \sqrt{Var(\gamma)}$
5. Degrees of freedom for the test are given by the number of simulation runs, r, minus one, $d.f. = (r-1)$

In addition to sidestepping the issue of possible dependencies between $\bar{K}_{\text{intra}}$ and $\bar{K}_{\text{inter}}$, the main feature of our modified t-test is that it remains computable even in cases where the standard error cannot be estimated by applying the usual scaling to the variance of the results. For example, in cases where the graph is disconnected, the variance s^2 (and standard error) of $\bar{K}_{\text{inter}}$ may be hard, even impossible, to estimate accurately. By using Monte-Carlo simulation to estimate standard error, we are able to overcome the obstacles posed by such situations.

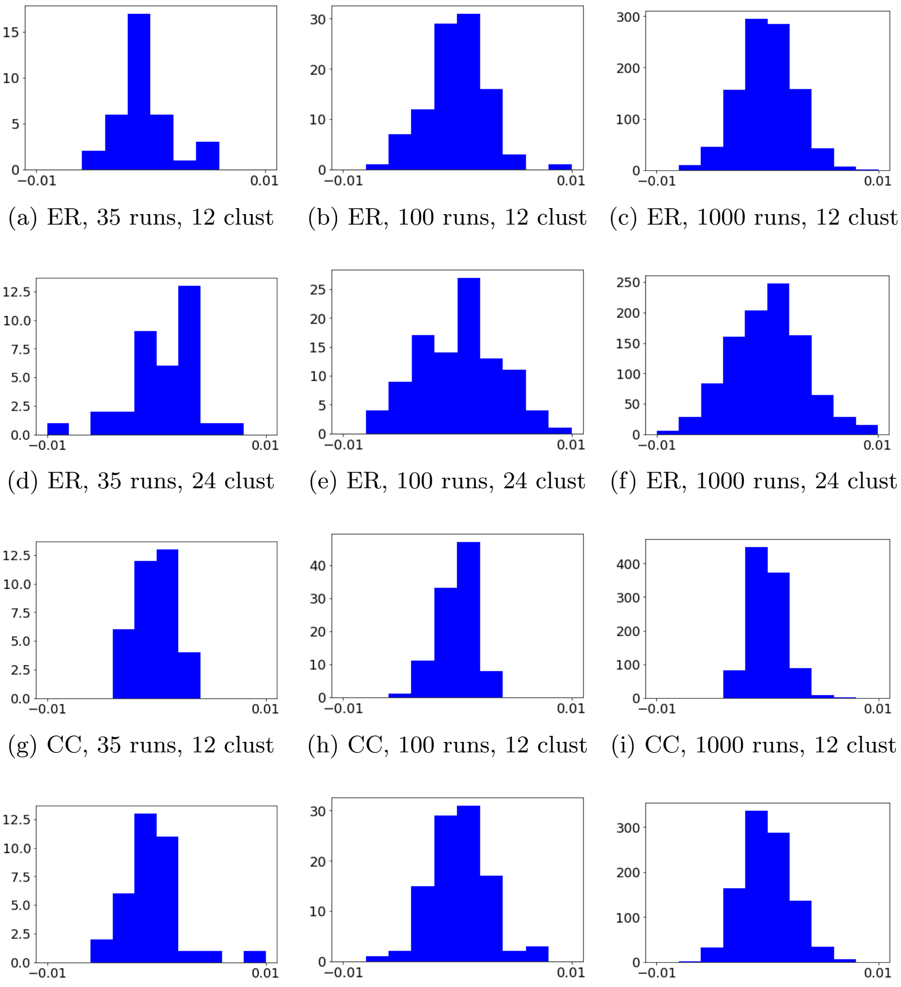
Of course, we could also apply the Monte-Carlo method to obtain p-values directly, without using t-statistics and the Student distribution. Proceeding in this way would not only circumvent any possible dependencies but would also be independent of any distributional assumptions on the null hypothesis. Unfortunately, such a procedure would also be very computationally demanding and could render testing infeasible. Fortunately, as we show in the next section, the Gaussian distribution offers a very good model of our null hypothesis, even when sample sizes (number of clusters) are small ($N < 30$).

4.3 Empirical Examination of the Null Distribution

Under our null hypothesis, our canonical example of poor clustering, the difference $\gamma = \bar{K}_{\text{intra}} - \bar{K}_{\text{inter}}$ has an expected value of zero. It is also approximately Gaussian. Indeed, under the null, cluster labels are assigned to vertices randomly. In this case, the cluster label of a vertex is independent of its connections to other vertices. The symmetry of the distribution of γ stems from the fact it is a difference of two sample means.

To empirically validate our statements about the distribution of γ under the null, we simulate random node labelings on two different synthetic graphs. The first graph is an Erdős-Rényi (ER) graph of $1,000$ vertices and edge probability of $\frac{1}{3}$. The second is a connected caveman (CC) graph of 10 (quasi-)cliques of 100 vertices. Each (quasi-)clique has one edge re-assigned so it connects to one vertex in another cluster. These graphs were chosen, because they lie at either end of the spectrum of structured-unstructured graphs. We then simulate a random vertex labeling of 12 and 24 clusters, repeat r times and compute the means and standard deviations of γ. To complete our comparisons, we examine the sample statistics of our simulated data and its empirical distribution. We also superimpose its percentiles over the percentiles of a Gaussian distribution with the same mean and standard deviation. The results are shown in Table 1, Figures 3 and 4.

The histograms in Figure 3 reveal that γ is roughly symmetrically distributed about its mean of zero, even with a small number of runs. That symmetry becomes more apparent as the number of runs increases. The numerical results in Table 1 confirm its mean is always roughy equal to zero



(a) ER, 35 runs, 12 clust

(b) ER, 100 runs, 12 clust

(c) ER, 1000 runs, 12 clust

(d) ER, 35 runs, 24 clust

(e) ER, 100 runs, 24 clust

(f) ER, 1000 runs, 24 clust

(g) CC, 35 runs, 12 clust

(h) CC, 100 runs, 12 clust

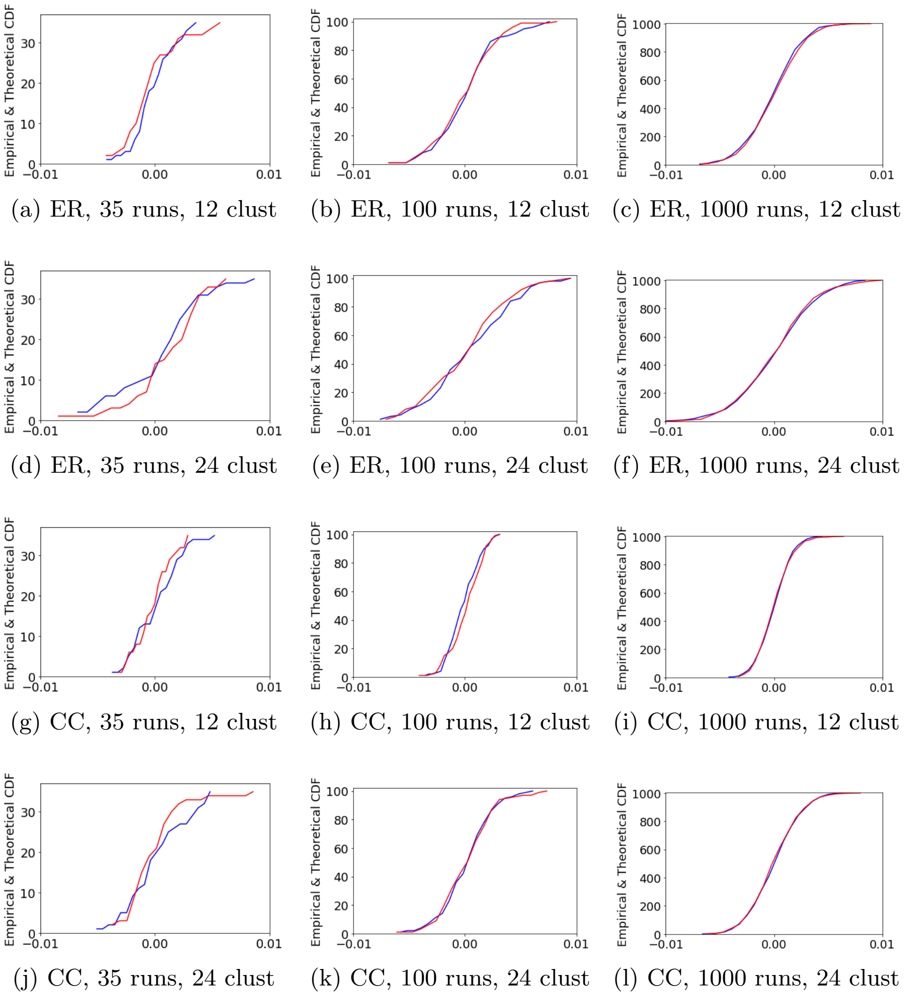
(i) CC, 1000 runs, 12 clust

(j) CC, 35 runs, 24 clust

(k) CC, 100 runs, 24 clust

(l) CC, 1000 runs, 24 clust

Figure 3: Null Distributions of γ, Number of Runs and Number of Clusters



(a) ER, 35 runs, 12 clust

(b) ER, 100 runs, 12 clust

(c) ER, 1000 runs, 12 clust

(d) ER, 35 runs, 24 clust

(e) ER, 100 runs, 24 clust

(f) ER, 1000 runs, 24 clust

(g) CC, 35 runs, 12 clust

(h) CC, 100 runs, 12 clust

(i) CC, 1000 runs, 12 clust

(j) CC, 35 runs, 24 clust

(k) CC, 100 runs, 24 clust

(l) CC, 1000 runs, 24 clust

Figure 4: Null Distributions of γ in Red, Number of Runs and Number of Clusters, Gaussian with Same Mean and Std Dev in Blue

Table 1: Mean and Std Dev of γ Under the Null

Graph	Num Runs	Num Clusters	Mean gamma	Std gamma
ER1000	35	12	-0.0005	0.0023
ER1000	100	12	-0.0001	0.0026
ER1000	1000	12	0.0000	0.0025
ER1000	35	24	0.0010	0.0030
ER1000	100	24	0.0001	0.0035
ER1000	1000	24	0.0001	0.0034
CC1000	35	12	-0.0002	0.0016
CC1000	100	12	0.0001	0.0015
CC1000	1000	12	0.0000	0.0015
CC1000	35	24	-0.0004	0.0024
CC1000	100	24	0.0001	0.0023
CC1000	1000	24	-0.0001	0.0022
Mean			**0.0000**	**0.0024**

and standard deviation also roughly equal to zero. More importantly, our results also confirm the empirical percentiles of simulated γ and those of a Gaussian with the same mean and standard deviation approximately coincide, especially when the number of runs is large. This coincidence can be seen in Figure 4, where the percentiles of γ converge towards theoretical Gaussian percentiles, as the number of runs increases. It should be noted that cases where the histograms appear skewed are due to the relatively small number of runs with respect to the number of clusters.

In summary, our node labelling simulations reveal that even a very small number of runs (e.g., 35 runs) can offer a very accurate estimate of the standard deviation under the null, of the standard error of our modified t-statistic. They also confirm that under the null γ has an expected value of zero. However, the most interesting result is that under the null γ remains approximately Gaussian, even in instances where the cluster number is small (12 and 24) and well below the usual sample size required to invoke the Central Limit Theorem ($N \geq 30$).

4.4 Scalability of the Modified t-test

With the mass of large datasets that are now commonly studied, it is important to consider the scalability of any clustering performance measurement technique. Indeed, any test that cannot be applied to larger graphs is not suited to the emerging area of complex networks. However, because of the stability of the null and its Gaussian distribution, our performance variables and associated test statistics are indeed applicable to such data sets.

In particular, the stability of the standard distribution of the null eases the estimation of our t-statistic's standard error, since its computation doesn't require a large number of simulation runs. More importantly, it also allows us to use a modified t-statistic to estimate p-values, instead of resorting to costly direct empirical Monte-Carlo estimation. Additionally, it should be noted that computing $\bar{K}_{\text{intra}}$ and $\bar{K}_{\text{inter}}$ can be obtained by computing each cluster's internal density and each cluster pairs' common density separately. Breaking up computations in such a way allows for easy parallelization.

5 Axioms for a Good Clustering Quality Function

In this section, we review the axioms that define a good clustering quality function and describe how our Kappas meet these axioms. Multiple authors have presented axioms defining a good clustering quality function [25, 1, 26, 24]. Although these publications use different terminology, their axioms share common features and are rooted in the seminal work of Kleinberg [25].

We combine the recent work of Van Laarhoven and Marchiori [26] and the more recent work of Kehagias and Pitsoulis [24] to draw a list of axioms defining the desirable properties of a clustering quality function for graphs. Our combined list of axioms contains ten axioms which characterize a good clustering quality function.

5.1 Axioms of Van Laarhoven and Marchiori

Most of the axioms that a good clustering quality function must meet are presented by Van Laarhoven and Marchiori [26] and are listed below. In their work, these authors also prove modularity fails to meet two of these axioms, locality and monotonicity.

1. **Permutation Invariance**: Clustering quality should not depend on specific labels. Instead, it should depend on global labeling structure. (*"isomorphism invariance"* in Kehagias and Pitsoulis [24])

2. **Scale Invariance**: Clustering quality should remain unaffected when edge weights are scaled uniformly. Mathematically, this means that for a graph G, any clusterings (C_i, C_j), constant $\alpha > 0$ and given quality

function $Q(G, C)$, the following equivalence holds:

$$Q(G, C_i) \leq Q(G, C_j) \Leftrightarrow Q(\alpha G, C_i) \leq Q(\alpha G, C_j) \, .$$

While this axiom appears to apply mostly to weighted graphs, the idea behind it can be generalized to unweighted graphs. In the case of unweighted graphs, this axiom can be understood as meaning the equivalence above must hold when edge probabilities are uniformly scaled.

3. **Richness**: Given a finite set of vertices V it is possible to re-arrange edges (or edge weights) to obtain an optimal partition.

4. **Monotonicity and Consistent Improvement**: A "consistent improvement" means an increase in edge density (edge probability) within clusters or a decrease in inter-cluster edge density (edge probability) should not decrease the quality function.

5. **Locality**:

   (a) Although Ackerman and Ben-David [1] have also defined locality in a very similar manner, we prefer the definition of Van Laarhoeven and Marchiori [26].

   (b) The latter is more flexible and does not rely on the assumption the number of clusters is known in advance. After all, when we are evaluating the performance of an algorithm, we do not want to impose a fixed number of clusters. First, in most cases, we don't know this number. Second, we also want to assess the validity of the number of clusters fed into or identified by the clustering algorithm under examination.

   (c) We use the definition that follows: "(...) the contribution of a single cluster to the total quality should only depend on nodes in the neighborhood of that cluster (...) On the other hand, a quality function that is written as a sum over clusters, where each summand depends only on properties of nodes and edges in one cluster and not on global properties, is local." [26].

6. **Continuity**: Here, the authors say "A quality function Q is continuous if a small change in the graph leads to a small change in the quality." Essentially, this property ensures quality functions remain robust to noise in the graph structure.

7. **Resolution-limit Free**: The partition remains optimal for any induced subgraph of the optimal partition. It was described as “the limitations in detecting small community structures in a large network”, by McSweeney et al. [34]. In essence, a resolution-limit free quality function’s output is not disproportionately influenced by larger clusters to a point where the effect of smaller clusters is washed away and hidden.

5.2 Additional Axioms from Kehagias and Pitsoulis

Many of the axioms of Kehagias and Pitsoulis [24] are similar to those of Van Laarhoven and Marchiori [26]. In this section, we include some additional ones which are specific to the work of Kehagias and Pitsoulis.

7. **Perfectness**: “(...) is based on the intuition that a union of disjoint complete graphs should exhibit perfect community structure”

8. **Connectivity**: “(...) is based on the intuition that a minimum requirement for a cluster to be classified as a community is that the associated induced subgraph should be connected”

9. **Complementarity**: Let Q be a uniformly scaled quality function on the interval $[0, 1]$ and G^c the complement graph of G, then if Q is complementary, the following holds

$$Q(G, C) = 1 - Q(G^c, C)\,.$$

5.3 Axioms of Good Clustering Applied to the Kappas and Its Accompanying Tests

In this section, we apply the axioms of a good clustering quality function to our Kappas and their significance test. We describe how they meet all ten axioms we just listed.

1. **Permutation Invariance**: Cluster labels are only used to aggregate edge and vertex counts. A label permutation, swapping all node cluster labels between an arbitrary number of pairs of clusters, does not affect edge or vertex counts. Therefore, our estimates of intra- or inter-cluster density or the graph’s global density remain unaffected.

2. **Scale Invariance**: In the context of our tests, scale invariance must hold on two levels. Not only must the relative differences in the Kappas remain unaffected by the scaling, but the test statistics must also

remain unaffected. A multiplication of edge probabilities or weights by a constant $\alpha > 0$ does not affect the relative differences in the Kappas, the t-test statistics and their degrees of freedom. Consequently, the conclusions of our tests are also unaffected. A full proof of this statement is trivial but lengthy. Essentially, it is based on the fact that our Kappas are non-negative numbers and the fact standard error scales linearly. Therefore, a multiplication by a non-zero positive constant does not affect the inequalities:

$$\frac{a}{b} \lesseqgtr \frac{c}{d} \Leftrightarrow \frac{\alpha a}{\alpha b} \lesseqgtr \frac{\alpha c}{\alpha d}\,.$$

3. **Richness**: This property means that the optimum is an achievable quantity. It is obvious that increasing intra-cluster density or decreasing inter-cluster density improves our test statistics' value and implies an improvement of clustering. In other words, the better the clustering, the greater the gap between global, intra- and inter-cluster density, the higher the t-statistic and the lower the p-value. For example, if we increase intra-cluster density and decrease inter-cluster density sufficiently, intra-cluster density will eventually reach its maximum value of one (case of unweighted graphs), while inter-cluster will reach its minimum value of zero (also unweighted graph case). Their difference, the quantity γ, will also reach its maximum value of one. The significancel test statistics will also reach their optima (i.e., minimum p-value), if we modify densities sufficiently.

4. **Monotonicity and Consistent Improvement**: This property is a consequence of the previous one.

5. **Locality**: Our quality measurement quantities are means. They are sums over clusters or cluster pairs scaled by a constant. Each summand depends exclusively on its cluster or cluster pair. Our tests meet this property by definition: "(..) a quality function that is written as a sum over clusters, where each summand depends only on properties of nodes and edges in one cluster and not on global properties, is local." [26].

6. **Continuity**: Continuity is a property of the mean, which provides a smoothed summary of a data set. A small improvement in clustering cannot result in a large fluctuation in the inter- or intra-cluster means, by construction. For example, a higher intra-cluster density within

one or a few clusters cannot have a drastic effect on our graph-wide quantities.

7. **Perfectness**: The union of disjoint complete graphs would indeed have perfect scores. In such a case, each cluster's intra-cluster density would be equal to one and all inter-cluster densities would be qual to zero, by definition.

$$\bar{K}_{\text{intra}} = \frac{1}{|C|}\underbrace{\sum 1}_{|C|\times 1} = 1 > K > \bar{K}_{\text{inter}} = \frac{1}{0.5 \times |C|\,(|C|-1)}\sum 0 = 0$$

Both necessary conditions would be met. The inequalities between global, intra- and inter-cluster densities would hold and γ would reach its maximum value. The statistical test would classify the resulting γ as significantly different from zero with a p-value approaching 0:

$$t = \frac{(1-0)}{s.e.}, \text{ with } s.e \lll 0$$

8. **Connectivity**: A set of disjoint vertices v_i would have internal density

$$\kappa_i = 0 \leq K, \bar{K}_{\text{inter}}$$

and would not meet our test for being a valid cluster. Such a set would trivially be classified as degenerate.

9. **Complementarity**: Both our test statistics have number of edges in their numerators.

$$\kappa_i = \frac{|E_{ii}|}{0.5 \times n_i(n_i-1)},\ \kappa_{ij} = \frac{|E_{ij}|}{n_i n_j}$$

$$\begin{aligned}
\kappa_{i^c} &= \frac{|E_{i^c i^c}|}{0.5 \times n_i(n_i-1)} \\
&= \frac{0.5 \times n_i(n_i-1) - m_{ii}}{0.5 \times n_i(n_i-1)} \\
&= 1 - \frac{m_{ii}}{0.5 \times n_i(n_i-1)} = 1 - \kappa_i \\
\frac{1}{|C|}\sum_{i=1}^{|C|}(1-\kappa_i) &= \frac{1}{|C|}\left(|C| - \sum_{i=1}^{|C|}\kappa_i\right) \\
&= 1 - \frac{1}{|C|}\sum_{i=1}^{|C|}\kappa_i = 1 - \bar{K}_{\text{intra}} \ \square
\end{aligned}$$

Let i^c, j^c be the complement of the bi-clique formed by clusters i, j, $|E_{ij}| = m_{ij}$ be the number of edges with one end in each cluster, n_i be the number of nodes in cluster i and the total number of cluster-cluster pairs be denoted as $N = 0.5 \times |C|\,(|C| - 1)$.

$$\begin{aligned}
\kappa_{i^c j^c} &= \frac{|E_{i^c j^c}|}{n_i n_j} \\
&= \frac{n_i n_j - m_{ij}}{n_i n_j} = 1 - \frac{m_{ij}}{n_i n_j} \\
\frac{1}{N}\left(\sum_{i,j=i+1}^{N} (1 - \kappa_{ij})\right) &= \frac{1}{N}\left(N - \sum_{i,j=1+1}^{N} \kappa_{ij}\right) \\
&= 1 - \frac{1}{N}\sum_{i,j=1+1}^{N} \kappa_{ij} = 1 - \bar{K}_{\text{inter}} \;\Box
\end{aligned}$$

10. **Resolution-limit Free**: If our first necessary condition (set of inequalities) is met, $\bar{K}_{\text{inter}} < K < \bar{K}_{\text{intra}}$, then we also expect $E(\kappa_{ij}) < K < E(\kappa_i)$ for all clusters i and cluster pairs i, j. If our second necessary condition is met and we reject the null that $\bar{K}_{\text{intra}} = \bar{K}_{\text{inter}}$, then the corresponding null on subset statistics, $\bar{K}^{(s)}_{\text{intra}} = \bar{K}^{(s)}_{\text{inter}}$, will tend to be rejected as well.

    More concretely, it is important to note that neither $\bar{K}_{\text{intra}}$ nor $\bar{K}_{\text{inter}}$ are affected by individual cluster size relative to network size. They do not suffer from the resolution limit observed in modularity [15, 23, 11, 16]. Very large (very small) clusters do not influence their values more than smaller (larger) ones, as with modularity. All terms in the sums are scaled by the total number of possible edges within each cluster or pair of clusters, which ensures they remain within the same order of magnitude regardless of size. In the case of unweighted graphs, they always lie on the $[0, 1]$ interval. In the weighted case, they are always proportional to edge weight. This feature makes these measures robust to large "mega-clusters" that are often observed in real-world networks and to the problematic tendency of clustering algorithms to lump all vertices together in a few very large clusters [16, 39]. (Naturally, K is a graph-wide measure that remains completely unaffected by cluster labelings and individual cluster sizes.)

6 Computational Experiments

To empirically compare each competing performance measure's accuracy and responsiveness to various graph structures and cluster labelings, we subject them to a number of numerical stress test scenarios. We use simulated graphs and cluster labels. The full experimental set-up of our tests and scenario details are described in the next section.

Overall, our goal is to test the accuracy and robustness of our clustering measures and compare their behavior to that of the two main clustering measures in the literature, modularity and conductance. Simulation is used to generate test scenarios where the clustering structure is known in advance and can be modified easily. These test scenarios are then used to examine and compare the sensitivities of the Kappas, modularity and conductance. Our scenarios include a number of contrived instances, but these are useful to stress test our metrics through extreme degenerate examples and compare their behavior to those of the more established measures.

The overarching logic guiding our tests is that a good measure of inter- or intra- cluster connectivity should accurately reflect the simulated graph's structures. We expect measures of intra-cluster connectivity, $\bar{K}_{\text{intra}}$ and modularity to increase in step with the simulated graph's intra-cluster connectivity levels, by definition. Meanwhile we expect conductance to display the inverse behavior. We also expect $\bar{K}_{\text{inter}}$ to follow the fluctuations of inter-cluster connectivity, by construction.

It should also be mentioned that some authors have used so-called "ground-truth" data sets, as benchmarks for clustering algorithm performance (e.g., [51, 36, 50]). These are data sets where the nodes' cluster memberships are known in advance. Our approach is more general, since it is data set independent. Arguably, the fact that an algorithm anecdotally provided accurate clustering on one labeled instance is no guarantee it will perform equally well on another (likely unlabeled) instance. In addition, our experiments provide us with an understanding of each measure's sensitivity and response to graph structure.

6.1 Experimental Set-up and Results

We experiment with variations in edge probability, both within and between clusters. Here, we slightly modify the procedure to generate inter-cluster edges. In our previous article [35], we varied the proportion of vertices inside and outside each cluster that shared an edge. Here, we vary edge probabilities.

Our data generation process consists of modifying intra-cluster and inter-cluster edge probabilities of a planted partition model [14] and generating graphs with clusters of varying sizes. We begin with increases in intra-cluster connectivity in steps of 25%, while maintaining inter-cluster edge probability at 0%. For example, in the second column of Table 2, approximately 25% of all possible edges within a cluster are added, but nodes only have connections to other nodes within their assigned cluster. Each cluster remains a connected component disconnected from the rest of the graph.

We conduct these tests with unweighted graphs, but also repeat them with weighted ones. While there are no formal and universally accepted definitions of weighted stochastic block models or planted partition models, we generate data that is consistent with the logic of the planted partition model. In the weighted case, the intra-cluster edge probability also corresponds to edge weight. For example, when edge probability is 25%, edge weight is also set to 0.25. All edge weights are between 0 and 1.

We then examine the effect of inter-cluster connectivity on each measure. We begin with no inter-cluster connectivity and then increase it in steps of 25%. We increase edge probability between nodes in different clusters in steps of 25%, while keeping intra-cluster connectivity at 0%. In other words, clusters are just sets of non-adjacent vertices. In these scenarios, we imagine an algorithm, a very poorly performing one, that groups non-adjacent vertices into clusters with different levels of inter-connection to other clusters but with an intra-cluster connectivity that remains constant at 0%. Here again, we also repeat our tests on weighted graphs, with edge weights corresponding to the inter-cluster connectivity percentage. Results are shown in Table 3.

We acknowledge these synthetic networks are unrealistic. Our goal is not to study common network structures. Our goal is to examine the effects of drastic degenerate structures on our quality measures and compare them to the effects on the competing measures.

Finally, in order to assess our measures' robustness, we repeat all the tests described above, but with the introduction of noise in the connectivity patterns. Noise is introduced in the form of 100% intra-(inter-) cluster connectivity (edge probability). Results are shown in Table 4 and Table 5.

6.2 Interpretation of Empirical Comparisons

As shown in Section 6.1, our Kappas behave exactly as expected, even when subjected to noise. In all instances where the labeling of clusters reflects a good partition, the inequalities $\bar{K}_{\text{intra}} > K > \bar{K}_{\text{inter}}$ hold and they do not

Table 2: Varying Intra-Cluster Connectivity, No Noise from Inter-Cluster Connectivity

Pct Inter = 0, Pct Intra varies					
Pct Intra	0	25	50	75	100
UNWEIGHTED					
N	10,048	9,725	10,374	9,490	9,700
$\|C\|$	200	200	200	200	200
$\|E\|$	0	77,043	173,221	224,723	313,955
K	0.0000	0.0016	0.0032	0.0050	0.0067
$\bar{K}_{\text{intra}}$	0.0000	0.2640	0.4995	0.7523	0.9900
$\bar{K}_{\text{inter}}$	0.0000	0.0000	0.0000	0.0000	0.0000
Φ	nan	0.0000	0.0000	0.0000	0.0000
Q	nan	0.9908	0.9911	0.9906	0.9907
WEIGHTED					
N	10,048	9,725	10,374	9,490	9,700
$\|C\|$	200	200	200	200	200
$\|E\|$	0	19,261	86,611	168,542	313,955
K	0.0000	0.0004	0.0016	0.0037	0.0067
$\bar{K}_{\text{intra}}$	0.0000	0.0660	0.2497	0.5642	0.9900
$\bar{K}_{\text{inter}}$	0.0000	0.0000	0.0000	0.0000	0.0000
Φ	nan	0.0000	0.0000	0.0000	0.0000
Q	nan	0.9908	0.9911	0.9906	0.9907

Table 3: Varying Inter-Cluster Connectivity, No Noise from Intra-Cluster Connectivity

Pct Intra = 0, Pct Inter varies					
Pct Inter	0	25	50	75	100
UNWEIGHTED					
N	10,048	10,048	10,048	10,048	10,048
$\|C\|$	200	200	200	200	200
$\|E\|$	0	12,210,800	24,864,800	37,291,600	50,142,500
K	0.0000	0.2419	0.4926	0.7388	0.9934
$\bar{K}_{\text{intra}}$	0.0000	0.0000	0.0000	0.0000	0.0000
$\bar{K}_{\text{inter}}$	0.0000	0.2366	0.4907	0.7336	1.0000
Φ	nan	1.0000	1.0000	1.0000	1.0000
Q	nan	-0.0067	-0.0067	-0.0067	-0.0067
WEIGHTED					
N	10,048	10,048	10,048	10,048	10,048
$\|C\|$	200	200	200	200	200
$\|E\|$	0	3,052,700	12,432,400	27,968,700	50,142,500
K	0.0000	0.0605	0.2463	0.5541	0.9934
$\bar{K}_{\text{intra}}$	0.0000	0.0000	0.0000	0.0000	0.0000
$\bar{K}_{\text{inter}}$	0.0000	0.0592	0.2454	0.5502	1.0000
Φ	nan	1.0000	1.0000	1.0000	1.0000
Q	nan	-0.0067	-0.0067	-0.0067	-0.0067

Table 4: Varying Intra-Cluster Connectivity, with Noise from Inter-Cluster Connectivity

Pct Inter = 100, Pct Intra varies					
Pct Intra	0	25	50	75	100
UNWEIGHTED					
N	10,048	9,725	10,374	9,490	9,700
$\|C\|$	200	200	200	200	200
$\|E\|$	50,142,500	47,052,100	53,631,600	44,950,500	47,040,200
K	0.9934	0.9951	0.9968	0.9983	1.0000
$\bar{K}_{\text{intra}}$	0.0000	0.2640	0.4995	0.7523	0.9900
$\bar{K}_{\text{inter}}$	1.0000	1.0000	1.0000	1.0000	1.0000
Φ	1.0000	0.9974	0.9952	0.9922	0.9899
Q	-0.0067	-0.0050	-0.0033	-0.0018	-0.0001
WEIGHTED					
N	10,048	9,725	10,374	9,490	9,700
$\|C\|$	200	200	200	200	200
$\|E\|$	50,142,500	46,994,300	53,545,000	44,894,300	47,040,200
K	0.9934	0.9939	0.9952	0.9971	1.0000
$\bar{K}_{\text{intra}}$	0.0000	0.0660	0.2497	0.5642	0.9900
$\bar{K}_{\text{inter}}$	1.0000	1.0000	1.0000	1.0000	1.0000
Φ	1.0000	0.9994	0.9976	0.9941	0.9899
Q	-0.0067	-0.0062	-0.0049	-0.0030	-0.0001

Table 5: Varying Inter-Cluster Connectivity, with Noise from Intra-Cluster Connectivity

Pct Intra = 100, Pct Inter varies					
Pct Inter	0	25	50	75	100
UNWEIGHTED					
N	9,700	9,700	9,700	9,700	9,700
$\|C\|$	200	200	200	200	200
$\|E\|$	313,955	11,612,400	23,403,200	34,964,100	47,040,200
K	0.0067	0.2469	0.4975	0.7433	1.0000
$\bar{K}_{\text{intra}}$	0.9900	0.9900	0.9900	0.9900	0.9900
$\bar{K}_{\text{inter}}$	0.0000	0.2294	0.4842	0.7271	1.0000
Φ	0.0000	0.9595	0.9798	0.9864	0.9899
Q	0.9907	0.0202	0.0066	0.0022	-0.0001
WEIGHTED					
N	9,700	9,700	9,700	9,700	9,700
$\|C\|$	200	200	200	200	200
$\|E\|$	313,955	3,138,570	11,858,600	26,301,500	47,040,200
K	0.0067	0.0667	0.2521	0.5591	1.0000
$\bar{K}_{\text{intra}}$	0.9900	0.9900	0.9900	0.9900	0.9900
$\bar{K}_{\text{inter}}$	0.0000	0.0574	0.2421	0.5453	1.0000
Φ	0.0000	0.8556	0.9603	0.9820	0.9899
Q	0.9907	0.0930	0.0196	0.0051	-0.0001

hold in instances where the partition reflects poor clustering. For example, in Table 4, all instances are cases of poor clustering. Similarly, in Table 5, instances where the percentage of inter-cluster connectivity is below 75% are examples of good clustering. Decimal mismatches with the expected edge probabilities are due to rounding in the sampling procedure. In contrast, we note modularity and conductance display very counterintuitive behaviors.

The case of modularity is of particular interest, given it is the most widely used clustering quality measure. To illustrate the breakdown of modularity observed in Tables 2-5, we use a hypothetical example of an unweighted graph with k clusters all containing the same number of vertices ($|v_i| = v$, $\forall i$) and with equal inter- and intra-cluster edge probability for all individual clusters and cluster pairs.

6.2.1 Case 1 (Table 2), varying intra-cluster edge probability, with inter-cluster edge probability of zero:

In this case, inter-cluster edge probability is zero and intra-cluster edge probability is denoted by p (≥ 0). In this case, modularity is approximated as

$$\begin{aligned} Q &= \sum_{i=1}^{k} \left(\underbrace{e_{ii} - a_i^2}_{q_i} \right) \\ &\approx k \times \left[e_{11} - a_1^2 \right] . \end{aligned}$$

If $p = 0$, modularity is undefined, because $m = 0$. If we let $p > 0$, since we have only intra-cluster edges, a constant edge probability and number of vertices for all clusters, all e_{ii} and all a_i are not only equal within a given cluster. They also take on the same approximate value, across all clusters:

$$\begin{aligned} e_{ii} = a_i &= \frac{1}{2m} \times 2 \times |E_{ii}| \quad \forall i \in C \\ |E_{ii}| &\approx p \times 0.5 \times v \times (v-1) \approx \frac{m}{k} \quad \forall i \in C \\ \sum_{i=1}^{k} |E_{ii}| &= m . \end{aligned}$$

As soon as p increases to any non-zero value, modularity reaches its maximum and remains unaffected by variations in connectivity, as shown here.

$$\begin{aligned} Q &\approx k \times \left[\frac{1}{2m}\frac{2m}{k} - \frac{1}{4m^2}\frac{4m^2}{k^2}\right] \\ &\approx 1 - \frac{1}{k} \approx 1 \quad \text{(if } k \text{ is large)} \end{aligned}$$

6.2.2 Case 2 (Table 3), varying inter-cluster edge probability, intra-cluster edge probability of zero:

Here, since there are no intra-cluster edges, we have $e_{ii} = 0$ for each cluster i. Since the inter-cluster probability is uniform across all cluster pairs (p) and intra-cluster edge probability is zero, the approximate number of connections (edge stubs) originating from each cluster is $s_i \approx (p \times v \times (k-1) \times v) \approx \frac{2m}{k}$. The approximate number of stubs is equal to the probability of inter-cluster edge multiplied by the total number of possible inter-cluster connections originating in cluster i and connecting to vertices in all other remaining $(k-1)$ clusters. Here, we note there are v vertices in each cluster i and $(k-1)v$ vertices in the remaining clusters. Summed over the graph, it corresponds to two times the number of inter-cluster edges originating in each cluster:

$$\sum_{i=1}^{k} s_i = 2m, \; s_i \approx \frac{2m}{k} \; .$$

Modularity with no intra-cluster edges but a probability of inter-cluster edge p then becomes approximately:

$$\begin{aligned} Q &\approx k \times \left[0 - \frac{1}{4m^2}\left(p \times v \times (k-1) \times v\right)^2\right] \\ &\approx k \times \left[-\frac{1}{4m^2}\left(\underbrace{p \times v \times (k-1) \times v}_{s_i}\right)^2\right] \\ &\approx k \times \left[-\frac{1}{4m^2}\left(\frac{2m}{k}\right)^2\right] = -\frac{1}{k} \\ &\approx 0 \quad \text{(if } k \text{ is large)}\,. \end{aligned}$$

Here again, modularity remains fixed, this time around zero, and unaffected by connectivity patterns.

6.2.3 Case 3 (Table 4), varying intra-cluster edge probability, inter-cluster edge probability of 1:

In Table 4, we begin our experiment with an intra-cluster edge probability of zero. This situation takes us back to the hypothetical example described in Case 2. In that example, modularity was approximately equal to zero.

Increasing intra-cluster edge probability has only a very minor effect on modularity because there are far more inter-cluster edges than intra-cluster ones. At each step, the denominator, number of edges, goes up minimally as the number of intra-cluster edges increases. The numerator also increases, but it remains smaller than the denominator. For example, increasing intra-cluster edge probability from zero to some p (> 0) results in the changes below. In this example, m is the number of edges when intra-cluster edge probability is zero.

$$\begin{aligned}
&e_{ii} = \frac{p \times v \times (v-1)}{2m + p \times v \times (v-1)} \approx \epsilon_e \\
&a_i^2 = \frac{\left(\frac{2m}{k} + p \times v \times (v-1)\right)^2}{\left(2m + p \times v \times (v-1)\right)^2} \approx \epsilon_a \\
&p \times v \times (v-1)^2 \ll 2m \\
&\epsilon_e - \epsilon_a = \epsilon \ \sim 0 \\
&k \times \epsilon \sim 0
\end{aligned}$$

As a result, modularity only increases in a trivial manner and remains at a value of approximately zero.

6.2.4 Case 4 (Table 5), varying inter-cluster edge probability, intra-cluster edge probability of 1:

In Table 5, modularity begins at a value approximately equal to one. This graph structure is the same as the last example in Table 2, where intra-cluster edge probability is equal to one and inter-cluster edge probability is equal to zero.

When inter-cluster edge probability increases to $p > 0$, the numerator in the e_{ii} remain constant, since intra-cluster edge probability is one. However the denominator increases dramatically and decreases the e_{ii} very drastically. Let m denote the number of edges with inter-cluster edge probability of zero. When inter-cluster edge probability increases to $p > 0$, the number of edges becomes

$$|E| = m + k \times \left(\frac{1}{2} \times p \times v \times [(k-1) \times v]\right) .$$

In this case, each cluster adds $\frac{1}{2} \times p \times v \times (k-1) \times v$ edges. This increase in the number of edges causes e_{ii} to collapse, as soon as $p > 0$.

$$e_{ii} = \frac{v \times (v-1)}{2m + (k \times p \times v \times [(k-1) \times v])} \approx 0$$

Because,

$$v \times (v-1) \ll 2m + (k \times p \times v \times [(k-1) \times v])$$

Meanwhile, when inter-cluster edge probability increases to $p > 0$, a_i^2 increases slightly.

$$a_i^2 = \frac{[v \times (v-1) + p \times v \times (k-1) \times v]^2}{[2m + k \times p \times v \times (k-1) \times v]^2} = \epsilon(\gtrsim e_{ii})$$

6.2.5 Link to Axioms

The axioms listed in Section 5, which define the properties of good clustering quality functions, may offer some clues to interpret modularity's counter-intuitive results (Tables 2-5). While modularity does not violate the axioms of richness, monotonicity and consistent improvement and perfectness, it only barely meets them. It also appears to violate continuity, in our experiments.

Richness essentially means the optimum must be achievable through a re-arrangement of edges. Although not defined as such, we would also expect that a "rich" function's minimum also be achievable, in routine cases, not just extremely rare degenerate trivial cases. However, the minimum for modularity, $Q = -\frac{1}{2}$, is only achievable in very rare degenerate and trivially identifiable cases. This fact is documented by Brandes et al. [6]. Van Mieghen et al. also corroborate this claim about the lower bound, when they state *"In conclusion, the modularity of any graph is never smaller than* $-\frac{1}{2}$ *, and this minimum is obtained for the complete bipartite graph"* [48]. Empirically, this weak realization of richness can be seen in all cases where intra-cluster edge probability is zero. Arguably, a good quality function should be at its minimum, when intra-cluster edge probability is zero. Unfortunately, in all of our tests, modularity never reaches its minimum, regardless of how poor the clustering. These cases can be seen in the first column of Table 2 where modularity (Q) is undefined, in Table 3 where Q is either undefined or roughly equal to zero and in the first column of Table 4.

Once again, while it does not violate the axiom of monotonicity and consistent improvement modularity only meets it weakly. Monotonicity and consistent improvement mean that an increase in edge density within clusters or decrease in inter-cluster edge density should not decrease the quality function. While modularity does not decrease with improvements in clustering, it rapidly meets its maximum after trivial improvements and then fails to accurately reflect any further improvements. This inconsistent behavior can be seen in Table 2 and Table 4 where modularity meets its maximum of one, with an intra-cluster edge probability of 25% and then fails to reflect any additional increase in intra-cluster edge probability.

Continuity is an axiom that is clearly violated by modularity. According to Van Laarhoven and Marchiori, “A quality function Q is continuous if a small change in the graph leads to a small change in the quality”. However, as mentioned previously, modularity spikes from a value of zero to its maximum of one, after only a modest improvement in clustering, an increase of 25% intra-cluster edge probability.

Perfectness is another axiom that is only barely met by modularity. While a set of disjoint complete graphs does indeed get a “perfect” modularity score, the reverse is not true. We would expect a “perfect” clustering quality function to reach its minimum when clustered vertices form a complete k-partite graph (where $k > 2$). Unfortunately, modularity does not reach its minimum in such cases. In our experiments, modularity of such poorly clustered graphs is shown in the fifth column of Table 3 and the first column of Table 4.

7 Illustrative Example: Comparing Clustering Quality of the Louvain and Asynchronous Label Propagation Algorithms

To illustrate the application of each step of our work, we test the Louvain [5] and the Asynchronous Label Propagation (ALP) [43] algorithms, as implemented in the Networkx library’s [18] “Communities” module. We use these algorithms to cluster the SNAP “email-Eu-core network” graph [30, 52], which we converted into an undirected graph with no self-loops (EUC).

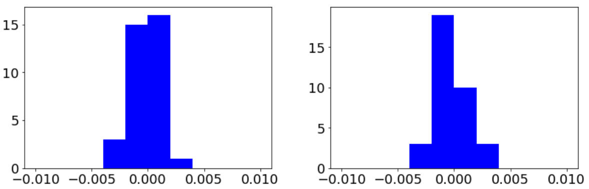
Table 6: Key Characteristics of the Graph and Simulations of the Null Hypotheses

Graph	**Num Vertices**	**Num Edges**	**Num Clusters**	**Num Runs**	**Mean γ**	**Std γ**
EUC	1,005	16,064	20	35	-0.0001	0.0013
EUC	1,005	16,064	27	35	-0.0002	0.0014

To obtain standard errors and to verify our claims about the null distribution in small sample (number of clusters) cases, once more, we simulate 35 instances of random cluster assignments. The number of clusters used to simulate the null distribution is the same as the number of clusters identified by each of the two clustering algorithms. Sample statistics of the null and graph characteristics are shown in Table 6. Corresponding histograms are shown in Figure 5. Full statistical test results are shown in Table 7.

Table 7: Clustering Results: Test Statistics

Algorithm	**Num Clusters**	$\bar{K}_{\textbf{intra}}$	$\bar{K}_{\textbf{inter}}$	**K**	γ	**t-stat**	**df**	**p-value**
ALP	20	0.0017	0.0000	0.0318	0.0017	1.3077	34	≈ 0.1
Louvain	27	0.0513	0.0011	0.0318	0.0502	35.8571	34	≈ 0.0



(a) EUC, 35 runs, 20 clust (b) EUC, 35 runs, 27 clust

Figure 5: Null Distributions of γ, EUC Graph

7.1 Analysis

In the case of the Louvain algorithm, we begin by observing that the inequalities $\bar{K}_{\text{intra}} > K > \bar{K}_{\text{inter}}$ hold numerically. To pursue our analysis with the second necessary condition, we first verify that the null distribution is centered at zero and roughly symmetric. We then compute our statistical test and observe the hypothesis that $\bar{K}_{\text{intra}} = \bar{K}_{\text{inter}}$ is rejected. We conclude the clustering returned by the Louvain algorithm is of good quality and statistically significant.

Meanwhile, for the ALP algorithm, we see that the inequalities $\bar{K}_{\text{intra}} > K > \bar{K}_{\text{inter}}$ do not hold numerically. In fact, mean intra-cluster density is lower than global density, $\bar{K}_{\text{intra}} < K$. Our analysis could end here, with a conclusion that the ALP algorithm offers a very poor clustering.

Notwithstanding this violation of our first necessary condition, we pursue our analysis for illustration purpose. Our significance test reveals the null hypothesis that $\bar{K}_{\text{intra}} = \bar{K}_{\text{inter}}$ is not rejected, at a confidence level of approximately 0.1. For these reasons, we conclude the clustering returned by the ALP algorithm is of poor quality.

Finally, although it is unnecessary in this specific case, we apply our two-algorithm heuristic test. We examine the difference

$$p_{\text{ALP}} - p_{\text{Louvain}} \approx 0.10 \ (\gg 0)$$

and obtain further evidence the Louvain algorithm identifies more meaningful clusters than the ALP algorithm.

The case of the ALP algorithm also illustrates the value of the standard error estimation technique in our modified t-test. The ALP algorithm lumps 986 of the 1,005 vertices into one single cluster. All remaining clusters, 19 of them are isolated vertices which have no edges connecting them to the rest of the graph. In this specific case, the standard error for the mean inter-cluster density, $\bar{K}_{\text{inter}}$, can only be computed through our Monte-Carlo technique. Because there are no inter-cluster edges in this clustering, the standard deviation for inter-cluster density cannot be estimated. Consequently it would be impossible to compute the standard error for $\bar{K}_{\text{inter}}$ in the usual way. Fortunately, our modified t-test allows us to overcome this obstacle and compute a t-statistic and p-value.

8 Conclusion

We described a new set of statistically-rooted clustering quality measures that allow formal clustering quality assessments and comparisons of clustering algorithm performances. Our measures are shown to be more robust than the commonly used modularity and conductance. In particular, our measures appear to be more responsive to cluster labeling and less sensitive to sample size and breakdowns during numerical stress testing. We also adapted Student's two-sample t-test to circumvent any possible correlation and degeneracies.

Future work will be focused on developing quality measures for clusterings with overlapping clusters. Overlapping clusters offer a more realistic model of most real-world networks. Unfortunately, quality measures to evaluate algorithms that identify overlapping clusters are sparse and a thorough examination is still required. Some authors have extended modularity to overlapping cluster quality measurements (e.g., [29, 7]). However, a thorough evaluation of it is still required. It is highly likely that the overlapping cluster extension suffers from many of the same shortcomings as its non-overlapping parent measure, including not meeting the axioms defining good clustering functions.

Funding

P. Miasnikof was supported by Mitacs-Accelerate PhD award IT05806.

Acknowledgements

- P. Miasnikof thanks Leonidas Pitsoulis, Liudmila Prokhorenkova, Andrei M. Raigorodskii, Cris Moore, Aaron Clauset and Mark Newman for their helpful comments.

- The research of P.M. Pardalos was conducted within the framework of the Basic Research Program at the National Research University Higher School of Economics (HSE).

References

[1] M. Ackerman and S. Ben-David. Measures of clustering quality: A working set of axioms for clustering. *Advances in Neural Information Processing Systems 21 - Proceedings of the 2008 Conference*, pages 121–128, 01 2008.

[2] H. M. Almeida, D. O. Guedes, W. Meira Jr., and M. J. Zaki. Is there a best quality metric for graph clusters? In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011. Proceedings, Part I*, pages 44–59, 2011.

[3] D. Aloise, G. Caporossi, P. Hansen, L. Liberti, S. Perron, and M. Ruiz. Modularity maximization in networks by variable neighborhood search. In D. A. Bader, H. Meyerhenke, P. Sanders, and D. Wagner, editors, *Graph Partitioning and Graph Clustering, 10th DIMACS Implementation Challenge Workshop, Georgia Institute of Technology, Atlanta, GA, USA, February 13-14, 2012. Proceedings*, pages 113–128, Providence, Rhode Island, 2012. American Mathematical Society.

[4] A. Biswas and B. Biswas. Defining quality metrics for graph clustering evaluation. *Expert Systems with Applications*, 71:1–17, 2017.

[5] V. D. Blondel, J-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, oct 2008.

[6] U. Brandes, D. Delling, M. Gaertler, R. Gorke, M. Hoefer, Z. Nikoloski, and D. Wagner. On modularity clustering. *IEEE Trans. on Knowl. and Data Eng.*, 20(2):172–188, February 2008.

[7] M. Chen, K. Kuzmin, and B. K. Szymanski. Extension of Modularity Density for Overlapping Community Structure. *arXiv e-prints*, page arXiv:1507.04606, Jul 2015.

[8] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *preprint*, 2004.

[9] J. Creusefond, T. Largillier, and S. Peyronnet. On the evaluation potential of quality functions in community detection for different contexts. *ArXiv e-prints*, October 2015.

[10] H. Djidjev and M. Onus. Using graph partitioning for efficient network modularity optimization. In D. A. Bader, H. Meyerhenke, P. Sanders, and D. Wagner, editors, *Graph Partitioning and Graph Clustering, 10th DIMACS Implementation Challenge Workshop, Georgia Institute of Technology, Atlanta, GA, USA, February 13-14, 2012. Proceedings*, pages 103–112, Providence, Rhode Island, 2012. American Mathematical Society.

[11] S. Emmons, S. Kobourov, M. Gallant, and K.K. Börner. Analysis of network clustering algorithms and cluster quality metrics at scale. *PLOS ONE*, 11(7):1–18, 07 2016.

[12] N. Fan and P. M. Pardalos. Robust optimization of graph partitioning and critical node detection in analyzing networks.

[13] N. Fan and P. M. Pardalos. Linear and quadratic programming approaches for the general graph partitioning problem. *J. of Global Optimization*, 48(1):57–71, September 2010.

[14] S. Fortunato. Community detection in graphs. *Physics Reports*, 486:75–174, February 2010.

[15] S. Fortunato and M. Barthélemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41, 2007.

[16] S. Fortunato and D. Hric. Community detection in networks: A user guide. *ArXiv e-prints*, November 2016.

[17] B. H. Good, Y.-A. de Montjoye, and A. Clauset. Performance of modularity maximization in practical contexts. *preprint*, 81(4):046106, April 2010.

[18] A.A. Hagberg, D.A. Schult, and P.J. Swart. Exploring network structure, dynamics, and function using networkx. In G. Varoquaux, T. Vaught, and J. Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA, USA, 2008.

[19] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning, Second Edition: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer-Verlag New York, 2nd ed. 2009. edition, 2009.

[20] L. B. Holder, R. Caceres, D. F. Gleich, J. Riedy, M. Khan, N. V. Chawla, R. Kumar, Y. Wu, C. Klymko, T. Eliassi-Rad, and A. Prakash. Current and future challenges in mining large networks: Report on the second sdm workshop on mining networks and graphs. *SIGKDD Explor. Newsl.*, 18(1):39–45, August 2016.

[21] H. Huang, Y. Liu, D.N. Hayes, A. Nobel, J.S. Marron, and C. Hennig. (15) Significance Testing in Clustering. In C. Hennig, M. Meila, F. Murtagh, and R. Rocci, editors, *Handbook of Cluster Analysis*, pages 315–335, Boca Raton, FL, USA. Chapman and Hall/CRC.

[22] R. Kannan, S. Vempala, and A. Vetta. On clusterings: Good, bad and spectral. *J. ACM*, 51(3):497–515, May 2004.

[23] A. Kehagias and L. Pitsoulis. Bad communities with high modularity. *The European Physical Journal B*, 86(7):330, Jul 2013.

[24] A. Kehagias and L. Pitsoulis. Graph Clustering Quality Functions. *Unpublished*, 2018.

[25] J.M. Kleinberg. An impossibility theorem for clustering. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 463–470. MIT Press, 2003.

[26] T. Van Laarhoven and E. Marchiori. Axioms for graph clustering quality functions. *J. Mach. Learn. Res.*, 15(1):193–215, January 2014.

[27] A. Lancichinetti and S. Fortunato. Community detection algorithms: A comparative analysis. *ArXiv e-prints*, 80(5):056117, November 2009.

[28] A. Lancichinetti, S. Fortunato, and F. Radicchi. Benchmark graphs for testing community detection algorithms. *Physical Review. E, Statistical, nonlinear, and soft matter physics*, 78:046110, 11 2008.

[29] A. Lázár, D. Ábel, and T. Vicsek. Modularity measure of networks with overlapping communities. *EPL (Europhysics Letters)*, 90(1):18001, Apr 2010.

[30] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data*, 1(1):2es, 2007.

[31] J. Leskovec, K. J. Lang, and M. W. Mahoney. Empirical Comparison of Algorithms for Network Community Detection. *ArXiv e-prints*, April 2010.

[32] J. Leskovec, K.J. Lang, A. Dasgupta, and M. W. Mahoney. Statistical properties of community structure in large social and information networks. In *In 7th International Conference on WWW*, New York, NY, USA, 2008. Association for Computing Machinery.

[33] S. Mancoridis, B. Mitchell, C. Rorres, Y-F. Chen, and E. Gansner. Using automatic clustering to produce high-level system organizations of source code. *Program Comprehension, Workshop Proceedings*, pages 45 – 52, 07 1998.

[34] P. J. McSweeney, K. Mehrotra, and J. C. Oh. *Game-Theoretic Framework for Community Detection*, pages 573–588. Springer New York, New York, NY, 2014.

[35] P. Miasnikof, A.Y. Shestopaloff, A.J. Bonner, and Y. Lawryshyn. *A Statistical Performance Analysis of Graph Clustering Algorithms*, chapter 11. Lecture Notes in Computer Science. Springer, Cham, 6 2018.

[36] C. N. Moschopoulos, G. A. Pavlopoulos, E. Iacucci, J. Aerts, S. Likothanassis, R. Schneider, and S. Kossida. Which clustering algorithm is better for predicting protein complexes? *BMC Research Notes*, 4(1):549, Dec 2011.

[37] M. Nascimento and L. Pitsoulis. Community detection by modularity maximization using GRASP with path relinking. *Computers and Operations Research*, 40:3121–3131, 12 2013.

[38] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 69:026113, 03 2004.

[39] C. Nicolini, C. Bordier, and A. Bifone. Community detection in weighted brain connectivity networks beyond the resolution limit. *NeuroImage*, 146:28 – 39, 2017.

[40] L. Prokhorenkova. Using synthetic networks for parameter tuning in community detection. *arXiv e-prints*, page arXiv:1906.04555, Jun 2019.

[41] L. Ostroumova Prokhorenkova, P. Prałat, and A. Raigorodskii. Modularity of complex networks models. In A. Bonato, F.C. Graham., and P. Prałat, editors, *Algorithms and Models for the Web Graph*, pages 115–126, Cham, 2016. Springer International Publishing.

[42] L. Ostroumova Prokhorenkova, P. Prałat, and A. Raigorodskii. Modularity in several random graph models. *Electronic Notes in Discrete Mathematics*, 61:947 – 953, 2017. The European Conference on Combinatorics, Graph Theory and Applications (EUROCOMB'17).

[43] U. N. Raghavan, R. Albert, and S. Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E*, 76:036106, Sep 2007.

[44] J. Reichardt and S. Bornholdt. When are networks truly modular? *Physica D: Nonlinear Phenomena*, 224(1):20 – 26, 2006. Dynamics on Complex Networks and Applications.

[45] P. Sanders and C. Schulz. High quality graph partitioning. In D. A. Bader, H. Meyerhenke, P. Sanders, and D. Wagner, editors, *Graph Partitioning and Graph Clustering, 10th DIMACS Implementation Challenge Workshop, Georgia Institute of Technology, Atlanta, GA, USA, February 13-14, 2012. Proceedings*, pages 1–18, Providence, Rhode Island, 2012. American Mathematical Society.

[46] D. A. Spielman and S.-H. Teng. A local clustering algorithm for massive graphs and its application to nearly linear time graph partitioning. *SIAM Journal on Computing*, 42(1):1–26, Jan 2013.

[47] V. A. Traag, G. Krings, and P. van Dooren. Significant Scales in Community Structure. *Scientific Reports*, 3:2930, Oct 2013.

[48] P. Van Mieghem, X. Ge, P. Schumm, S. Trajanovski, and H. Wang. Spectral graph analysis of modularity and assortativity. *Phys. Rev. E*, 82:056113, Nov 2010.

[49] P. Wagenseller and F. Wang. Size matters: A comparative analysis of community detection algorithms. *CoRR*, abs/1712.01690, 2017.

[50] J. Yang and J. Leskovec. Defining and Evaluating Network Communities based on Ground-truth. *CoRR*, abs/1205.6233, 2012.

[51] J. Yang and J. Leskovec. Overlapping Community Detection at Scale: A Nonnegative Matrix Factorization Approach. *WSDM'13*, ACM 978-1-4503-1869-3/13/02, 2013.

[52] H. Yin, A.R. Benson, J. Leskovec, , and D. F. Gleich. Local higher-order graph clustering. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2017*, 2017.