Hierarchical mutual information for the comparison of hierarchical community structures in complex networks

Juan Ignacio Perotti,[1] Claudio Juan Tessone,[2] and Guido Caldarelli[1, 3, 4]

[1] *IMT Institute for Advanced Studies Lucca, Piazza San Francesco 19, I-55100, Lucca, Italy*[*]
[2] *URPP Social Networks, Universität Zürich, Andreasstrasse 15, CH-8050 Zürich, Switzerland*[†]
[3] *Institute for Complex Systems CNR, via dei Taurini 19, I-00185, Roma, Italy*
[4] *London Institute for Mathematical Sciences, 35a South St. Mayfair, London W1K 2XF UK*[‡]

(Dated: August 19, 2015)

The quest for a quantitative characterization of community and modular structure of complex networks produced a variety of methods and algorithms to classify different networks. However, it is not clear if such methods provide consistent, robust and meaningful results when considering hierarchies as a whole. Part of the problem is the lack of a similarity measure for the comparison of hierarchical community structures. In this work we give a contribution by introducing the *hierarchical mutual information*, which is a generalization of the traditional mutual information, and allows to compare hierarchical partitions and hierarchical community structures. The *normalized* version of the hierarchical mutual information should behave analogously to the traditional normalized mutual information. Here, the correct behavior of the hierarchical mutual information is corroborated on an extensive battery of numerical experiments. The experiments are performed on artificial hierarchies, and on the hierarchical community structure of artificial and empirical networks. Furthermore, the experiments illustrate some of the practical applications of the hierarchical mutual information. Namely, the comparison of different community detection methods, and the study of the the consistency, robustness and temporal evolution of the hierarchical modular structure of networks.

PACS numbers: 89.75.Hc,89.75.-k,89.75.Fb

I. INTRODUCTION

Many complex systems exhibit some degree of organization at different physical scales. Often, the organization is hierarchical. There exist examples of this fact in variegated fields, like biological, social and technological systems. Among the former, and starting from complex molecules (such as lipids, proteins, RNA or DNA) while increasing the scale of observation, new levels of organization are found: organelles, cells, tissues, organs, anatomical systems, organisms, populations and ecosystems. In the social context, human societies organize from the level of individuals, groups, cities, up to the global scale of countries or continents. Finally, among technological systems, computer networks are also arranged at different scales from the local network level up to the domain level routing systems that constitute the backbone of internet. Hierarchical organizations seem ubiquitous in complex systems and, despite the early interest of the scientific community about the subject [1–3], it is far from being fully understood. The description of hierarchical organization of complex systems remains, to a great extent, at the semantic level. This is mainly because the following difficulties: the existence of several relevant physical scales, the existence of a variety of organizing principles, the large number of components, and the lack of a generally enough and well defined formal theory for the identification of hierarchies.

The study of complex networks [4–7] plays a central role in the characterization of the organization of complex systems. In essence, networks are used to represent the structure of the interactions between the components of the system under consideration. Therefore, it is reasonable to assume that some complex networks have hierarchically organized topologies, reflecting the underlying hierarchical organization of the associated complex systems. A natural way of thinking about hierarchical network topologies is that of hierarchical community structures; i.e. communities within communities of nodes [8–10]. Typically, the identification of the communities of a network is computationally intensive and a statistically difficult problem [11]. Although a large number of community detection methods have been developed already [12–17] – including methods for the identification of hierarchical community structures [8, 9, 17–23] – not all methods provide comparable results. This is true, specially for hierarchical community structures. Therefore, similarity measures for the comparison of hierarchical community structures are of crucial importance. The aim of this paper is to introduce an information-theoretic tool which can be used to compare hierarchies, or trees, which might be composed of network communities. We further show that this tool can be employed to trace the evolution of hierarchies when temporal networks are analyzed.

A standard way to quantify the similarity of two community structures is to compute the mutual information

[*] E-mail: juanignacio.perotti@imtlucca.it
[†] E-mail: claudio.tessone@business.uzh.ch
[‡] E-mail: guido.caldarelli@imtlucca.it

between the associated node partitions [24, 25]. Extending the idea, the present paper introduces a *hierarchical mutual information*, generalizing the traditional mutual information to work with hierarchical partitions. In principle, there might be different ways in which the mutual information can be generalized into a hierarchical mutual information. In this work, hierarchies are considered to be of divisive nature; i.e. the whole is divided into parts, each of which is sub-divided into sub-parts, an so on, following a top-down approach. As a consequence, in this context hierarchies are represented by trees with branches of varying length. Other possible generalization approaches might exist. For example, generalizations that consider agglomerative hierarchies – i.e. bottom up approaches – or overlapping communities. Alternatively, related methods exists for the comparison of phylogenetic trees [26–28]. However, to the best of our knowledge, no previous method – based on information-theoretic measures – exist for the comparison of hierarchies. These alternative methods, and the previously mentioned generalization approaches, are not discussed further in this paper, but can be considered in future works.

The outline of the paper is the following. In section II, the hierarchical mutual information is motivated and introduced. In section III, this measure is tested on different synthetic setups. More specifically, in subsection III A, the behavior of the hierarchical mutual information is tested in artificial hierarchies; while in subsection III B, the hierarchical mutual information is used to analyze the hierarchical community structure of artificial networks, or network models. A similar procedure is performed on empirical networks in section III C, including the case of a temporal one. Finally, the discussion and conclusions are summarized in section IV.

II. THEORY

A. Hierarchical Partitions

A hierarchical partition is a generalization of the traditional concept of partition. Here, each element of the partition can be recursively partitioned into others, yielding a hierarchy. The formal definition is as follows. Consider a set of elements, or universe, denoted by Ω. An element in Ω is denoted by i. The set Ω splits into a hierarchy of sub-sets, denoted by v. The number of elements in the sub-set v is written as $|v|$. The *hierarchical partition*, or simply hierarchy, is represented by a tree denoted by $\mathcal{T}$. The root $v_\Omega \in \mathcal{T}$ is the "oldest ancestor" of the various vertices, or descendants in the tree $\mathcal{T}$. As a sub-set, the root contains the whole set of elements, i.e. $v_\Omega \equiv \Omega$. For any sub-set $v \in \mathcal{T}$, $\triangle_v^{\mathcal{T}}$ denotes the set of descendants of v. A sub-set v is at the l-th level (or depth) of the hierarchy if l is the topological distance from v to v_Ω. When there is no confusion, we simplify the notation to $\triangle_v$, i.e. by omitting the reference to $\mathcal{T}$.

Consider a network of nodes i and links (weighted or not) $w_{ii'}$. Here, the terms *elements* and *nodes* are used interchangeably; both, referring to the entities denoted by i. Traditionally, the community structure of a network is represented by a node partition. In many cases, these communities present a hierarchical organization. In particular, if the hierarchy is constituted by sub-communities within communities, then the structure can be mapped to a hierarchical partition $\mathcal{T}$. Depending on the context, $\mathcal{T}$ is referred to as a tree, as a hierarchical community structure, or simply as a hierarchy; i.e. the terms are used interchangeably. Each sub-set $v \in \mathcal{T}$ corresponds to one and only one sub-community of the network hierarchical community structure (see Fig. 1a). The root v_Ω represents the set of all nodes in the network. The children $u \in \Delta_v$ correspond to a partition of the sub-community v into sub-communities u. The leaves of $\mathcal{T}$ are the smallest sub-communities of the network. Finally, each sub-community $v \in \mathcal{T}$ has an associated sub-network with links $w_{ii'}^{(v)}$, between the pair of nodes $i, i' \in v$.

B. Uncertainty Reduction

In this section, the definition of the *hierarchical mutual information* is motivated. Only Shannon-based information measures are used throughout the rest of the paper [29].

Consider how the uncertainty about the identification of a specific node i is reduced when going down a tree $\mathcal{T}$. As the root $v_\Omega \in \mathcal{T}$ represents the set of all nodes, to look for a specific node i requires checking $\cong \log_2 |v_\Omega|$ binary choices. In other words, the uncertainty is reduced by $\ln |v_\Omega|$ nats when a node i is unequivocally identified (a nats is a unit of information equals to $1/\ln 2 \approx 1.44$ bits), and there is no uncertainty left. Sometimes the information pointing towards a specific node is not precise, and the uncertainty reduction is not complete. For example, if node i is specified to be in the sub-community v, the uncertainty reduction is $\ln |v_\Omega| - \ln |v| = -\ln(|v|/|v_\Omega|)$ nats, and $\ln |v|$ nats of uncertainty still remains.

Transversing a hierarchy along descendants is similar to a sequential reduction of uncertainty. More specifically, it is possible to write

$$\begin{aligned} -\ln 1/|v_\Omega| = & -\ln |v_1|/|v_\Omega| - \ln |v_2|/|v_1| - ... \\ & ... - \ln |v_l|/|v_{l-1}| - \ln 1/|v_l| ... \\ & .. - \ln |v_{L_i}|/|v_{L_i - 1}| - \ln 1/|v_{L_i}|, \end{aligned} \tag{1}$$

where L_i is the deepest level at which node i can be found. Each term $-\ln |v_l|/|v_{l-1}|$ can be considered ed a conditional uncertainty reduction. Specifically, how much the uncertainty is reduced when new information is gained (that $i \in v_l$), given that some other information was already available (that $i \in v_{l-1}$).

It is possible to average over nodes i using an appropriate weighted version of the expression in Eq. (1). More specifically, the average uncertainty reduction along the tree $\mathcal{T}$ is defined as

$$\langle H\rangle_{\mathcal{T}} = \sum_{v_1\in\triangle_{v_\Omega}} -\frac{|v_1|}{|v_\Omega|}\ln\frac{|v_1|}{|v_\Omega|} + ... \qquad (2)$$
$$... + \sum_{v_l\in\triangle_{v_{l-1}}} -\frac{|v_l|}{|v_{l-1}|}\ln\frac{|v_l|}{|v_{l-1}|} + ...$$
$$+ \sum_{i\in v_{L_i}} -\frac{1}{|v_{L_i}|}\ln\frac{1}{|v_{L_i}|}.$$

In Eq. 2, every reduction step is weighted by the fraction of nodes that are found by following the corresponding branch of the tree $\mathcal{T}$. Using similar ideas, the hierarchical mutual information is defined in the next section.

C. The Hierarchical Mutual Information

In community detection problems, it is customary to quantify the similarity between two inferred community structures using the mutual information between the corresponding node partitions [11, 15, 24]. Here, the goal is to introduce the hierarchical mutual information to quantify the similarity between two hierarchical partitions, or trees, associated to corresponding hierarchical community structures.

Consider two trees $\mathcal{T}$ and $\mathcal{T}'$ and two sub-communities $v \in \mathcal{T}$ and $v' \in \mathcal{T}'$, both at the same topological distance, or level l, from the roots of their corresponding trees. It is not necessary for the trees $\mathcal{T}$ and $\mathcal{T}'$, nor the sub-communities v and v' to contain the same elements. Let $\mathcal{T}_v$ represent the sub-tree of root v obtained from $\mathcal{T}$. The analogous holds for $\mathcal{T}'_{v'}$. The hierarchical mutual information between the sub-trees $\mathcal{T}_v$ and $\mathcal{T}'_{v'}$ is denoted by $I(\mathcal{T}_v;\mathcal{T}'_{v'})$. By definition, it is assumed that $I(\mathcal{T}_v;\mathcal{T}'_{v'}) = 0$ if either v or v' is a leaf of the corresponding tree. Otherwise, $I(\mathcal{T}_v;\mathcal{T}'_{v'})$ is recursively defined by the formula

$$I(\mathcal{T}_v;\mathcal{T}'_{v'}) := I(\triangle_v;\triangle_{v'}|v\cap v') + \sum_{\substack{u\in\triangle_v,u'\in\triangle_{v'}\\ |v\cap v'|\neq 0}} \frac{|u\cap u'|}{|v\cap v'|} I(\mathcal{T}_u;\mathcal{T}'_{u'}). \qquad (3)$$

In Eq. 3, the first term of the r.h.s. is called the *one step mutual information*, and is defined as

$$I(\triangle_v;\triangle_{v'}|v\cap v') := H(\triangle_v|v\cap v') + H(\triangle_{v'}|v\cap v') -H(\triangle_v\cap\triangle_{v'}|v\cap v'),$$

where $H(\cdot)$ represents the Shannon entropy. These terms are computed as

$$H(\triangle_v|v\cap v') \qquad (4)$$
$$:= \begin{cases} \sum_{u\in\triangle_v} -\frac{|u\cap v'|}{|v\cap v'|}\ln\frac{|u\cap v'|}{|v\cap v'|} & \text{if } |v\cap v'|\neq 0 \\ 0 & \text{otherwise} \end{cases},$$

and

$$H(\triangle_v \;\cap\; \triangle_{v'}|v\cap v') \qquad (5)$$
$$:= \begin{cases} \sum_{\substack{u\in\triangle_v\\ u'\in\triangle_{v'}}} -\frac{|u\cap u'|}{|v\cap v'|}\ln\frac{|u\cap u'|}{|v\cap v'|} & \text{if } |v\cap v'|\neq 0 \\ 0 & \text{otherwise} \end{cases}.$$

In all cases, the convention $0\ln 0 = 0$ is adopted. Finally, the hierarchical mutual information of two full trees $\mathcal{T}$ and $\mathcal{T}'$ is denoted and defined by

$$I(\mathcal{T};\mathcal{T}') := I(\mathcal{T}_{v_\Omega};\mathcal{T}'_{v'_\Omega}) \qquad (6)$$

where v_Ω and v'_Ω are the roots of $\mathcal{T}$ and $\mathcal{T}'$, respectively.

Each term involved in $I(\mathcal{T};\mathcal{T}')$ is non-negative, and thus, the hierarchical mutual information is a non-negative quantity. Also, $I(\mathcal{T};\mathcal{T}') = I(\mathcal{T}';\mathcal{T})$, i.e. it is a symmetric function of its arguments. When the trees $\mathcal{T}$ and $\mathcal{T}'$ are just stars, i.e. a root plus one generation of descendants, it is possible to think of them as standard partitions. In this case, the hierarchical mutual information reduces to the standard mutual information.

Note, the hierarchical mutual information is not a measure of the similarity between the corresponding final partitions of the nodes at the leaves of the trees (except when both trees are stars). Rather, it is a summation of weighted local one-step contributions, measuring how similar the partitions are at each corresponding point in both trees. For example, if two nodes i and i' are separated at level l in tree $\mathcal{T}$ and at level $l'\neq l$ in tree $\mathcal{T}'$ then, the separation of i and i' contributes with zero to the value of the hierarchical mutual information.

For practical purposes, a *normalized* hierarchical mutual information is defined as

$$i(\mathcal{T};\mathcal{T}') = \frac{I(\mathcal{T};\mathcal{T}')}{\sqrt{I(\mathcal{T};\mathcal{T})I(\mathcal{T}';\mathcal{T}')}}. \qquad (7)$$

Its value lays in the interval $[0, 1]$, and attains the maximum 1 if and only if $\mathcal{T} = \mathcal{T}'$. We conjecture the truth of the previous statement, supported by the results of extensive numerical exploration reported in the following sections. More precisely, it remains to be proved that $i(\mathcal{T};\mathcal{T}') \leq 1$ for all pairs $\mathcal{T}, \mathcal{T}'$.

To help better understand the hierarchical mutual information, a simple example is worked out explicitly. Consider the set of nodes $\{a, b, c, d, e, f\}$, and the two hierarchical partitions $\mathcal{T} = \{\{\{a\}, \{b, c\}\}, \{d, e, f\}\}$ and $\mathcal{T}' = \{\{a\}, \{b, c\}, \{d, e, f\}\}$ (see Figs. 1b and 1c). Here, $v_\Omega = v'_\Omega = \{a, b, c, d, e, f\}$. Also, $\Delta_{v_\Omega} = \{\{a, b, c\}\}, \{d, e, f\}\}$ and $\Delta_{v'_\Omega} = \{\{a\}, \{b, c\}\}, \{d, e, f\}\}$. In the tree $\mathcal{T}$, there is an intermediate sub-community $\{a, b, c\}$ which is not on the other tree $\mathcal{T}'$. As a consequence, the one-step mutual information at level $l = 1$ (see Eq. 4) is $I(\triangle_{v_\Omega};\triangle_{v'_\Omega}|\{a, b, c, d, e, f\}) \cong 0.693$. All other terms corresponding to levels $l > 1$ contribute with zero because they involve leaves. This is because the tree $\mathcal{T}'$ is just a star which has only one level. Adding

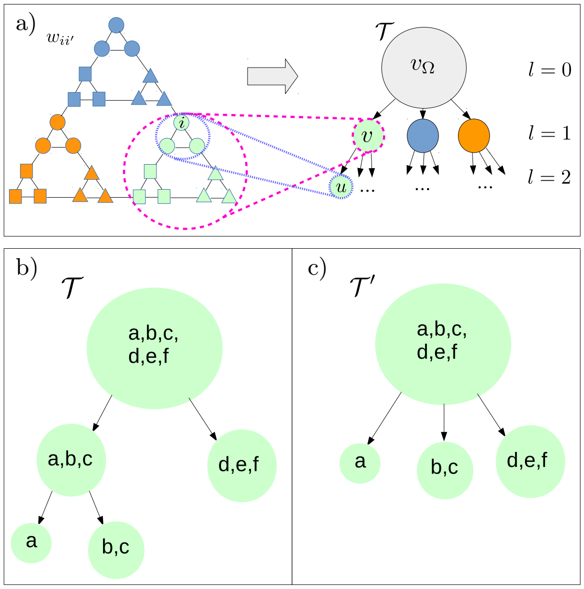

FIG. 1. (Color online). a) Illustration of how a hierarchy of communities obtained from a Sierpinski network $w_{ii'}$ corresponds to a hierarchical partition, or tree $\mathcal{T}$. The root $v_\Omega \in \mathcal{T}$ contains all the nodes of the network $w_{ii'}$, and v represents a sub-community level $l = 1$. In b) and c), two simple hierarchical partitions, or trees, of the same set of nodes, $\{a, b, c, d, e, f, g\}$, are presented. On the tree $\mathcal{T}$, the node a is separated from the other nodes $\{b, c\}$ at the level $l = 2$, while on the tree $\mathcal{T}'$ the separation occurs at level $l = 1$. This difference implies a normalized hierarchical mutual information smaller than one, even if the partition at the bottom of both trees is the same.

all together, $I(\mathcal{T};\mathcal{T}') \cong 0.693$. On the other hand, the *self-hierarchical mutual informations* are $I(\mathcal{T};\mathcal{T}) \cong 1.242$ and $I(\mathcal{T}';\mathcal{T}') \cong 1.011$. Therefore, the normalized hierarchical mutual information yields $i(\mathcal{T};\mathcal{T}') \cong 0.618$; a value smaller than one. In other words, these trees share only a fraction of the information they contain. This holds in spite that the partitions at the bottom are the same for both trees.

To facilitate future research, collaboration and scientific reproducibility, we provide Python [30] code implementing the hierarchical partition data-structure and the hierarchical mutual information function, as an open-source package [31].

III. RESULTS

A. Testing the Hierarchical Mutual Information in Artificial Hierarchies

Before focusing on the hierarchical community structures of networks, we analyze the behavior of the hierarchical mutual information when used to compare artificially generated hierarchical partitions. More specifically, hierarchies composed of binary trees $\mathcal{T}$ containing $N = 2^L$ elements i, L levels, and $2^{L+1} - 1$ sub-communities including the root. Each tree has one element i per sub-community at the bottom level $l = L$, two elements per sub-community at the previous level $l = L - 1$, and so on until it has N elements at the root.

In the experiments, the original trees are compared against correspondingly randomized ones. The idea is to show how the normalized hierarchical mutual information decays with respect to the level of randomization. Two different randomization procedures are used.

In the first randomization procedure, pairs of elements are randomly chosen from the tree, and consecutively swapped until a fraction f of them is affected. This is called the *basic* randomization procedure. In Fig. 2, the average normalized hierarchical mutual information $\langle i \rangle_L$ is plotted vs the fraction f of randomized elements. The average is computed over 100 repetitions of the randomization procedure, for each value of f and L. Notice, $\langle i \rangle_L$ decays approximately in an exponential way with respect to f; further, it is almost independent of L except for large values of f, where finite size effects become important. In particular, when the hierarchy is fully randomized, i.e. $f = 1$, the $\langle i \rangle_L$ is non-zero. Although *a priori* this may be attributed to an error, it is indeed an expected result for finite size hierarchies: random coincidences produce a non-zero amount of shared information. A similar result is known to hold for the traditional mutual information [25].

In the second procedure, the elements are also shuffled by swapping pairs chosen at random. However, a given pair is swapped only if both elements belong to the same sub-community at depth l. In other words, the randomization procedure preserves the classification of the elements at the levels $0, 1, ..., l - 1$, while in the subsequent levels $l, l + 1, ..., L$, the original classification is destroyed. Again, the swapping procedure runs until a fraction f of the elements is affected. This second procedure is called the *level-preserving* randomization procedure. In Fig. 3, the average normalized hierarchical mutual information $\langle i \rangle_l$ is plotted as a function of f for the level-preserving randomization procedure. Here, experiments are repeated for different values of l and fixed $L = 7$. Averages are computed as it was done with the basic randomization procedure. In line with the previous result of Fig. 2, $\langle i \rangle_l$ also decreases with f following approximately an exponential decay. Now the smaller is the shuffling level l, the slower is the decay. In particular, for $l = 6$ no decay at all is observed, i.e. $\langle i \rangle_l = 1$ for all f. This is expected because trees have $L = 7$ levels and only one element per sub-community at the bottom level, which do not contribute to the hierarchical mutual information.

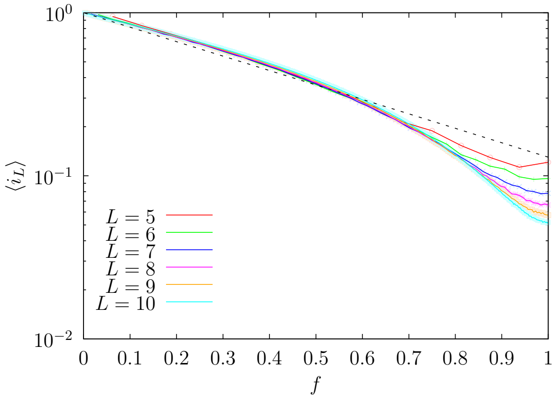

FIG. 2. (Color Online). The normalized hierarchical mutual information, $\langle i_L \rangle$, comparing hierarchical partitions represented by binary trees with L levels, and corresponding randomized partitions with a fraction f of the elements shuffled at random. The average is computed over 100 realizations of the shuffling procedure, and different colors correspond to trees with different number of levels L. The black dashed line corresponds to an exponential fit, $\langle i_L \rangle = \exp(-f/f_0)$ with $f_0 = 0.490 \pm 0.004$ and $R^2 = 0.968$, for the case $L = 10$. Error bars are not plotted for clarity.

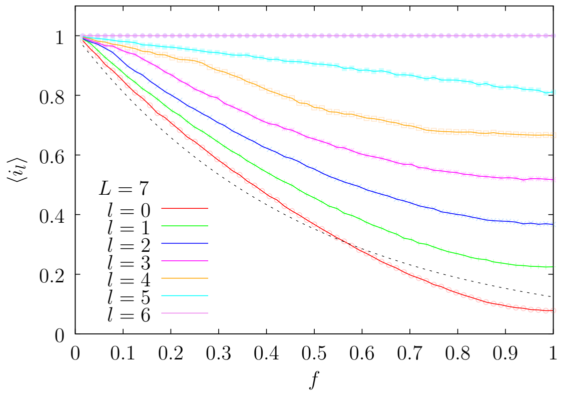

FIG. 3. (Color Online). The normalized hierarchical mutual information comparing hierarchical partitions represented by binary trees with $L = 7$ levels, and corresponding randomized partitions where a fraction f of the elements are randomly shuffled. The randomization procedure preserves the element classification of levels $0, 1, ..., l-1$, but affects the rest of the levels. The dashed line is the best fit of an exponential decay $\langle i_l \rangle = \exp(-f/f_0)$, with $f_0 = 0.478 \pm 0.008$ and $R^2 = 0.978$, for $l = 0$. For clarity, error bars are not shown.

B. Comparing Community Detection Algorithms on Artificial Networks

1. *Community Detection Methods*

One of the interesting applications of the normalized hierarchical mutual information is comparing the results yielded by different community detection methods. In this paper, three community detection methods are compared: *Infomap* [8], which find a hierarchy of communities through the minimization of the description length of the path traversed by a random walker; the *Hierarchical Stochastic Block Model* method (HSBM) [9], which fits a hierarchy of stochastic block models to the network topology; a *Recursive Louvain* method (RL), which recursively splits the network into a hierarchy of network modules using, at each step, the well-known Louvain community detection algorithm [32]. In what follows, the relevant aspects of the different methods are considered in more detail.

Infomap return hierarchies that are consistent with a *divisive* algorithm, i.e. the branches of the corresponding trees may have different depths. The algorithm itself uses both approaches, repeatedly. Communities are split and merged until a minimum description length is attained. In the hierarchies obtained by this method, the leaves have one and only one node. For the sake of comparison with the other methods, these communities of size equal to one are ignored, except if same level communities of size larger than one exists.

At difference with Infomap, the HSBM merge nodes to generate super-nodes or communities, which are further merged to obtain the communities at the contiguous higher level, and so on. As a consequence, all the branches of the returned trees have the same depth. Moreover, the HSBM may return trees containing sub-communities with descendants but no further subdivisions, i.e., sub-communities with only one child. Although the hierarchies produced by the HSBM can be compared using the hierarchical mutual information – as they are hierarchical partitions – the comparison are not fully appropriate. This is because the hierarchical mutual information is based on a divisive approach, while the HSBM is based on an agglomerative approach. The experiments involving the HSBM show how important is the difference between both kind of approaches.

The recursive application of the Louvain method is a mixed agglomerative-divisive algorithm. The standard Louvain method is an agglomerative algorithm; a community structure is obtained by merging modules until Q reaches a maximum value [32]. On the other hand, the recursive use of Louvain presented here, is a divisive method. More specifically, given a network $w_{ii'}$ (defining the level $l = 0$), a standard Louvain method is applied to obtain a partition into sub-communities v at level $l = 1$. Then, Louvain is applied again on each sub-community, to split each sub-network $w_{ii'}^{(v)}$ into sub-communities at level $l = 2$, and so on. In this way, a tree $\mathcal{T}$ is generated. The division of a particular sub-community stops when the standard Louvain returns a modularity $Q \leq 0$. Importantly, the Louvain method is not deterministic, leading to stochastic differences from run to run. Two important points have to be stressed: First, the use of Louvain is circumstantial, any other modularity maximization procedure would produce similar results. Second, the idea of a recursive application of a modularity based community detection algorithm is not new, and more elaborate algorithms do exist [10, 17, 33]. However, here RL is chosen for its simplicity. Our main goals are: to show how the hierarchical mutual information behaves, and to illustrate how it can be used, without aiming to find the best community detection method.

2. *Artificial Hierarchical Networks*

In order to analyze the performance of the different community detection methods, in this section they are run on specific networks. Here, two well-known benchmark network models are used to generate the networks necessary for the experiments. In principle, these network models are able to generate network samples with underlying hierarchical community structures. Clearly, the specific characteristics of the generated networks depend on the parameter values chosen.

The first network model is the hierarchical planted partition model (HPM) [20], a generalization of the planted partition model [34] where the network obtained is hierarchically arranged. In this model, N nodes are connected according to a hierarchical structure of L levels and a branching factor B. For practical purposes, we chose $N = 512$ nodes, $L = 3$ levels, and a branching factor $B = 4$ and (see Fig. 4). At the root level, $l = 0$, all nodes belong to the same community. At level $l = 1$, there exist $B = 4$ communities with 128 nodes each. Consecutively, at the final level $l = 2$, there are $B^2 = 16$ communities, with 32 nodes each. Each node has an average of K_l links to nodes exclusively within the community they belong at level l, i.e. K_2, K_1, K_0 to other nodes in the same communities at levels 2, 1, 0, respectively. Therefore, the total average degree of the nodes is $\langle k \rangle = K_0 + K_1 + K_2$. In principle, networks sampled from the HPM have the expected hierarchical community structure whenever $K_0 < K_1 < K_2$ [20].

The second network model consists of *Sierpinski* networks with L levels. Fig. 1a illustrates a Sierpinski network with $L = 3$. These networks have a natural self-similar and hierarchical modular structure. A Sierpinski network with a single level is just a clique with 3 nodes, i.e. a triangle. A network of this type with $L + 1$ levels is obtained by by replacing each node of a Sierpinski network with L levels by a clique of size 3. It is worth to point out that a Sierpinski network with L levels has $N(L) = 3^L \sim e^L$ nodes, $M(L) = 3[M(L-1)+1] \sim e^L$ links, and its average degree $\langle k \rangle \to 3$ when $L \to \infty$. To make the analysis more interesting, a fraction f of the links in the Sierpinski networks are randomly rewired. The rewiring procedure is well-known [35]. Essentially, successive pairs of links, each of which is chosen at random, swap the nodes at their extremes until a fraction f of the links is affected. In this way, there is a well-defined hierarchy of communities for $f = 0$, which is progressively blurred out as f increases.

In the following sections, the different community detection methods, and these two network models are combined into a set of experiments analyzed using the normalized hierarchical mutual information.

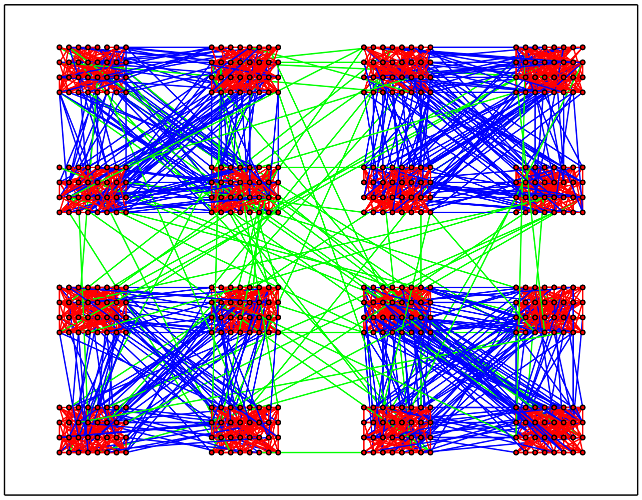

FIG. 4. (Color Online). A sample network obtained from the hierarchical planted network model (HPM), for $K_0 = 0.25, K_1 = 2$ and $K_2 = 8$. The network contains $N = 512$ nodes, 4 big communities of 128 nodes each, and 16 small communities of 32 nodes. The red links connect pairs of nodes sharing the same small community at level $l = 2$, the blue links nodes sharing intermediate communities at level $l = 1$, but not sharing the same small communities. Finally, the green links connect pairs of nodes at level $l = 0$, but not sharing communities at levels $l = 1$ and $l = 2$.

3. *Hierarchical-fidelity*

Each network model has an associated natural or *reference* hierarchy, denoted as $\mathcal{T}^*$. Specifically, the reference hierarchy for the Sierpinski and HPM models are shown in Figs. 1a and 4, respectively. The hierarchies identified by the community detection methods are not necessarily equal to the reference ones, and in some cases, they don't even resemble it. The degree of fidelity of the community detection method measures how similar are the identified communities to the reference ones. Formally, given a community detection method, a network sample $w_{ii'}$ and a reference hierarchy $\mathcal{T}^*$, the *hierarchical-fidelity* – or simply, *fidelity* – of the community method is defined as the average normalized hierarchical mutual information, $\langle i(\mathcal{T}^*; \mathcal{T}) \rangle$. The average is computed summing over an ensemble of hierarchies $\{\mathcal{T}\}$, obtained by repeatedly identifying the hierarchical community structure of the network $w_{ii'}$, using the chosen community detection method. In the results shown, the ensemble $\{\mathcal{T}\}$ was composed by 100 hierarchies. Furthermore, the fidelity is averaged by sampling 100 networks from each network model. The procedure is repeated for different values of the network models parameters, and using the different community detection methods.

For the case of the HPM, two different model re-parameterizations are used. In one case the whole network structure change simultaneously, while in the other case, only one level is affected [20]. More specifically, in case 1 all parameters $K_0 = 7.75\mu + 2$, and $K_1 = 6\mu + 2$ are linearly re-parameterized by $\mu \in [0, 1]$, while K_2 is kept constant. In case 2, the parameters $K_0 = K_2 = 8$ are kept constant, while $K_1 = 8\mu + 4$ changes linearly with μ. For the case of the Sierpinski network model, the parameter is the fraction of randomized links, f, as mentioned in Section III B 2. In what follows the results are presented and commented.

First, the results of the fidelity for Infomap are shown in Fig. 5a. In the HPM, case 1, Infomap detects the reference hierarchy almost always for $\mu = 0$, and the fidelity is ≈ 1. On the other extreme, at $\mu = 1$, Infomap typically finds a one-level hierarchy composed of 4 communities with 128 nodes. The 4 communities are the right ones at the level $l = 1$, and the fidelity decays to $\approx 1/\sqrt{2} \cong 0.707$. The decay in the fidelity is expected because all K_l converge to the same value $K_l = 8$ when $\mu \to 1$, making the generated network hierarchies less defined. In case 2, the same scenario occurs for $\mu \geq 1/2$, i.e. the same 4 communities are identified. On the other hand, for small μ, the structure of the network is dominated by links at levels 0 and 2. As a consequence, and depending on the particular network realization, Infomap finds a one-level hierarchy with either 1 or ≈ 16 communities, resulting in a small fidelity value. For the Sierpinski networks, the behavior can be more easily interpreted. For small f, Infomap finds an approximately accurate representation of the exact hierarchy of communities. However, as f grows, the hierarchy is quickly blurred out and the fidelity decays accordingly.

The findings of the fidelity for the HSBM method are shown in Fig. 5b. For the HPM, the fidelity is almost a constant function of μ, for both cases 1 and 2. A closer inspection reveals that, typically, the HSBM method splits the network samples into two communities at level $l = 1$, which are then further subdivided, giving rise to a hierarchy with 3 levels. Interestingly, the identified hierarchies are similar, regardless of the value of K_1. Therefore, the resulting fidelity is relatively small because the identified hierarchies are significantly different for the reference one. In essence, the two communities identified at level $l = 1$ mean a significant difference with respect to the expected value of 4. For the case of the Sierpinski model, the HSBM typically detects only one community, except for vanishing f and $L = 5$ where the network splits into two big ones. For this second model, again, overall the fidelity is small.

Thirdly, the fidelity is computed for the recursive Louvain method, and the corresponding results are shown in Fig. 5c. For the HPM, the fidelity is not a monotonic function of μ, instead it displays a maximum at an intermediate value of μ. Interestingly, in general this method tends to find the right communities at the first level $l = 1$. However, the random fluctuations of the network samples become meaningful information for RL, and therefore it tends to split the networks into more communities than the originally found in the reference hierarchy. As a consequence, the normalized hierarchical mutual information yields values smaller than one. However, because the information shared at level $l = 1$ is non-trivial and fairly accurate, the normalized hierarchical mutual information is far from being negligible. On the other hand, for the case of the Sierpinski network model, RL has a poor performance. In essence, this method finds significantly more communities than expected, even at level $l = 1$, resulting in small fidelity values for all f.

4. Hierarchical-consistency

In the previous section, it was shown that each community detection method returns hierarchies different from the expected ones; therefore, some questions arise. How mutually consistent are the returned hierarchies? Do these hierarchies represent noise, or represent a specific detected bias? The following set of experiments addresses these questions. More specifically, the idea is to analyze how mutually similar, or consistent are the communities detected by the methods. Formally, the *hierarchical-consistency* – or just, *consistency* – of a method is defined as the average normalized hierarchical mutual information $\langle i(\mathcal{T}; \mathcal{T}')\rangle$, where the average is computed over an ensemble of pairs of hierarchies, $\{(\mathcal{T}, \mathcal{T}')\}$. The hierarchies in the pairs are randomly chosen, without repetition, from the ensembles of hierarchies generated in the previous experiments about the fidelity. The procedure is repeated for each network sample in order to average the consistency. The whole procedure is repeated for the different network models and corresponding parameters.

In Fig. 6a, the consistency is analyzed when the hierarchical communities are detected by using Infomap. For the HPM, case 1 (see Section III B 2), the consistency is ≈ 1 for all values of μ. In other words, in this initial setting, Infomap provides very consistent results over all the parameter range. For case 2, the fidelity is also close to 1 when μ is large; however, the consistency becomes small for small μ. This is expected, as it was already mentioned Infomap's detection is largely bimodal: either it finds one or ≈ 16 communities depending on the network sample, and these two cases are very inconsistent with each other. For the Sierpinski networks, the consistency is large when $f \approx 0$ and decays to a non-zero value for larger values of f. In other words, network randomization becomes important for large f, but still, part of the information captured by Infomap is already contained even in this case.

The results of the consistency for the HSBM are shown in Fig. 6b. For the HPM, the observed consistency is large in both cases, 1 and 2, despite the small fidelity with respect to the natural hierarchies shown in Fig. 5b. This means that the HSBM return hierarchies similar to each other, but significantly different from the reference

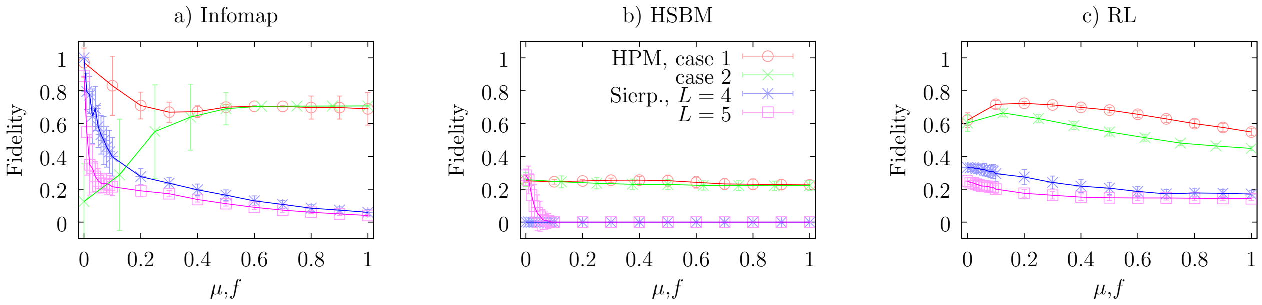

FIG. 5. (Color Online). The fidelity compares the hierarchical community structure of the networks generated with the models against corresponding reference hierarchies. The topology of the generated networks changes as a function of the parameter, $\mu \in [0, 1]$. For the HPM, case 1, μ parameterizes the network model according to $K_0(\mu) = 7.75\mu + 0.25$, $K_1(\mu) = 6\mu + 2$ and $K_2 = 8$. Similarly, in case 2, $K_0 = K_2 = 8$ and $K_1(\mu) = 8\mu + 4$. For the Sierpinski model, $f \in [0, 1]$ is the fraction of rewired links and L is the number of network levels. Each panel corresponds to one of the community detection methods discussed in the text: a) Infomap, b) HSBM, and c) RL.

one. On the other hand, the returned hierarchies share similarities at level $l = 1$, but at the following levels, the differences become important – except for case 1 at $\mu = 1$ where the consistency remain ≈ 1. For the Sierpinski network model, the consistency is negligible in most of the range of f. This is expected because a flat hierarchy conveys no information, and the HSBM typically returns trivial hierarchies for the Sierpinski networks, i.e. hierarchies with only one community, the root one. Only for small values of f, for the case $L = 5$, the consistency is non-zero, but still with small values. Here, only two communities are identified, agreeing only over a small fraction of the nodes.

The consistency for the RL method is shown in Fig. 6c. For the HPM, the curves look similar to the ones corresponding to the fidelities in Fig. 5c. In essence, the computed hierarchies are very similar to each other, and to the reference hierarchy. For the Sierpinski network model, the consistency can be large, even if the fidelity is small. This means that the detected structure is invariably the same, although not the reference expected one.

5. *Hierarchical-similarity*

It is clear that the different community detection methods return different results. However, it remains to analyze how similar are the results of one detection method with respect to one another. To address this point, the *hierarchical-similarity* between two community detection methods is defined as the average normalized hierarchical mutual information, $\langle i(\mathcal{T}_1; \mathcal{T}_2)\rangle$. In shorthand, we speak about the *similarity*, and he average is computed over pairs of trees, where the trees $\mathcal{T}_1$ are computed with one of the methods, while the trees $\mathcal{T}_2$ with the other method. Both set of trees are computed from the same network sample. Later, the similarity is averaged by sampling networks from the different network models. The procedure is repeated for each set of chosen values of the corresponding model parameters. In practice, the network samples and corresponding sampled trees used to compute the fidelities are the ones used to compute the similarities (see Figs. 5 and 6).

Combining the methods of Infomap, HSBM and RL, three different comparisons are possible: Infomap vs. HSBM, Infomap vs. RL, and HSBM vs. RL. These are presented in Figs. 7a, 7b and 7c, respectively. The HSBM method shares a small similarity with the other two. This is expected, because the other methods lead to relatively large fidelities, while the HSBM does not.

The similarity between Infomap and the RL method is the largest among the three possibilities. However, the similarity cannot be as large as the consistency. This is not surprising as Infomap is able to return consistencies as large as 1, while RL is not. The largest similarity value is ≈ 0.6, occurring at $\mu = 0$ for the case 1 in the HPM. Also, the similarity is ≈ 0.5 at $\mu = 1$ for both cases, 1 and 2. For the Sierpinski network, the similarity reaches a maximum value ≈ 0.5 for small f, and it decays slowly up to ≈ 0.2 for large f.

C. Analysis of the Hierarchical Modular Structure of Complex Networks

The experiments of the previous section can be repeated using empirical networks – as opposed to network models – except for the computation of fidelity because, a priori, it is not clear which one is the concomitant reference hierarchy. Notice however, this last possibility is not necessarily impossible for all empirical networks. Many empirical networks have associated a hierarchical decomposition that can be used as a “ground-truth” about its hierarchical structure. Let us remark here that by ground-truth we refer to the practical use of the term [36]. For example, the *NAICS* [37] codes for the case of financial networks [10, 38–40], and the *Harmo-*

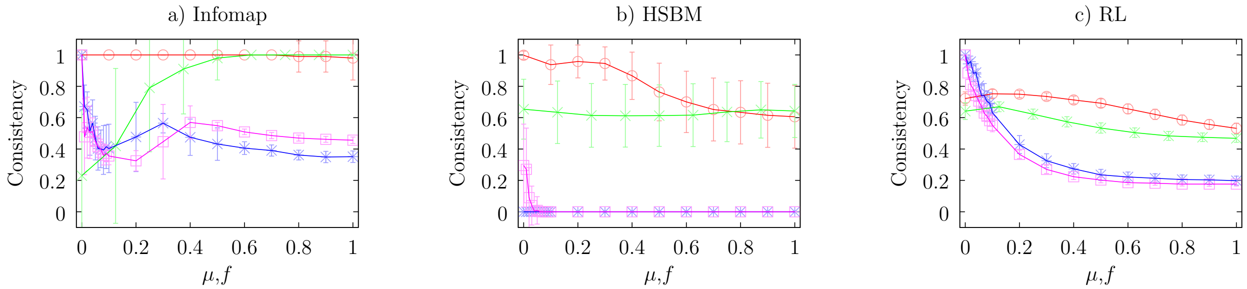

FIG. 6. The consistency measures how similar are the different hierarchies obtained from a given community detection method with respect to each other in a specific network sample. The computations are repeated for several network samples, for each network model, and different values of the parameters μ and f. See Fig. 5 for specific details of the simulation parameters. The panels, (a), (b) and (c) correspond, respectively, to the different community detection methods: Infomap, HSBM and RL.

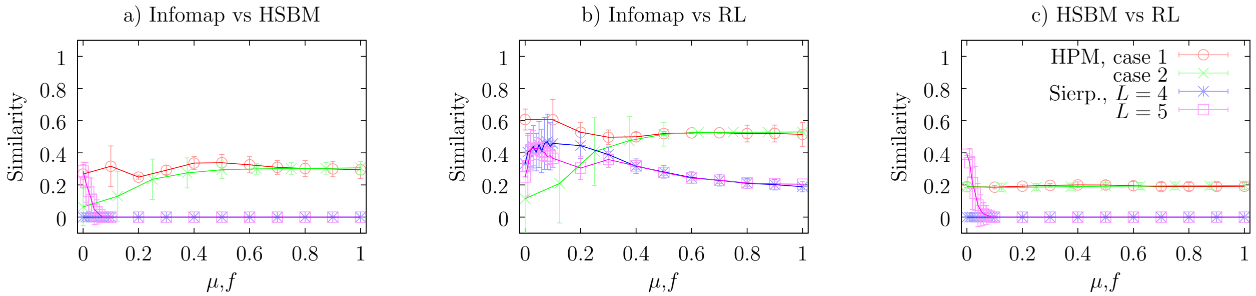

FIG. 7. (Color Online). The similarity compares how similar are the hierarchies obtained by two different community detection methods methods. Here, we compare: a) Infomap vs HSBM, b) Infomap vs RL, and c) HSBM vs RL. The hierarchical community structures used to compute the similarities are the same as those used in Fig. 5. Results are shown as a function of the parameters μ and f of the corresponding network models.

nized System [41] for the case of the international trade network [42–44]. However, these studies are left open for future research and, in what follows, only consistencies and similarities are analyzed in different empirical networks.

The networks in Table I (referenced therein) are the ones studied in the following analysis. All of these networks have convenient characteristics: they are large enough to show relatively rich hierarchical community structures (e.g. Infomap returns up to five hierarchical levels for the case of the Power-grid [8]), with diverse shape (e.g. compare the case of the Power-grid in Fig. 8a, with the case of the Network-science in Fig. 11a), and small enough to keep the computation time bounded. Originally, some of these networks had link weights, or self-loops. For the sake of simplicity, such attributes are removed from the networks. As an illustration of how different are the hierarchical community structures identified by the different community detection methods, Fig. 8 shows the results for the Power-grid network. In this figure, it is apparent that the different methods provide substantially different results.

In order to enrich the analysis, the topology of the empirical networks is shuffled, following the same procedure applied to the Sierpinski networks (cf. Section III B 2). In this way, the obtained hierarchies are analyzed as a function of the fraction f of randomized links.

Firstly, Infomap is used to study the consistency of the empirical networks. The results are shown in Fig. 9a. For some networks, like the Power-grid and the Erdős networks, the identified hierarchies are largely affected by the randomization procedure, i.e. the consistency quickly decays with f. This is particularly reasonable for the Power-grid network, as its hierarchy is embedded into space; reshuffling the links attenuates the embedding, rapidly destroying its spatial nature [45–47]. There exist other networks like EVA, Geometry and Network-science, for which the consistencies of the identified hierarchies seem quite robust to the randomization procedure. This can be interpreted in two ways: on the one hand, this suggests that the hierarchical community structure is mainly determined by the node degrees in the networks or some other topological property that is not destroyed by the randomization procedure. On the other other hand, it may indicate that the the relatively large values of consistency are not significant from a hierarchical point of view. A closer inspection to the Network-science network reveals that the latter possibility is the cause. Specifically, just a relatively small fraction of the network has a rich hierarchical structure with up to 4 levels. The rest of the network nodes are identified as communities at depth $l = 1$, which have no children sub-communities (see Fig. 11a). The hierarchical part is washed out as f grows, eventually leading to a star-like structure (see Figs. 11b, 11c and 11d). The relatively large consistency values for large f are the outcome of random coincidences occurring for these star-like structures.

Secondly, the HSBM method is used for the analysis and the results are shown in Fig. 9b. Overall, a small consistency is obtained. This is because the HSBM method often finds a single community, except for the Geometry network. This suggests that the HSBM finds a rich hierarchical structure for the Geometry network in the form of nested block models. However, a closer inspection indicates that the HSBM identifies simply two large communities, i.e. there is no hierarchy, similar to what is found for the Network-science network for the case of Infomap. This explains the slow decay of the consistency curve.

Thirdly, the consistency is studied using the RL method. The results are shown in Fig. 9c. In all cases, the consistency presents a smooth decay as a function of the randomization f. This is not a surprise because RL tends to return trees with a large number of sub-communities and levels. Therefore, the small changes occurring for increasing f lead to small changes in the consistencies.

In Figs. 10a and 10c, the average similarity between the HSBM method and the other two methods is shown as a function of f. Not surprisingly, the values obtained are small. However, it is interesting to note that, in certain cases, the similarity is larger than the corresponding values for the consistency. For example, cf. the Network-science network in Fig. 9b and Fig. 10c, for small values of f. Even though at a first glance this may seem contradictory, the explanation is simple. The HSBM tends to return trivial hierarchies, yielding a value of zero for the hierarchical mutual information. Then, when the consistency is computed, the number of terms contributing with zero to the average $\langle i(\mathcal{T};\mathcal{T}')\rangle$ is proportional to $1-p^2$, where p is the probability for the HSBM to produce a non-trivial hierarchy. On the other hand, such probability is $1-p$ for the case of the similarity because neither Infomap and nor RL produce trivial hierarchies. In other words, the chances for zero terms to occur in the case of the consistency is significantly larger than for the case of the similarity.

In Fig. 10b, the similarity compares the results for Infomap and RL. A sharp peak can be appreciated at $f \approx 0.05$. This is because Infomap returns a sudden change over the number of identified hierarchies. Namely, the hierarchies pass from having ≈ 4 communities at level $l = 1$, to up to ≈ 40. This large number of communities at level $l = 1$ is always present for the RL. Therefore, the sharp increase occurs when the number of communities at level $l = 1$ becomes large for Infomap, i.e. when it becomes similar for both methods.

TABLE I. Information summary about the empirical network datasets used in the calculations. N is number of nodes and M number of links. Erdős, Network-science and Geometry are scientific-collaboration networks. The Power-grid is technological, and EVA is a network of corporate inter-relationships. The networks marked with * were originally weighted.

Network	N	M	Ref.
Power-grid	4,941	6,594	[48]
Erdős	6,927	11,850	[49, 50]
Network-science*	1,589	2,742	[13, 50]
Geometry*	7,343	11,898	[50, 51]
EVA	8,497	7,970	[50, 52]

1. *Temporal Networks*

In this section, the subject of study is slightly modified. Specifically, the study of traditional complex networks is replaced by the study of correlation matrices computed from the log-returns of stock prices in the S&P500 [38, 39, 54]. The data is obtained from *Yahoo! Finance* [55]. In general, the correlation matrices can be considered as weighted dense networks.

Complex networks are not necessarily static, but change in time [56]. The temporal aspect of a complex network could have dramatic consequences for the behavior of the associated system [57–59]. The correlation matrices of the S&P500 – and the associated hierarchical community structures – can be studied in their time evolution [10, 60, 61]. Therefore, we use the hierarchical mutual information to investigate the evolution of the hierarchical community structure of the financial activity in the S&P500.

The data encompasses the 390 stocks which uninterruptedly cover the 3522 working days from January 1$^{\rm st}$, 1998 until December 31$^{\rm st}$, 2011, according to Yahoo! Finance. Each matrix entry of the correlation matrices is given by

$$C_{ss'} = \frac{\mathrm{Cov}(X_s, X_{s'})}{\sqrt{\mathrm{Var}(X_s)\mathrm{Var}(X_{s'})}}. \tag{8}$$

Specifically, the r.h.s. of Eq. 8 is the *cross-correlation* between the time series $X_s(t)$ and $X_{s'}(t)$, corresponding to the stocks s and s', respectively. In general, cross correlation matrices have off-diagonal entries in $[-1, 1]$, while diagonal entries are equal to one. To simplify the analysis, the correlation matrices are transformed according to the expression [14], $w_{ss'} = |C_{ss'}| - \delta_{ss'}$. The transformation returns a weighted network of non-negative entries and zero diagonals. The transformed networks are the ones used for the computation of the hierarchies. Even though more sophisticated approaches exist (see for example Ref. [10]), for the sake on simplicity the approach taken is the one described above.

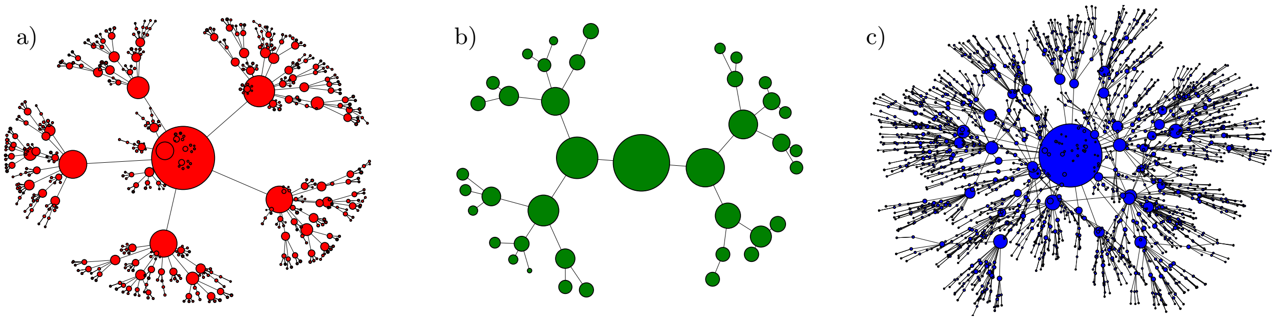

FIG. 8. (Color Online). Hierarchical partition samples, or trees $\mathcal{T}$, computed from the Power-grid empirical network using: a) Infomap (red), b) the HSBM method (green) and c) RL (blue). The trees contain 1099, 40 and 2879 sub-communities, respectively. The size of the sub-communities are proportional to the number of network nodes they contain. The spring-layout is used to distribute the nodes on the plot [53]. Clearly, the different community detection methods find significantly different hierarchies.

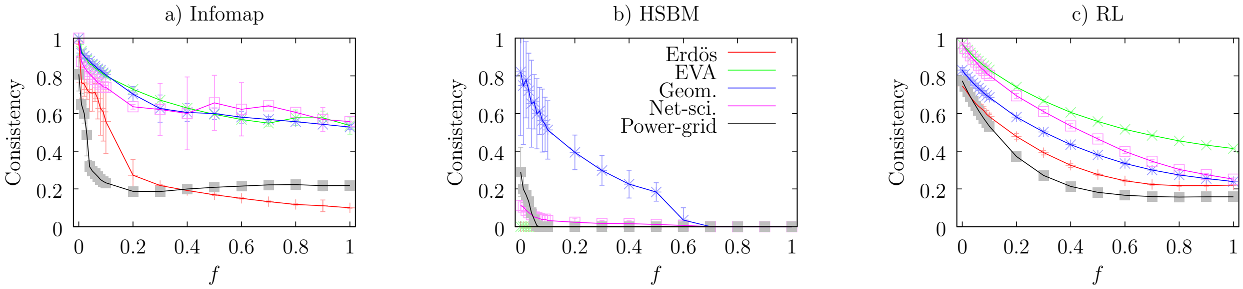

FIG. 9. (Color Online). The consistency is plotted for the different empirical networks in Table I, as a function of the fraction of randomly rewired links, f, and for the different community detection methods: a) Infomap, b) HSBM, and c) RL.

To perform a temporal analysis, different correlation matrices, or weighted networks, are computed by processing the data over different time windows $[t, t+T]$, where t is the initial day of the time window, and T the window duration, measured in days.

In the following analysis, only the RL community detection method is used, this is because the other two methods typically return trivial communities. More specifically, the other two methods fail to find communities because the correlation networks are dense [62]. On the other hand, RL is more sensitive to small link-weights differences, and therefore, it is able to find communities in the dense matrices, but at risk of over-fitting (see section III B 3). As it was already mentioned, more sophisticate methods can be used to mitigate these undesired tendencies (see section III B 1). However, such experiments are left for future works.

Two sets of experiments are analyzed; in both cases, for each computed weighted network, 50 hierarchical community structures, or trees, are computed. The first analysis studies how the integration time, or time windows length T, affects the detected hierarchies. For this purpose, we compute the following average normalized hierarchical mutual information,

$$\langle i_T \rangle := \langle i(\mathcal{T}_{T_{\max}}; \mathcal{T}_T) \rangle \,.$$

We call this quantity, the *temporal-scale hierarchical similarity*, or simply, the *scale-similarity*. It compares hierarchies obtained from the full-length time window, against hierarchies obtained from time windows of length T. In all cases, the initial time is chosen to be the first day, $t = 0$. In Fig. 12a, $\langle i_T \rangle$ is plotted as a function of T. As it can be seen, the larger is T, the larger is $\langle i_T \rangle$. In other words, the expected behavior is observed because, the larger is T the more similar $\mathcal{T}_T$ and $\mathcal{T}_{T_{\max}}$ become in average. In particular, a *plateau* exists for $1000 \lesssim T \lesssim 3000$. This last observation suggests that changes do not occur smoothly, but different hierarchical structural properties emerge at different time scales.

In the second set of experiments, T is fixed at 1500 days and trees are computed out of networks corresponding to different regions in the time line. More specifically, we introduce the *temporal hierarchical auto-similarity* – or *auto-similarity* – which is defined as

$$\langle i_{t,\tau} \rangle := \langle i(\mathcal{T}_t; \mathcal{T}_{t+\tau}) \rangle \,.$$

The auto-similarity compares two set of hierarchies. The first set is computed from the data in the time window $[t, t+T]$, and the other set from the time window defined τ days after. We analyze the auto-similarity varying τ for fixed $t = 1$, and varying t for fixed $\tau = 100$. In the

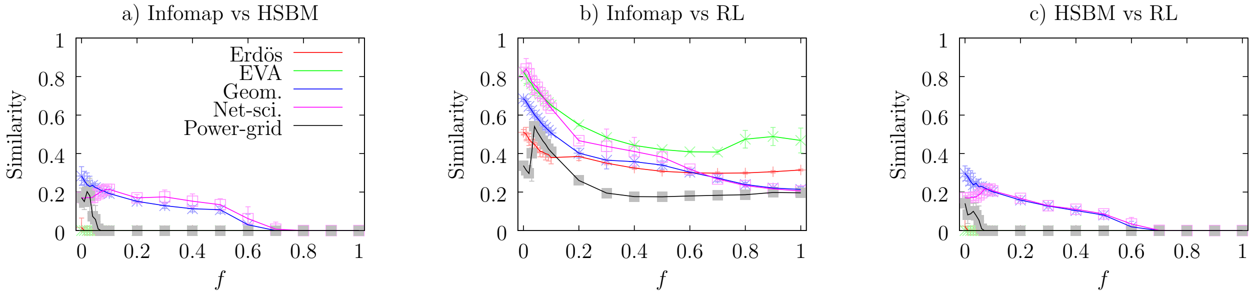

FIG. 10. (Color Online). The similarity is plotted for different empirical networks as a function of the fraction of randomly rewired links f, and the different pairs of community detection methods: a) Infomap vs HSBM, b) Infomap vs HSBM, and c) HSBM vs RL.

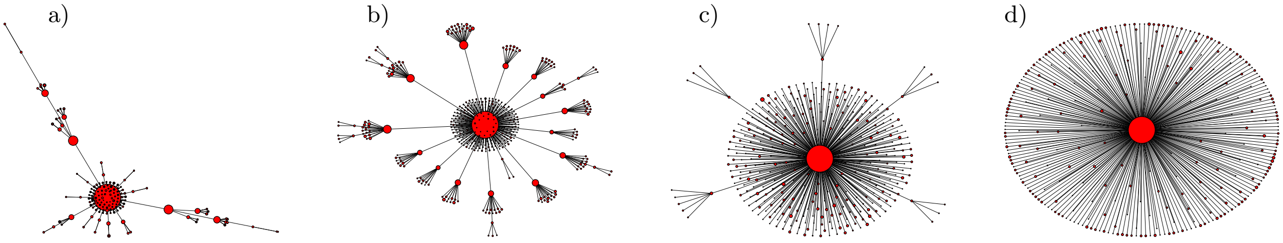

FIG. 11. (Color Online). Hierarchical partition samples $\mathcal{T}$, computed from the Network-science empirical network using Infomap. Each panel correspond to a different level of link randomization: a) $f = 0$, b) $f = 0.2$, c) $f = 0.5$ and d) $f = 1$. In Network-science network, the hierarchy is dominated by branches with almost no children at $f = 0$, but two branches have considerable size and depth. Then, as f grows, the hierarchy evolves towards a simple star.

first case, we study how the time separation τ affects the hierarchy, and in the second case we compare hierarchies corresponding to consecutive time windows. In Fig. 12b, both quantities are plotted. On the one hand, the auto-similarity $\langle i_{t=1,\tau}\rangle$ decays as the time separation τ grows (red curve), i.e. the hierarchy drift in time from the initial structure. On the other hand, the auto-similarity fluctuates around $\langle i_{t,\tau=100}\rangle \approx 0.7$ (blue curve), indicating that the hierarchies of consecutive time windows always share a significant amount of information.

IV. DISCUSSION, CONCLUSIONS AND FUTURE WORK

In this work, the hierarchical mutual information has been introduced, a tool that generalizes the standard mutual information for the comparison of hierarchies. More specifically, for the comparison of hierarchical partitions, which take the form of trees where parts are subsequently subdivided further into sub-parts and so on. The hierarchical mutual information can be used to compare the hierarchical community structure of complex networks, in analogous way as the standard mutual information can be used to compare standard community structures.

We define here a normalized hierarchical mutual information. The traditional normalized mutual information satisfy certain properties; it is a quantity lying in $[0, 1]$, and is equal to one if and only if the compared partitions are exactly equal. If the normalized hierarchical mutual information behaves correctly, it should satisfy analogous properties. The appropriate behavior of the normalized hierarchical mutual information is extensively tested in numerical experiments. The test include artificially generated hierarchical partitions, and the hierarchical community structure of artificially and empirical complex networks. In all the experiments, the normalized hierarchical mutual information is found to behave correctly. However, it should be mentioned that a formal proof of the correct behavior is not provided in the present work.

The experiments also illustrate the overall behavior of the hierarchical mutual information. On the one hand, when comparing artificially generated hierarchies against correspondingly randomized ones, the normalized hierarchical mutual information was found to decrease with the level of randomization. On the other hand, a level by level randomization analysis of the hierarchies indicated that, the larger the number of randomized levels, the faster the normalized hierarchical mutual information decays with the randomization. Another interesting finding was that the normalized hierarchical mutual information never decays to zero. This effect, also present in the standard normalized mutual information, occurs because random (hierarchical) partitions in finite systems share information just by chance.

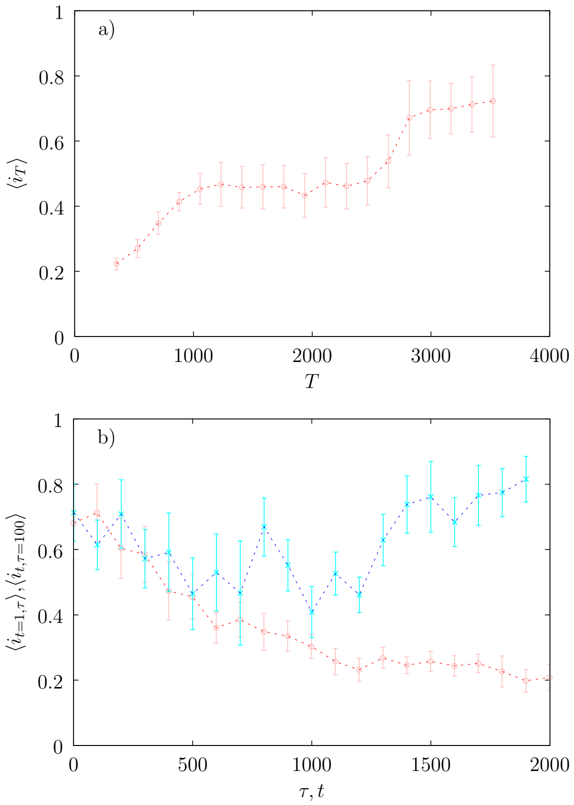

FIG. 12. (Color Online). Temporal analysis of the hierarchical community structure of correlation matrices. The matrices are computed from log-returns of the time series of stock prices in the S&P500. In a), the scale-similarity $\langle i_T \rangle$, determines how the hierarchies change with the time length T of the time window over which the data is processed. In b), two different comparisons are presented using the auto-similarity $\langle i_{t,\tau} \rangle$. In red, $\langle i_{t=1,\tau} \rangle$ determines how similar are the hierarchies at day one, with the hierarchies τ days after. In blue, $\langle i_{t,\tau=100} \rangle$ determines how similar are the hierarchies of consecutive time windows, separated by 100 days, as time t evolves.

The experiments also constitute examples of how the hierarchical mutual information can be used to analyze the hierarchical community structure of complex networks. Specifically, the hierarchical community structure of artificial and empirical networks were studied. In the analysis, different popular community detection methods were utilized, and the results compared. The results were tested on two network models and five empirical networks. It was found that the different methods can return significantly different hierarchical community structures. The normalized hierarchical mutual information correctly identifies these differences. It was also shown that the normalized hierarchical mutual information can be used to compare the detected hierarchies against the natural, reference ones in the different network models. In particular, when the parameters of the network models are appropriate, and the network models tend to generate networks with the expected hierarchical structures, the normalized mutual information between the identified hierarchies and the expected ones tends to grow.

In another set of experiments, the normalized hierarchical mutual information was used to compare the hierarchical community structure of the different networks – the networks generated by the models, and the empirical networks – against that of correspondingly randomized networks. As expected, the normalized mutual information was found to decay with the level of randomization. In a final example, the time evolution of the hierarchical community structure of correlation matrices was analyzed. Specifically, we considered correlation matrices computed from the log returns of stock prices in the S&P500. This final example epitomizes how the hierarchical mutual information is useful to study the evolution of temporal networks. In the analysis, the normalized hierarchical mutual information showed that the hierarchical community structure of the correlations of stocks slowly changes in time, but exhibiting important changes at different times-scales.

The present work opens several possibilities for future research. The mathematical framework behind the hierarchical mutual information can be used to generalize other information measures, like generalizing the variation of information [25]. On a different line of research, the normalized hierarchical mutual information can be used to systematically benchmark, and compare, the different community detection methods in existence. Another interesting future line of research concerns the comparison of phylogenetic trees [26, 28, 63, 64], where the hierarchical mutual information could have useful applications. Finally, the normalized hierarchical mutual information can be used to compare the identified hierarchies against corresponding *ground-truth hierarchies* that different data sets might have available. The above examples go without mentioning the ample possibilities of using and extending this methodology in the many fields where hierarchical communities structures are identified.

V. ACKNOWLEDGMENTS

J.I.P and G.C. acknowledge support from: FET IP Project MULTIPLEX nr. 317532. FET Project SIMPOL nr. 610704, FET project DOLFINS nr. 640772. CJT acknowledges financial support from the URPP on Social Networks, Universität Zürich, Switzerland.

[1] H. A. Simon, Proc. Am. Phil. Soc. **106**, 467 (1962).

[2] P. W. Anderson *et al.*, Science **177**, 393 (1972).

[3] J. H. Holland, *Emergence: from chaos to order* (Oxford University Press, 1998).

[4] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, Phys. Rep. **424**, 175 (2006).

[5] G. Caldarelli, *Scale-Free Networks: complex webs in nature and technology* (Oxford University Press, 2007).

[6] M. Newman, *Networks: An Introduction* (Oxford University Press, 2010).

[7] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter, J. Complex Netw. **2**, 203 (2014).

[8] M. Rosvall and C. T. Bergstrom, PloS ONE **6**, e18209 (2011).

[9] T. P. Peixoto, Phys. Rev. X **4**, 011047 (2014).

[10] M. MacMahon and D. Garlaschelli, Phys. Rev. X **5**, 021006 (2015).

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

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

[13] M. E. J. Newman, Phys. Rev. E **74**, 036104 (2006).

[14] T. Heimo, J. M. Kumpula, K. Kaski, and J. Saramäki, J. Stat. Mech. Theor. Exp. **2008**, P08007 (2008).

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

[16] V. Zlatić, A. Gabrielli, and G. Caldarelli, Phys. Rev. E **82**, 066109 (2010).

[17] P. Zhang and C. Moore, Proc. Natl. Acad. Sci. USA **111**, 18144 (2014).

[18] M. Sales-Pardo, R. Guimera, A. A. Moreira, and L. A. N. Amaral, Proc. Natl. Acad. Sci. USA **104**, 15224 (2007).

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

[20] A. Lancichinetti, S. Fortunato, and J. Kertész, New J. Phys. **11**, 033015 (2009).

[21] A. Lancichinetti, F. Radicchi, J. J. Ramasco, and S. Fortunato, PloS ONE **6**, e18961 (2011).

[22] C. Granell, S. Gomez, and A. Arenas, Int. J. Bifurcat. Chaos **22**, 1250171 (2012).

[23] C. Granell, S. Gomez, and A. Arenas, Int. J. Bifurcat. Chaos **22**, 1230023 (2012).

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

[25] M. Meilă, J. Multivar. Anal. **98**, 873 (2007).

[26] B. DasGupta, X. He, T. Jiang, M. Li, J. Tromp, and L. Zhang, in *Proceedings of the eighth annual ACM-SIAM symposium on Discrete algorithms* (Society for Industrial and Applied Mathematics, 1997) pp. 427–436.

[27] J. Nielsen, A. K. Kristensen, T. Mailund, and C. N. Pedersen, Algorithm. Mol. Biol. **6**, 15 (2011).

[28] F. Shi, Q. Feng, J.-J. Chen, L. Wang, and J. Wang, Tsinghua Sci. Technol. **18**, 490 (2013).

[29] T. M. Cover and J. A. Thomas, *Elements of information theory* (Wiley-Interscience, Hoboken, N.J, 2006).

[30] `https://www.python.org/`.

[31] `http://hierpart.readthedocs.org`.

[32] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, J. Stat. Mech. Theor. Exp. **2008**, P10008 (2008).

[33] P. Pons and M. Latapy, Theor. Comput. Sci. **412**, 892 (2011).

[34] A. Condon and R. M. Karp, Random Struct. Algor. **18**, 116 (2001).

[35] S. Maslov and K. Sneppen, Science **296**, 910 (2002).

[36] D. Hric, R. K. Darst, and S. Fortunato, Phys. Rev. E **90**, 062805 (2014).

[37] `http://www.census.gov/eos/www/naics/`.

[38] R. N. Mantegna, Eur. Phys. J. B **11**, 193 (1999).

[39] G. Bonanno, G. Caldarelli, F. Lillo, and R. N. Mantegna, Phys. Rev. E **68**, 046130 (2003).

[40] G. Bonanno, G. Caldarelli, F. Lillo, S. Miccichè, N. Vandewalle, and R. N. Mantegna, Eur. Phys. J. B , 363 (2004).

[41] `http://www.wcoomd.org`.

[42] C. A. Hidalgo, B. Klinger, A.-L. Barabási, and R. Hausmann, Science **317**, 482 (2007).

[43] M. Barigozzi, G. Fagiolo, and D. Garlaschelli, Phys. Rev. E **81**, 046104 (2010).

[44] G. Caldarelli, M. Cristelli, A. Gabrielli, L. Pietronero, A. Scala, and A. Tacchella, PloS ONE **7**, e47278 (2012).

[45] H. D. Rozenfeld, C. Song, and H. A. Makse, Phys. Rev. Lett. **104**, 025701 (2010).

[46] M. Barthélemy, Phys. Rep. **499**, 1 (2011).

[47] M. Popović, H. Štefančić, and V. Zlatić, Phys. Rev. Lett. **109**, 208701 (2012).

[48] D. J. Watts and S. H. Strogatz, Nature **393**, 440 (1998).

[49] J. W. Grossman, *Erdos Number Project* (2002).

[50] `http://vlado.fmf.uni-lj.si/pub/networks/data/`.

[51] `http://jeffe.cs.illinois.edu/compgeom/biblios.html`.

[52] K. Norlen, G. Lucas, M. Gebbie, and J. Chuang, in *Proc. Inter. Telec. Soc.* (2002).

[53] `https://networkx.github.io/`.

[54] M. Tumminello, F. Lillo, and R. N. Mantegna, J Econ. Behav. Organ. **75**, 40 (2010).

[55] `http://finance.yahoo.com`.

[56] P. Holme and J. Saramäki, Phys. Rep. **519**, 97 (2012).

[57] M. Starnini, A. Baronchelli, A. Barrat, and R. Pastor-Satorras, Phys. Rev. E **85**, 056115 (2012).

[58] R. Pfitzner, I. Scholtes, A. Garas, C. J. Tessone, and F. Schweitzer, Phys. Rev. Lett. **110**, 198701 (2013).

[59] I. Scholtes, N. Wider, R. Pfitzner, A. Garas, C. J. Tessone, and F. Schweitzer, Nat. Commun. **5** (2014).

[60] M. Bazzi, M. A. Porter, S. Williams, M. McDonald, D. J. Fenn, and S. D. Howison, arXiv:1501.00040 (2014).

[61] C. Granell, R. K. Darst, A. Arenas, S. Fortunato, and S. Gómez, Phys. Rev. E **92**, 012805 (2015).

[62] T. Kawamoto and M. Rosvall, Phys. Rev. E **91**, 012809 (2015).

[63] D. Robinson and L. R. Foulds, Math. Biosci. **53**, 131 (1981).

[64] L. van Iersel, S. Kelk, N. Lekic, and L. Stougie, SIAM J. Discrete Math. **28**, 49 (2014).