Adaptive Cuts reveal multiscale complexity in networks

Louis Boucherie,[1,2] Kyriaki Ilektra Zarafeta,[1] Yong-Yeol Ahn,[3] and Sune Lehmann[1,2,*]

[1]Department of Applied Mathematics and Computer Science, Technical University of Denmark, Lyngby, Denmark
[2]Copenhagen Center for Social Data Science (SODAS), University of Copenhagen, Denmark
[3]Center for Complex Networks and Systems, Luddy School of Informatics, Computing and Engineering, Indiana University, Bloomington, IN, USA.

(Dated: September 3, 2024)

The identification of communities within networks represents a fundamental challenge in the field of network science, with significant implications for the understanding of complex systems. The application of link-based community detection methods, such as link clustering, provide valuable insights. However, their reliance on a single-level cut limits their capacity to identify communities with varying densities. In this study, we introduce the Adaptive Cut, which addresses this limitation by optimising the partition function over the multiple levels of the hierarchical dendrogram with a Markov chain Monte Carlo and simulated annealing scheme. Moreover, we introduce the concept of 'balanceness' as a quantitative measure to evaluate the balance of the dendrograms. We demonstrate that the balanceness metric quantifies the extent to which the Adaptive multi-level Cut method offers a substantial improvement over the previously proposed single-level cut approach. Our results demonstrate that the Adaptive Cut method significantly increases the objective function, thereby improving link community detection. We show our method can also improve the communities obtain from the Louvain algorithm. Additionally, the Adaptive Cut can be applied to any hierarchical clustering, not only community detection in networks. This methodological advancement makes a significant contribution to the field, providing a tool for the analysis and interpretation of any kind of clustering.

I. INTRODUCTION

The identification of groups of objects that are closely related, which is commonly referred to as clustering or community detection in networks, is an important problem in many scientific fields. A plethora of clustering methods have been proposed in the literature [1–5]. Community detection and cluster analysis methods frequently exhibit hierarchical characteristics, reflecting the multi-scale nature of the data [6–8]. Hierarchical clustering constructs a binary merge tree (or dendrogram) originating from leaves containing data elements and culminating at the root, which encompasses the entire dataset. Dendrograms are thus a common feature of both clustering and community detection methods. Our focus is on hierarchical clustering and community detection. However, the approach can be applied to any clustering process that produces a dendrogram.

The most common method for obtaining clusters (or communities) is to cut the dendrogram at a single level or a constant height cutoff value [9]. There are several methods for selecting this value, including optimizing an objective function [9, 10], , obtaining a specific number of clusters (through the elbow method or the silhouette method) [11], , or achieving clusters with high-intra similarity and high inter-cluster differences [12]. However, single-level cuts may not always accurately identify clusters. In particular, many dendrograms are unbalanced, and a single-level cut cannot effectively resolve the trade-off between over-aggregation in one part of the dendrogram and providing an excessively granular view in another part. Consequently, single-level cuts are incapable of optimally separating distinct clusters.

Single-level cuts are unable to use all of the information contained within the hierarchical struture of the dendrogram. In unbalanced dendrograms, the selection of a single cut point can result in the formation of clusters with significantly different sizes. For instance, statistical tests that assume equal cluster sizes may be unsuitable when employed in the analysis of unbalanced clusters []. Similarly, machine learning algorithms that rely on balanced training sets may underperform [].

We propose the Adaptive Cut, a novel method for cutting dendrogram with a multi-level cut. Our approach enhances a broad range of tasks that involves hierarchical tree structures. To demonstrate the robustness of our results, we evaluate the method on over 200 real networks. The adaptive cut optimizes an objective function along the dendrogram using a Monte Carlo Markov chain with a simulated annealing scheme. We demonstrate the generality of the method generality across two distinct use cases of community detection for network nodes and edges.

We also present a new measure of dendrogram balance. Numerous tree balance indices have been proposed in the literature [13–15]. Our novel balance index, based on information theory, is defined at each level of the dendrogram, computationally efficient, and satisfies the axioms required for membership in the class of robust, universal tree balance indices [16]. Previous work on multi-level cuts includes a visualization tool for exploring various clustering scenarios by offering different cut levels [17]. However, this approach is dependent on expert knowledge and does not provide a quantitative solution. In a more quantitative manner, [18] proposes a heuristic based on dendrogram shape, akin to the silhouette coefficient [19].

Markov chain Monte Carlo with simulated annealing has been employed for community detection in networks to optimize modularity [20, 21], description length [7], or fit stochas-

* sljo@dtu.dk

tic block models [22]. Although these examples do not utilize the tree structure to optimize the objective function, they operate on a large state space, making convergence to the optimal partition challenging and with high convergence time. In contrast, our Adaptive Cut method leverages the tree structure, reducing the size of the state space and enabling a more efficient optimization of the objective function and faster convergence to an optimal solution.

II. BALANCENESS

A. Definition

We introduce the *Balanceness* score, an information theory based metric that quantifies the balance of a dendrogram. The balanceness metric compares the actual branching structure of a dendrogram (real entropy) with both a perfectly balanced scenario (maximal entropy) and a highly skewed one (minimal entropy) to quantify how balanced the dendrogram is. At each level l of the dendrogram, the partition of the n leaves in each of the k branches is noted as follows:

$$\pi_l = B_1, B_2, \ldots, B_k \tag{1}$$

where B_i represents the set of leaves having branch B_i as an ancestor. To account for balanceness, we first define the maximal entropy of the leaf distribution,

$$H_{max}(k,n) = -\sum_{i=1}^{k} \frac{1}{k}\log_2\left(\frac{1}{k}\right) = \log_2(k)\,. \tag{2}$$

then we define the minimal entropy as,

$$H_{min}(k,n) = -\sum_{i=1}^{k-1} \frac{1}{n}\log_2\left(\frac{1}{n}\right) - \frac{n-(k-1)}{n}\log_2\left(\frac{n-(k-1)}{n}\right) \tag{3}$$

Additionally, we determine the actual realised entropy of the leaf partition π_l at level l as follows:

$$H(\pi_l) = -\sum_{i=1}^{|\pi_l|} p(B_i)\log_2(p(B_i))\,, \tag{4}$$

where $p(B_i) = |B_i|/n$. The Balanceness score corresponds to the average value of the ratio between the realized entropy minus the minimal entropy and the maximum entropy minus the minimal entropy across all levels,

$$B = \frac{1}{L}\sum_{i=1}^{L} \frac{H(\pi_l) - H_{min}(\pi_l)}{H_{max}(\pi_l) - H_{min}(\pi_l)}\,. \tag{5}$$

The balanceness metric ranges from 0 (unbalanced dendrogram, Fig.1a) to 1 (perfectly balanced dendrogram, Fig. 1c). Moreover, the balanceness score satisfies the axioms of a robust and universal measure of tree balance [16].

B. Examples

In this paragraph we examine the balanceness score of two real networks. Figure 1d,g illustrates the dendrogram obtained by the link clustering of two real networks, the character network of Les Miserables (unbalanced) and the street network of Brasilia (balanced, 1g). To determine the balanceness of the dendrograms, we need to compute the maximum, minimal and real entropy values at each level as defined in (Eq.2 -5). These values are displayed at each level of the dendrogram on figure 1e,h. The balanceness score is equal to the proportion of the area between the two black curves that is under the pink curve. For further details, refer to Equation 5. Consequently, if the real entropy is equal to the maximum entropy, the balanceness score is equal to one. Conversely, if the real entropy is equal to the minimum entropy, the balanceness score is equal to zero.

A majority of real-world networks yield unbalanced dendrograms, as illustrated in Figure 1f. Moreover, we show on Figure 1i that the balanceness measure independent of the size of the network and therefore the size of the dendrogram.

III. ADAPTIVE CUT

To identify the multi-levels of the Adaptive Cut, we optimize an objective function over the partitions (or cluster membership). We employ a Markov chain Monte Carlo (MCMC) approach to optimize the objective function f within a finite search space X, using a softmax distribution,

$$\pi^\star(x) = \frac{e^{f(x)/T}}{Z}\,, \tag{6}$$

where x represents a state within the search space X, T denotes the temperature parameter, and Z is the partition function.

A. Markov Chain

The Markov chain we define can walk up or down the dendrogram. As it goes up, it merges two neighboring partitions into a larger one; as it goes down, it splits one partition into two smaller partitions. For each step, given a partition, a cluster is selected uniformly at random, followed by a direction (up or down). Subsequently, we obtain a new partition by merging the cluster with its neighbour (up) or dividing the cluster in two (down). Each move is accepted or rejected with a probability given as a function of the objective function difference Δf. To ensure that the Adaptive Cut MCMC converges to the optimal partition we must verify the Markov chain is ergodic (or irreducible), implying that every network partition present in the dendrogram is accessible from every other partition in the dendrogram, and that detailed balance is maintained, meaning each step is reversible. After a sufficiently long equilibration time, each observed partition must occur with the desired probability $\pi^\star$.

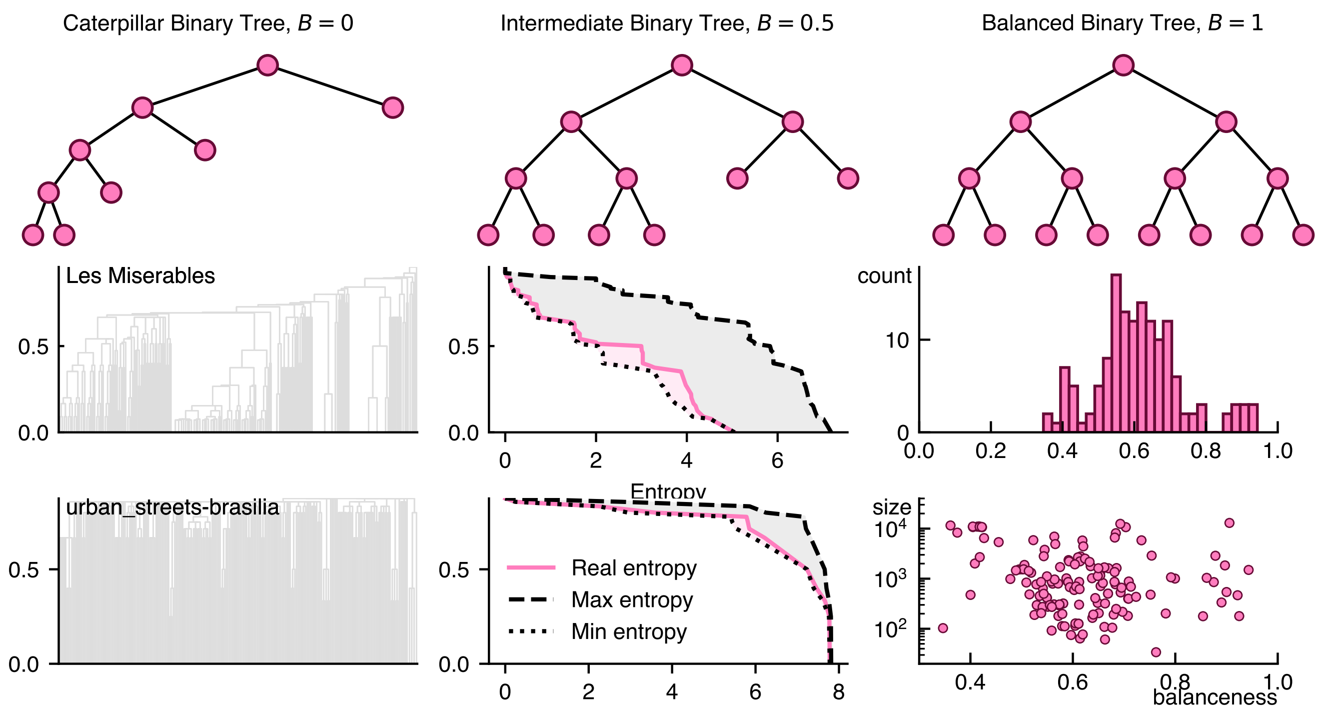

FIG. 1. **Explanation of the Balanceness Measure.** **(a, b, c)** Illustrations of different tree structures: (a) an unbalanced caterpillar tree, (b) an intermediate tree, and (c) a balanced tree. **(d)** Dendrogram representing the "Les Miserables" character network based on link similarities [9]. The dendrogram is unbalanced, as shown in (e). **(e)** The progression of the real, maximal and minimal entropies (x axis) across different similarity levels (y axis). The three entropies are used to compute the balanceness metric (Eq. 5). **(f)** The distribution of balanceness scores for 200 real networks **??**. **(g)** A balanced dendrogram for the urban street network of Brasilia. **(h)** The progression of the real, maximal and minimal entropies across different levels. **(i)** A plot of the balanceness metric against network size (number of node), demonstrating that the balanceness score is independent of network size.

We define the following Markov chain that is ergodic. Given a partition that correspond to the list of branch $\{B_1, B_2, \ldots, B_n\}$, we select uniformly a branch $i \sim \mathcal{U}\{1, .., n\}$. The choice of the direction is not uniform, indeed, if we go down one level, there are two paths that can bring back the chain to the initial state. Consequently, the probability to go down should be twice the probability to go up. Although the number of branches also changes, it increases (down) or decreases (up) by one. Therefore, the probability to go up/down from a level l that contains n branches is:

$$Q_{i\to\text{up}} = \frac{1}{3} \times \frac{3n}{3n-2}, \tag{7}$$

$$Q_{i\to\text{down}} = \frac{2}{3} \times \frac{3(n-1)}{3n-2}. \tag{8}$$

The probability to move from level i to j is,

$$Q_{i\to j} = \begin{cases} \frac{1}{3n-2}, & \text{if } j \text{ is up from } i, \\ \frac{2(n-1)}{n(3n-2)}, & \text{if } j \text{ is down from } i, \\ 0, & \text{if we cannot attain } j \text{ from } i \text{ in one step.} \end{cases} \tag{9}$$

The process can transition from any state to any state, irrespective of the number of steps required, rendering the Markov chain defined by Q ergodic. However, it does not fulfill detailed balance. This condition can be enforced using the Metropolis-Hastings algorithms [23, 24].

B. Metropolis-Hasting

To sample the Markov chain we use the Metropolis–Hastings algorithm [23]. Therefore, at each step of the Markov chain, we accept a move with a probability α given by,

$$\alpha = \min\left\{1, \frac{\pi(x_\star)Q_{x_\star\to x_i}}{\pi(x_i)Q_{x_i\to x_\star}}\right\}, \tag{10}$$

with probability α, we move to state $x_{\text{i+1}} = x_\star$, otherwise, $x_{\text{i+1}} = x$. As Q is symmetric and $\pi \propto \exp(f(x)/T)$ then if the objective function increase between the two states, i.e. $f(x_\star) > f(x)$ we always move to $x_\star$, else we move with probability $\exp(-|\Delta f|/T)$. In Eq. (10), the parameter T represents the temperature, which can assist in escaping local maxima. Therefore, the Markov chain converges to the desired distribution, as defined in Eq. (6).

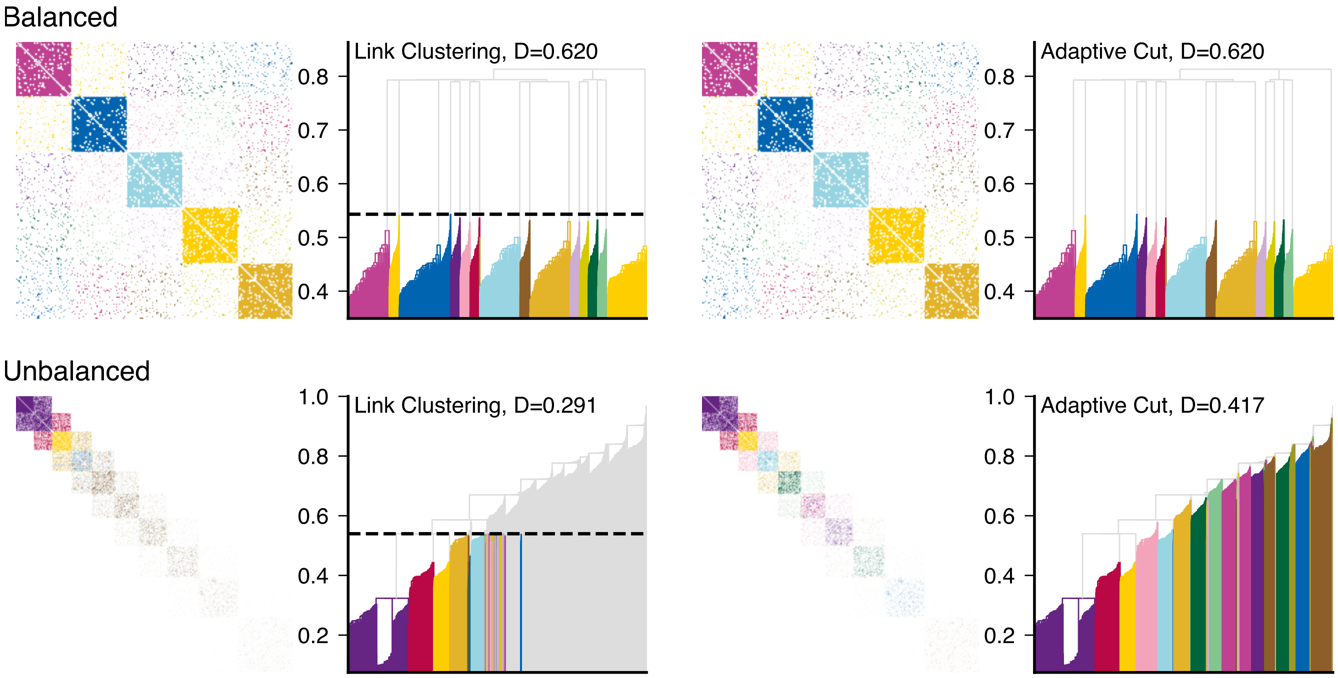

FIG. 2. **Comparison between Link Clustering and the Adaptive Cut.** **(a)** The adjacency matrix of a stochastic block model network, with nodes colored according to edge communities identified by the link clustering method [9]. **(b)** The corresponding dendrogram for the same network, with partitions or communities defined by the single-level link clustering cut [9]. The similarity level at which the cut is made is indicated by the dashed line. **(c, d)** Similar network to (a) and (b), but using an adaptive cut method instead of link clustering. **(e, f)** The same type of analysis applied to a stochastic block model with decreasing density

C. Simulated Annealing

In practice, however, the mixing time may be long. To prevent the Markov chain from getting trapped in a local maximum, we employ a simulated annealing scheme [25]. The principle behind simulated annealing is to initiate with a high temperature and gradually reduce it over time. We opt for Fast Annealing [26], where the temperature cooling scheme is,

$$T_k = \frac{T_0}{k}, \tag{11}$$

with T_k is the temperature at iteration k and T_0 is the initial temperature.

D. Initialization

We initiate the Markov chain at the single-level cut obtained from the clustering algorithms. Indeed a critical practical aspect of Markov chain Monte Carlo (MCMC) methods is the selection of the initial state, as it has a significant impact on the mixing time [27]. The mixing time is strongly influenced by the proximity of the initial state of the Markov chain to the equilibrium partitions. While it is common practice to start with a random partition, in the present case, a local maximum is already provided by the single-level cut. Utilizing MCMC to optimize community detection methods is not novel [20–22]. However, the main issue is the long converging time as the state space of these methods is very large. By restricting our method to the structure of the dendrogram, we optimize the moves and achieve significant improvement over random alternatives when networks become larger. The Adaptive Cut has a smaller state space, allowing it to converge to the global optimum in a shorter time.

[SI ADD CONVERGENCE TRAJECTORIES]

IV. RESULTS

A. Link Clustering

The Link Clustering method [9] utilizes hierarchical clustering based on link similarity (Jaccard index on neighbors) to create a dendrogram, where each link is a leaf, and branches represent link communities. Communities are extracted by cutting the dendrogram at a specific similarity threshold, allowing nodes to belong to multiple overlapping communities (see VI B). To identify the most relevant communities, the method introduces partition density D, which measures link density within communities and does not suffer from resolution limits [9, 28, 29]. The optimal cut is determined by maximizing D along the levels of the dendrogram.

However, determining the appropriate cut level can be challenging. The Adaptive Cut method optimizes D further, potentially revealing more accurate community structures, especially in networks with communities of varying sizes and densities. This approach addresses the limitations of a single-level cut by better capturing complex network structures (Fig. 2e-h).

1. *Toy examples*

Varying Density Stochastic Block Model

To illustrate the limitations of the single level cut, we focus on two toy networks with a given community structure, a stochastic block model [30, 31] and a varying density stochastic block model that we introduce. We simplify the stochastic block model (SBM) by assuming that the probability of an edge between two nodes depends only on whether the nodes belong to the same community or different communities (see the adjacency matrix Fig. 2a) The model is described by the following equation:

$$P(A_{ij} = 1 \mid z_i = z_j) = \theta_{\text{intra}}, \quad P(A_{ij} = 1 \mid z_i \neq z_j) = \theta_{\text{inter}} \tag{12}$$

where A_{ij} is the adjacency matrix entry for nodes i and j, where $A_{ij} = 1$ indicates the presence of an edge between these nodes, and $A_{ij} = 0$ otherwise. The variables z_i and z_j denote the community assignments of nodes i and j respectively. The term θ_{intra} represents the probability of an edge between any two nodes in the same community, while θ_{inter} represents the probability of an edge between nodes in different communities. In this model, θ_{intra} is the same for all communities, and θ_{inter} is the same for all pairs of different communities. Our varying density stochastic block model is the same, except that the θ_{intra} decreases and the θ_{inter} also decreases. In the Varying Density Stochastic Block Model, we generalise the traditional stochastic block model by varying the intra-community and inter-community densities. This approach allows the creation of communities with different internal structures and varying degrees of connectivity between them. In particular, the intra- and inter-community densities decrease across communities (see VI A and adjacency matrix Fig. 2e).

Link Clustering vs Adpative Cut

The stochastic block model (adjacency matrix in Fig. 2a) exhibits clear communities of identical size and density, thus exhibiting identical similarities with respect to link clustering. The single-level cut is capable of distinguishing between the communities (see Fig. 2b), while the adaptive cut does not offer a superior partitioning (see Fig. 2d). Indeed, the partition density of both cuts is $D = 0.620$. Conversely, although the Varying Density Stochastic Block Model still exhibits a clear community structure, as evidenced by the adjacency matrix in Fig. 2e, the single-level cut is unable to provide a partition that reflects these communities. This is due to the inability of the cut to identify communities with varying densities. The adaptive cut yields a notable enhancement in performance, with a value of $D = 0.417$, in comparison to the single level cut, which has a value of $D = 0.291$, Fig. 2g,h), the communities are clearly aligned with the blocks of the adjacency matrix, as illustrated in Figure 2g. Finally, it can be observed that the stochastic block model dendrogram is more balanced than the dendrogram of the varying density stochastic block model. This is evidenced by the respective balanceness values of $B = 0.6$ and $B = 0.4$.

2. *Community Structure Evaluation*

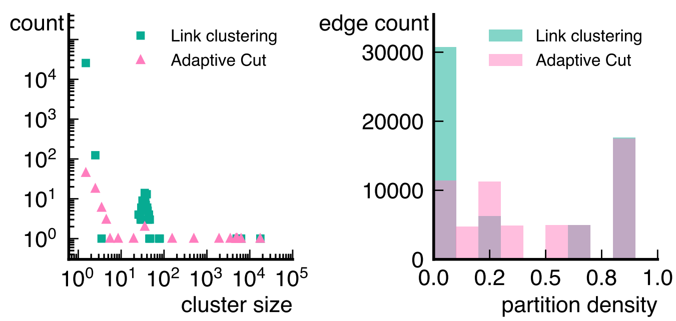

FIG. 3. Comparison between single and multi-level cuts. **(a)** Distribution of community sizes for link clustering and adaptive cut. Link Clustering results in 20,000 single node communities, while Adaptive Cut effectively merges these into larger, more meaningful clusters. **(b)** Distribution of partition density across edges. Link Clustering tends to produce clusters with either very high (¿0.7) or very low (¡0.1) partition densities due to the limitations of a single level cut. In contrast, the adaptive cut better captures the varying densities present in the network model.

Figure 3a illustrates the distribution of community sizes for both Link Clustering and Adaptive Cut for the varying density stochastic block model. The Link Clustering method results in a large number of single node communities (21254). This observation is indicative of the tendency for the single level cut to over partitioned the network. In contrast, the Adaptive Cut method successfully merges these small communities into larger, more meaningful clusters, reflecting a more accurate representation of the structure of the network, as can be seen in the adjacency matrix (see Fig. 2e).

Figure 3b shows the distribution of partition densities for the edges within the network. The link clustering method tends to produce clusters with a bimodal partition densities, they are either high (> 0.7) or very low (< 0.1). This outcome is attributable to the intrinsic density trade-offs imposed by the single-level cut, which is unable to achieve a trade-off between the varying densities across the network. Conversely, the adaptive cut method is able to accommodate the varying densities of the network, thereby capturing the a wider amplitude of partition densities and thus detecting communities that are close to the ground truth. [AMI NMI ?]

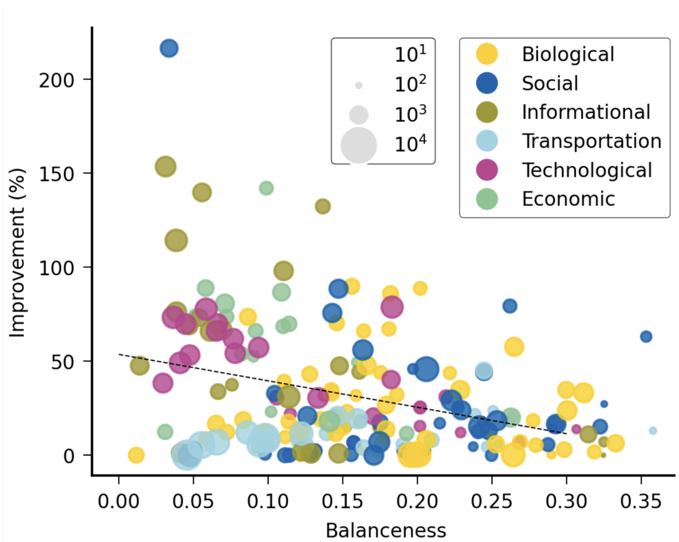

FIG. 4. Improvement (in %) of the partition density between the single-level cut (Link Clustering [9]) and the multi-level Adaptive Cut as a function of the dendrograms balanceness. The symbol colors indicate the domain of the network, and the size the number of nodes in the network, as show in legends.

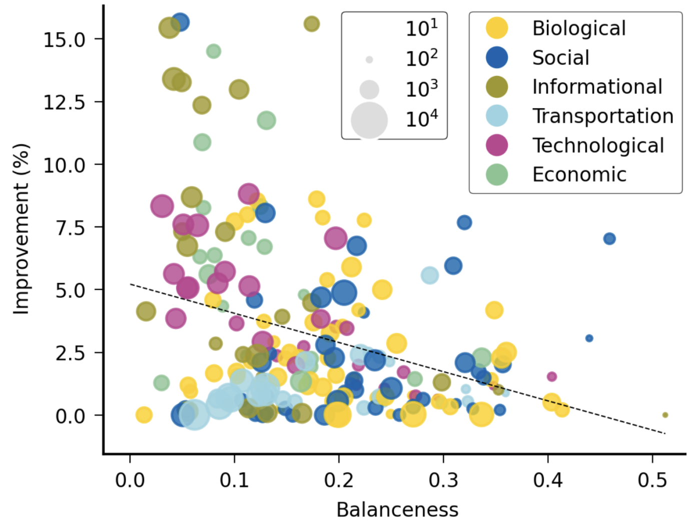

FIG. 5. Improvement (in %) of the modularity between the single-level cut (Louvain) and the multi-level cut (Louvain + Adaptive Cut) as a function of Balanceness. The symbol colors indicate the domain of the network, and the size the number of nodes in the network, as show in legends.

3. *Real Networks Examples*

We compare the Link Clustering and Adaptive Cut across 200 real-world networks to evaluate the influence of balanceness on community detection outcomes. Figure 4 presents the extent to which Adaptive Cut enhances partition density (expressed as a percentage) relative to the balanceness of each dendrogram. Our findings indicate that as the balanceness of a network decreases, the likelihood of achieving significant improvements with Adaptive Cut increases. Importantly, this relationship is consistent across various types of networks, including economic, transportation, informational, biological, and social networks.

B. Louvain

The Louvain method [32] is a community detection algorithm that aims to identify the hierarchical structure of communities in a network. The algorithm iteratively optimizes a modularity function that measures the quality of the community structure. The Louvain method generates a dendrogram by recursively merging the communities with the highest modularity gain. In each iteration, the algorithm considers all potential pairs of adjacent communities and calculates the modularity gain associated with their merger into a single community. The modularity gain is defined as the difference in modularity between the merged communities and the original communities. Subsequently, the pair of communities with the highest modularity gain is merged into a single community. The algorithm proceeds to merge communities until no further increase in modularity is observed, employing a greedy approach. The resulting community structure can be represented as a dendrogram, with each level of the tree corresponding to a distinct community partition. The leaves of the tree represent the individual nodes in the network, whereas the internal nodes represent the merged communities. As the Louvain method is designed to optimise modularity in a greedy manner, it may not always identify the globally optimal community structure and become stuck in a locally optimal community structure.

To apply the Adaptive Cut, we continue building the dendrogram until the top, following the greedy approach, and allowing modularity to decrease when merging two branches. We can then apply the Adaptive Cut on this dendrogram.

In comparison to Link Clustering, the Louvain method generates a more balanced dendrogram. This is due to the first step of the algorithm that forces each node/community to merge with another, resulting in branches of more uniform size and a more balanced dendrogram overall.

V. DISCUSSION

The Adaptive Cut is limited by the dendrogram structure, in the sense that the set of possible partitions consists of the partition that are in the dendrogram, but force monte carlo methods [20–22] are not limited by that. On the other hand their optimization space is much more larger and therefore necessitate more computation to converge.

ACKNOWLEDGMENTS

We thank Morten Mørup, Nikolaos Nakis and Abdulkadir Çelikkanat for stimulating discussions. This work was funded

by .

VI. SUPPLEMENTARY

A. vdSBM

The model begins by defining a set of communities, each with a specific size and intra-community density. The intra-community density, $\theta_{\text{intra}}^{(c)}$, for community c is given by:

$$\theta_{\text{intra}}^{(c)} = \frac{1}{k_c} \sum_{i \in \text{Community } c} p_i$$

where k_c is the size of the community, and p_i represents the probability of an edge between any two nodes within community c. As the community index c increases, $\theta_{\text{intra}}^{(c)}$ decreases linearly or non-linearly, depending on the specific configuration used.

For the inter-community density, $\theta_{\text{inter}}^{(c,c+1)}$, which represents the probability of an edge between nodes in adjacent communities c and $c+1$, the density is also designed to decrease as a function of the intra-community densities:

$$\theta_{\text{inter}}^{(c,c+1)} = \left(\theta_{\text{intra}}^{(c+1)}\right)^2$$

B. Dendrogram Construction and Link Similarity

To construct a dendrogram representing link communities [9], we start by defining the similarity between pairs of links. For an undirected, unweighted network, let $n_1(i)$ denote the set of neighbors of node i. The similarity $S(e_{ik}, e_{jk})$ between two links e_{ik} and e_{jk}, sharing a common node k, is calculated using the Jaccard index:

$$S(e_{ik}, e_{jk}) = \frac{|n_1(i) \cap n_1(j)|}{|n_1(i) \cup n_1(j)|}, \tag{13}$$

Here, $n_1(i)$ and $n_1(j)$ represent the sets of neighbors of nodes i and j, respectively, with the shared node k excluded. This measure, known as the Jaccard index, provides a normalized similarity score based on the overlap between the sets of neighbors.

The similarity between links can be easily extended to networks with weighted, directed, or signed links (without self-loops), as the Jaccard index generalizes to the Tanimoto coefficient [33]. Consider a vector $\mathbf{a_i} = (\tilde{A}_{i1}, \ldots, \tilde{A}_{iN})$, where

$$\tilde{A}_{ij} = \frac{1}{k_i} \sum_{i' \in n(i)} w_{ii'} \delta_{ij} + w_{ij}, \tag{14}$$

with w_{ij} representing the weight on edge e_{ij}, $n(i) = \{j | w_{ij} > 0\}$ being the set of all neighbors of node i, $k_i = |n(i)|$ denoting the degree of node i, and $\delta_{ij} = 1$ if $i = j$ and zero otherwise. The similarity between edges e_{ik} and e_{jk}, analogous to Eq. 13, is now defined by the Tanimoto coefficient:

$$S(e_{ik}, e_{jk}) = \frac{\mathbf{a_i} \cdot \mathbf{a_j}}{|\mathbf{a_i}|^2 + |\mathbf{a_j}|^2 - \mathbf{a_i} \cdot \mathbf{a_j}}, \tag{15}$$

This formula generalizes the similarity measure to handle various types of networks, including those with weighted, directed, or signed edges, offering greater flexibility in the analysis.

C. Single-Linkage Clustering for Dendrogram Creation

Using the similarity matrices $S(e_{ik}, e_{jk})$ for the unweighted case, or $S(e_{ik}, e_{jk})$ for the weighted, directed case as defined in Eq. 15, we perform single-linkage hierarchical clustering. This method initializes each link as its own cluster and iteratively merges clusters based on the highest similarity until a single cluster remains. The resulting hierarchical structure is represented as a dendrogram, where each leaf corresponds to an original network link, and the branches depict the formation of link communities.

D. Partition Density for Evaluating Clusters

To identify the most meaningful level of clustering within the dendrogram, we employ partition density D, a metric that assesses the density of links within communities. Given a network with M links and N nodes, partitioned into C subsets $P = \{P_1, \ldots, P_C\}$, where m_c is the number of links in subset P_c, and n_c is the number of nodes connected by these links, the link density D_c for a community c is defined as:

$$D_c = \frac{m_c - (n_c - 1)}{n_c(n_c - 1)/2 - (n_c - 1)}, \tag{16}$$

This expression normalizes the number of links in P_c by the minimum and maximum possible number of links that could exist among n_c nodes. The overall partition density D is then computed as the weighted average of D_c over all communities:

$$D = \frac{2}{M} \sum_c m_c \frac{m_c - (n_c - 1)}{(n_c - 2)(n_c - 1)}. \tag{17}$$

This approach avoids the resolution limits that often challenge other community detection methods, making it an effective measure for evaluating the hierarchical structure in a network.

E. Adaptive Cut for Optimal Clustering

To optimize the clustering process, we introduce an adaptive cutting technique that selects the optimal dendrogram cut by maximizing the partition density D as defined in equation 17. This method ensures that the communities identified are both meaningful and reflective of the network's underlying structure.

1. State Space

The state space of MCMC on the dendrogram is smaller than if we simply optimize the objective function over all possible partitions to find the best one. The number of ways to partition n vertices into k non-empty groups is given by the Stirling number of the second kind [34], denoted as $S(n,k)$. Therefore, the total number of distinct community divisions is given by the sum over all k from 1 to n of $S(n,k)$, which can be written as:

The total number of partitions of a set of n objects, regardless of the number of subsets, is given by the sum of the Stirling numbers of the second kind for k from 1 to n:, which is known as the Bell number $B(n)$,

$$B(n) = \sum_{k-1}^{n} S(n,k) \tag{18}$$

This sum does not have a known closed form, but for all $n \leq 1$, the sum of the Stirling numbers of the second kind for $k = 1$ and $k = 2$ is 2^{n-1}, i.e.,$S(n,1)+S(n,2) = 2^{n-1}$. Therefore, the sum over all k from 1 to n of $S(n,k)$ must increase at least exponentially in n.

On the other hand, the number of partitions allowed by the dendrogram, i.e that respect the structure of the binary tree, is much smaller. When considering multi-cuts on a dendrogram, the state space is defined by the possible partitions of the tree, which can be obtained by making cuts at different levels of the tree. For a perfectly balanced binary tree with n leaf nodes, the number of possible partitions (or states) is $2^{n-1} - 1$. Indeed, each cut separates the subtree below that node from the rest of the tree, creating a new cluster. The number of ways to cut the tree is equivalent to the power set of the set of internal nodes, minus the empty set (since we don't consider the situation with no cuts as a valid partition). The power set of a set is the set of all possible subsets of that set. If n is the number of leaf nodes in the tree, then there are $n-1$ internal nodes in a complete binary tree. Therefore, the number of possible partitions is $2^(n-1)-1$ which is also $S(n,2)$.

This state space size is significantly smaller than the Bell number, which represents the number of ways to partition a set of n objects into any number of subsets. The Bell number grows much faster than $2^{n-1}-1$, and for large n, the difference between the two becomes increasingly pronounced.

The smaller state space when considering multi-cuts on a dendrogram has important implications for the performance of MCMC methods. In particular, it suggests that MCMC algorithms should converge faster when applied to dendrograms compared to more general partition problems. This is because the algorithm has fewer states to explore, allowing it to more quickly and efficiently sample the state space and converge to the equilibrium distribution.

Moreover, the tree structure of the dendrogram allows for more informed decisions about where to make cuts, potentially leading to more efficient exploration of the state space. Therefore, in practice, MCMC methods applied to dendrograms with multi-cuts can offer significant advantages in terms of both computational efficiency and convergence speed.

VII. SUPPLEMENTARY TABLES

TABLE I: Result for the link clustering method

Name	Nodes	Edges	LC	AC	Balanceness	Category
caenorhabditis elegans	453	2025	0.144	0.199	0.535	Biological
central chilean power grid reduced	218	527	0.541	0.592	0.697	Technological
chicago road network	12979	20627	0.122	0.122	0.906	Transportation
clements long plant pollinator web	275	906	0.036	0.039	0.551	Biological
contact	274	2124	0.533	0.543	0.559	Social
david copperfield	112	425	0.063	0.070	0.584	Informational
euairtransportation multiplex	417	2953	0.174	0.187	0.549	Transportation
ego facebook	2888	2981	0.000	0.000	0.754	Social
email	1133	5451	0.102	0.125	0.649	Social
email europe	986	16064	0.135	0.152	0.478	Social
epinions	964	14214	0.329	0.392	0.564	Economic
erdos	6927	11850	0.006	0.006	0.564	Informational
football	115	613	0.550	0.562	0.605	Social
gallery proximity	410	2765	0.362	0.411	0.644	Social
haggle human proximity network	274	2124	0.533	0.543	0.559	Social
high school dynamic contacts 1	126	1709	0.446	0.476	0.611	Social
high school dynamic contacts 1 copy	126	1709	0.446	0.476	0.611	Social
high school dynamic contacts 2	180	2220	0.326	0.331	0.572	Social
hospital ward dynamic contacts	75	1139	0.466	0.470	0.596	Social
hypertext 2009 dynamic contact network	113	2196	0.368	0.370	0.579	Social
jazz	198	2742	0.416	0.427	0.579	Social

Table I (Continued)

Name	Nodes	Edges	LC	AC	Balanceness	Category
les miserables	77	254	0.577	0.598	0.619	Informational
oregon 0	10670	22002	0.006	0.007	0.423	Technological
oregon 1	10729	21999	0.006	0.006	0.405	Technological
oregon 2	10790	22469	0.007	0.007	0.422	Technological
oregon 3	10859	22747	0.007	0.007	0.419	Technological
oregon 4	10886	22493	0.006	0.006	0.418	Technological
oregon 5	10943	22607	0.007	0.009	0.414	Technological
oregon 6	11011	22677	0.006	0.006	0.405	Informational
oregon 7	11051	22724	0.006	0.008	0.417	Technological
political books	105	441	0.287	0.306	0.677	Informational
power grid	4941	6594	0.137	0.137	0.841	Technological
pretty good privacy	10680	24316	0.265	0.329	0.703	Technological
route views	6474	12572	0.008	0.013	0.427	Informational
stem concept networks researchers	1616	3045	0.019	0.019	0.669	Informational
song of ice and fire	796	2823	0.120	0.161	0.560	Informational
star wars	109	398	0.321	0.326	0.662	Informational
train	64	243	0.557	0.628	0.614	Transportation
trumpworld	2669	3380	0.024	0.024	0.610	Economic
us airport network	500	2980	0.238	0.264	0.543	Transportation
workplace contacts	92	755	0.325	0.331	0.559	Social
yeast spliceosome	103	4119	0.915	0.915	0.346	Biological
zachary	34	78	0.285	0.377	0.762	Social
arxiv collab hep th 1999	5835	13815	0.349	0.350	0.734	Social
bible nouns	1707	9059	0.178	0.236	0.575	Informational
bitcoin trust	5875	21489	0.023	0.023	0.523	Social
board directors net1m 2011 08 01	854	2745	0.849	0.892	0.875	Social
board directors net2m 2011 08 01	1066	1148	0.016	0.016	0.792	Social
cintestinalis	205	2575	0.165	0.172	0.538	Biological
copenhagen fb friends	800	6418	0.147	0.211	0.571	Social
eu procurements alt at 2008	1684	1921	0.002	0.003	0.662	Economic
eu procurements alt cz 2008	1777	2110	0.002	0.003	0.635	Economic
eu procurements alt dk 2011	2219	2904	0.003	0.005	0.614	Economic
eu procurements alt ee 2008	480	540	0.006	0.008	0.752	Economic
eu procurements alt es 2008	8097	13286	0.002	0.003	0.684	Economic
eu procurements alt fi 2008	2517	3701	0.005	0.007	0.624	Economic
eu procurements alt gr 2008	1933	2495	0.002	0.002	0.604	Economic
eu procurements alt hu 2015	2163	2929	0.006	0.008	0.633	Economic
eu procurements alt it 2008	6622	9121	0.002	0.003	0.682	Economic
eu procurements alt lt 2008	1359	2486	0.015	0.019	0.682	Economic
eu procurements alt lv 2008	1628	2269	0.006	0.010	0.685	Economic
eu procurements alt pl 2008	12456	23991	0.015	0.017	0.693	Economic
eu procurements alt pt 2008	692	776	0.004	0.005	0.713	Economic
eu procurements alt se 2008	4421	5905	0.002	0.002	0.620	Economic
eu procurements alt sk 2008	660	773	0.009	0.011	0.695	Economic
euroroad	1039	1305	0.068	0.082	0.861	Transportation
facebook friends	329	1954	0.414	0.436	0.648	Social
facebook organizations l1	5793	30753	0.133	0.149	0.619	Social
facebook organizations m1	1429	19357	0.126	0.151	0.514	Social
facebook organizations s1	320	2369	0.174	0.196	0.581	Social
facebook organizations s2	165	726	0.219	0.254	0.638	Social
foursquare nyc restaurant checkin	4906	13457	0.003	0.003	0.566	Social
foursquare nyc restaurant tips	5372	8852	0.001	0.001	0.455	Social
genetic multiplex candida	303	314	0.048	0.048	0.612	Biological
genetic multiplex gallus	205	234	0.034	0.035	0.651	Biological
genetic multiplex humanhiv1	1005	1155	0.000	0.000	0.799	Biological
genetic multiplex humanherpes4	189	191	0.002	0.002	0.525	Biological
genetic multiplex plasmodium	1158	2402	0.010	0.014	0.653	Biological
genetic multiplex rattus	2350	3484	0.053	0.067	0.592	Biological
interactome figeys	2217	6418	0.008	0.015	0.593	Biological
interactome stelzl	1615	3106	0.010	0.010	0.650	Biological

Table I (Continued)

Name	Nodes	Edges	LC	AC	Balanceness	Category
interactome vidal	2783	6007	0.054	0.054	0.615	Biological
kegg metabolic aae	880	2296	0.045	0.067	0.559	Biological
kegg metabolic afu	861	2011	0.036	0.055	0.538	Biological
kegg metabolic ana	1314	3552	0.030	0.030	0.519	Biological
kegg metabolic ape	769	1858	0.030	0.055	0.529	Biological
kegg metabolic atc	1538	4315	0.028	0.051	0.501	Biological
kegg metabolic ath	1531	4285	0.027	0.037	0.495	Biological
kegg metabolic atu	1542	4323	0.028	0.038	0.504	Biological
kegg metabolic bas	619	1501	0.052	0.075	0.613	Biological
malaria genes hvr 1	307	2812	0.443	0.522	0.563	Biological
malaria genes hvr 5	298	2684	0.263	0.337	0.555	Biological
malaria genes hvr 6	291	3251	0.572	0.610	0.574	Biological
malaria genes hvr 7	306	11688	0.589	0.635	0.574	Biological
malaria genes hvr 8	273	3933	0.701	0.719	0.541	Biological
malaria genes hvr 9	297	7562	0.503	0.530	0.530	Biological
new zealand collab	1463	4246	0.061	0.081	0.489	Social
physics collab pierreauger	475	6426	0.878	0.894	0.400	Social
plant pol robertson	1882	15254	0.017	0.019	0.505	Biological
product space hs	866	2532	0.225	0.245	0.681	Economic
product space sitc	774	1779	0.216	0.255	0.683	Economic
software dependencies colt	504	1139	0.138	0.170	0.669	Technological
software dependencies jmail	192	488	0.224	0.256	0.680	Technological
software dependencies jung	398	943	0.163	0.177	0.709	Technological
software dependencies jung c	879	2047	0.139	0.139	0.709	Technological
software dependencies org	486	2717	0.128	0.161	0.515	Technological
software dependencies scolt	504	1139	0.138	0.170	0.669	Technological
software dependencies sjbullet	249	802	0.182	0.182	0.685	Technological
software dependencies sjung	399	946	0.159	0.166	0.708	Technological
software dependencies slucene	2811	10741	0.092	0.129	0.543	Technological
tree of life 333990	178	282	0.263	0.288	0.854	Biological
tree of life 335543	641	4334	0.517	0.575	0.541	Biological
tree of life 338966	595	3140	0.478	0.545	0.603	Biological
tree of life 338969	879	5873	0.458	0.542	0.590	Biological
tree of life 339670	1360	9646	0.450	0.500	0.654	Biological
tree of life 340177	434	1477	0.457	0.496	0.723	Biological
tree of life 366602	861	4970	0.443	0.444	0.612	Biological
tree of life 406817	714	3283	0.413	0.488	0.654	Biological
tree of life 452652	1021	7516	0.470	0.523	0.637	Biological
tree of life 469383	838	7232	0.465	0.484	0.509	Biological
ugandan village friendship 1	202	547	0.071	0.075	0.781	Social
ugandan village friendship 2	181	688	0.081	0.126	0.707	Social
ugandan village friendship 3	192	1060	0.048	0.048	0.640	Social
ugandan village friendship 4	320	2076	0.035	0.060	0.663	Social
unicodelang	858	1249	0.016	0.022	0.591	Informational
urban streets ahmedabad	2870	4375	0.156	0.160	0.878	Transportation
urban streets bologna	541	771	0.124	0.131	0.900	Transportation
urban streets brasilia	179	230	0.106	0.120	0.925	Transportation
urban streets cairo	1496	2252	0.145	0.145	0.943	Transportation
urban streets paris	335	494	0.135	0.135	0.890	Transportation
urban streets venice	1840	2397	0.115	0.115	0.897	Transportation
urban streets vienna	467	691	0.087	0.087	0.922	Transportation
us agencies alabama	1115	4983	0.121	0.142	0.549	Social
us agencies florida	2010	27570	0.338	0.360	0.409	Social
us agencies indiana	1017	6505	0.148	0.148	0.587	Social
wiki science	677	6517	0.355	0.392	0.594	Informational
word adjacency french	8308	23832	0.001	0.001	0.374	Informational
word adjacency japanese	2698	7995	0.002	0.004	0.418	Informational
word adjacency spanish	11558	43050	0.002	0.002	0.361	Informational

TABLE II: Result for the Louvain method

Name	Nodes	Edges	L	AC	Balanceness	Category
caenorhabditis elegans	453	2025	0.357	0.362	0.868	Biological
central chilean power grid reduced	218	527	0.765	0.777	0.932	Technological
chicago road network	12979	20627	0.919	0.925	0.990	Transportation
clements long plant pollinator web	275	906	0.291	0.294	0.950	Biological
contact	274	2124	0.119	0.123	0.926	Social
euairtransportation multiplex	417	2953	0.228	0.235	0.864	Transportation
email	1133	5451	0.426	0.429	0.880	Social
email europe	986	16064	0.346	0.348	0.804	Social
epinions	964	14214	0.495	0.495	0.801	Economic
erdos	6927	11850	0.661	0.670	0.903	Informational
football	115	613	0.525	0.569	0.921	Social
gallery proximity	410	2765	0.650	0.660	0.855	Social
haggle human proximity network	274	2124	0.119	0.123	0.926	Social
high school dynamic contacts 2	180	2220	0.500	0.507	0.831	Social
hypertext 2009 dynamic contact network	113	2196	0.085	0.088	0.906	Social
jazz	198	2742	0.410	0.412	0.826	Social
les miserables	77	254	0.533	0.571	0.913	Informational
messel shale food web	700	6395	0.242	0.243	0.894	Biological
oregon 0	10670	22002	0.565	0.567	0.860	Technological
oregon 1	10729	21999	0.565	0.567	0.845	Technological
oregon 2	10790	22469	0.545	0.549	0.856	Technological
oregon 3	10859	22747	0.538	0.542	0.854	Technological
oregon 4	10886	22493	0.539	0.544	0.859	Technological
oregon 5	10943	22607	0.542	0.545	0.856	Technological
oregon 6	11011	22677	0.543	0.547	0.854	Informational
oregon 7	11051	22724	0.536	0.539	0.857	Technological
political books	105	441	0.468	0.477	0.874	Informational
power grid	4941	6594	0.923	0.934	0.985	Technological
pretty good privacy	10680	24316	0.849	0.867	0.931	Technological
route views	6474	12572	0.571	0.575	0.871	Informational
stem concept networks researchers	1616	3045	0.555	0.558	0.961	Informational
song of ice and fire	796	2823	0.511	0.518	0.876	Informational
star wars	109	398	0.420	0.423	0.905	Informational
trumpworld	2669	3380	0.759	0.769	0.894	Economic
us airport network	500	2980	0.319	0.325	0.842	Transportation
workplace contacts	92	755	0.291	0.302	0.924	Social
zachary	34	78	0.378	0.423	0.939	Social
arxiv collab hep th 1999	5835	13815	0.792	0.797	0.966	Social
bible nouns	1707	9059	0.435	0.436	0.857	Informational
bitcoin trust	5875	21489	0.422	0.423	0.832	Social
board directors net1m 2011 08 01	854	2745	0.856	0.862	0.960	Social
board directors net2m 2011 08 01	1066	1148	0.906	0.909	0.970	Social
copenhagen fb friends	800	6418	0.367	0.367	0.866	Social
eu procurements alt at 2008	1684	1921	0.844	0.875	0.924	Economic
eu procurements alt cz 2008	1777	2110	0.824	0.834	0.935	Economic
eu procurements alt dk 2011	2219	2904	0.752	0.764	0.933	Economic
eu procurements alt ee 2008	480	540	0.767	0.860	0.900	Economic
eu procurements alt es 2008	8097	13286	0.682	0.685	0.929	Economic
eu procurements alt fi 2008	2517	3701	0.683	0.695	0.916	Economic
eu procurements alt gr 2008	1933	2495	0.778	0.819	0.911	Economic
eu procurements alt hu 2015	2163	2929	0.781	0.790	0.938	Economic
eu procurements alt it 2008	6622	9121	0.722	0.737	0.931	Economic
eu procurements alt lt 2008	1359	2486	0.608	0.625	0.924	Economic
eu procurements alt lv 2008	1628	2269	0.751	0.770	0.916	Economic
eu procurements alt pl 2008	12456	23991	0.695	0.712	0.923	Economic
eu procurements alt pt 2008	692	776	0.843	0.899	0.935	Economic
eu procurements alt se 2008	4421	5905	0.764	0.806	0.911	Economic
eu procurements alt sk 2008	660	773	0.820	0.824	0.913	Economic
euroroad	1039	1305	0.852	0.855	0.983	Transportation
facebook friends	329	1954	0.655	0.664	0.885	Social

Table II (Continued)

Name	Nodes	Edges	L	AC	Balanceness	Category
facebook organizations l1	5793	30753	0.690	0.712	0.885	Social
facebook organizations m1	1429	19357	0.422	0.423	0.804	Social
facebook organizations s1	320	2369	0.344	0.346	0.863	Social
facebook organizations s2	165	726	0.443	0.444	0.870	Social
foodweb baywet	128	2075	0.110	0.114	0.901	Biological
foursquare nyc restaurant tips	5372	8852	0.599	0.602	0.979	Social
genetic multiplex gallus	205	234	0.479	0.490	0.742	Biological
genetic multiplex plasmodium	1158	2402	0.485	0.500	0.972	Biological
genetic multiplex rattus	2350	3484	0.688	0.709	0.878	Biological
interactome figeys	2217	6418	0.425	0.429	0.918	Biological
interactome vidal	2783	6007	0.583	0.584	0.954	Biological
kegg metabolic aae	880	2296	0.478	0.483	0.943	Biological
kegg metabolic afu	861	2011	0.529	0.537	0.948	Biological
kegg metabolic ana	1314	3552	0.483	0.485	0.950	Biological
kegg metabolic ape	769	1858	0.514	0.520	0.947	Biological
kegg metabolic atc	1538	4315	0.479	0.482	0.949	Biological
kegg metabolic atu	1542	4323	0.470	0.478	0.943	Biological
kegg metabolic bas	619	1501	0.518	0.532	0.955	Biological
malaria genes hvr 1	307	2812	0.605	0.608	0.830	Biological
malaria genes hvr 5	298	2684	0.346	0.356	0.872	Biological
malaria genes hvr 6	291	3251	0.494	0.501	0.868	Biological
malaria genes hvr 8	273	3933	0.451	0.458	0.831	Biological
new zealand collab	1463	4246	0.433	0.435	0.848	Social
physics collab pierreauger	475	6426	0.559	0.564	0.828	Social
plant pol robertson	1882	15254	0.272	0.272	0.944	Biological
product space hs	866	2532	0.692	0.728	0.907	Economic
product space sitc	774	1779	0.723	0.748	0.922	Economic
software dependencies colt	504	1139	0.622	0.623	0.914	Technological
software dependencies jmail	192	488	0.570	0.602	0.901	Technological
software dependencies jung	398	943	0.697	0.709	0.920	Technological
software dependencies jung c	879	2047	0.714	0.724	0.904	Technological
software dependencies org	486	2717	0.371	0.376	0.872	Technological
software dependencies scolt	504	1139	0.622	0.623	0.914	Technological
software dependencies sjbullet	249	802	0.517	0.521	0.900	Technological
software dependencies sjung	399	946	0.695	0.707	0.920	Technological
software dependencies slucene	2811	10741	0.670	0.682	0.901	Technological
tree of life 331678	531	1848	0.698	0.708	0.907	Biological
tree of life 333990	178	282	0.801	0.871	0.945	Biological
tree of life 338966	595	3140	0.614	0.621	0.859	Biological
tree of life 338969	879	5873	0.570	0.575	0.852	Biological
tree of life 339670	1360	9646	0.632	0.642	0.838	Biological
tree of life 340177	434	1477	0.671	0.685	0.903	Biological
tree of life 365044	820	4408	0.685	0.694	0.861	Biological
tree of life 366602	861	4970	0.697	0.715	0.851	Biological
tree of life 406817	714	3283	0.686	0.688	0.880	Biological
tree of life 452652	1021	7516	0.682	0.686	0.837	Biological
tree of life 469383	838	7232	0.616	0.619	0.816	Biological
ugandan village friendship 1	202	547	0.386	0.394	0.968	Social
ugandan village friendship 2	181	688	0.299	0.315	0.939	Social
ugandan village friendship 3	192	1060	0.238	0.240	0.931	Social
ugandan village friendship 4	320	2076	0.215	0.217	0.915	Social
unicodelang	858	1249	0.681	0.717	0.926	Informational
urban streets ahmedabad	2870	4375	0.900	0.904	0.988	Transportation
urban streets bologna	541	771	0.810	0.826	0.983	Transportation
urban streets cairo	1496	2252	0.868	0.882	0.985	Transportation
urban streets paris	335	494	0.772	0.834	0.980	Transportation
urban streets venice	1840	2397	0.903	0.914	0.987	Transportation
urban streets vienna	467	691	0.796	0.805	0.982	Transportation
us agencies alabama	1115	4983	0.464	0.465	0.887	Social
us agencies florida	2010	27570	0.487	0.490	0.798	Social

Table II (Continued)

Name	Nodes	Edges	L	AC	Balanceness	Category
us agencies indiana	1017	6505	0.326	0.329	0.919	Social
wiki science	677	6517	0.550	0.554	0.897	Informational
word adjacency french	8308	23832	0.375	0.376	0.901	Informational
word adjacency japanese	2698	7995	0.343	0.344	0.916	Informational
word adjacency spanish	11558	43050	0.295	0.295	0.855	Informational

[1] S. C. Johnson, Hierarchical clustering schemes, Psychometrika **32**, 241 (1967).

[2] 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 (2008).

[3] K. A. Heller and Z. Ghahramani, Bayesian hierarchical clustering, in *Proceedings of the 22nd international conference on Machine learning* (ACM, 2005) pp. 297–304.

[4] R. Xu and D. Wunsch, Survey of clustering algorithms, IEEE Transactions on Neural Networks **16**, 645 (2005).

[5] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, Defining and identifying communities in networks, Proceedings of the national academy of sciences **101**, 2658 (2004).

[6] 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**, P10008 (2008).

[7] M. Rosvall and C. T. Bergstrom, Maps of random walks on complex networks reveal community structure, Proceedings of the National Academy of Sciences **105**, 1118 (2008), https://www.pnas.org/doi/pdf/10.1073/pnas.0706851105.

[8] F. Murtagh and P. Contreras, Algorithms for hierarchical clustering: an overview, WIREs Data Mining and Knowledge Discovery **2**, 86 (2012), https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/widm.53.

[9] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann, Link communities reveal multiscale complexity in networks, Nature **466**, 761 (2010).

[10] J. C. Dunn, Well separated clusters and optimal fuzzy partitions, Journal of Cybernetics **4**, 95 (1974).

[11] G. W. Milligan and M. C. Cooper, An examination of procedures for determining the number of clusters in a data set, Psychometrika **50**, 159 (1985).

[12] D. L. Davies and D. W. Bouldin, A cluster separation measure, IEEE Transactions on Pattern Analysis and Machine Intelligence **1**, 224 (1979).

[13] K.-T. Shao and R. R. Sokal, Tree Balance, Systematic Biology **39**, 266 (1990), https://academic.oup.com/sysbio/article-pdf/39/3/266/39805211/sysbio_39_3_266.pdf.

[14] M. Fischer, L. Herbst, S. Kersting, L. Kühn, and K. Wicke, Tree balance indices: a comprehensive survey (2021).

[15] K.-T. Shao and R. R. Sokal, Tree Balance, Systematic Zoology **39**, 266 (1990), publisher: [Oxford University Press, Society of Systematic Biologists, Taylor & Francis, Ltd.].

[16] J. Lemant, C. Le Sueur, V. Manojlović, and R. Noble, Robust, Universal Tree Balance Indices, Systematic Biology **71**, 1210 (2022), https://academic.oup.com/sysbio/article-pdf/71/5/1210/48454082/syac027.pdf.

[17] A. Vogogias, J. Kennedy, D. Archambault, V. A. Smith, and H. Currant, MLCut: Exploring Multi-Level Cuts in Dendrograms for Biological Data, in *Computer Graphics and Visual Computing (CGVC)*, edited by C. Turkay and T. R. Wan (The Eurographics Association, 2016).

[18] P. Langfelder, B. Zhang, and S. Horvath, Defining clusters from a hierarchical cluster tree: the Dynamic Tree Cut package for R, Bioinformatics **24**, 719 (2007), https://academic.oup.com/bioinformatics/article-pdf/24/5/719/16885227/btm563.pdf.

[19] P. J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, Journal of computational and applied mathematics **20**, 53 (1987).

[20] S. Lehmann and L. K. Hansen, Deterministic modularity optimization, The European Physical Journal B **60**, 83 (2007).

[21] R. Guimerà and L. A. N. Amaral, Functional cartography of complex metabolic networks, Nature **433**, 895 (2005).

[22] T. P. Peixoto, Efficient monte carlo and greedy heuristic for the inference of stochastic block models, Physical Review E **89**, 012804 (2014).

[23] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, Equation of state calculations by fast computing machines, The Journal of Chemical Physics **21**, 1087 (1953).

[24] W. K. Hastings, Monte carlo sampling methods using markov chains and their applications, Biometrika **57**, 97 (1970).

[25] S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi, Optimization by simulated annealing, science **220**, 671 (1983).

[26] H. H. Szu and R. Hartley, Fast simulated annealing, Physics Letters A **122**, 157 (1987).

[27] R. M. Neal, Probabilistic inference using markov chain monte carlo methods, (1993).

[28] S. Fortunato and M. Barthelemy, Resolution limit in community detection, Proceedings of the national academy of sciences **104**, 36 (2007).

[29] A. Lancichinetti and S. Fortunato, Limits of modularity maximization in community detection, Physical Review E—Statistical, Nonlinear, and Soft Matter Physics **84**, 066122 (2011).

[30] P. W. Holland, K. B. Laskey, and S. Leinhardt, Stochastic blockmodels: First steps, Social networks **5**, 109 (1983).

[31] B. Karrer and M. E. Newman, Stochastic blockmodels and community structure in networks, Physical Review E—Statistical, Nonlinear, and Soft Matter Physics **83**, 016107 (2011).

[32] 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**, P10008 (2008).

[33] T. T. Tanimoto, Elementary mathematical theory of classification and prediction, (1958).

[34] M. E. Newman, Fast algorithm for detecting community structure in networks, Physical review E **69**, 066133 (2004).