Spanning tree separation reveals community structure in networks

Jongkwang Kim[1,*] and Thomas Wilhelm[2,†]
[1]*School of Electrical and Computer Engineering, Hanyang University, Ansan, Kyunggi-Do 425-791, Korea*
[2]*Theoretical Systems Biology, Institute of Food Research, Norwich Research Park, Norwich NR4 7UA, United Kingdom*

(Received 8 November 2012; published 29 March 2013)

We present a simple, intuitive, and effective approach for network clustering. It is based on basic concepts of linear algebra such as efficient calculation of spanning trees, and can be implemented in a few lines of code. We introduce the node separation measure spanning tree separation (STS) and the corresponding graph distance measure spanning tree vector similarity distance (STVSD). We demonstrate that the STS is a link salience measure able to identify the backbone of networks. The STVSD is used to reveal the hierarchical community structure of networks. We show that it, together with the clustering quality measure partition density, is on a par with the best graph or network clustering methods known, in terms of both quality and efficiency. In perspective, we note that our approach could also handle weighted and directed networks and could be used for identification of overlapping communities.

DOI: 10.1103/PhysRevE.87.032816 PACS number(s): 89.75.Hc

I. INTRODUCTION

Networks are simple and informative representations of natural and social systems. Here we focus on the simplest form of unweighted and undirected networks, i.e., we consider graphs $G = (V,E)$ with n vertices (nodes) V and m edges E, but the method can easily be extended to also deal with directed and/or weighted edges.

Community structure is one of the most important features of networks. A community might represent a group of friends or colleagues or molecules involved in processes of a specific cellular function. Not surprisingly, much has been found about network or graph communities and many corresponding clustering methods have been developed [1,2], in particular during the last decade, starting with the famous Girvan-Newman approach [3]. The latter provides a hierarchy of clusters. Different objective functions have been developed to identify the optimal level of the hierarchy, i.e., the "right" number of clusters. The first well known one is the modularity measure Q which is high if there are many more intercluster edges than would be expected in a corresponding random version [4]. However, it was found that Q suffers from a bias towards the same cluster size and a resolution limit, a failure to identify small clusters [5]. Information theoretic approaches have been developed to overcome Q's same-size bias, such as the minimum description length (MDL) [6], but the MDL still suffers from a resolution limit [7]. It has been shown that multiresolution versions of Q can overcome the resolution limit [8,9]. However, for networks with very different cluster sizes these methods still struggle, and it was recently conjectured that this might be a general flaw of all methods based on the optimization of a global measure [10]. The partition density (PD) seems the best cluster objective function known so far, overcoming both of Q's problems [11]. Hierarchical clustering of nodes assigns a unique cluster to each node, but in reality communities often overlap [1,2,12]. The PD has already been used to reveal hierarchical and overlapping communities [11].

Here we discuss four "distance" measures on graphs: the known resistance distance (RD) [13], as well as three additional measures: the spanning tree separation (STS), an easier to calculate relative of the RD; and two deduced distance measures based on vector similarities, the resistance vector similarity distance (RVSD) and the spanning tree vector similarity distance (STVSD). We use all four measures for graph clustering and demonstrate that the STVSD, together with the objective function PD, provides a graph clustering method that is on a par with Infomap [14], the best corresponding method available, in terms of both quality and efficiency. Note that our method the STVSD-PD is not globally optimizing the partition density, but it establishes a cluster dendrogram and just applies the PD for identification of the best cluster level. Accordingly, we demonstrate that the STVSD-PD method does not suffer from the conjectured problem of global measure optimization methods [10]. Finally, we show that the STS is also a valuable measure to quantify the importance of edges and to identify a corresponding network backbone.

II. METHODS

A. Spanning tree separation and resistance distance

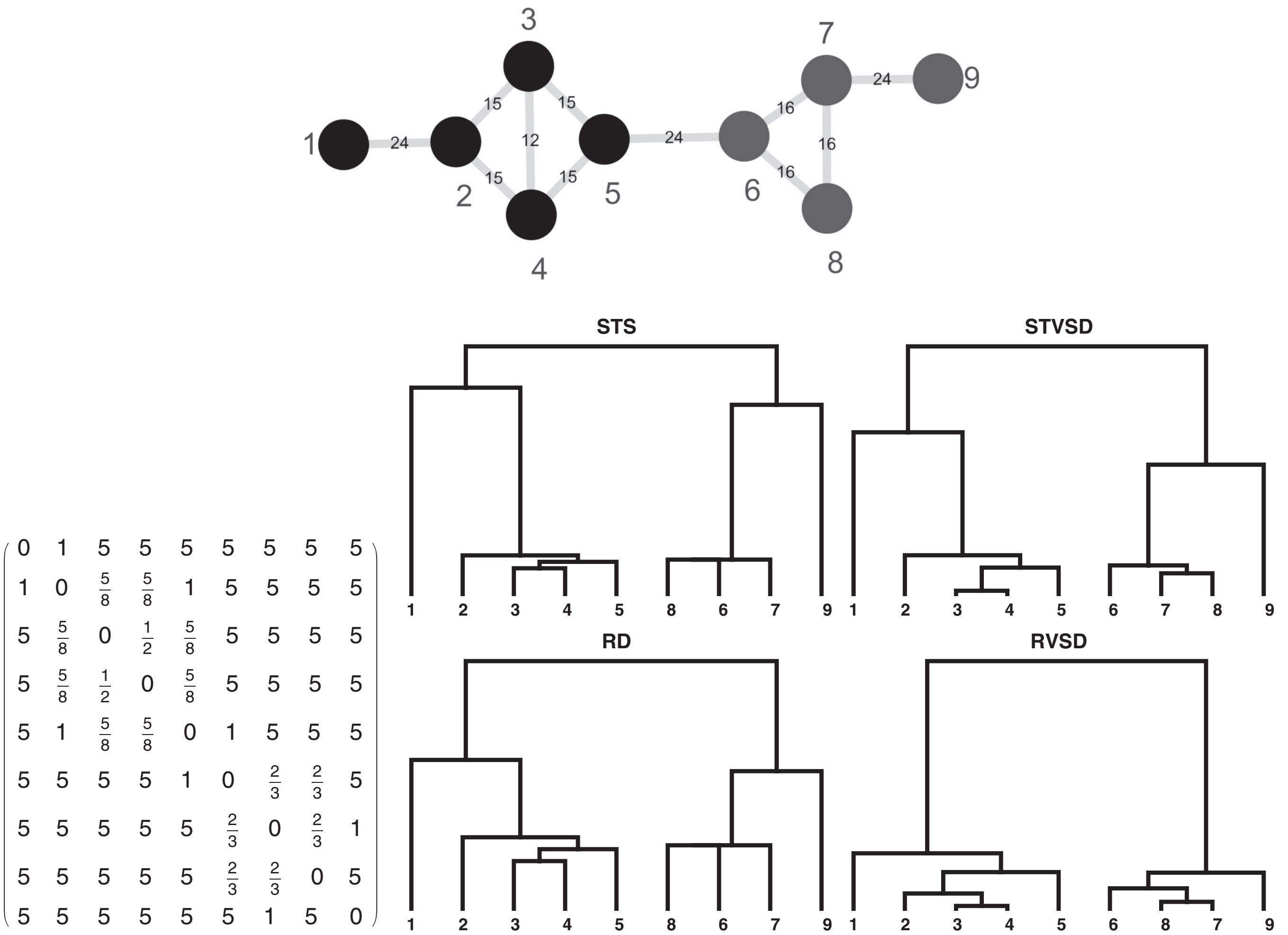
Kirchhoff's matrix tree theorem allows efficient calculation of the number of spanning arborescences rooted at a vertex v in a digraph D: it is equal to the determinant of the Kirchhoff matrix $\mathbf{K}$ (the degree matrix minus the adjacency matrix) of D, with the rows and columns corresponding to v being deleted [15]. In the simpler case of a graph G the corresponding result is as follows: the number $\tau(G)$ of spanning trees (STs) of G is equal to the determinant of the augmented Kirchhoff matrix $\mathbf{K}_a$ of G (the augmented graph contains an added leaf node, i.e., $\mathbf{K}_a$ is identical to $\mathbf{K}$, with 1 added to any main diagonal entry).

We define the spanning tree separation STS of two nodes corresponding to an edge e_{ij}

$$\frac{\tau(G)_{e_{ij}}}{\tau(G)} \quad (\{i,j\} \in E) \tag{1}$$

[*] Present address: Department of Life and Pharmaceutical Science, Ewha Woman's University, 52 Ewhayeodae-gil, Seodaemun-gu, Seoul 120-750, Korea.

[†] Corresponding author: thomas.wilhelm@ifr.ac.uk



$$\begin{pmatrix} 0 & 1 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ 1 & 0 & \frac{5}{8} & \frac{5}{8} & 1 & 5 & 5 & 5 & 5 \\ 5 & \frac{5}{8} & 0 & \frac{1}{2} & \frac{5}{8} & 5 & 5 & 5 & 5 \\ 5 & \frac{5}{8} & \frac{1}{2} & 0 & \frac{5}{8} & 5 & 5 & 5 & 5 \\ 5 & 1 & \frac{5}{8} & \frac{5}{8} & 0 & 1 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 1 & 0 & \frac{2}{3} & \frac{2}{3} & 5 \\ 5 & 5 & 5 & 5 & 5 & \frac{2}{3} & 0 & \frac{2}{3} & 1 \\ 5 & 5 & 5 & 5 & 5 & \frac{2}{3} & \frac{2}{3} & 0 & 5 \\ 5 & 5 & 5 & 5 & 5 & 5 & 1 & 5 & 0 \end{pmatrix}$$

FIG. 1. A simple example graph. Edge labels denote the number of spanning trees running through the edge. Node color corresponds to the two different clusters as consistently identified by all methods (see Table I). The matrix shows the spanning tree distance (STS, $c = 4$) and the dendrograms the average linkage hierarchical clustering for the STS, STVSD, RD, and RVSD.

as the normalized number of STs of G that contain the edge e_{ij}. If all STs of G contain e_{ij}, the separation is maximum, i.e., 1, if few STs pass e_{ij} it is low (see Fig. 1). Note that the STS is also a betweenness measure for the edges, so it could directly be used for Girvan-Newman- (GN-) like graph clustering [3]. Obviously, leaf edges also have high betweenness according to the STS, which might be considered as inappropriate. However, although one could easily remove leaves before clustering, our experience shows that this is not necessary because generally leaves do not corrupt our graph clustering (using average linkage and/or vector similarity distance; see below).

Fortunately, we can calculate the STS for all m edges efficiently: it was shown that Kirchhoff's matrix tree theorem can be used to obtain efficiently the number of arborescences containing an arc, for all arcs of a digraph [16]. In the simpler case of a graph the number of spanning trees containing an edge e_{ij} is

$$\tau(G)_{e_{ij}} = b_{ii} + b_{jj} - 2b_{ij} \tag{2}$$

with $\mathbf{B} = \{b_{ij}\} = \mathrm{Det}(\mathbf{K}_a) \cdot \mathbf{K}_a^{-1}$ ($\mathbf{K}_a$ is the augmented Kirchhoff matrix).

Of course, based on the matrix $\mathbf{B}$, one can calculate numbers for all node pairs, not just for edges. This all-node-pairs distance is identical to the resistance distance [13], a distance function defined for undirected graphs. The RD for two nodes corresponding to an edge e_{ij} can most easily be calculated by

$$p_{ii} + p_{jj} - 2p_{ij}, \tag{3}$$

where $\mathbf{P}$ denotes the Moore-Penrose pseudoinverse of the Kirchhoff matrix $\mathbf{K}$ [17]. The maximum RD for linked nodes is 1 (see above); unlinked nodes typically have RD greater than 1. We define the spanning tree separation of two unlinked nodes i and j as

$$1 + c(c > 0)(\{i,j\} \notin E). \tag{4}$$

Note that the size of c can modulate the clustering results. However, c has rarely any effect when average linkage clustering is used. For sufficiently large c the clustering is always stable. We set $c = 10^6$ for all random network benchmarking and real world graph calculations.

Note that the RD fulfills the axioms of a distance metric [13], but the STS does not (it can violate the triangle inequality). We therefore denote STS as a separation, instead of a distance. However, we demonstrate that the STS is a much better basis for graph clustering than the full RD (see Sec. III). We also show that the STS is a link salience measure [18] able to identify the backbone of networks (see Sec. III).

B. Spanning tree vector similarity distance and resistance vector similarity distance

Based on the STS we calculate a proper distance matrix by calculating vector similarities between all row pairs of the STS. We use the Spearman rank correlation r_{Spear} as a robust similarity measure. Note that using r_{Spear} any $c > 0$ gives the same result; the distance rank of unlinked edges is then always maximal. The vector similarity distance (VSD) is $1 - r_{\text{Spear}}$ (Fig. 1). The RVSD is defined accordingly, using the RD instead of the STS. We checked a number of other vector similarity measures, but r_{Spear} gave the best results.

C. Hierarchical clustering

We use the distance matrices STS and STVSD and RD and RVSD for corresponding hierarchical clustering. Note that these matrices could also provide a basis for identification of overlapping communities, but here we do not elaborate on this. We use average linkage clustering [the unweighted pair group method with arithmetic mean (UPGMA)], providing cluster hierarchies which can be visualized by corresponding dendrograms (Fig. 1). We also tested other different agglomerative linkages such as single (equivalent to the minimal spanning tree corresponding to the distance matrix) and complete as well as divisive clustering, but the benchmark curves for graph clustering (results not shown here) were generally inferior to the results from the UPGMA method (shown in Sec. III). Note that single linkage yields identical results for the STS and RD.

D. Optimal number of clusters

Based on the cluster hierarchies, we applied the following three measures for identification of the optimal cut in the hierarchy (discussed in the Introduction), i.e., identification of the optimal number of communities: the modularity Q [4], the minimum description length [6], and the partition density [11].

E. Time complexity

Matrix inversion is the bottleneck for STS and RD calculations. Using the Williams algorithm this can be done in $O(n^{2.3727})$ time steps [19], so this is the time complexity of RD clustering. However, off-diagonal entries of the inverse of a large symmetric sparse matrix (such as the augmented Kirchhoff matrix of an undirected graph) can be computed much more efficiently [20]. So the STS can be obtained in $\sim O(m)$ time steps (the precise numbers depend on graph topology).

Generally, the complete distance matrix based on vector similarities needs $O(n^2k)$ time steps, where k denotes the number of entries per row (the number of features used for similarity calculation). However, given the general importance of similarity matrices it is not surprising that much work has been done to improve the performance of corresponding calculations. Using fingerprints for each vector and corresponding hash functions a distance matrix can be calculated in $O(nk)$ time steps [21]. Average linkage hierarchical clustering (the UPGMA method) needs $O(n^2)$ steps [22].

A straightforward search for the maximum Q, MDL, or PD, given the complete cluster hierarchy, needs $O(n^3)$ time steps. Fortunately, we can recursively calculate the PD during average linkage clustering in $O(n^2)$ time steps, as follows. Given a graph with p clusters, the link density of cluster i is

$$D_i = [m_i - (n_i - 1)] \Big/ \left[\frac{n_i(n_i - 1)}{2} - (n_i - 1)\right],$$

normalized by the minimum and maximum numbers of links possible (m_i and n_i are the numbers of edges and nodes in cluster i) [11]. Partition density (PD) is the average of D_i weighted by the fraction of present links $(1/m)\sum_i m_i D_i$ [11]. Agglomerative UPGMA starts with a singleton partition, then successively connects clusters separated by the lowest average distance. The only difference in the PD for two successive partitions arises due to the one cluster arising from merging of two smaller clusters, so the PD at merging level l can be calculated from the previous PD at merging level $l - 1$ in $O(1)$:

$$\mathrm{PD}_l = \mathrm{PD}_{l-1} - \frac{m_a D_a + m_b D_b}{m} + \frac{m_{\mathrm{cl}} D_{\mathrm{cl}}}{m}$$

(a and b refer to the two subsets being merged to the larger cluster "cl").

Summing up, the time complexity for PD graph clustering based on the STS and STVSD is $O(n^2)$ [based on the RD and RVSD time complexity is $O(n^{2.3727})$].

F. Cluster quality: Benchmark curves

We use a standard procedure [2] to test the quality performance of our method. Specifically, the GN method [3] (Fig. 2) benchmarks clustering of random computer-generated

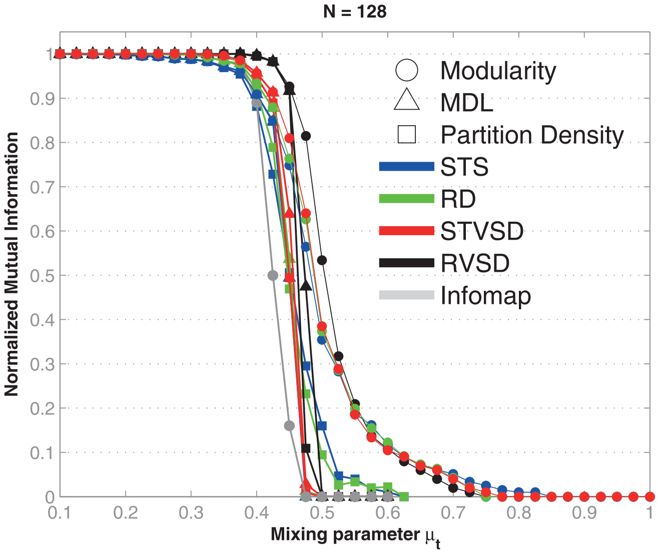


FIG. 2. (Color) GN benchmark ($n = 128$).

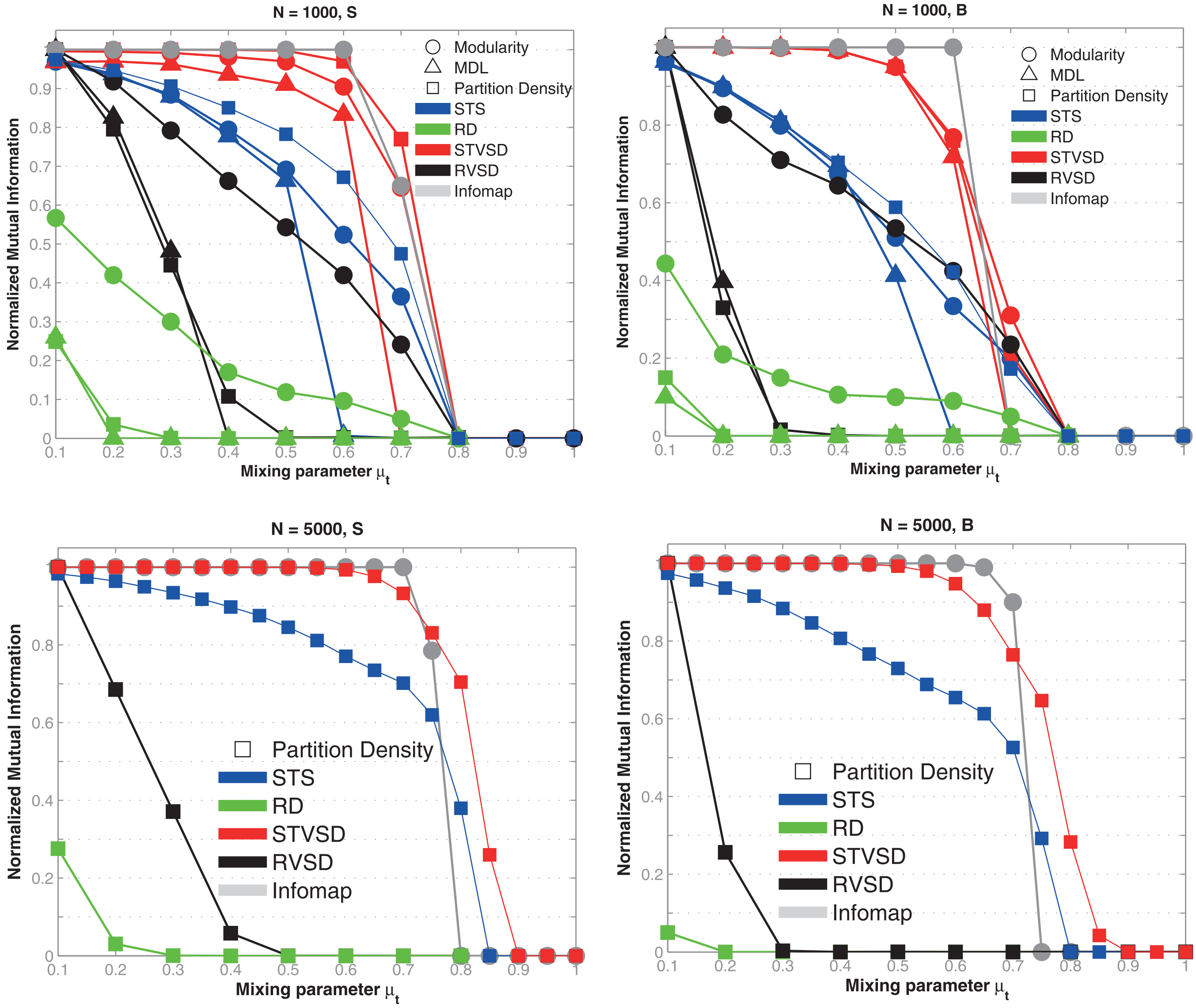


FIG. 3. (Color) LFR benchmark ($n = 1000$ and 5000).

graphs with four clusters of size $n = 32$. The higher the mixing parameter μ_t gets, the less well defined the clusters are. The LFR benchmark [23] (Figs. 3 and 4) evaluates clustering of random graphs with clusters of different sizes and nodes of different degrees.

The mutual information (MI) is used to quantify the similarity of the obtained clustering to the underlying known one [24]. The MI is zero if the whole graph is assigned to one cluster. In the case of GN benchmarking and $\mu_t = 1$ all three quality measures Q, MDL, and PD are maximum for one cluster; the benchmark curves approach zero for high μ_t (Fig. 2). However, in the case of LFR benchmarking with 1000- and 5000-node graphs, even for $\mu_t = 1$ the maximum Q, MDL, and/or PD was sometimes obtained for more than one cluster (but always a low number), implying positive MI. So for Fig. 3 we adjusted the MI as follows: we always assumed the optimum number of clusters to be 1 (MI of zero) if the maximum quality measure (M) was lower than for the value obtained for one cluster plus a constant ($\text{M}_{\text{max}} < \text{M}|_{1\ \text{cluster}} + \tilde{c}$). To obtain an appropriate constant $\tilde{c}$ for a given μ_t we simulated 100 random graphs and considered the distributions of all positive differences $\mathcal{D} = \text{M} - \text{M}|_{1\ \text{cluster}}$. Figure S1 in the Supplemental Material [25] shows (for the quality measure PD and $n = 5000$) that this distribution becomes narrower and moves closer to zero for increasing μ_t. We take as the constant $\tilde{c}$ the maximum $\mathcal{D}$ actually obtained in the 100 simulations for $\mu_t = 1$. Note that this adjustment affects (decreases) only the benchmark curves for high μ_t (greater than 0.7 for $n = 1000$; greater than 0.8 for $n = 5000$). For lower μ_t M_{max} is always larger than $\text{M}|_{1\ \text{cluster}} + \tilde{c}$ (Fig. S1).

III. RESULTS

A. Simple example

We start with a simple example where all 24 spanning trees can easily be found by hand. Figure 1 shows the (undirected) graph, the STS matrix, and the average linkage hierarchical clustering for the STS, STVSD, RD, and RVSD. All methods

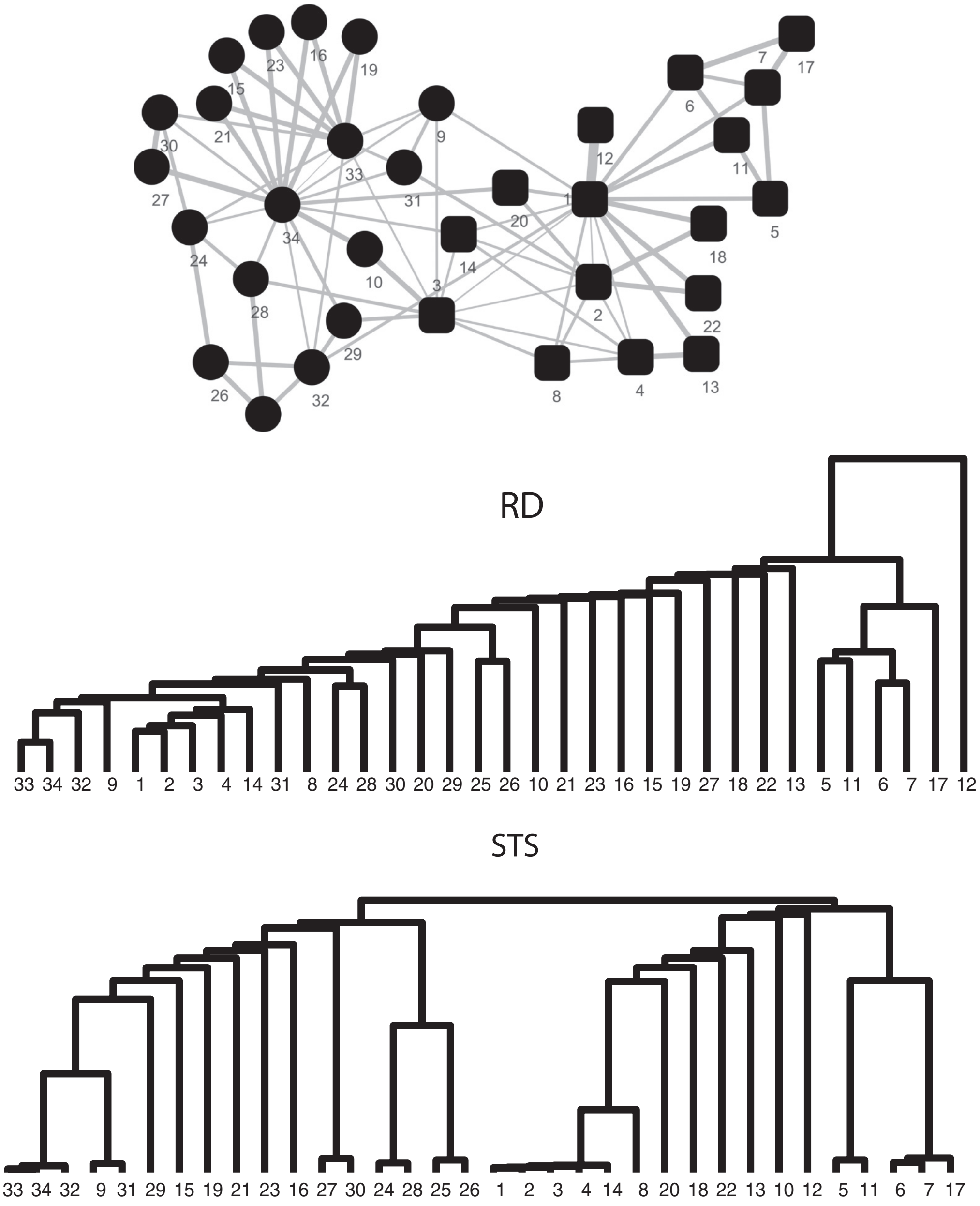


FIG. 4. The Karate network [26]; node shapes indicate the two known communities. The dendrograms show UPGMA clustering results, according to the RD and STS ($c = 10$).

yield 2 as the “correct” number of clusters (see Table I). Note that a rather small constant c was chosen for the STS ($c = 4$), such that the fine structure of the clusters remains visible. However, a higher c does not change the cluster hierarchy at all.

B. Karate network

Figure 4 shows the well-known Karate network [26] and UPGMA clustering dendrograms according to the RD and STS. In contrast to the RD, the STS is able to identify the known two communities. This example demonstrates that the STS is a better basis for clustering than the RD because it weights direct connections by graph edges much more highly than indirect ones, whereas the RD also assigns low distances to nonconnected nodes. Consider leaf node 12 as an example cluster to understand the advantage of the STS (the leaves themselves are never a problem; see Sec. II A). It has a high average distance from all other node clusters. For the RD it is the outlier node, but for the STS it correctly comes closer to node 1.

C. Benchmark curves

Figures 2, 3, and 5 show benchmark graph clustering curves according to the methods of GN [3] (same cluster sizes, Fig. 2) and LFR [23] (different cluster sizes, Figs. 3 and 5). All data points show mean results of 100 corresponding random networks. According to the simple GN benchmarking, all four of our considered distance matrices (STS, STVSD, RD, RVSD) perform better than the state-of-the-art method Infomap [14], which is “often considered the most accurate method

TABLE I. Predicted number of communities for two example graphs and five well known real-world graphs, using four separation or distance metrics and three cluster quality measures. The STVSD-PD results are highlighted in bold.

	Method												
	STS			RD			**STVSD**			RVSD			
Network([a])	Q	MDL	PD	Q	MDL	PD	Q	MDL	**PD**	Q	MDL	PD	Infomap
Example (2)	2	2	2	2	2	2	2	2	**2**	2	2	2	2
Ring (15)	9	2	15	9	2	15	8	2	**15**	8	2	15	15
Karate (2) [26]	3	2	3	3	1	1	4	2	**4**	3	2	3	3
Football (12) [3]	10	10	12	15	11	1	10	10	**12**	9	5	5	12
Chesapeake (3) [27]	4	2	2	19	1	1	4	2	**2**	8	1	1	4
Jazz (2) [28]	23	6	1	13	9	1	5	9	**2**	4	3	3	7
Dolphins (2) [29]	15	2	2	17	1	1	4	3	**5**	17	2	2	6

[a]known number of partitions

available" [11]. STS-based clustering outperforms RD-based clustering in the more complex and more realistic case of unequal cluster sizes (as addressed by LFR benchmarking). According to Fig. 3, the STVSD-PD results are on a par with those of Infomap. It has been shown already that one can find parameters for multiresolution methods [8,9] outperforming Infomap in these benchmarks [30]. However, it was demonstrated recently that these global measure optimization methods perform well because the spread of different cluster sizes is still small in these benchmark problems [10]. More demanding LFR benchmarking with cluster sizes varying from 10 to 1000 reveals problems of the multiresolution methods [8,9]. Figure 5 shows that the STVSD-PD method deals well also with these very challenging problems. In fact, comparison with Fig. 9 in Ref. [10] shows that the STVSD-PD methods apparently outperforms [8,9] and that it is even better than two other tested modern methods, the constant Potts model [30] and Order Statistics Local Optimization Method (OSLOM) [31].

Newman's modularity Q, known for its bias towards same cluster size [5], performs best in GN benchmarking and even surprisingly well in LFR benchmarking. However, the partition density is the best measure in the latter case.

D. Real world graphs

Table I shows average linkage clustering results for five well-known real world graphs and two simple example networks (the example from Fig. 1 and a ring connecting 15 four-node loops), all with "known" community structure (given in parentheses). It shows that the two spanning tree PD methods, STS and STVSD, perform best. Only Infomap and our four PD methods are able to cluster the ring correctly; the MDL and Q methods clearly suffer from the resolution limit. Figures S2–S6 in the Supplemental Material [25] show for the five real world graphs the STVSD hierarchical clustering dendrogram, together with the corresponding quality measure curves PD, MDL and Q, as well as the graphs with already known (shape) and STVSD-PD (color) communities.

In summary, taking the benchmark results (Figs. 2, 3, and 5) and the real world graphs clustering together we see that the vector similarity distance clustering based on the spanning

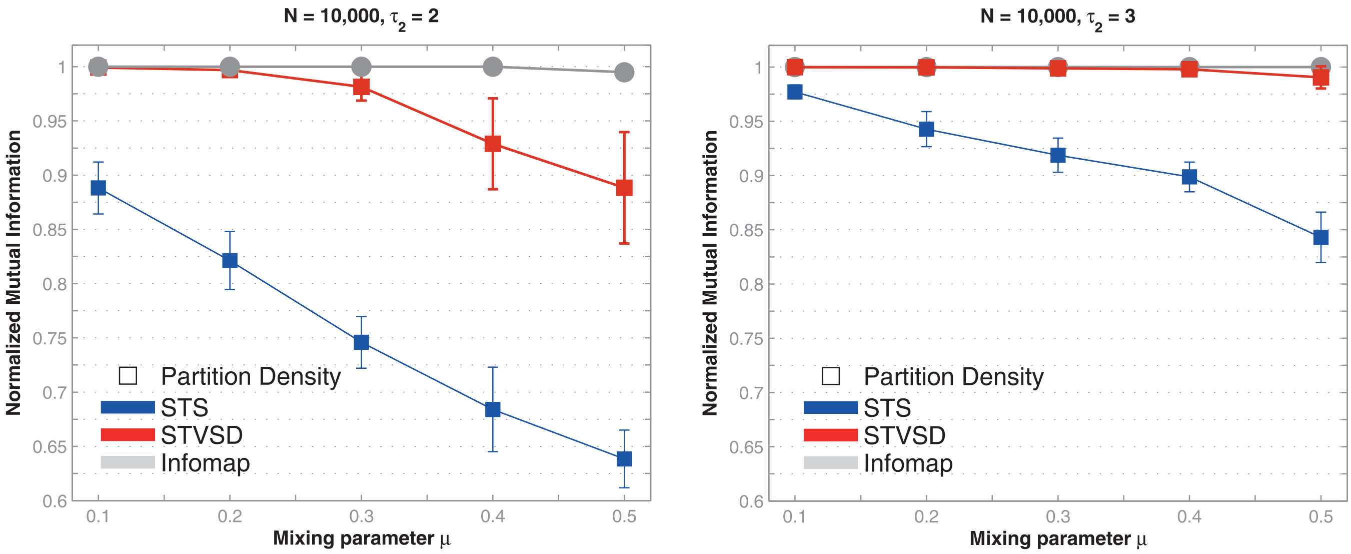


FIG. 5. (Color) Challenging LFR benchmark with cluster sizes from 10 to 1000 vertices ($n = 10\,000$).

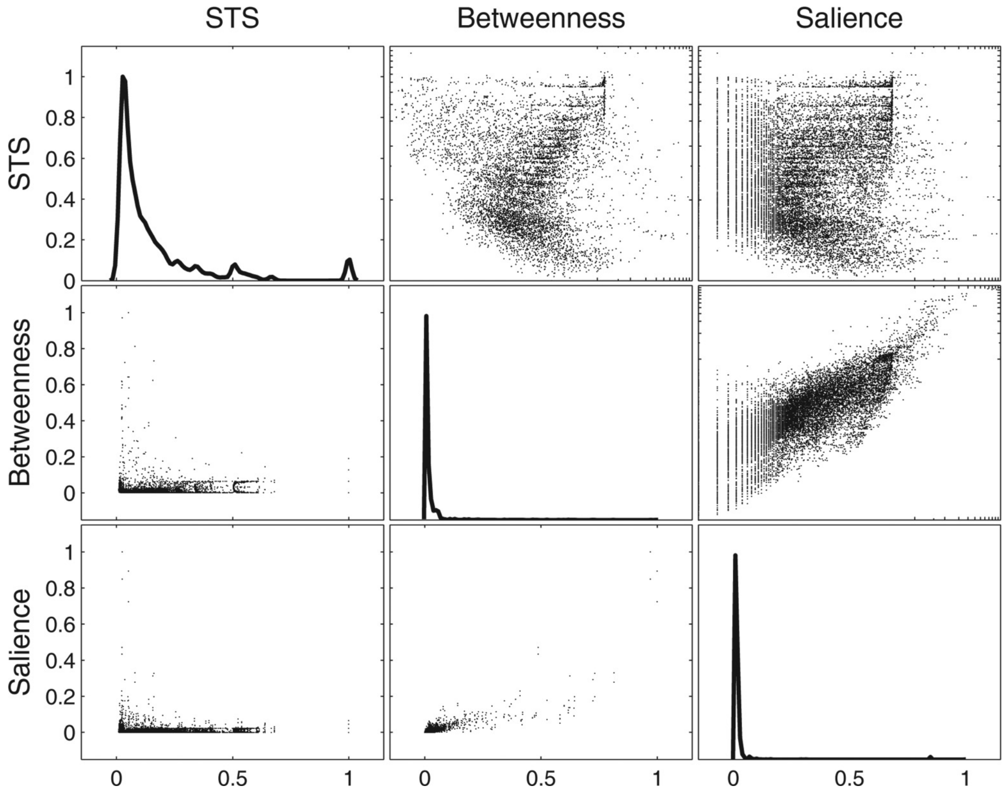


FIG. 6. Comparison of the betweenness (edge importance) measures STS, Newman's betweenness [35], and link salience [18], for the US air traffic network. The main diagonal shows the distribution of the three measures, the upper triangle scatter plots on a log-log scale, and the lower triangle the same scatter plots on a linear scale.

tree distance together with partition density optimization (the STVSD-PD method) reliably reveals community structure in natural and social systems.

E. STS is a link importance (salience) measure

Recently, a new measure has been presented to quantify the importance of links in weighted networks, called link salience [18]. Studying different examples, a bimodal distribution was found: links are either highly or not important. The important links represent the informative backbone of a network. Link salience shares the underlying edge centrality idea [3] with the STS: the more pathways go through an edge, the more important it is. The STS uses spanning trees for pathways; link salience is focused on shortest paths which can be calculated by Dijkstra's algorithm [32]. As was noted, in unweighted networks typically many different shortest paths exist for a given node pair [18]. By calculating all of them it seems one could even reproduce the STS results, but for a much higher computational cost of course: the Dijkstra algorithm alone needs $O(n^3)$ time steps for all node pairs, and all alternative shortest paths have to be calculated as well. Interestingly, link salience has already been used for some specific network clustering, but using completely different similarity measures [33].

We analyzed the unweighted US air traffic network (air traffic is a central example in Ref. [18]) (data from Ref. [34]) to study the STS distribution and its relation to link salience and another well-known betweenness measure [35]. Figure 6 shows that the STS follows a broad multimodal distribution, implying that it well separates edges according to their network importance. The link salience and betweenness distributions are unimodal, i.e., showing no distinct network backbone. This is in agreement with the previous finding that unweighted networks do not exhibit bimodal link salience distributions [18]. According to our results, the missing bimodality of link salience in unweighted networks results from the fact that link salience considers only one shortest path between two nodes (per shortest path tree), but the typically existing many alternatives are not considered. It therefore seems that the STS is the better link importance measure for unweighted networks. Interestingly, there is only weak correlation of the STS with betweenness ($r = 0.22$) and link salience ($r = 0.16$), but the latter two are highly correlated ($r = 0.81$). This indicates that the STS captures different aspects of link centrality. Figure S7 in Ref. [25] shows the complete US air traffic network. Solid red lines indicate the backbone, i.e., all edges with STS greater than or equal to 0.45 (edges with STS of 1 are just leaves in this example network, so not highlighted). Note that only 0.56% of all edges belong to the backbone. Figure S8 in Ref. [25] shows the STVSD-PD clustering, resulting in two large and three small communities [841, 306, 44, 7, and 3 (a triangle with two US Virgin Islands and one Puerto Rico airport) nodes]. Only backbone edges are shown; the node size corresponds to the node degree.

IV. DISCUSSION

Graph or network clustering is a complex task having many facets. Different methods focus on different aspects and "it remains impossible to decide which algorithm does the best job" [1]. It is advantageous to have different high-quality alternative methods to test for results best fitting to the specific problems in question.

We have presented an approach for network clustering that is based on the efficient calculation of the number of spanning trees. We had used spanning tree calculations before to define the complexity of graphs [36]. Here we show that it also provides a valuable basis for graph clustering.

Since the STS can be interpreted as a centrality measure of edges we also tested the original GN approach [3] of hierarchical divisive clustering (but without recalculation). We removed successively edges with the highest betweenness (according to the STS); the corresponding benchmark performance was (sometimes only slightly) inferior to the shown best agglomerative clustering (Figs. 2 and 3). However, it is worth noting that this approach is still much better than the original GN method [3], in terms of both quality and speed.

The resistance distance is a well-known graph distance measure [13], but it seems the RD has not been used for graph clustering. That might be due to performance problems, as we have shown for the Karate network and in the LFR benchmark curves (Fig. 3). Interestingly, we have demonstrated that the simpler spanning tree separation is very appropriate for graph clustering. It is equivalent to the RD for node pairs linked by an edge but assigns higher distances to unconnected nodes. In contrast to the RD, the STS does not fulfill the axioms of a distance metric, so we call it a separation instead of a distance. Note that the calculation of the STS matrix needs only time $\sim O(m)$ which is much faster than the calculation of the RD, which needs time of $O(n^{2.3727})$. So, the STS-based clustering is both more efficient and of higher quality than RD clustering.

Benchmark curves (Figs. 2, 3, and 5) show that STS clustering is on a par with Infomap results [14], "often considered the most accurate method available" [11]. More precisely, it is the method of spanning tree vector similarity distance–partition density that generally gives the best results.

Newman's Q is by far the most studied modularity measure [1]. In addition to Q's well-known problems mentioned in the Introduction two more problems have been found recently. Sparse networks tend to have unreasonably high modularity [37], and the Q value typically lacks a clear global maximum, having many different high-scoring solutions [38]. We have systematically tested the three modularity measures Q, MDL, and PD and found that the PD is not only easiest to calculate but also gave the best results. Given all the effort spent on better understanding Q our results imply that focus might better shift towards the PD.

We have demonstrated that STS clustering can reliably identify known community structure in real world networks. Moreover, real world networks typically possess a hierarchical organization where large communities are composed of smaller communities which in turn include smaller communities [1]. So it is important to note that our method, in contrast to Infomap, efficiently provides a complete community hierarchy of high quality which is more informative than just the one most appropriate cluster level.

We have also shown that the STS is a useful link salience measure [18], quantifying the importance of links and identifying corresponding network backbones. In the example studied here (US air traffic), the STS is more informative than link salience [18] and the well-known betweenness measure [35]. Given that link salience is typically unimodular for unweighted networks [18], it is likely that the STS will be advantageous for such networks in general.

The presented network clustering methods need only a few lines of code (in a high-level programming language) for a straightforward implementation to get the demonstrated high-quality results. We think this is an attractive feature of our approach, but we have also shown that highly sophisticated algorithms could be employed which are harder to implement but lead to significantly reduced time complexity.

In this article we have focused on disjoint node clustering of graphs, but more general cases can also be handled. As we have shown in Sec. II, our approach works for digraphs as well. Moreover, after calculating the spanning tree distances corresponding to all edges it would be possible to put additional weightings on these entries in the STS or RD results, according to the weights of the weighted directed or undirected edges (e.g., the shortest path distance), so such networks could be handled as well. Finally, we note that overlapping communities are reflected by corresponding structures in the STS or RD matrix, so we conjecture that this is also an appropriate basis for identification of overlapping network communities.

ACKNOWLEDGMENTS

We thank an anonymous referee for indicating the link salience. T.W. gratefully acknowledges the support of the Biotechnology and Biological Sciences Research Council (BBSRC). This research was funded by the BBSRC Institute Strategic Program Grant No. BB/J004529/1. J.K. acknowledges support by the Korean Research Foundation funded by the Korean Government (MOEHRD) (Grant No. KRF-612-2006-1-C-00031).

[1] S. Fortunato, Phys. Rep. **486**, 75 (2010).

[2] A. Lancichinetti and S. Fortunato, Phys. Rev. E **80**, 056117 (2009).

[3] M. Girvan and M. E. J. Newman, Proc. Natl. Acad. Sci. USA **99**, 7821 (2002).

[4] M. E. J. Newman and M. Girvan, Phys. Rev. E **69**, 026113 (2004).

[5] S. Fortunato and M. Barthelemy, Proc. Natl. Acad. Sci. USA **104**, 36 (2007).

[6] M. Rosvall and C. T. Bergstrom, Proc. Natl. Acad. Sci. USA **104**, 7327 (2007).

[7] L. K. Branting, in *Advances in Social Network Mining and Analysis*, edited by C. L. Giles, M. Smith, J. Yen, and H. Zhang, Lecture Notes in Computer Science, Vol. 5498 (Springer, Heidelberg, 2010), pp 114–130.

[8] J. Reichardt and S. Bornholdt, Phys. Rev. E **74**, 016110 (2006).

[9] A. Arenas, A. Fernandez, and S. Gomez, New J. Phys. **10**, 053039 (2008).

[10] A. Lancichinetti and S. Fortunato, Phys. Rev. E **84**, 066122 (2011).

[11] Y.-Y. Ahn, J. P. Bagrow, and S. Lehrmann, Nature (London) **466**, 761 (2010).

[12] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, Nature (London) **435**, 814 (2005).

[13] D. J. Klein and M. J. Randic, J. Math. Chem. **12**, 81 (1993).

[14] M. Rosvall and C. T. Bergstrom, Proc. Natl. Acad. Sci. USA **105**, 1118 (2008).

[15] G. Kirchoff, Ann. Phys. Chem. **72**, 497 (1847).

[16] P. T. Fu-Shang, T.-Y. Sung, M.-Y. Lin, L.-H. Hsu, and W. Myrvold, IEEE Trans. Reliability **43**, 600 (1994).

[17] en.wikipedia.org/wiki/resistance_distance.

[18] D. Grady, C. Thiemann, and D. Brockmann, Nat. Commun. **3**, 864 (2012).

[19] V. V. Williams, http://www.cs.berkeley.edu/∼virgi/matrixmult.pdf.

[20] S. Eastwood and J. W. L. Wan, Numer. Lin. Alg. Appl. **20**, 74 (2013).

[21] D. Ravichandran, P. Pantel, and E. Hovy, Proc. Annual Meeting Assoc. Comput. Linguistics **43**, 622 (2005).

[22] F. Murtagh, Comput. Stat. Q. **1**, 101 (1984).

[23] A. Lancichinetti, S. Fortunato, and F. Radicchi, Phys. Rev. E **78**, 046110 (2008).

[24] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas, J. Stat. Mech.: Theory Exp. (2005) P09008.

[25] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevE.87.032816 for details concerning adjustment of the mutual information in Fig. 4 (Fig. S1) and the real world networks of Table I, together with details of clustering (Figs. S2–S6), and the US air traffic network with its salient STS backbone (Figs. S7 and S8).

[26] W. W. Zachary, J. Anthropol. Res. **33**, 452 (1977).

[27] D. Baird and R. E. Ulanowicz, Ecol. Monographs **59**, 329 (1989).

[28] P. M. Gleiser and L. Danon, Adv. Complex Syst. **6**, 565 (2003).

[29] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, Behav. Ecol. Sociobiol. **54**, 396 (2003).

[30] V. A. Traag, P. Van Dooren, and Y. Nesterov, Phys. Rev. E **84**, 016114 (2011).

[31] A. Lancichinetti, F. Radicchi, J. J. Ramasco, and S. Fortunato, PLoS One **6**, e18961 (2011).

[32] E. W. Dijkstra, Numer. Math. **1**, 269 (1959).

[33] C. Thiemann, F. Theis, D. Grady, R. Brune, and D. Brockmann, PLoS One **5**, e15422 (2010).

[34] http://www.levmuchnik.net/Content/Networks/NetworkData.html

[35] M. E. J. Newman, Social Netw. **27**, 39 (2005).

[36] J. Kim and T. Wilhelm, Physica A **387**, 2637 (2008).

[37] J. P. Bagrow, Phys. Rev. E **85**, 066118 (2012).

[38] B. H. Good, Y.-A. de Montjoye, and A. Clauset, Phys. Rev. E **81**, 046106 (2010).