Exploring the randomness of directed acyclic networks

Joaquín Goñi,[1] Bernat Corominas-Murtra,[2] Ricard V. Solé,[2,3,4] and Carlos Rodríguez-Caso[2,*]
[1]*Functional Neuroimaging Laboratory, Department of Neurosciences, Center for Applied Medical Research, University of Navarra, Pamplona, Spain*
[2]*ICREA-Complex Systems Laboratory, Universitat Pompeu Fabra (PRBB), Dr. Aiguader 88, 08003 Barcelona, Spain*
[3]*Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, New Mexico 87501, USA*
[4]*Institut de Biologia Evolutiva, CSIC-UPF, Passeig Maritim de la Barceloneta 37-49, 08003 Barcelona, Spain*
(Received 10 June 2010; revised manuscript received 27 October 2010; published 17 December 2010)

The feed-forward relationship naturally observed in time-dependent processes and in a diverse number of real systems—such as some food webs and electronic and neural wiring—can be described in terms of the so-called directed acyclic graphs (DAGs). An important ingredient of the analysis of such networks is a proper comparison of their observed architecture against an ensemble of randomized graphs, thereby quantifying the *randomness* of the real systems with respect to suitable null models. This approximation is particularly relevant when the finite size and/or large connectivity of real systems make inadequate a comparison with the predictions obtained from the so-called *configuration model*. In this paper we analyze two methods of DAG randomization as defined by the desired combination of two topological invariants (directed degree sequence and component distributions) aimed to be preserved. A highly ordered DAG, called *snake* graph, and an Erdös-Rényi DAG were used to validate the performance of the algorithms. Finally, three real case studies, namely, the *C. elegans* cell lineage network, a Ph.D. student-supervisor network, and the Milgram's citation network, were analyzed using each randomization method. Results show how the interpretation of degree-degree relations in DAGs with respect to their randomized ensembles depends on the topological invariants imposed.

DOI: 10.1103/PhysRevE.82.066115 PACS number(s): 89.75.Fb, 89.75.Hc, 05.65.+b

I. INTRODUCTION

Many relevant properties of complex systems can be described by an appropriate network representation of their elements and interactions [1–5]. Most of these networks are directed, i.e., there is a directional relationship between two elements defining who influences who in a given order. Among the class of directed networks, directed acyclic graphs (DAGs) are an important subset lacking feedback loops. This is especially suitable for the representation of evolutionary, developmental, and historical *processes* in which the time asymmetry determines a feed-forward (acyclic) flow of causal relations. In this context, DAGs constitute a formal representation of causal relations that display the direct effects of earlier events over latter ones. Citation networks are among their most paradigmatic cases [6,7]. In these networks nodes are scientific articles and directed links (arcs in the terminology of graph theory) stand for bibliographic citations among them. According to a chronological order, arcs are established from former articles to newer ones in a feed-forward manner.

In general, time-dependent processes have been formalized as DAGs. Examples of that comprehend article and patent citation networks [8,9], decision jurisprudence processes [10,11], and tree genealogies and phylogenies. Moreover, other relevant systems such as standard electric circuits [12], feed-forward neural [13], and transmission networks [14] are also suitably represented as DAGs.

The main objective of this paper is to explore the randomness—in topological terms—of real systems displaying a directed acyclic structure by the definition of randomization methods that preserves a fixed number of topological invariants. To this end, the design of null models to highlight the particular features characterizing a system with respect to a neutral or *random* scenario [2] is needed. In this context, the so-called configuration model [2,3,15,16] has been probed as a fruitful approximation to provide a null-model scenario of what is expected by *chance* in complex networks under the assumptions of sparseness, infinite size, and lack of correlations. However, little attention has been paid concerning DAGs. Indeed, a rigorous definition of random DAG from its directed degree sequence has been only recently proposed [17], rising the interest for its study through the configuration model approach. Borrowing the methodology to build random undirected graphs [16,18], degree sequence is visualized as a set of edge *stubs*. Hence, a random DAG is constructed by matching stubs according to certain order constraints until they are completely canceled [17]. Although this is an important contribution, some problems arise in using this methodology as the null-model reference of real nets. First, this methodology is dependent on how probable is to construct a graph from a degree sequence since not all of them produce a graph, i.e., they are not *graphical*. Additionally, configuration model assumptions are not fulfilled in real systems due to their finite size and the presence of densely connected regions.

An alternative approach used in this work is based on iterative processes of arc rewiring over the graph, keeping the graphical condition during the whole process of randomization. This is a relevant issue since the degree sequence, either directed or undirected, imposes a particular space of topological configurations rather limited—as we shall see in this work—for DAGs. Attending to this approach we can estimate where a real graph is placed preserving a graphical

*carlos.rodriguez@upf.edu

ensemble that holds some topological invariants. The two fundamental topological invariants considered in this work for a null-model comparison are the directed degree sequence and the component structure. Degree sequence (either directed or undirected) and the degree distribution have been typically chosen as invariants in the construction of random models [2,16,17,19]. However, as is well known in random graph theory, the existence of some graph satisfying a given degree sequence does not guarantee a single connected component containing the whole set of nodes, except at high connectivities. Therefore, sparse networks representing connected systems are expected to be fragmented during a randomization process. For instance, this may be an undesirable effect when studying historical processes since it breaks the flow of causality. Besides, there are also real systems that display more than a single connected component. Those disconnected components do not interact among them and, arguably, can be considered to be independent systems in terms of causality. It is worth to note that preservation of connected components in graph randomization processes has recently raised the interest of the network community [20].

According to the above considerations, in order to produce comparable ensembles for the evaluation of the randomness of a real DAG, we describe two randomization methods for DAGs paying attention to the conservation of directed degree sequence. These two models differ in the requisite of the connected component structure conservation. In order to test the success of these randomization techniques, algorithms were applied to two extreme—in terms of degree-degree relations—network models: an Erdös-Rényi (ER) DAG and a highly ordered graph denominated snake DAG. Their respective graph ensembles were evaluated by measuring the loss of degree-degree relations along different rewiring processes.

It is arguable to think that as the number and complexity of topological invariants increases, the space of possible random network is consequently reduced. Then, doing a comparison of graph ensembles keeping different topological invariant can provide a good insight about the effectiveness of our randomization methods. According to this, extreme models were also randomized with two additional raw methods based on the conservation of the undirected degree sequence (and one of them also on the connected component structure). Therefore, a total of four types of randomization were applied to these two extreme models. Randomized ensembles were compared among them in terms of their degree-degree relations. Once evaluated the success of our two methods they were applied to three real DAGs: a citation network, a Ph.D. student-supervisor network, and the cell lineage in the development of *Caebnorhabditis elegans* worm.

The paper is organized as follows: Sec. II offers the basic concepts related to DAGs. Section III explicitly defines the set of four randomization algorithms according to different topological invariants. Section IV describes and characterizes the randomness indicators, and it applies the randomization processes to the systems under study: two toy models—which enable us to validate the performance of the algorithms—and three real systems. Section V discusses the relevance of the obtained results.

II. ORDERING, CAUSALITY, AND FORMAL DEFINITION OF DAGs

In this section we discuss some mathematical properties of DAGs, their interpretation in terms of causal relations, and the problem of component structure conservation.

A. Basic definitions

Let $\mathcal{G}(V,E)$ be a directed graph, with $V=\{v_1,\ldots,v_N\}$ being the set of nodes, and the set of ordered pairs $E=\{\langle v_k,v_i\rangle,\ldots,\langle v_j,v_l\rangle\}$ the set of arcs—where the order $\langle v_k,v_i\rangle$ implies that there is an arrow in the following direction: $v_k\rightarrow v_i$. The *underlying graph* $\mathcal{G}_u$ of a directed graph $\mathcal{G}$ is an undirected graph with the same set of nodes $\mathcal{G}$, but whose arrows are neglected. Formally, if $\langle v_k,v_j\rangle\in E$, then the unordered pair—the edge—$\{v_k,v_j\}\in E_u$. Given a node $v_i\in V$, the number of outgoing arcs, to be denoted by $k_o(v_i)$, is called the *out-degree* of v_i. Similarly, the number of ingoing arcs of v_i is called the *in-degree* of v_i, denoted by $k_i(v_i)$.

A DAG is a directed graph characterized by the absence of cycles: if there is a *directed path* from v_i to v_k (i.e., there is a finite sequence $\langle v_i,v_j\rangle,\langle v_j,v_l\rangle,\langle v_l,v_s\rangle,\ldots,\langle v_m,v_k\rangle\in E$), then there is no directed path from v_k to v_i. Borrowing concepts from order theory [21,22], we refer to nodes with $k_i=0$ as *maximals* and those with $k_o=0$ as *minimals*. The absence of cycles ensures that at least there is one minimal node and one maximal node. Maximal nodes can be seen as inputs of a given computational or sequential process while minimal—or terminal—ones are the outputs of such a process. Furthermore, the acyclic nature permits us to define a node ordering by labeling all the nodes with sequential natural numbers. Thus, in a DAG there is at least one numbering of the nodes such that

$$(\forall\langle v_i,v_j\rangle\in E)\Rightarrow(i<j). \tag{1}$$

For this reason, DAGs have been also referred as *ordered graphs* [17].

B. Random DAGs

The theoretical roots of the concept of a random DAG are based on the so-called directed degree sequence [17]—as well as on the concept of random graph [16]. A random DAG $\mathcal{G}$ is a randomly chosen element of an ensemble of DAGs which share the directed degree sequence, denoted by $d(\mathcal{G})$, which is defined as follows:

$$d(\mathcal{G})=(k_i(v_1),k_o(v_1)),\ldots,(k_i(v_i),k_o(v_i)),\ldots. \tag{2}$$

The two numerical quantities composing every element of such a sequence, $k_i(v_k)$ and $k_o(v_k)$, encode the pattern of connectivity of every node of the graph. In general, the ensemble of random graphs containing N nodes is composed by all possible graphs whose connectivity pattern satisfies the directed degree sequence. If we only pay attention to the number of edges connected to a given node v_i—i.e., regardless the direction of the arrows—we define the *degree* of the node v_i as $k(v_i)=k_i(v_i)+k_o(v_i)$ [23] and, consistently, the *undirected degree sequence* of $\mathcal{G}$, is defined as

$$d_u(\mathcal{G}) = k(v_1), \ldots, k(v_i), \ldots . \tag{3}$$

However, it is clear that not any sequence of N pairs of natural numbers—or N natural numbers in the case of the undirected degree sequence—represents the degree sequence of an ensemble of some kind of random graphs containing N nodes [16,17]. There are indeed several restrictions that a (un)directed degree sequence must satisfy in order to represent a proper graph, i.e., a sequence to be graphical or *feasible* [16]. In the case of directed graphs, in- and out-degrees of the whole sequence must be consistent with the number of arcs, i.e.,

$$\sum_{i \le n} k_i(v_i) = \sum_{i \le n} k_o(v_i) = |E|. \tag{4}$$

Nevertheless, such a condition does not avoid the presence of cycles in the network structure. Consistent with the claim that DAGs depict systems where some unavoidable ordering among nodes is at work, we can ensure the generation of a given DAG if and only if there is a labeling of the nodes such that $v_i \to v_j$ implies $i < j$ [24]. Taking into account this ordering to build the graph, the directed degree sequence must also hold two conditions. First,

$$k_o(v_n) = k_i(v_1) = 0, \tag{5}$$

and, second,

$$(\forall v_l \in V) \quad \sum_{j>l} k_i(v_j) - \sum_{j \ge l} k_o(v_j) \ge 0. \tag{6}$$

Under conditions (4)–(6) it is ensured that a directed degree sequence will be graphical and able to represent the degree sequence of a given nonempty ensemble of DAGs.

C. Component structure and causal relations

The concept of component structure stems from the notion of undirected path: given two pairs of nodes $v_i, v_k \in V$, there is an *undirected path* among them if there is a finite sequence of edges such that it can be ordered sequentially, for example, $\{v_i,v_j\},\{v_j,v_l\},\{v_l,v_s\},\ldots,\{v_m,v_k\} \in E_u$. A *component* of $\mathcal{G}$ is a (maximal) subset of V by which an undirected path can be defined among any pair of nodes. The special features of a DAG impose constraints on the number of—DAG-like—components from a given directed degree sequence. Indeed, let $M(\mathcal{G})=\{v_k \in V: k_i(v_k)=0\}$ be the set of maximal nodes of a given DAG $\mathcal{G}$ and $\mu(\mathcal{G})=\{v_k \in V: k_o(v_k)=0\}$ be the set of its minimal nodes. Let $d(\mathcal{G})$ be the directed degree sequence of $\mathcal{G}$ which, by assumption, is graphical. Then, the number of (DAG) components of the graph $c(\mathcal{G})$ is bounded by

$$c(\mathcal{G}) \le \min\{|M|,|\mu|\}, \tag{7}$$

since any connected DAG must have, at least, $|\mu|=|M|=1$. Another constraint must be satisfied. There must exist a partition of the directed degree sequence $d(\mathcal{G})$ by which all $c(\mathcal{G})$ subsequences are graphical, i.e., they satisfy Eqs. (4)–(6).

III. RANDOMIZATION METHODS

This section describes four algorithms that provide randomized ensembles of DAGs considering the conservation of two topological constraints: the (un)directed degree sequence and the connected component structure of the original DAG. From a methodological point of view, we propose two algorithms of randomization that keeps the directed degree sequence of a DAG, one of them preserving the number and size of the connected components of the original DAG and the other one not considering this requisite. Furthermore, two additional randomization methods keeping the undirected degree sequence (analogously, one of them conserving component structure) are also included in this work. From a technical perspective, they are simpler and quite intuitive since they are derived from randomizing methods of the underlying graph. For the sake of completeness, they are also included in this methodological section. As we shall see, these latter methods allow us to evaluate how random can be a DAG given directed degree sequence (and component structure). The four algorithms described in this work are illustrated in Fig. 1. The order of presentation of the methods is the same for all figures and tables, and thus the letters used in Fig. 1 equivocally identify the methods of randomization.

A. Generating the ensemble from the undirected degree sequence

The simplest method of randomization preserving components consists of applying a random numbering to $\mathcal{G}^u$. This allows us to define order criteria to establish the direction of arrows. In this case, given an undirected pair $\{v_i,v_j\}$ we say that if $i<j$ we defined the order pair as $\langle v_i,v_j\rangle$, otherwise $\langle v_j,v_i\rangle$. Since $\mathcal{G}^u$ is preserved, undirected degree-degree relations are also conserved. Then, a suitable randomization should require a rewiring process on the underlying network able to destroy the presence of degree-degree relations. Once the underlying network is randomized, a simple numbering preceding an arrow reassignment provides a randomized DAG. Arguably, due the destructive nature of this method any possible degree-degree relation should be broken.

The first randomization method, denoted by the letter **a**, is depicted in Fig. 1(a). The steps of the algorithm are scheduled in the following:

(1) Given a DAG, $\mathcal{G}(V,E)$, we obtain its respective underlying network, $\mathcal{G}_u$.

(2) We obtain a random network conserving the undirected degree sequence of $\mathcal{G}_u$ and its component structure by a randomization process of denominated *local swap* [20]. Local swap is performed as follows: we randomly select an existing edge $\{v_i,v_j\}$ of $\mathcal{G}_u$ such that the two additional edges $\{v_i,v_k\}$ and $\{v_j,v_l\}$ also exist in $\mathcal{G}_u$, provided that v_k,v_i,v_j,v_l are all different. Then we proceed to make the rewiring by generating the edges $\{v_i,v_l\}$ and $\{v_j,v_k\}$ and removing the edges $\{v_i,v_k\}$ and $\{v_j,v_l\}$. If $\{v_i,v_l\}$ or $\{v_j,v_k\}$ already exist in $\mathcal{G}_u$, we abort the operation and we randomly select another edge satisfying the above described conditions to perform the local swap. According to [20], local-swap method can perform all edge rewiring except those that imply breaking the component structure. This process is iteratively repeated until achieving a suitable randomization of $\mathcal{G}_u$ or after a predefined number of iterations.

(3) Once the local-swap randomization of $\mathcal{G}_u$ is done, we label every node with an arbitrary natural number, from 1 to

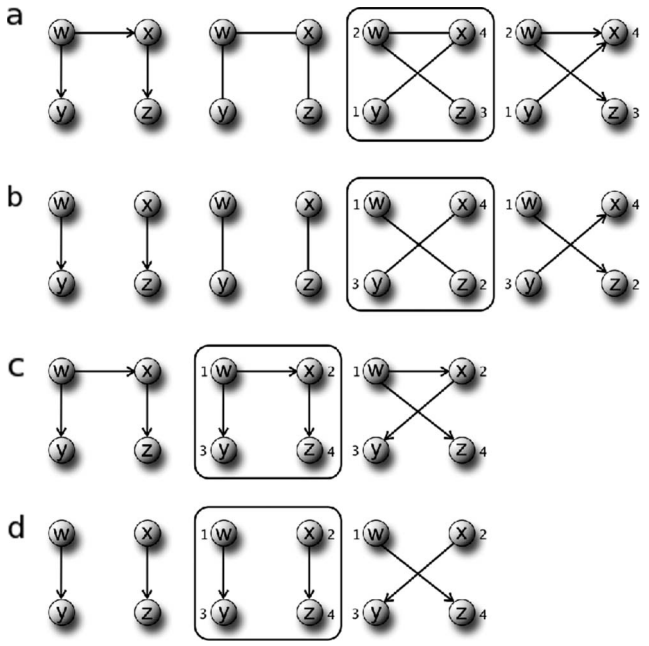

FIG. 1. Schematic representation of the four different DAG randomizations proposed in this work. Methods are alphabetically denoted [(a)–(d)]. Method **a**: randomization preserving undirected degree sequence and component size distribution. Method **b**: DAG randomization only preserving the undirected degree sequence. Method **c**: randomization preserving directed degree sequence and component size distribution. Method **d**: randomization preserving only directed degree sequence. Boxes represent where an iterative process is applied. Note that while vertices have invariable labels their associated numbers can be changed. From a computational point of view, every vertex (noted by its label) is associated to a fixed position in the adjacency matrix. In this way vertex reallocations in the matrix are avoided. As we shall see later this is an important aspect for the computation of dissimilarity.

N, with N being the size of the graph. No repetitions are allowed.

(4) We now proceed to defining the arrows taking into account the numbering of the nodes defined in the previous step. For every pair of connected nodes in the randomized version of $\mathcal{G}_u$, we define the arrow from lower to higher number's nodes. Formally, given an undirected pair $\{v_i,v_j\}$, where i,j are the respective labels obtained through the random numbering, if $i<j$ then $v_i \rightarrow v_j$, otherwise $v_i \leftarrow v_j$. The total order of natural numbers ensures the absence of cycles.

Method **b** consists of preserving the undirected degree sequence [19] but not preserving the component structure. The component structure is ensured by step (2) in method **a**. In this case, step (2) is replaced with a rewiring process: (i) selecting a pair of different edges $\{v_i,v_k\},\{v_j,v_l\}$ of $\mathcal{G}_u$, (ii) generating with probability $p=1/2$ either the edges $\{v_i,v_l\},\{v_j,v_k\}$ or the edges $\{v_i,v_j\},\{v_l,v_k\}$ (provided that both two edges are not already present), and (iii) removing the edges $\{v_i,v_k\},\{v_j,v_l\}$ [see Fig. 1(b)].

B. Generating the ensemble from the directed degree sequence

Once described the randomization of the undirected topological structure of the real DAG conserving component structure, we are going to consider the preservation of the directed degree sequence [see Eq. (2)]. This has an important physical interpretation since it implies that every node has an invariant number of inputs and outputs, as it happens with the components of an electronic device. Under such a restriction we can no longer work with the underlying graph but with the directed graph.

The proposed algorithm, denoted by method **c** [see Fig. 1(c)], begins with a numbering of the nodes resulting from the application of a *leaf-removal algorithm* [25] and a rewiring operation constrained by this numbering. Let us briefly revise how a leaf-removal algorithm works: from the original graph $\mathcal{G}$, we iteratively remove the nodes with $k_o=0$ until the complete pruning of the graph. According to this, a DAG can be *layered*, and thus a partial order between nodes can be easily established. Formally, the ith iteration of the leaf-removal algorithm defines the set $V_i \subseteq V$ of nodes where V_i corresponds the ith *layer* of the DAG. Then, any DAG can be redefined in terms of the resulting (ordered) layers of a leaf-removal algorithm, i.e.,

$$W(\mathcal{G}) = \{W_1, \ldots, W_l\}, \tag{8}$$

where no arc between nodes of the same layer is established.

Method **c** [see Fig. 1(c)] is defined as follows:

(1) Generate the set $W(\mathcal{G})$ by applying the leaf-removal algorithm.

(2) Perform a random numbering of the nodes in such a way that, given $v_i \in W_u$ and $v_k \in W_s$,

$$(u > s) \rightarrow (i < k). \tag{9}$$

(3) Select at random an arc $\langle v_k,v_j\rangle \in E$. Then we look for the presence of two nodes $v_i,v_l \in V$ by which either

$$\text{(i)} \quad \langle v_i,v_k\rangle,\langle v_l,v_j\rangle \in E, \tag{10}$$

or

$$\text{(ii)} \quad \langle v_k,v_i\rangle,\langle v_j,v_l\rangle \in E. \tag{11}$$

Notice that the absence of cycles makes these two options mutually exclusive.

(4) If condition (10) is satisfied, the arcs $\langle v_l,v_k\rangle$ and $\langle v_i,v_j\rangle$ are generated, and $\langle v_i,v_k\rangle$ and $\langle v_l,v_j\rangle$ are deleted, provided that the following conditions are satisfied: (1) $\langle v_l,v_k\rangle\langle v_i,v_j\rangle \notin E$ and (2) $l<k$ and $i<j$. If any of these three conditions does not hold, the rewiring event is aborted and another arc is newly selected at random.

If condition (11) is satisfied, the pairs $\langle v_k,v_l\rangle$ and $\langle v_j,v_i\rangle$ are generated, deleting $\langle v_k,v_i,\rangle\langle v_j,v_l\rangle$ arcs, provided that $\langle v_k,v_l\rangle\langle v_j,v_i\rangle \notin E$ and $k<l$ and $j<i$ conditions are satisfied. Again, if one of these two conditions does not hold the rewiring event is restarted.

Finally, the randomization method **d** preserves the directed degree sequence but does not preserve the component structure. In this case, step (3) is replaced with the following procedure: (i) select two arcs at random $\langle v_k,v_i\rangle,\langle v_l,v_j\rangle \in E$; (ii) generate the arcs $\langle v_k,v_j\rangle,\langle v_l,v_i\rangle$ provided that $\langle v_k,v_j\rangle,\langle v_l,v_i\rangle \notin E$ and that $k<j,l<i$. If some of these conditions do not hold, the process is aborted and we restart the rewiring event. (iii) If conditions are satisfied, $\langle v_k,v_i\rangle,\langle v_l,v_j\rangle$ are removed [see Fig. 1(d)].

IV. EXPLORING THE RANDOMNESS OF DAGs

In this section we apply the above-defined algorithms to some real topologies to construct an ensemble of randomized networks (also known as *surrogate data* in other scientific communities) preserving the defined topological invariants. First of all, we need to define proper measures to evaluate the level of randomness of our systems.

A. Testing the success of the randomization process

As it is described above, randomizations are subject to very restrictive constraints since not all (un)directed degree sequence configurations are graphical. Therefore, the success of DAG randomization processes must be properly evaluated. Two estimators were measured for this purpose. First, as a control of evolution process, a *dissimilarity* parameter $\mathcal{D}$ is proposed to measure how the graph differs along the iterations with respect to the original one. Second, the deterioration of present degree-degree relations is also reported by means of an estimator borrowed from information theory, the so-called *joint entropy* [26]. As we shall see, the combination of these two measures provides a good insight about the randomization process, especially when the degree-degree relations of a graph are close to the expected at random or an extreme (un)directed degree sequence with a very limited number of graphical solutions. In such cases, minor changes in the joint entropy might be expected, and then the increase in the dissimilarity would indicate that randomization method is working and the graph is not stacked in a particular configuration. According to this, we are ready to define a working criterion of ending the randomization process. We consider that a DAG has acquired a stable randomized configuration if dissimilarity has been stabilized along effective rewiring events during the randomization process. For practical issues randomization methods were iterated and evaluated in powers of 2 until dissimilarity showed an increase with respect to the previous evaluation below 5%. Additionally, *efficiency* or *effectiveness* of the randomization process was evaluated by the relative frequency of successful iterations. In this case, each point represents the percent of successful iterations in the current randomization interval, i.e., the iterations performed from the previous point, which is the previous power of 2. An iteration is said to be successful or effective when the selected nodes and links (according to each method as expressed in Fig. 1) permit a change in the network.

1. Dissimilarity

The dissimilarity parameter $\mathcal{D}$ between two graphs is the relative frequency of arc mismatches between them, i.e., the *Hamming distance* of their adjacency matrices. In the context of a randomization process, let us define $\mathcal{A}$ as the adjacency matrix of an original graph ($\mathcal{G}$) and $\mathcal{A}^t$ as the adjacency matrix of the graph resulting from the application of t randomization iterations ($\mathcal{G}^t$). We consider that vertex position in the adjacency matrix must be kept invariable along randomization process [27]. Under such a restriction, any change in the matrix will be only due to an arc rewiring event. Then, dissimilarity can be expressed as

$$\mathcal{D}(\mathcal{G},\mathcal{G}^t) \equiv \frac{1}{2|E|}\sum_{i,j} 1 - \delta(\mathcal{A}_{ij},\mathcal{A}^t_{ij}), \tag{12}$$

where δ is the Kronecker delta and $|E|$ denotes the number of arcs of both $\mathcal{G}$ and $\mathcal{G}^t$, since the undirected degree sequence is preserved in the four randomization methods.

2. Degree-degree joint entropy

Given two random variables X,Y, the joint entropy between X and Y, $H(X,Y)$, is given by

$$H(X,Y) = -\sum_{x,y} \mathbb{P}(x,y)\log \mathbb{P}(x,y), \tag{13}$$

with $\mathbb{P}(X,Y)$ being the joint probability of the pair of outcomes x,y happening together—throughout this paper, $\log_2$ will be used for the joint entropy measurements. Let us detail how this concept is translated in a useful way to produce a set of four measurements that describe the different degree-degree relations within a graph.

Joint entropy for the evaluation of degree-degree relations can be expressed as

$$H(\mathcal{G}_u) = \sum_{i\geq j} \mathbb{P}(i,j)\log \mathbb{P}(i,j), \tag{14}$$

where $\mathbb{P}(i,j)$ defines the probability of finding a randomly selected edge that connects two nodes $v_m, v_l \in V$ such that $k(v_m)=i$, $k(v_l)=j$. This measurement was found to be more appropriate than other existing alternatives for the purpose of monitoring the degree-degree interplay along the randomization processes [28]. The subscript "u" emphasizes that such a measure does not take into account the directed nature of the graph. Joint entropy quantification for degree-degree considering the directed degree sequence can be easily derived. In this case three additional joint entropies attending directedness can be considered, namely, the ones accounting for k_ik_o, k_ik_i, and k_ok_o relations. Although more elaborated definitions of these three probabilities could be proposed, for the sake of simplicity, we assessed whether two nodes with given degrees tend to be connected, no matter what the direction of the arrow connecting them is. Then the (k_ik_i)-joint entropy of a directed graph $\mathcal{G}$, $H^{i,i}(\mathcal{G})$, is expressed as

$$H^{i,i}(\mathcal{G}) = \sum_{k\geq j} \mathbb{P}_{i,i}(k,j)\log \mathbb{P}_{i,i}(k,j), \tag{15}$$

where $\mathbb{P}_{i,i}(k,j)$ is the probability of that an edge chosen at random connects a node with $k_i=i$ to another with $k_i=j$. A similar expression is obtained for $H^{o,o}(\mathcal{G})$. Finally, $H^{i,o}(\mathcal{G})$ is defined as

$$H^{i,o}(\mathcal{G}) = \sum_{k,j} \mathbb{P}_{i,o}(k,j)\log \mathbb{P}_{i,o}(k,j). \tag{16}$$

Notice that this is the only case where $\mathbb{P}_{i,o}(k,j) \neq \mathbb{P}_{i,o}(j,k)$.

The ensemble of random graphs produced from a original graph $\mathcal{G}$ after t iterations can be associated to the undirected degree-degree joint entropy distribution of its conforming graphs, which can be characterized by its mean $\langle H(\mathcal{G}^t_u)\rangle$ and

its standard deviation $\sigma(H(\mathcal{G}_u^t))$. The closeness of the joint entropy value of the original graph to the ensemble distribution can be quantified by means of the Z score, which reads

$$Z(\mathcal{G}_u) = \frac{H(\mathcal{G}_u) - \langle H(\mathcal{G}_u^t)\rangle}{\sigma(H(\mathcal{G}_u^t))}. \tag{17}$$

The statistical significance level was set at $p < 0.001$, which for two tails corresponds to $|Z| > 3.27$. Significant values were denoted by Z^* in the tables describing joint entropy values of graphs. Values of $Z > 3.27$ mean that the degree-degree relations at the original network $\mathcal{G}$ are significantly high with respect to the Z distribution of its random ensemble. Values of $Z < -3.27$ mean that the degree-degree relations at the original network $\mathcal{G}$ are significantly low with respect to the Z distribution of its random ensemble. Finally, values within the range $[-3.27, 3.27]$ indicate that no significant differences in the degree-degree relations were found between the original graph and its randomized ensemble. Analogously, we can compute $\langle H^{i,o}(\mathcal{G}^t)\rangle$, $\langle H^{i,i}(\mathcal{G}^t)\rangle$, $\langle H^{o,o}(\mathcal{G}^t)\rangle$, and its associated Z scores at the step t of the randomization process.

B. Extreme graphs

Prior to study the randomness of real DAGs, we construct two extreme topologies in order to evaluate the behavior of the algorithms using the measures defined above. The first model, random DAG, permits us to test the randomization methods in a highly disordered degree-degree scenario. In terms of degree-degree joint entropy, minimal changes are expected along the randomization processes. The second model, snake DAG, permits the same test but in a highly ordered scenario where large increments of joint entropy values should be observed in the obtained randomized ensembles.

1. Random-DAG model

The first one is a completely degree-degree disordered DAG up to finite-size effects. Let V be a set of N nodes, by which the probability p for two of them to be connected is constant. This is the definition of the ER graph. Once we have an ER graph, we randomly label the nodes of V sequentially, from 1 to N. Finally, we define the direction of the arrows by looking at the labeling of the nodes and observing condition (1). We will refer to this model as random DAG.

For methodology evaluation we created a random DAG of $|V| = 600$ and $|E| = 6000$ and an ensemble of 500 randomized graphs for each method. See Fig. 2(a) for an example of this graph. As shown in Table I and Fig. 3 the degree-degree relations of the random DAG are neither significantly low nor significantly high with respect to any of its randomized ensembles. This result indicates that the four randomization methods proposed here did not produce significant biases in the degree-degree relations with respect to the original random DAG. It is worth noting that dissimilarity is a useful indicator for noncorrelated graphs due to a nonsignificant variation of the joint entropy that can be used as an estimator of the success of the random process. In this case, dissimilarity variation indicates that randomization processes have not been initially stacked in a particular graph configuration.

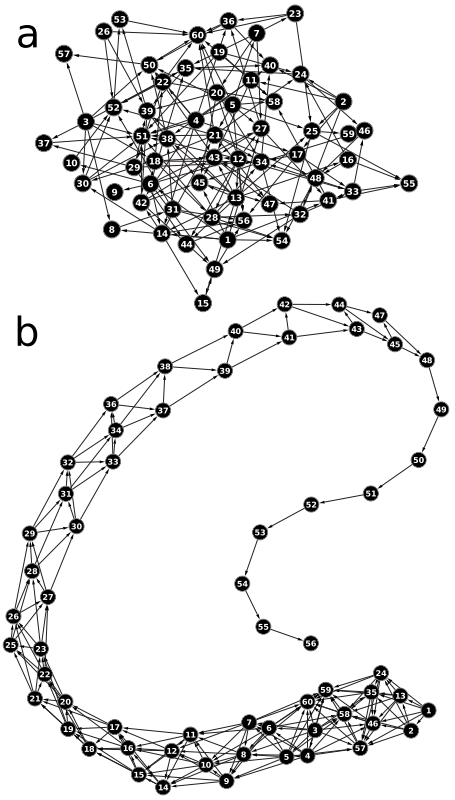

FIG. 2. (a) Illustration of a random DAG with $N=60$ and $\langle k\rangle = 3$. (b) Illustration of a snake DAG with $N=60$ and $K=6$. A particular numbering satisfying the DAG condition is indicated for these graphs. Note that arrows go from nodes with smaller numbers to nodes with larger numbers.

It is worth to mention that since graph fragmentation is unlikely to happen under the values of $\langle k\rangle$ used for graph construction, methods **a** and **b** on a hand and methods **c** and **d** on the other are indistinguishable in terms of joint entropy. Interestingly, Table I shows that standard deviations produced by methods **a** and **b** were one order of magnitude higher than for methods **c** and **d**. Another interesting result is that $H^{i,i}$ and $H^{o,o}$ exhibited higher differences, respecting the original graph for methods **a** and **b** than for methods **c** and **d**. These two facts agree with that, although random DAG is a noncorrelated network, the conservation of the directed degree sequence of the original network by the use of methods **c** and **d** reduces the space of graphical solutions. As expected, this does not occur in **a** and **b** methods. Interestingly, observing the differences among methods we can conclude that the solutions of more restrictive methods (**c** and **d**) are included within the space of solutions of the methods with softer topological restrictions (**a** and **b**).

2. Snake-DAG model

In order to complement the random-DAG model with an opposite extreme case, we construct a highly degree-degree ordered acyclic graph, hereafter called snake DAG. In this graph, vertices of the same degree tend to be connected

TABLE I. Joint entropy values for a random DAG of $|V|=600$ and $|E|=6000$ and a snake DAG ($|V|=600$ and $K=6$) compared with sets of 500 randomized graphs after 2^{16} iterations of each of the four randomization methods (alphabetically denoted). Symbol (*) denotes significant differences.

Method	$\mathcal{D}$	$H^u(\mathcal{G}_u^t)$	$H^{i,o}(\mathcal{G}^t)$	$H^{i,i}(\mathcal{G}^t)$	$H^{o,o}(\mathcal{G}^t)$
Random graph		8.147	9.224	9.234	9.202
a	0.98	8.145 ± 0.003 ($Z=0.56$)	9.20 ± 0.02 ($Z=1.13$)	9.19 ± 0.04 ($Z=0.93$)	9.19 ± 0.04 ($Z=0.27$)
b	0.98	8.145 ± 0.003 ($Z=0.67$)	9.20 ± 0.02 ($Z=1.13$)	9.19 ± 0.04 ($Z=1.01$)	9.20 ± 0.04 ($Z=0.14$)
c	0.96	8.144 ± 0.003 ($Z=0.87$)	9.219 ± 0.003 ($Z=1.61$)	9.226 ± 0.004 ($Z=2.08$)	9.202 ± 0.004 ($Z=-0.06$)
d	0.96	8.144 ± 0.003 ($Z=0.90$)	9.217 ± 0.003 ($Z=2.41$)	9.223 ± 0.004 ($Z=2.76$)	9.200 ± 0.004 ($Z=0.47$)
Snake graph		2.998	4.694	4.524	4.462
a	0.99	5.16 ± 0.002 ($Z=-110.74^*$)	6.97 ± 0.02 ($Z=-142.06^*$)	6.97 ± 0.03 ($Z=-77.85^*$)	6.97 ± 0.03 ($Z=-91.33^*$)
b	0.99	5.281 ± 0.003 ($Z=-674.41^*$)	6.98 ± 0.02 ($Z=-137.3^*$)	6.97 ± 0.03 ($Z=-82.06^*$)	6.97 ± 0.03 ($Z=-71.45^*$)
c	0.93	4.87 ± 0.03 ($Z=-64.46^*$)	5.47 ± 0.02 ($Z=-37.38^*$)	5.59 ± 0.02 ($Z=-52.99^*$)	5.29 ± 0.03 ($Z=-28.43^*$)
d	0.94	4.97 ± 0.03 ($Z=-77.62^*$)	5.47 ± 0.01 ($Z=-52.37^*$)	5.61 ± 0.02 ($Z=-66.26^*$)	5.25 ± 0.02 ($Z=-39.20^*$)

among them, giving rise to a high degree-degree relation and thus very low joint entropy values. In the following lines we outline the construction of this network.

Let us consider $K \equiv k_o^{max}$ as the highest out-degree to appear in the resulting graph. Let V be the set of nodes such that there exists an integer n by which $nK=|V|$. We then perform a partition of V in K different subsets:

$$\mathcal{P}(V) = \{V_1, \ldots, V_K\}. \tag{18}$$

In this partition, for any $V_i \in \mathcal{P}(V)$, $|V_i|=n$. We sequentially number the nodes of the set V in the following way: for the subset of nodes V_1, the label will run from 1 to n, thus obtaining

$$V_1 = \{v_1, \ldots, v_n\}.$$

For the subset of nodes V_2, the label will run from $n+1$ to $2n$:

$$V_2 = \{v_{n+1}, \ldots, v_{2n}\}.$$

We follow the numbering by using the criteria that the nodes of subset V_m will be labeled from $(m-1)n+1$ to in, until all the nodes of V are numbered. We then identify the label of the partition with the out-degree of the nodes belonging to it, namely,

$$(v_i \in V_m) \Rightarrow (k_o(v_i) = K - m). \tag{19}$$

Now we proceed to defining the connections: for any $v_i \in V_m$, we will have the following arcs:

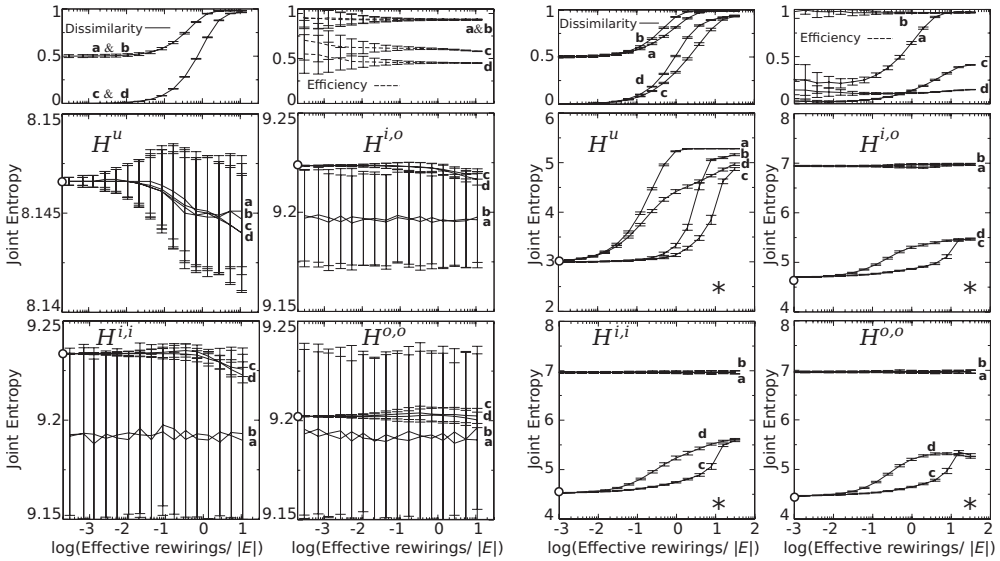

FIG. 3. Evolution of the randomization process along effective rewiring showing dissimilarity, efficiency, and joint entropy values for a random DAG of $|V|=600$ and $|E|=6000$ (six panels, left) and a snake DAG ($|V|=600$ and $K=6$) (six panels, right). Sets of 500 randomized graphs were produced until 2^{16} iterations of the four randomization methods. The mean and the standard deviation of 500 are shown for each point. The joint entropies with significant Z values are highlighted with $(*)$. Open circles in the vertical axis indicate the joint entropy value of the original graph in each case. X axes are expressed in $\log_{10}$. See Table I for numerical details.

$\langle v_i, v_{i+1}\rangle, \ldots, \langle v_i, v_{i+K-m}\rangle$. This process excludes node $v_{|V|}$ which will only receive an arc from $v_{|V|-1}$. We observe that, in general, both v_1 and v_2 belong to V_1. Finally, to break the extreme symmetry of the obtained net, we introduce a minimal source of noise by renumbering a small fraction ~ 0.05 of the nodes with a further arrow orientation, consistent with the new numbering, as depicted in Eq. (1). This modification was required since the snake-graph model without noise was computationally not affordable when trying to obtain a complete randomization. However, time execution was drastically reduced with this little fraction noise in the network.

Analogous to the experiment performed with a random DAG, we create a snake DAG of $|V|=600$ and $|E|=2099$ ($K=6$) and, for each method, an ensemble of 500 randomized graphs product of 2^{16} iterations. See Fig. 2(b) for an example of this graph. As shown in Fig. 3 (right) and Table I, all the degree-degree relations of the snake DAG were highly significant with respect to any of the randomized ensembles. This result indicates that the four randomization methods proposed here are able to successfully deteriorate the high degree-degree relations present at the snake DAG.

As mentioned above, methods **a** and **b** preserve the undirected degree sequence, while methods **c** and **d** do the same with the directed degree sequence. While the latter methods are randomization processes that iteration by iteration modify the DAG under study, the former ones proceed in a different way. They can be seen as iterative processes acting over the underlying graph. In this sense, H^u is iteratively modified by means of destroying presumable degree-degree relations. However, $H^{i,i}$, $H^{i,o}$, and $H^{o,o}$ are only altered in the last step of the algorithm, i.e., the numbering process. In this process any directed degree-degree relation is lost and the directed graph is, on average, modified by 50% since the direction of arcs are randomly changed (see dissimilarity in Fig. 3). Hence, methods **a** and **b** break, somehow, the iterative concept of a randomization process on a DAG since this really happens on its underlying graph. This can be observed in Fig. 3 where methods **a** and **b** exhibit a different value of entropy and dissimilarity at the first iteration. By contrast, methods **c** and **d** gradually differ from the initial value along the production of effective rewiring. Interestingly, not only directed joint entropies are informative for DAG randomization; we can observe that in the snake-graph randomization significant variation of H can be found even when the underlying network has not been randomized. This is especially dramatic for methods **a** and **b** as Fig. 3 shows. By this reason, dissimilarity and also both undirected and directed versions of joint entropy should be observed for the evaluation of the randomization process.

C. Real biological and social DAGs

Results of previous section have checked the behavior of methods **c** and **d** by comparing them with methods **a** and **b** in two extreme models: snake DAG and random DAG with high and low degree-degree relations, respectively. In this section we proceed to evaluating three DAGs representing real systems: the *C. elegans* cell lineage network, the Milgram's citation network, and a Ph.D. student-supervisor network (Fig. 4) with methods **c** and **d**, i.e., preserving the directed degree sequence with and without conserving the connected component structure of the original graph.

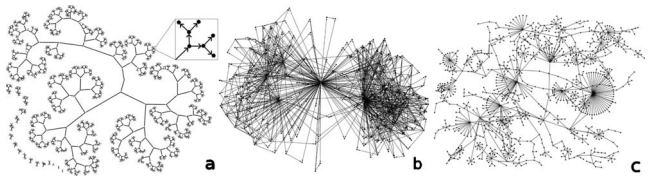

FIG. 4. Real biological and social DAGs studied in this work. *Caebnorhabditis elegans* cell lineage network (left), Milgram's citation network (center), and Ph.D. student-supervisor network (right).

1. C. elegans cell lineage network

The first system chosen is a cell lineage network. Briefly, it captures the genealogical pedigree of cells related through mitotic division during its development in a treelike structure. The cell lineage network of *C. elegans* was retrieved from the WormBase [29] *C. elegans* repository. In this network the initial egg division (the giant component) and alternative variants of neural postembryonic cell lines are included in an 18-component graph representation. The randomization methods were applied up to fulfill the ending criterion. The dissimilarity values reached were over 0.95 in all cases, indicating a successful alteration of most of the arcs under the different topological invariants.

Figure 5 shows representative DAGs obtained from **c** and **d** randomization methods. Interestingly, Fig. 5 (right panel) shows that the treelike structure and the number of graph components are conserved by just only preserving the directed degree sequence invariant. The reason is that the regular pattern of $k_i=1$ for all nonmaximal nodes in its directed degree sequence is graphical only in a tree structure. In this particular situation, the number of DAG components coincides with the number of maximal nodes but not with their size of components [see Fig. 4(a)]. This contrasts with the big fragmentation (up to 200 components) obtained from the application of the raw method **b** (data not shown). Figure 5 and Table II show that although methods produced effective rewiring and increased the dissimilarity, no significant deviations in joint entropies were found due to the very restrictive differences between the original and randomized ensembles, even at very small standard deviations.

2. Milgram's citation network

The second system is a sample of the process of article citation. The chosen system used to illustrate this process is the resulting network containing the papers that (1) cite "S. Milgram's 1967 Psychology Today" paper or (2) use *Small World* in the title. This network was retrieved from Pajek's network data set [30].

Figure 6 and Table III show that this DAG displays statistical deviations of undirected and directed joint entropies to all randomized ensembles, except the case of $H^{i,i}$.

This example illustrates how a randomization process destroys local associations and the heterogeneous partition ob-

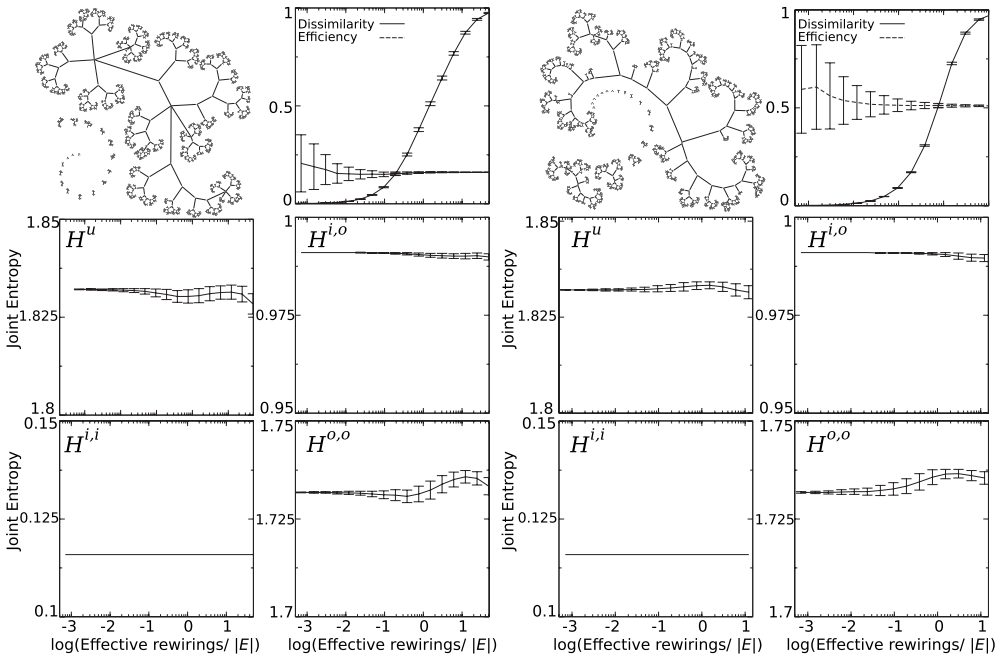

FIG. 5. Randomization process for *Caebnorhabditis elegans* network using methods **c** (left) and **d** (right). A six-panel composition for each method shows a prototypic randomized network and the evolution of dissimilarity, efficiency, and the undirected and directed joint entropies. Evolution of the randomization process is shown along effective rewiring. Randomization for method **c** was completed after 2^{17} iterations, while method **d** required 2^{15} iterations according the ending criterion for randomization processes. The mean and the standard deviation of 500 graph randomizations are shown for each point. The joint entropies with significant Z values are highlighted with $(*)$. X axes are expressed in $\log_{10}$. See Table II for numerical details.

served in the original DAG [Fig. 4(b)]. In this case, due to the high connectivity of the original DAG—it is worth to note that such a graph contains several nodes whose connectivity is $\mathcal{O}(|V|)$—fragmentation is unlikely to happen due to a high average connectivity. In such a situation, it is arguable to think that both methods must show the same behavior. Interestingly, Table III shows the same values for joint entropies and dissimilarities. Differences come from the efficiency of the method, which in general for all the real networks analyzed in this work method **c** was lower than **c**. This result indicates that the proposed methods are indistinguishable when DAGs have enough average degree to avoid fragmentation.

Furthermore, as a side effect of this high degree, an upper boundary below maximal value of dissimilarity is imposed depending on the randomization method used, even in the simplest case of the randomization of the underlying network (data not shown). A limit case of that is provided by a clique conformation where a directed acyclic condition is obtained by described numbering process. In this situation no rewiring is possible since all possible arc combinations satisfying the directed acyclic condition are already present in the network.

3. Ph.D. student-supervisor network

The last system evaluated in this paper contains the ties between Ph.D. students and their supervisors in theoretical computer science. Each arc points from a supervisor to one of its Ph.D. students [Fig. 4(c)]. Data were retrieved from Pajek's network data set [30]. This network illustrates just the intermediate situation between the two previous examples. It is a DAG able to be fragmented (when DAG component conservation is not imposed) but with *just right* connectivity: too low to avoid a large fragmentation but not too high to impose an upper bounding in dissimilarity, being, jointly to the random DAG studied above, the DAG structure fairly closer to the assumptions of the configuration model. Interestingly, contrasting with the *C. elegans* behavior when randomized with method **d**, little fragmentation is found. In other words, strictly speaking, the tree architecture but not a softer configuration, as that observed in Ph.D. student-

TABLE II. Joint entropy values for the original *C. elegans* network and a set of 500 randomized networks originated from methods **c** and **d**. Symbol $(^*)$ denotes significant differences and $(^a)$ indicates that Z score is not computable due to $\sigma=0$.

Method	$\mathcal{D}$	$H^u(\mathcal{G}_u^t)$	$H^{i,o}(\mathcal{G}^t)$	$H^{i,i}(\mathcal{G}^t)$	$H^{o,o}(\mathcal{G}^t)$
$\mathcal{G}$ orig.		1.832	0.991	0.116	1.732
c	0.97	1.828 ± 0.003 $(Z=1.41)$	0.990 ± 0.001 $(Z=1.31)$	0.116 ± 0.0^a	1.733 ± 0.002 $(Z=-0.72)$
d	0.98	1.832 ± 0.002 $(Z=0.36)$	0.990 ± 0.001 $(Z=-1.50)$	0.116 ± 0.0^a	1.735 ± 0.002 $(Z=-2.37)$

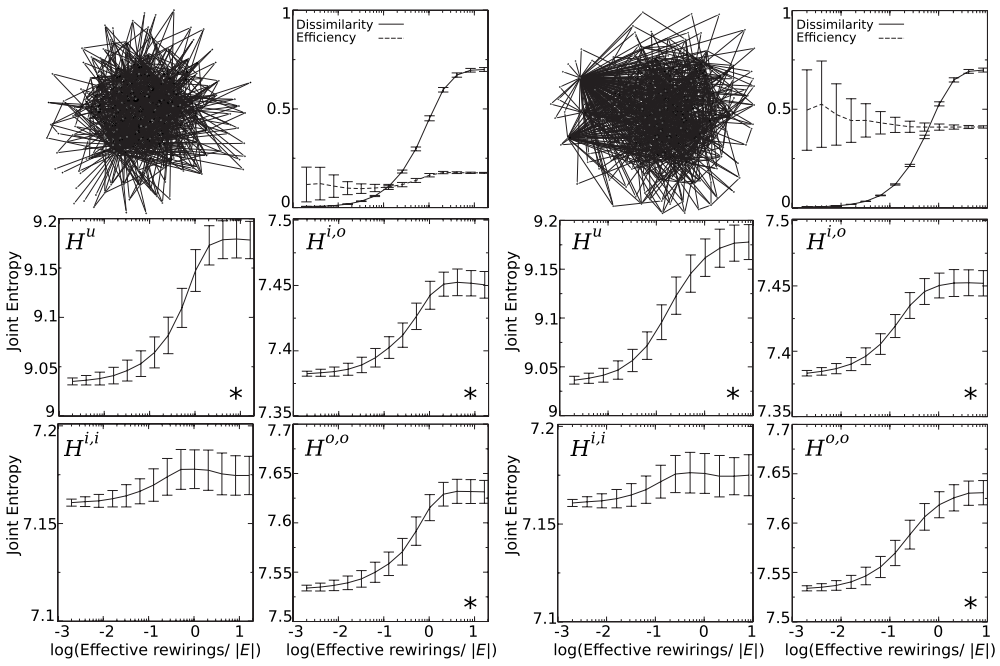

FIG. 6. Randomization process for Milgram's citation network using methods **c** (left) and **d** (right). A six-panel composition for each method shows a prototypic randomized network and the evolution of dissimilarity, efficiency, and the undirected and directed joint entropies. Evolution of the randomization process is shown along effective rewiring. Randomization for method **c** was completed after 2^{14} iterations, while method **d** required 2^{13} iterations according the ending criterion for randomization processes. The mean and the standard deviation of 500 graph randomizations are shown for each point. The joint entropies with significant Z values are highlighted with $(*)$. X axes are expressed in $\log_{10}$. See Table III for numerical details.

supervisor network, is able to preserve the component distribution (Fig. 7). In this network, the dissimilarity value reached was 0.98 in both cases, indicating a successful alteration of most of the arcs independent of graph fragmentation.

Table IV displays statistically significant low joint entropy values for H^u and $H^{o,o}$. A less clear behavior was observed for $H^{i,i}$ where Z values were close to signification. In these cases joint entropies exhibited higher values for method **d** than for **d** illustrating the softer restriction that operates in the randomization in method **d**. Contrasting to this $H^{i,o}$ does not exhibit any significant difference. Note that contrasting with Milgram's citation network, these methods, although fairly similar, are not strictly indistinguishable when fragmentation occurs.

V. DISCUSSION

Is there any differential trait in the architecture of causal relations representing historical processes, formal chains of reasoning, or flows in gradient-dissipative systems? The first step dealing with this question begins with the generation of random null-model metaphors. Tackling this problem from the perspective of graph theory, the construction of random graph ensembles provides a framework to observe what is expected by simple combinatorics under a number of constraints to be satisfied. In this paper we present a set of two algorithms based on iterative processes of rewiring for the construction of DAG random models. The difference between algorithms stems from the conservation or not of the connected component distribution. In contrast to other methods of random model construction, this approach works within the space of graphical solutions providing a feasible computational approximation for the exploration of such a graphical space, considering a defined number of topological invariants in the null-model ensemble generation.

Two methods proposed in this paper (**c** and **d**) preserve the directed degree sequence. First, they were compared with two alternative raw methods where only undirected degree sequence was conserved (methods **a** and **b**). This comparison was evaluated through the analyses of two extreme graph models: random DAG and snake DAG. Second, three real

TABLE III. Joint entropy values for the original *Milgram's citation* network and two sets of 500 randomized networks originated from methods **c** and **d**. Symbol $(^*)$ denotes significant differences.

Method	$\mathcal{D}$	$H^u(\mathcal{G}_u^t)$	$H^{i,o}(\mathcal{G}^t)$	$H^{i,i}(\mathcal{G}^t)$	$H^{o,o}(\mathcal{G}^t)$
$\mathcal{G}$ orig.		9.03	7.38	7.16	7.53
c	0.70	9.18 ± 0.02 $(Z=-7.63^*)$	7.45 ± 0.01 $(Z=-6.75^*)$	7.17 ± 0.01 $(Z=-1.44)$	7.63 ± 0.01 $(Z=-8.49^*)$
d	0.70	9.18 ± 0.02 $(Z=-8.04^*)$	7.45 ± 0.01 $(Z=-7.25^*)$	7.17 ± 0.01 $(Z=-1.38)$	7.63 ± 0.01 $(Z=-7.97^*)$

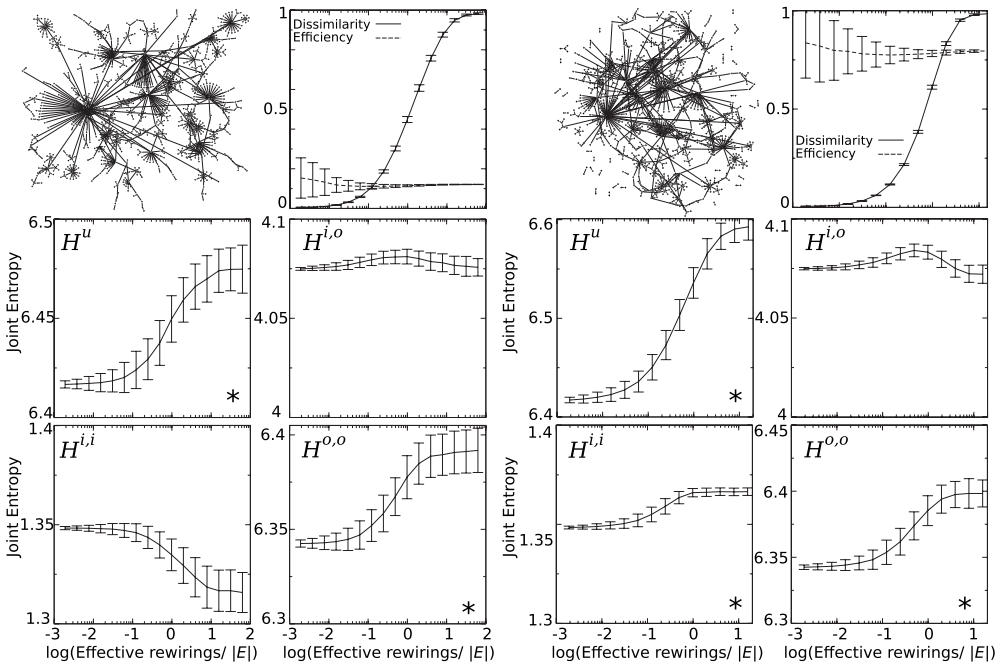

FIG. 7. Randomization process for Ph.D. student-supervisors network using methods **c** (left) and **d** (right). A six-panel composition for each method shows a prototypic randomized network and the evolution of dissimilarity, efficiency, and the undirected and directed joint entropies. Evolution of the randomization process is shown along effective rewiring. Randomization for method **c** was completed after 2^{16} iterations, while method **d** required 2^{14} iterations according the ending criterion for randomization processes. The mean and the standard deviation of 500 graph randomizations are shown for each point. The joint entropies with significant Z values are highlighted with $(*)$. X axes are expressed in $\log_{10}$. See Table III for numerical details.

systems were studied by the evaluation of dissimilarity and joint entropy. While the former indicates whether connections have been actually changed after randomization, the latter quantifies the *disorder* or uncertainty in the degree-degree relations, thereby being an indicator of randomness. In this context, it is worth to mention that other measures such as assortative mixing [31,32] or mutual information [33] have been suggested for the evaluation of degree-degree correlations. In essence, these measures compare these correlations with the expected one obtained from the *remaining degree* information. A problem arises when a proper definition of remaining degree attending directedness needs the information of the directed degree sequence because the latter is a topological feature not preserved in all of our methods (methods **a** and **b**). Therefore, measures based on remaining degree information, although extensively used as estimators of degree-degree relations in the network literature [31–33], cannot be applied in this work for a comparative evaluation of our methodology.

To overcome these limitations, joint entropy was used as a raw measure of uncertainty once defined to be applied to directed graphs leading to four alternative descriptors according to in- and out-degree information. Furthermore, the significance of the variation of degree-degree relations between the random ensembles and the original graph was evaluated using a Z-score estimator. The analysis of network models verified that our methods do not produce a bias when applied to the random-DAG model while they produced a significant increase in disorder of the degree-degree relations on the snake-DAG model when randomized (see Table I). When studying real systems, our analyses revealed that all the methods produced an H^u greater than its respective original value, indicating that the randomizations performed affect the pattern of arrows.

Additionally, our results show that preserving the component size structure is an important aspect to take into account since it has dramatic effects when the network is markedly sparse. This is the case of Ph.D. student-supervisor DAG, by which randomizations not preserving the component size produced a graph fragmentation. However, this does not occur in a treelike structure as is the case of *C. elegans* network, in which the number of maximals imposes the number

TABLE IV. Joint entropy values for the *Ph.D. student-supervisor* network and a set of 500 randomized networks originated from methods **c** and **d**. Symbol $(^*)$ denotes significant differences.

Method	$\mathcal{D}$	$H^u(\mathcal{G}_u^t)$	$H^{i,o}(\mathcal{G}^t)$	$H^{i,i}(\mathcal{G}^t)$	$H^{o,o}(\mathcal{G}^t)$
$\mathcal{G}$ orig.		6.42	4.075	1.348	6.34
c	0.98	6.47 ± 0.01 $(Z=-4.83)$	4.076 ± 0.005 $(Z=-0.23)$	1.32 ± 0.01 $(Z=3.22)$	6.39 ± 0.01 $(Z=-4.25^*)$
d	0.98	6.59 ± 0.01 $(Z=-13.06^*)$	4.072 ± 0.005 $(Z=0.58)$	1.366 ± 0.002 $(Z=-9.81^*)$	6.39 ± 0.01 $(Z=-5.50^*)$

of components, but not their size when applying method **d**. Finally, it is noteworthy that high average degree guarantees the preservation of the giant component in both randomization methods. Hence, DAGs with this feature contain an *intrinsic preserver* of components that leads, in those cases, to almost indistinguishable topologies.

Another important observation is related to the small values displayed by standard deviations in joint entropies. When extreme graphs were evaluated, methods **c** and **d** are one order of magnitude lower than the ones obtained for methods **a** and **b**. This suggests that just directed degree sequence conservation is enough to severely reduce the space of graphical configurations. Consistently, it was observed that, in general, methods **c** and **d** provided lower Z values than methods **a** and **b**. However, the small divergence of the obtained values is not explained by a noneffective rewiring since high values of dissimilarity were reached. An interesting exception was found in the Milgram's citation network where dissimilarity values after processes of randomization were markedly lower than those observed in the other real networks, as well as in the extreme models. An explanation can be found in the presence of *superhubs*, nodes whose connectivity is $\sim\mathcal{O}(|V|)$. This introduces a strong constraint in the rewiring, difficult—even impossible—to overcome. Nevertheless, Fig. 6 illustrates that the original network seems to be clustered in two regions. Using the same layout for randomized graphs, we did not found this clustered view, suggesting that rewiring process has actually produced an impact on the topological structure. By contrast, *C. elegans* randomized ensembles were completely shuffled—as indicated by the high values of dissimilarity—but degree-degree relations were not significantly altered. In this context, correct evaluation of the randomization process comes from the observation of both dissimilarity and joint entropies. In fact, this is a consequence of the limited space of possibilities permitted by the extreme directed degree sequence. Therefore, very small variations were found in the joint entropies (notice the case of $H^{i,i}=0$ for method **c**). On the one hand, when directed degree sequence and component structure are not preserved, tree configuration is unlikely to happen by chance. On the other hand, tree structure is practically the only solution when directed degree sequence is preserved even not conserving the component size distribution. In this latter context, although dissimilarity is telling us that our method has not been stacked (and thus effective rewiring is happening), the randomization process does not produce any variation in topological terms.

The choice of topological constraints (i.e., the particular method) for a desired randomization process depends on the question the researcher wants to explore, rather than on a technical issue. Preserving the directed degree sequence captures the necessity to fix the number of inputs and outputs for every element. Randomizations attending to this constraint (for example, in a technological system) may be interpreted as a rewiring of an electronic circuit by a random assembling of integrated devices (e.g., chips) but preserving the inputs and outputs of the components. This contrasts with the softer undirected degree sequence invariant produced by preserving just the number of connections in every node. In this case, the relevance relies on the number of relations instead of the arrow orientation—i.e., the undirected degree sequence. Furthermore, the conservation of the *connected components* is essential in a graph describing a *process*, since fragmentation can be interpreted as a break of the flow of causality. Since causal flow can be preserved by method **c** in random ensembles, we are able to provide a random structure attending the input or output constraint of every element of a system. At this point, differences between the original and randomized graphs would provide a good insight about what is beyond the simple assembling of elements expected by simple combinatorics. Hence, the characterization of the topological differences between the original and its random counterparts would constitute the next step for the study of the organization of feed-forward causal structures. A further exploration in the hierarchical behavior of these networks may contribute to a better understanding of the causal relations present in many complex systems and processes.

Finally, we stress that an important feature of any randomization process is how or to what extent topological invariants restrict the space of graphical solutions. Our methodology provides valuable information about the randomness of a particular structure within the context of its graphical space of solutions. It is arguable to think that the higher the number of constraints the smaller the space of solutions. In any case, its complete exploration is not feasible beyond a graph containing more than a handful of nodes. In this context, our methodology provides a sampling of such space in order to estimate the randomness of a DAG given some topological constraints.

ACKNOWLEDGMENTS

This work was supported by the EU 6th Framework Project ComplexDis (Grant No. NEST-043241, C.R.-C. and J.G.), the UTE project CIMA (J.G.), the James McDonnell Foundation (B.C.-M. and R.V.S.), and the Santa Fe Institute (R.V.S.). We thank the Complex System Laboratory members for fruitful conversations.

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

[2] M. E. J. Newman, S. H. Strogatz, and D. J. Watts, Phys. Rev. E **64**, 026118 (2001).

[3] S. N. Dorogovtsev and J. F. F. Mendes, *Evolution of Networks: From Biological Nets to the Internet and WWW* (Oxford University Press, Oxford, 2003).

[4] R. V. Sole, R. Ferrer-Cancho, J. M. Montoya, and S. Valverde, Complexity **8**, 20 (2002).

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

[6] N. Hummon and P. Doreian, Soc. Networks **11**, 39 (1989).

[7] E. Garfield, I. H. Sher, and R. J. Torpie, Air Force Office of Scientific Research Technical Report No. F49(638)-1256 (The Institute for Scientific Information, Philadelphia, PA, 1964).
[8] S. Valverde, R. V. Sole, M. A. Bedau, and N. Packard, Phys. Rev. E **76**, 056118 (2007).
[9] G. Csardi, K. J. Strandburg, L. Zalanyi, J. Tobochnik, and P. Erdi, Physica A **374**, 783 (2007).
[10] J. H. Fowler and S. Jeon, Soc. Networks **30**, 16 (2008).
[11] S. J. Chandler, *The Network Structure of Supreme Court Jurispurdence*, University of Houston Law Center No. 2005–W-01 (2005), p. 2005-W-01, http://ssrn.com/abstract=742065.
[12] R. Clay, *Nonlinear Networks and Systems* (John Wiley & Sons Inc., New York, 1971).
[13] S. Haykin, *Neural Networks: A Comprehensive Foundation* (Prentice-Hall, London, 1999).
[14] H. Frank and I. T. Frisch, *Communication, Transmission and Transportation Networks* (Addison-Wesley, Reading, MA, 1971).
[15] *Random Graphs*, edited by W. Fulton, A. Katok, F. Kirwan, P. Sarnak, B. Simon, and B. Totaro (Cambridge University Press, Cambridge, England, 2001).
[16] M. Molloy and B. Reed, Random Struct. Algorithms **6**, 161 (1995).
[17] B. Karrer and M. E. J. Newman, Phys. Rev. Lett. **102**, 128701 (2009).
[18] W. Aiello, F. Chung, and L. Lu, *A random graph model for massive graphs*, Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, 2001 (unpublished), pp. 171–180.
[19] S. Maslov and K. Sneppen, Science **296**, 910 (2002).
[20] S. Hanhijarvi, G. Garriga, and K. Puolamaki, *Randomization Techniques for Graphs*, Proceedings of the 9th SIAM International Conference on Data Mining (2009), http://siam.org/proceeedings/datamining/2009/dm09.php.
[21] P. Suppes, *Axiomatic Set Theory* (Dover, New York, 1960).
[22] J. Kelley, *General Topology*, Graduate Texts in Mathematics Vol. 27 (Van Nostrand, New York, 1955), p. 1975.
[23] Such an equality is only general in DAGs since the absence of cycles avoids the existence of autoloops or situations like $\langle v_i, v_k \rangle, \langle v_k, v_i \rangle \in E$. Furthermore, within this formalism, it is assumed that we can neglect the probability that two arcs begin at a given node v_k and end in a given node v_i, due to the assumption of *sparseness*. As we shall see in Sec. IV, such an assumption does not hold for some real systems.
[24] We observe that we defined an ordering which is just the opposite of the one defined in [17]. The reason for this stems from the interesting role played by order theory to understand the particular properties of DAGs. In this way, in our ordering, a maximal will display a number smaller than any of its neighbors, and the opposite happens in the case of minimal, leading this definition of order to being more intuitive for the reader. It is clear, however, that any choice is equivalent, provided that the construction is internally consistent.
[25] C. Rodríguez-Caso, B. Corominas-Murtra, and R. V. Sole, Mol. Biosyst. **5**, 1617 (2009).
[26] C. E. Shannon, Bell Syst. Tech. J. **27**, 379 (1948).
[27] According to formalization, DAG numbering is used to identify the vertex of a graph, and implicitly we might construct $\mathcal{A}$ positioning vertices according to this numbering. However, by doing so, simple vertex reallocation in $\mathcal{A}$ may produce an increase in dissimilarity without an effective rewiring event in the graph. Then, the use of numbering can lead to an erroneous estimation of the process of randomization. To avoid this problem, the vertex position in $\mathcal{A}$ and the numbering must be considered as different things. Formally speaking, we might consider two indices for every vertex: one for the numbering (variable according to process of randomization) and another one for the position in the matrix. Then, the arc $\langle v_{i,m}, v_{j,n} \rangle$ is defined by $\mathcal{A}_{ij}=1$ position in $\mathcal{A}$ and m and n are the numbers obtained in the randomization method. For the sake of simplicity we avoid the use of this double notation saying that node position is invariant in the matrix.
[28] A measure quantifying *how random or how deterministic is a structure in relation to the space allowed by the topological invariants* was required. The degree-degree joint entropy of a graph holds this property. Other valuable measures, such as assortativity [31,32] or mutual information [33], have been pointed out. Assortativity measures degree-degree correlations and degree-degree mutual information quantifies the predictability of neighbors' degrees from the sole knowledge of the degree of a given node in relation to the *available* degree richness of the system. The former case strictly looks for linear relationships, and it is supposed to be a more appropriate measure for normally distributed data. Furthermore, both approaches naturally require certain degree-degree variance within the graph. For instance, a large feed-forward single chain of nodes has a strong degree-degree determinism that none of these two measurements would capture. The reason is that most of the degree-degree pairs would be (2,2) for undirected and (1,1) for any directed degree analyses. In this sense, degree-degree joint entropy provides a suitable measure of the relation or determinism of degree-degree relations with neither parametric assumptions nor degree-degree variance requisites. Accordingly, we used the concept of degree-degree relations instead of degree-degree correlations.
[29] Release WS202, http://www.wormbase.org
[30] V. Batagelj and A. Mrvar, Pajek Datasets, http://vlado.fmf.uni-lj.si/pub/networks/data/
[31] M. E. J. Newman, Phys. Rev. Lett. **89**, 208701 (2002).
[32] J. G. Foster, D. V. Foster, P. Grassberger, and M. Paczuski, Proc. Natl. Acad. Sci. U.S.A. **107**, 10815 (2010).
[33] R. V. Sole and S. Valverde, *Complex Networks*, Lecture Notes in Physics (Springer-Verlag, Berlin, 2004), pp. 189–210.