Unbiased sampling of network ensembles

Tiziano Squartini
Instituut-Lorentz for Theoretical Physics, University of Leiden,
Niels Bohrweg 2, 2333 CA Leiden (The Netherlands)
Institute for Complex Systems UOS Sapienza, "Sapienza" University of Rome, P.le Aldo Moro 5, 00185 Rome (Italy)

Rossana Mastrandrea
Institute of Economics and LEM, Scuola Superiore Sant’Anna, 56127 Pisa (Italy)
Aix Marseille Université, Université de Toulon, CNRS, CPT, UMR 7332, 13288 Marseille (France)

Diego Garlaschelli
Instituut-Lorentz for Theoretical Physics, University of Leiden,
Niels Bohrweg 2, 2333 CA Leiden (The Netherlands)
(Dated: June 12, 2014)

Sampling random graphs with given properties is a key step in the analysis of networks, as random ensembles represent basic null models required to identify patterns such as communities and motifs. A key requirement is that the sampling process is unbiased and efficient. The main approaches are microcanonical, i.e. they sample graphs that exactly match the enforced constraints. Unfortunately, when applied to strongly heterogeneous networks (including most real-world graphs), the majority of these approaches become biased and/or time-consuming. Moreover, the algorithms defined in the simplest cases (such as binary graphs with given degrees) are not easily generalizable to more complicated ensembles. Here we propose a solution to the problem via the introduction of a ‘maximize-and-sample’ (‘Max & Sam’) method to correctly sample ensembles of networks where the constraints are ‘soft’ i.e. they are realized as ensemble averages. Being based on exact maximum-entropy distributions, our approach is unbiased by construction, even for strongly heterogeneous networks. It is also more computationally efficient than most microcanonical alternatives. Finally, it works for both binary and weighted networks with a variety of constraints, including combined degree-strengths sequences and full reciprocity structure, for which no alternative method exists. Our method can also be turned into a microcanonical one, via a restriction to the relevant subset. We show various applications to real-world networks and provide a code implementing all our algorithms.

PACS numbers: 05.10.-a,89.75.Hc,02.10.Ox,02.70.Rr

Network theory is systematically used to address problems of scientific and societal relevance [1], from the prediction of the spreading of infectious diseases worldwide [2] to the identification of early-warning signals of upcoming financial crises [3]. More in general, several dynamical and stochastic processes are strongly affected by the topology of the underlying network [4]. This results in the need to identify the topological properties that are statistically significant in a real network, i.e. to discriminate which higher-order properties can be directly traced back to the local features of nodes, and which are instead due to more complicated factors.

To achieve this goal, one requires (a family of) randomized benchmarks, i.e. ensembles of networks where the local heterogeneity is the same as in the real one, and the topology is random in any other respect: this defines a null model of the original network. Nontrivial patterns can then be detected in the form of empirical deviations from the null model’s theoretical expectations [5]. Important examples of such patterns is the presence of motifs (recurring subgraphs of small size, like building blocks of a network [6]) and communities (groups of nodes that are more densely connected internally than with each other [7]). To detect these and many other patterns, one needs to correctly specify the null model and then calculate e.g. the average and standard deviation (or confidence interval) of any topological property of interest over the corresponding randomized ensemble of graphs.

Unfortunately, given the strong heterogeneity of nodes (e.g. the power-law distribution of vertex degrees), the solution to the above problem is not simple. This is most easily explained in the case of binary graphs, even if similar arguments apply to weighted networks as well. For simple graphs, the most important null model is the (Undirected Binary) Configuration Model (UBCM), defined as an ensemble of networks where the degree of each node is specified, and the rest of the topology is maximally random [8]. Since the degrees of all nodes (the so-called degree sequence) act as constraints, ‘maximally random’ does not mean ‘completely random’: in order to realize the degree sequence, interdependencies among vertices necessarily arise. These interdependencies affect other topological properties as well. So, even if the degree sequence is the only quantity that is enforced ‘on purpose’, other structural properties are unavoidably constrained as well. These higher-order effects are called ‘structural correlations’. In order to disentangle spurious structural correlations from genuine correlations of interest, it is very important to properly implement the
UBCM in such a way that it takes the observed degree sequence as input and generates expectations based on a uniform and efficient sampling of the ensemble. Similar and more challenging considerations apply to more complicated null models, defined e.g. for directed or weighted graphs and specified by more general constraints.

Several approaches have been proposed and can be roughly divided in two large classes: microcanonical and canonical methods. Microcanonical approaches [9,15] aim at artificially generating many randomized variants of the observed network in such a way that the constrained properties are identical to the empirical ones, thus creating a collection of graphs sampling the desired ensemble. In these algorithms the enforced constraints are ‘hard’, i.e. they are met exactly by each graph in the resulting ensemble. This strong requirement implies that most microcanonical approaches proposed so far suffer from various problems, including bias, lack of ergodicity, mathematical intractability, high computational demands, and poor generalizability.

On the other hand, in canonical approaches [5,16–26] the constraints are ‘soft’, i.e. they can be violated by individual graphs in the ensemble, even if the ensemble average of each constraint must match the enforced value exactly. Canonical approaches are generally introduced to directly obtain, as a function of the observed constraints (e.g. the degree sequence), exact mathematical expressions for the expected topological properties, thus avoiding the explicit generation of randomized networks [5]. However, this is only possible if the mathematical expressions for the topological properties of interest are simple enough to make the analytical calculation of the expected values feasible. In all other cases, one is led again to the problem of sampling many network configurations explicitly and average the properties of interest over such configurations.

In this paper, we show that canonical approaches can indeed be used also for the purpose of computationally sampling several ensembles of diverse types of networks (i.e. directed, undirected, weighted, binary) with various constraints (degree sequence, strength sequence, reciprocity structure, mixed binary and weighted properties, etc.). This computational use of canonical null models has not been implemented systematically so far, because the most popular approaches rely on highly approximated expressions leading to ill-defined or unknown probabilities that cannot be used to sample the ensemble. These approximations are in any case available only for the simplest ensembles (e.g. the UBCM), leaving the problem unsolved for more general constraints. By contrast, here we make use of a series of recent analytical results that generate the exact probabilities for a wide range of constraints of interest [5,19,27]. We show that, unlike microcanonical approaches, our canonical ones allow for a unified treatment of several different ensembles, fulfill all the standard requirements of unbiasedness and uniformity, and are computationally efficient. We consider various examples that show the power of our method when applied to real-world networks. We also discuss conditions under which our approach can be used microcanonically, via a restriction to the subset of realizations that match the constraints sharply.

I. PREVIOUS APPROACHES

In this section, we briefly discuss the main available approaches to the problem of sampling network ensembles with given constraints, and highlight the main obstacles and limitations. We consider both microcanonical and canonical methods. In both cases, since the UBCM is the most popular and most studied ensemble, we will discuss the problem by focusing mainly on the implementations of this model. The same kind of considerations extend to other constraints (e.g. strengths, reciprocity) and other types of networks (e.g. directed and/or weighted) as well.

A. Microcanonical methods

There have been several attempts to develop microcanonical algorithms that efficiently implement the UBCM. One of the earliest algorithm starts with an empty network having the same number of vertices of the original one, where each vertex is assigned a number of ‘half edges’ (or ‘edge stubs’) equal to its degree in the real network. Then, pairs of stubs are randomly matched, thus creating the final edges of a random network with the desired degree sequence [5]. Unfortunately, for most empirical networks, the heterogeneity of the degrees is such that this algorithm produces several multiple edges between vertices with large degree, and several self-loops [9]. If the formation of these undesired edges is forbidden explicitly, the algorithm gets stuck in configurations where edge stubs have no more eligible partners, thus failing to complete any randomized network.

To overcome this limitation, a different algorithm (which is still widely used) was introduced [9]. This ‘Local Rewiring Algorithm’ (LRA) starts from the original network, rather than from scratch, and randomizes its topology through the iteration of an elementary move that preserves the degrees of all nodes. While this algorithm always produces random networks, it is extremely time consuming since many iterations of the fundamental move are needed in order to produce just one randomized variant, and this must be repeated several times (the mixing time is still unknown [28]) in order to produce many variants.

Besides these practical limitations, the main problem of the LRA is the fact that it is biased, i.e. it does not sample the desired ensemble uniformly. This has been rigorously shown relatively recently [10,12]. For undirected networks, uniformity has been shown to hold, at least approximately, only when the degree sequence is
where $k_{\text{max}}$ is the largest degree in the network, $\overline{k}$ is the average degree, $\overline{k}^2$ is the second moment, and $N$ is the number of vertices. For directed networks, a similar condition must hold \[11\]. Clearly, the above condition sets an upper bound for the heterogeneity of the degrees of vertices, and is violated if the heterogeneity is strong. This is a first indication that the available methods break down for ‘strongly heterogeneous’ networks. As we discuss later, most real-world networks are known to fall precisely within this class.

For directed networks, where links are oriented and the constraints to be met are the numbers of incoming and outgoing links (in-degree and out-degree) separately, a condition similar to eq. \[1\] holds, but there is also the additional problem that the LRA is non-ergodic, i.e. it is in general not able to explore the entire ensemble of networks \[11\]. The violation of uniformity and ergodicity in the LRA implies that the average quantities over the graphs it generates are biased, i.e. they do not correspond to the correct expectations. It has been shown that, in order to restore ergodicity, it is enough to introduce an additional ‘triangular move’ inverting the direction of closed loops of three vertices \[11\]. However, in order to restore uniformity, one must do something much more complicated: at each iteration, the attempted ‘rewiring move’ must be accepted with a probability that depends on some complicated property of the current network configuration \[10–12\]. Since this property must be recalculated at each step, the resulting algorithm is extremely time consuming.

Other recent alternatives \[13\]–\[19\] rely on theorems, such as the Erdős-Gallai \[29\] one, that set necessary and sufficient conditions for a degree sequence to be graphic, i.e. realized by at least one graph. These ‘graphic’ methods exploit such (or related) conditions to define biased sampling algorithms in conjunction with the estimation of the corresponding sampling probabilities, thus allowing one to statistically reweight the outcome and sample the ensemble effectively uniformly \[13–15\]. Del Genio et al. \[13\] show that, for networks with power-law degree distribution of the form $P(k) \sim k^{-\gamma}$, the computational complexity of sampling just one graph using their algorithm is $O(N^2)$ if $\gamma > 3$. However, when $\gamma < 3$ the computational complexity increases to $O(N^{2.5})$ if
\[
k_{\text{max}} < \sqrt{N}
\]
and to $O(N^3)$ if $k_{\text{max}} > \sqrt{N}$. The upper bound $\sqrt{N}$ is a particular case of the so-called ‘structural cut-off’ that we will discuss in more detail later. For the moment, it is enough for us to note that eq. \[2\] is another indication that, for strongly heterogeneous networks, the problem of sampling gets more complicated. As we will discuss later, most real networks violate eq. \[2\] strongly.

So, while ‘graphic’ algorithms do provide a solution for every network, their complexity increases for networks of increasing (and realistic) heterogeneity. A more fundamental limitation is that they can only handle the problem of binary graphs with given degree sequence. The generalization of these methods to other types of networks and other constraints is not straightforward, as it would require the proof of more general ‘graphicity’ theorems, and ad hoc modifications of the algorithm.

From this discussion of microcanonical algorithms, it should be clear that it is not obvious how they should be extended to the case of weighted networks, where the constraint relevant to each vertex $i$ is the so-called strength $s_i$ (the sum of weights of the links reaching that vertex), and possibly also the purely binary degree $k_i$ \[23\]–\[26\]. Similarly, it is unclear how the available microcanonical algorithms should be generalized to accommodate additional constraints like the reciprocity structure in both binary \[22\] and weighted \[24\] directed networks. In all these more complicated ensembles, one should solve the problem of bias to ensure uniform sampling protocols.

B. Canonical methods

Canonical approaches aim at obtaining, as a function of the observed constraints (e.g. the degree sequence), mathematical expressions for the expected topological properties, avoiding the explicit generation of randomized networks. For canonical methods the requirement of uniformity is replaced by the requirement that the probability distribution over the enlarged ensemble has maximum entropy \[5\]–\[10\].

For binary graphs, since any topological property is a function of the adjacency matrix $A$ of the network (with entries $a_{ij} = 1$ if the vertices $i$ and $j$ are connected, and $a_{ij} = 0$ otherwise), the ultimate goal is finding a mathematical expression for the expected value of $a_{ij}$, which is nothing but the probability $p_{ij}$ that the vertices $i$ and $j$ are connected in the ensemble. Note that, since in the microcanonical ensemble all links are dependent on each other (the degree sequence must be reproduced exactly in each realization), finding analytical expressions for $p_{ij}$ in the microcanonical case is unfeasible. However, it turns out that in the canonical ensemble, for most constraints of interest, all pairs of vertices become independent, making the quest for $p_{ij}$ possible.

For binary undirected networks, the most popular specification for $p_{ij}$ is the factorized one \[11\]–\[30\]–\[31\]
\[
p_{ij} = \frac{k_i k_j}{k_{\text{tot}}}
\]
where $k_i$ is the degree of node $i$ and $k_{\text{tot}}$ is the total degree over all nodes), while for weighted undirected networks, where each link can have a non-negative weight $w_{ij}$ and each vertex $i$ is characterized by a given strength $s_i$, the corresponding assumption is that the expected
weight of the link connecting the vertices $i$ and $j$ is

$$\langle w_{ij} \rangle = \frac{s_is_j}{s_{tot}}$$

(4)

(where $s_{tot}$ is the total strength of all vertices).

Equations (3) and (4) are routinely used, and have become standard textbook expressions [1]. The most intense use of these expressions is perhaps encountered in the empirical analysis of communities, i.e., relatively denser modules of vertices in large networks [2]. Most community detection algorithms compare different partitions of vertices into communities (each partition being parametrized by a matrix $C$ such that $c_{ij} = 1$ if the vertices $i$ and $j$ belong to the same community, and $c_{ij} = 0$ otherwise) and look for the ‘optimal’ partition. The latter is the one that maximizes the modularity function which, for binary networks, is defined as

$$Q(C) \equiv \frac{1}{k_{tot}} \sum_{i,j} \left[ a_{ij} - \frac{k_i k_j}{k_{tot}} \right] c_{ij}$$

(5)

where eq. (3) appears explicitly as a null model for $a_{ij}$. For weighted networks, a similar expression involving eq. (4) applies. Other important examples where eq. (3) is used are the characterization of the connected components of networks [31], the average distance among vertices [30], and more in general the theoretical study of percolation [1] (characterizing the system’s robustness under the failure of nodes and/or links) and other dynamical processes [4] on networks.

Due to the important role they play in many applications, it is remarkable that the literature puts little emphasis on the fact that eqs. (3) and (4) are valid only under strict conditions that, for most real networks, are strongly violated. It is evident that eq. (3) represents a probability only if the largest degree $k_{max}$ in the network does not exceed the ‘structural cut-off’ $k_c \equiv \sqrt{k_{tot}}$, i.e. if

$$k_{max} < \sqrt{k_{tot}}$$

(6)

Obviously, the above condition sets an upper bound for the allowed heterogeneity of the degrees, since both $k_{max}$ and $k_{tot}$ are determined by the same degree distribution. If $k_{max}$ exceeds $k_c$, eq. (3) breaks down.

In principle, the knowledge of $p_{ij}$ allows one to sample networks from the canonical ensemble, by running over all pairs of nodes and connecting them with the appropriate probability (we will use this method later on). Unfortunately, the above discussion shows that, for strongly heterogeneous degree sequences, eq. (3) would produce ill-defined values of connection probabilities. This is why, so far, general algorithms to sample canonical, rather than microcanonical, ensembles of networks with constraints have not been implemented.

C. The ‘strong heterogeneity regime’ challenging most algorithms

Equations (1), (2) and (3), along with our discussion above, show that most methods run into problems when the heterogeneity of the network is strong enough. Strongly heterogeneous networks elude most microcanonical and canonical approaches proposed so far. As is well known, networks in the ‘strongly heterogeneity’ regime turn out to be ubiquitous, and represent the rule rather than the exception. A simple way to prove this is by directly checking whether the largest degree exceeds the structural cut-off $k_c$. As Maslov et al. first noticed [9], in real networks $k_c$ is strongly and systematically exceeded: for instance, for the Internet $k_{max} = 1458$ and $k_c \approx 159$, which means that the structural cut-off is exceeded tenfold! Consequently, if eq. (3) were applied to the two vertices with largest degree, the resulting connection ‘probability’ would be $p_{ij} = 43.5$, i.e. more than 40 times larger than any reasonable estimate for a probability! We also note that, when inserted into eq. (5), this value of $p_{ij}$ would produce, in the summation, a single term 40 times larger than any other ‘regular’ term (of order unity), thus significantly biasing the community detection problem. To the best of our knowledge, a study of the entity of this bias has never been performed.

The Internet is not a special case, and similar figures are found in the majority of real networks, making the problem entirely general. To see this, we recall that most real networks have a power-law degree distribution of the form $P(k) \sim k^{-\gamma}$ with exponent in the range $2 < \gamma < 3$. For these networks, the average degree $K = k_{tot}/N$ is finite but the second moment $K^2$ diverges. Therefore the structural cut-off scales as $k_c \sim N^{\frac{1}{2}}$ [22], which means that eqs. (2) and (3) coincide. By contrast, Extreme Value Theory shows that the largest degree scales as $k_{max} \sim N^{\frac{1}{(\gamma-1)}}$ [22]. This implies that the ratio $k_{max}/k_c$ diverges for large networks, i.e. the largest degree is infinitely larger than the allowed cut-off value. Unfortunately, many results and approaches that have been obtained by assuming $k_{max} < k_c$ are routinely extended to real networks where, in most of the cases, $k_{max} \gg k_c$. Therefore, although this might appear as an exaggerated claim, most analyses of real-world networks (including community detection) that have been carried out so far have relied on incorrect expressions, and have been systematically affected by an uncontrolled bias.

In theoretical and computational models of networks, the problem is routinely circumvented by enforcing the condition $k_{max} < k_c$ explicitly, e.g. by considering a truncated power-law distribution, using the justification that this condition should hold for ‘sparse’ networks where the average degree does not grow with $N$, as in most real networks [8] [33]. This is however misleading, since in real scale-free networks with $2 < \gamma < 3$ the average degree is finite, and this makes them sparse even without assuming a truncation in the degree distribution. Indeed, as in the example above, real networks systemat-
ically violate the cut-off value, and are therefore ‘strongly heterogeneous’. By the way, sparseness itself is not crucial, since dense but homogeneous networks (including the densest of all, i.e. the complete graph) are such that $k_{\text{max}} < k$, and are correctly described by eq. (3), just as sparse homogeneous networks. This confirms that the problem is in fact due to strong heterogeneity.

The above arguments extend also to other ensembles of networks with different constraints. The general conclusion is that, since real-world networks are generally strongly heterogeneous, the available approaches either break down or become computationally demanding. Moreover, it is difficult to generalize the available knowledge to modified constraints and different types of graphs.

II. THE ‘MAX & SAM’ METHOD

In what follows, we exploit a series of recent results [5][22][24][26] characterizing several canonical ensembles of networks and define a unified approach to sample these ensembles in a fast, unbiased and efficient way. In our approach, the functional form of the probability of each graph in the ensemble is derived by maximizing Shannon’s entropy [16], and the numerical coefficients of this probability are derived by maximizing the probability (i.e. the likelihood) itself [5]. A similar approach was already introduced [5] in order to derive analytical expectations for some topological quantities of interest, but that approach entirely avoids the problem of sampling. Here, we instead implement a sampling protocol explicitly, so that the expected value of any desired topological property (not only those that can be calculated analytically) can be obtained as an average over the sampled networks. We provide unified practical recipes to sample random networks efficiently, and from an expanded set of ensembles that have been treated only separately so far [5][19][22][24][27]. We also provide a code implementing all our sampling algorithms (see Appendix). Since our method is based on the idea that network ensembles can be efficiently sampled stochastically after a double maximization (first of Shannon’s entropy, and then of the likelihood), we call it the ‘Maximize and Sample’ (for short, ‘Max & Sam’) method.

In this section we consider the canonical implementation of our method, while in the next one we briefly discuss the microcanonical implementation. We will consider canonical ensembles of binary graphs with given degree sequence (both undirected [5][19] and directed [5][19][21]), of weighted networks with given strength sequence (both undirected [5][20] and directed [5][20][21][24]), of directed networks with given reciprocity structure (both binary [22][23] and weighted [24]), and of weighted networks with given strength sequence and degree sequence [25][27]. In all these cases, one can show that the probability of the entire network factorizes as a product of dyadic probabilities over pairs of nodes. This ensures that the computational complexity is always $O(N^2)$ in all cases considered here, irrespective of the level of heterogeneity of the real-world network being randomized (thus overcoming the ubiquitous problems discussed in sec. III). Since all the probability distributions involved in our method are obtained via an explicit maximization of Shannon’s entropy, the resulting sampling algorithms are unbiased by construction. This implies that our methods do not suffer from the limitations of the other methods discussed in sec. I, and are efficient and unbiased even for strongly heterogeneous networks.

A. Binary undirected graphs with given degree sequence

Let us start by considering binary, undirected networks (BUNs). A generic BUN is uniquely specified by its binary adjacency matrix $A$. The particular matrix corresponding to the observed graph that we want to randomize will be denoted by $A^\ast$. As we mentioned, the simplest non-trivial constraint is the degree sequence, $(k_i)_{i=1}^N$ (where $k_i \equiv \sum_j a_{ij}$ is the degree of node $i$), defining the Undirected Binary Configuration Model (UBCM). Throughout sec. II we have already discussed the crucial importance of the UBCM for the analysis of real-world networks (e.g. for the detection of communities [1]), and we have highlighted the limitations of the available implementations.

In our approach, the canonical ensemble of BUNs is the set of networks with the same number of nodes, $N$, of the observed graph and a number of (undirected) links varying from zero to the maximum value $\frac{N(N-1)}{2}$. Appropriate probability distributions on this ensemble can be fully determined by maximizing, in sequence, Shannon’s entropy (under the chosen constraints) and the likelihood function, as already pointed out in [5]. The result of the entropy maximization [5][16] is that the graph probability factorizes as

$$P(A|\vec{x}) = \prod_{i<j} p_{ij}^{x_{ij}} (1 - p_{ij})^{1-x_{ij}}$$

where $p_{ij} \equiv \frac{x_{ij}}{1+x_{ij}}$. The vector $\vec{x}$ of $N$ unknown parameters is to be determined either by maximizing the log-likelihood function

$$\lambda(\vec{x}) \equiv \ln P(A^\ast|\vec{x}) = \sum_i k_i(A^\ast) \ln x_i - \sum_{i<j} \ln(1 + x_i x_j)$$

or, equivalently, by solving the following system of $N$ equations (corresponding to the requirement that the gradient of the log-likelihood vanishes) [5]:

$$\langle k_i \rangle = \sum_{i \neq j} \frac{x_i x_j}{1 + x_i x_j} = k_i(A^\ast) \quad \forall i$$

where $k_i(A^\ast)$ is the observed degree of vertex $i$ and $\langle k_i \rangle$ indicates its ensemble average. From eq. (9) it is evident...
that only the observed values of the chosen constraints (the sufficient statistics of the problem) are needed in order to obtain the numerical values of the unknowns (the empirical degree sequence fixes the value of \( \vec{x} \), which in turn fix the value of all the \( \{p_{ij}\} \)). In any case, for the sake of clarity, in the code we allow the user to choose the preferred input-form (a matrix, a list of edges, a vector of constraints). This is true for all the models described here and implemented in the routine.

Note that the above form of \( p_{ij} \) represents the exact expression that should be used in place of eq.3. This reveals the highly non-linear and non-local character of the interdependencies among vertices in the UBCM: in random networks with given degree sequence, the correct connection probability \( p_{ij} \) is a function of the degrees of all vertices of the network, and not just of the end-point degrees as in eq.3. Only when the degrees are ‘weakly heterogeneous’ (mathematically, this happens when \( x_i x_j \ll 1 \) for all pairs of vertices, which implies \( p_{ij} \approx x_i x_j \)), these structural interdependencies become approximately local. Note that, in the literature, this is improperly called the ‘sparse graph’ limit [16], while from our previous discussion in sec.1C it is clear that low heterogeneity, and not sparsity, is required to produce this limit.

Unlike eq.3, the \( p_{ij} \) considered here always represents a proper probability ranging between 0 and 1, irrespective of the heterogeneity of the network. This implies that eq.7 provides us with a recipe to sample the canonical ensemble of BUNs under the UBCM. After the unknown parameters have been found, they can be put back into eq.7 to obtain the probability to correctly sample any graph \( A \) from the ensemble. The key simplification allowing this in practice is the fact that the graph probability is factorized, so that a single graph can be sampled stochastically by sequentially running over each pair of nodes \( i, j \) and implementing a Bernoulli trial (whose elementary events are \( a_{ij} = 0 \), with probability \( 1 - p_{ij} \), and \( a_{ij} = 1 \), with probability \( p_{ij} \)). This process can be repeated to generate as many configurations as desired. Note that sampling each network has complexity \( O(N^2) \), and that the time required to preliminarily solve the system of coupled equations to find the unknown parameters \( \vec{x} \) is independent on how many random networks are sampled and on the heterogeneity of the network. Thus this algorithm is always more efficient than the corresponding microcanonical ones described in sec.1A.

In fig.1 we show an example of this procedure on the network of liquidity reserves exchanges between Italian banks in 1999 [35]. For an increasing number of sampled graphs, we show the convergence of the sample average \( \bar{a}_{ij} \) of each entry of the adjacency matrix to its exact canonical expectation \( \langle a_{ij} \rangle \), analytically determined after solving the likelihood equations. This preliminary check is useful to establish that, in this case, generating 1000 matrices (bottom right) is enough to reach a high level of accuracy. If needed, the accuracy can be quantified rigorously (e.g. in terms of the maximum width around the identity line) and arbitrarily improved by increasing the number of sampled matrices. Note that this important check is impossible in microcanonical approaches, where the exact value of the ‘target’ probability is unknown.

We then select the sample of 1000 matrices and confirm (see the top panel of fig.2) that the imposed constraints (the observed degrees of all nodes) are excellently reproduced by the sample average, and that the confidence intervals are narrowly spread around the identity line. This is a crucial test of the accuracy of our sampling procedure. Again, the accuracy can be improved by increasing the number of sampled matrices if needed.

After this preliminary check, the sample can be used to compare the expected and observed values of higher-order properties of the network. Note that in this case we do not require (or expect) that these (unconstrained) higher-order properties are correctly reproduced by the null model. The entity of the deviations of the real network from the null model depends on the particular example considered, and the characterization of these deviations is precisely the reason why a method to sample random networks from the appropriate ensemble is needed in the first place. In the bottom panels of fig.2 we compare the observed value of two quantities of interest with their arithmetic mean over the sample. The two quantities are the average nearest neighbors degree (ANND), \( k_{nn}^n = \sum_j a_{ij} k_j / k_i \), and the clustering coefficient, \( c_i = \sum_{j,k} a_{ij} a_{jk} a_{kj} / k_i (k_i - 1) \) of each vertex.

Note that since our sampling method is unbiased, the sample mean automatically weighs the configurations according to their correct probability. In this particular case, we find that the null model reproduces the observed network remarkably well, which means that the degree sequence effectively ‘explains’ (or rather ‘generates’) the two empirical higher-order patterns that we have considered. This is consistent with other studies [5, 19, 20], but not true in general for other networks or other constraints, as we show later on. From the bottom panels of fig.2 we also note that the confidence intervals highlight a non-obvious feature: the fact that the (apparently) ‘strongest outliers’ (those further away from the identity line) turn out to be actually within (or at the border of) the chosen confidence intervals, while several (apparently) ‘weak outliers’ (closer to the identity) are instead found to be much more distant from the confidence intervals, and thus in an unexpectedly stronger disagreement with the null model. These counter-intuitive insights cannot be gained from the analysis of the expected values alone, e.g. using expressions like eq.3 or similar.

B. Binary directed graphs with given in-degree and out-degree sequences

For binary directed networks (BDNs), the adjacency matrix \( A \) is (in general) not symmetric, and each node
FIG. 1. Sampling binary undirected networks with given degree sequence (Undirected Binary Configuration Model). The example shown is the binary network of liquidity reserves exchanges between Italian banks in 1999 [35] (N = 215). The four panels show the convergence of the sample average $\langle a_{ij} \rangle$ of each entry of the adjacency matrix to its exact canonical expectation $\langle a_{ij} \rangle$, for 1 (top left), 10 (top right), 100 (bottom left) and 1000 (bottom right) sampled matrices. The identity line is shown in red.

$i$ is characterized by two degrees: the out-degree $k_{i}^{\text{out}} \equiv \sum_j a_{ij}$ and the in-degree $k_{i}^{\text{in}} \equiv \sum_j a_{ji}$. The Directed Binary Configuration Model (DBCM), the directed version of the UBCM, is defined as the ensemble of BDNs with given out-degree sequence $\{k_{i}^{\text{out}}\}_{i=1}^{N}$ and in-degree sequence $\{k_{i}^{\text{in}}\}_{i=1}^{N}$. The DBCM is widely used in order to detect communities [7] and other higher-order patterns [1] in directed binary networks. However, most approaches [1, 7] make use of a directed version of eq. (3) which, just like eq. (3) itself, is a poor approximation (especially for strongly heterogeneous networks, see sec. I C) to the correct expression that corresponds to an unbiased sampling of the ensemble [5]. As we now show, our ‘Max & Sam’ method automatically retrieves this expression, along with the full probability distribution that is unspecified by eq. (3) alone.

At a canonical level, the DBCM is defined on the ensemble of all BDNs with $N$ vertices and a number of links ranging from 0 to $N(N-1)$. Equation (7) still applies, but now with ‘$i < j$’ replaced by ‘$i \neq j$’ and $p_{ij} = \frac{x_{i}y_{j}}{1+x_{i}y_{j}}$, where the $2N$ parameters $\bar{x}$ and $\bar{y}$ are determined by either maximizing the log-likelihood function [5]

$$\lambda(\bar{x}, \bar{y}) \equiv \ln P(A^{*}|\bar{x}, \bar{y}) =$$

$$= \sum_i [k_i^{\text{out}}(A^{*}) \ln x_i + k_i^{\text{in}}(A^{*}) \ln y_i]$$

$$- \sum_{i \neq j} \ln(1 + x_iy_j)$$

(where $A^{*}$ is the real network) or, equivalently, by solving the corresponding system of $2N$ equations [5]:

$$\langle k_i^{\text{out}} \rangle = \sum_{i \neq j} \frac{x_iy_j}{1+x_iy_j} = k_i^{\text{out}}(A^{*}) \quad \forall i$$

$$\langle k_i^{\text{in}} \rangle = \sum_{i \neq j} \frac{x_jy_i}{1+x_jy_i} = k_i^{\text{in}}(A^{*}) \quad \forall i$$

This time, the ensemble can be efficiently sampled by considering each pair of vertices twice, and using (say) $p_{ij}$ and $p_{ji}$ to draw directed links in the two directions (these two events being statistically independent). Since this is a straightforward extension of the UBCM, we do not consider any specific example to illustrate the DBCM. However, the related algorithm has been implemented in the code (see Appendix).
C. Binary directed graphs with given degree sequences and reciprocity structure

A more constrained null model, the Reciprocal Binary Configuration Model (RBCM), can be defined for BDNs by enforcing, in addition to the two directed degree sequences considered above, the whole local reciprocity structure of the network \[5, 22, 23\]. Equivalently, this amounts to specify the three observed degree sequences defined as the vector of the numbers of non-reciprocated outgoing links, \( \{k_i^+\}_{i=1}^N \), the vector of the numbers of non-reciprocated incoming links, \( \{k_i^-\}_{i=1}^N \), and the vector of the numbers of reciprocated links, \( \{k_i^{\leftrightarrow}\}_{i=1}^N \). These numbers are defined as \( k_i^+ = \sum_j a_{ij}(1 - a_{ji}) \), \( k_i^- = \sum_j a_{ji}(1 - a_{ij}) \), and \( k_i^{\leftrightarrow} = \sum_j a_{ij}a_{ji} \) respectively \[22, 23\].

The RBCM is of crucial importance when analysing higher-order patterns that exist beyond the dyadic level in directed networks. The most important example is that of triadic motifs \[3, 6, 23\], i.e. patterns of connectivity (involving triples of nodes) that are statistically over- or under-represented with respect to a null model where the observed degree sequences and reciprocity structure are preserved (i.e. the RBCM). Note that in this case no approximate canonical expression similar to eq. (3) exists, therefore the null model is usually implemented micro-canonical using a generalization of the LRA that we have discussed in sec. \[4\]. Conceptually, this procedure suffers from the same problem of bias as the simpler procedures used to implement the UBCM and the DBCM through the LRA \[10–12\]. To our knowledge, in this case no correction analogous to that proposed in ref. \[11\] has been developed in order to restore uniformity.

In our ‘Max & Sam’ approach, we exploit known analytical results \[5, 22, 23\] showing that the probability of each graph \( A \) in the RBCM is

\[
P(A | \vec{x}, \vec{y}, \vec{z}) = \prod_{i<j} (p_{ij}^x)^{a_{ij}} (p_{ij}^y)^{a_{ij}} (p_{ij}^z)^{a_{ij}} (p_{ij}^w)^{a_{ij}}, \quad (13)
\]

where \( p_{ij}^x = \frac{x_iz_j}{x_iz_j + y_iy_j + z_iz_j + x_jz_j} \), \( p_{ij}^y = \frac{1}{x_iz_j + y_iy_j + x_jz_j + z_iz_j} \), \( p_{ij}^z = \frac{1}{x_iz_j + y_iy_j + x_jz_j + z_iz_j} \), and \( p_{ij}^w = \frac{1}{x_iz_j + y_iy_j + x_jz_j + z_iz_j} \), denote the probabilities of a single (non-reciprocated) link from \( i \) to \( j \), a single (non-reciprocated) link from \( j \) to \( i \), a double (reciprocated) link between \( i \) and \( j \), and no link at all respectively. The above four possible events are mutually exclusive. The greatest difference with respect to the DBCM lies in the fact that the two links that can be drawn between the same two nodes are no longer independent.
The 3N unknown parameters, \( \bar{x}, \bar{y}, \bar{z} \) must be determined by either maximizing the log-likelihood \[5\]

\[
\lambda(\bar{x}, \bar{y}, \bar{z}) = \ln P(\mathbf{A}^*|\bar{x}, \bar{y}, \bar{z}) = \sum_i \left[ k_i^{-}(\mathbf{A}^*) \ln x_i + k_i^{+}(\mathbf{A}^*) \ln y_i + k_i^{\leftrightarrow}(\mathbf{A}^*) \ln z_i \right] - \sum_{i<j} \ln (1 + x_i y_j + y_i x_j + z_i z_j)
\]

or, equivalently, solving the corresponding system of 3N equations \[5, 23, 23\]:

\[
\begin{align*}
\langle k_i^{-}\rangle &= \sum_{i \neq j} \frac{x_i y_j}{1 + x_i y_j + y_i x_j + z_i z_j} = k_i^{-}(\mathbf{A}^*) \quad \forall i \quad (15) \\
\langle k_i^{+}\rangle &= \sum_{i \neq j} \frac{y_i x_j}{1 + x_i y_j + y_i x_j + z_i z_j} = k_i^{+}(\mathbf{A}^*) \quad \forall i \quad (16) \\
\langle k_i^{\leftrightarrow}\rangle &= \sum_{i \neq j} \frac{z_i z_j}{1 + x_i y_j + y_i x_j + z_i z_j} = k_i^{\leftrightarrow}(\mathbf{A}^*) \quad \forall i \quad (17)
\end{align*}
\]

After the unknown parameters have been found, the four probabilities allow us to sample the ensemble correctly and very easily. In particular, we can consider each pair of vertices \( i, j \) only once and either: draw a single link directed from \( i \) to \( j \) with probability \( p_{ij}^{+} \), draw a single link directed from \( j \) to \( i \) with probability \( p_{ji}^{+} \), draw two oppositely oriented links with probability \( p_{ij}^{\leftrightarrow} \), or draw no link at all with probability \( 1 - p_{ij}^{+} \). Note that, despite the increased number of constraints, the computational complexity is still \( O(N^2) \). As for the DBCM, we do not show a specific illustration of the RBCM, but the procedure described above has been fully coded in order to sample the relevant ensemble in a fast and unbiased way (see Appendix).

D. Weighted undirected networks with given strength sequence

Let us now consider weighted undirected networks (WUNs). Differently from the binary case, link weights can now range from zero to infinity by (without loss of generality) integer steps. The number of configurations in the canonical ensemble is therefore infinite. Still, enforcing node-specific constraints implies that a proper probability measure can be defined over the ensemble, such that the average value of any network property of interest is finite \[5\]. A single graph in the ensemble is now specified by its (symmetric) weight matrix \( \mathbf{W} \), where the entry \( w_{ij} \) represents the integer weight of the link connecting nodes \( i \) and \( j \) (\( w_{ij} = 0 \) means that no link is there). We denote the particular real-world weighted network as \( \mathbf{W}^* \). Each vertex is characterized by its strength \( s_i = \sum_j w_{ij} \) representing the weighted analogue of the degree.

The weighted, undirected counterpart of the UBCM is the Undirected Weighted Configuration Model (UWCM). The constraint defining it is the observed strength sequence, \( \{s_i\}_{i=1}^N \). Like its binary analogue, the UBCM is widely used in order to detect communities and other higher-order patterns in undirected weighted networks. However, most approaches \[\overline{1}\] incorrectly assume that this model is characterized by eq. \[\overline{1}\], which is instead only a highly simplified expression \[\overline{3}\]. Again, our ‘Max & Sam’ method automatically retrieves the correct expression, corresponding to an unbiased sampling of the ensemble.

The probability of each weighted network \( \mathbf{W} \) in the canonical ensemble is \[5\]

\[
P(\mathbf{W}|\bar{x}) = \prod_{i<j} p_{ij}^{w}(1 - p_{ij})
\]

where now \( p_{ij} \equiv x_i x_j \), showing that the weights are drawn from geometric distributions \[34\]. As usual, the numerical values of the unknown parameters \( \bar{x} \) are found by either maximizing the log-likelihood function

\[
\lambda(\bar{x}) = \ln P(\mathbf{W}^*|\bar{x}) = \sum_i s_i(\mathbf{W}^*) \ln x_i + \sum_{i<j} \ln (1 - x_i x_j)
\]

or solving the corresponding system of \( N \) equations:

\[
\langle s_i \rangle = \sum_{i \neq j} \frac{x_i x_j}{1 - x_i x_j} = s_i(\mathbf{W}^*) \quad \forall i \quad (20)
\]

In this case, after finding the unknown parameters we can sample the canonical ensemble by drawing, for each pair of vertices \( i, j \) and \( k \), a link of weight \( w_k \) with geometrically distributed probability \( p_{ij}^k (1 - p_{ij}) \). Note that this correctly includes the case \( w_{ij} = 0 \), occurring with probability \( 1 - p_{ij} \), corresponding to the absence of a link. Equivalently, as discussed in \[34\], one can alternatively start with the disconnected vertices \( i \) and \( j \), draw a first link (of unit weight) with Bernoulli-distributed probability \( p_{ij} \), and (only if this event is successful) place a second unit of weight on the same link, again with probability \( p_{ij} \), and so on until a failure is first encountered. In this way, only repetitions of elementary Bernoulli trials are involved, a feature that can sometimes be convenient for coding purposes (e.g. if only uniformly random number generators need to be used). After all pairs of vertices have been considered and a single weighted network has been sampled, the process can be repeated to get the desired number of sampled matrices.

In fig. 3 we show an application of this method to the same interbank network considered previously in figs. 1 and 2, but now using its weighted representation \[35\]. In this case we plot, for increasing numbers of sampled networks, the convergence of the sample average \( \bar{w}_{ij} \) of each edge weight to its exact canonical expectation \( \langle w_{ij} \rangle \). As for the example considered for the UBCM, generating 1000 matrices (bottom right) turns out to be enough to obtain a high level of accuracy for this network. If needed, this accuracy can be quantified and improved by sampling more matrices. As in the binary case, this important check is impossible in microcanonical approaches, where there is no knowledge of the exact value of the expected weights.
FIG. 3. Sampling weighted undirected networks with given strength sequence (Undirected Weighted Configuration Model). The example shown is the weighted network of liquidity reserves exchanges between Italian banks in 1999 [35] (N = 215). The four panels show the convergence of the sample average \( \overline{w_{ij}} \) of each entry of the weight matrix to its exact canonical expectation \( \langle w_{ij} \rangle \), for 1 (top left), 10 (top right), 100 (bottom left) and 1000 (bottom right) sampled matrices. The identity line is shown in red.

Here as well, the average of the quantities of interest over the sample can be compared with the observed values. As a preliminary check, the top plot of fig. 4 confirms that, for the sample of 1000 matrices, the sample average of the strength of each node coincides with its observed value, and the confidence intervals are very narrow around the identity line. Thus the enforced constraints are correctly reproduced. We can then properly use the UWCM as a null model to detect higher-order patterns in the network.

In the bottom panels of fig. 4 we show the average nearest neighbor strength (ANNS), \( s_{in}^{an} = \frac{\sum_{j} a w_{ij} s_{j}}{\sum_{j} w_{ij}} \), and the weighted clustering coefficient, \( c_i^w = \frac{\sum_{j,k} w_{ij} w_{jk} w_{ki}}{\sum_{j \neq k} w_{ij} w_{ik}} \). In this case, in line with previous analyses on different networks [5, 19–21], we find that the UWCM is not as effective as its binary counterpart in reproducing the observed higher-order properties, as clear from the presence of many outliers in the plots. Since our previous checks ensure that the implementation of the null model is correct, we can safely conclude that the divergence between the null model and the real network is not due to an insufficient or incorrect sampling of the ensemble. Rather, it is a genuine signature of the fact that, in this network, the strength sequence alone is not enough in order to replicate higher-order quantities. So the strength sequence turns out to be less informative (about the whole weighted network) than the degree sequence is (about the binary projection of the same network). This is in line with various recent findings on several weighted networks [20, 21, 26].

E. Weighted directed networks with given in-strength and out-strength sequences

We now consider weighted directed networks (WDNs), defined by a weight matrix \( W \) which is in general not symmetric. Each node is now characterized by two strengths, the out-strength \( s_{out}^i = \sum_j w_{ij} \) and the in-strength \( s_{in}^i = \sum_j w_{ji} \). The Directed Weighted Configuration Model (DWCM), the directed version of the UWCM, enforces the out- and in-strength sequences, \( \{s_{out}^i\}_{i=1}^N \) and \( \{s_{in}^i\}_{i=1}^N \), of a real-world network \( W^* \) [5, 20, 21]. The model is widely used to detect modules and communities in real WDNs [4].

In its canonical version, the DWCM is still characterized by eq. (18) where ‘\( i < j \)’ is replaced by ‘\( i \neq j \)’ and now \( p_{ij} = \bar{x}_i \bar{y}_j \). The 2N unknown parameters \( \bar{x} \) and \( \bar{y} \) can be fixed by either maximizing the log-likelihood...
FIG. 4. Sampling weighted undirected networks with given strength sequence (Undirected Weighted Configuration Model). The example shown is the weighted network of liquidity reserves exchanges between Italian banks in 1999 [35] (N = 215). The three panels show, for each node in the network, the comparison between the observed value and the sample average of the (constrained) strength (top), the (unconstrained) ANNS (bottom left) and the (unconstrained) weighted clustering coefficient (bottom right), for 1000 sampled matrices. The 95% confidence intervals of the distribution of the sampled quantities is shown in pink for each node.

function [5]
\[
\lambda(x, y) \equiv \ln P(W^*|x, y) = \sum_i \left[ s_{i\text{out}}^W(W^*) \ln x_i + s_{i\text{in}}^W(W^*) \ln y_i \right] + \sum_{i \neq j} \ln(1 - x_i y_j)
\]

or solving the the corresponding 2N equations [5]:

\[
\langle s_{i\text{out}}^W \rangle = \sum_{i \neq j} \frac{x_i y_j}{1 - x_i y_j} = s_{i\text{out}}^W(W^*) \quad \forall i \tag{22}
\]

\[
\langle s_{i\text{in}}^W \rangle = \sum_{i \neq j} \frac{x_j y_i}{1 - x_j y_i} = s_{i\text{in}}^W(W^*) \quad \forall i \tag{23}
\]

Once the unknown variables are found, we can implement an efficient and unbiased sampling scheme in the same way as for the UWCM, by running over each pair of vertices twice (i.e. in both directions). One can establish the presence and weight of a link from vertex i to vertex j using the geometric distribution \(p_{ij}^W(1 - p_{ij})\), and the presence and weight of the reverse link from j to i using the geometric distribution \(p_{ji}^W(1 - p_{ji})\), these two events being independent. Alternatively, as for the undirected case, one can construct these random events as a combination of fundamental Bernoulli trials with success probability \(p_{ij}\) and \(p_{ji}\).

This generalization is straightforward, not requiring any special application to be shown. However, we have explicitly included this model in the code (see Appendix).

F. Weighted directed networks with given strength sequences and reciprocity structure

In analogy with the binary case, we now consider the Reciprocal Weighted Configuration Model (RWCM), which is a recently proposed null model that for the first time allows one to enforce the reciprocity structure in weighted networks [24]. The RWCM constrains the whole local reciprocity structure of a weighted, directed network, by enforcing three strengths for each node, mimicking the binary ones: the non-reciprocated incoming strength, \(s_i^-\equiv \sum_j w_{ij}^-\), the non-reciprocated outgoing strength, \(s_i^+\equiv \sum_j w_{ij}^+\), and the reciprocated strength, \(s_i^{\leftrightarrow}\equiv \sum_j w_{ij}^{\leftrightarrow}\) [24]. Such quantities are defined by means
of three pair-specific variables: \( w_{ij}^{\pi} \equiv \min[w_{ij}, w_{ji}], \)
\( w_{ij}^{\pi} \equiv w_{ij} - w_{ij}^{\pi} \) and \( w_{ij}^{\pi} \equiv w_{ij} - w_{ij}^{\pi} \), extending the binary \([22, 23]\) definitions.

Despite its complexity, the RWCM is analytically solvable \([24]\) and the graph probability factorizes as:

\[
P(W|\tilde{x}, \tilde{y}, z) = \prod_{i<j} \left[ (x_{ij} y_{ij})^{w_{ij}^{\pi}} (z_{ij} z_{ij})^{w_{ij}^{\pi}} / Z_{ij}(x_{ij}, x_{ji}, y_{ij}, y_{ji}, z_{ij}, z_{ji}) \right]
\]

(24)

where \( Z_{ij}(x_{ij}, x_{ji}, y_{ij}, y_{ji}, z_{ij}, z_{ji}) \) is the node-pair partition function. The \( 3N \) unknown parameters \( \tilde{x}, \tilde{y} \) and \( \tilde{z} \) must be determined either by maximizing the log-likelihood function

\[
\lambda(\tilde{x}, \tilde{y}, \tilde{z}) \equiv \ln P(W^*|\tilde{x}, \tilde{y}, \tilde{z}) = \sum_i \left[ s_i^{\pi} (W^*) \ln x_i + s_i^{\pi} (W^*) \ln y_i \right] + \sum_{i<j} \ln Z_{ij}(x_{ij}, x_{ji}, y_{ij}, y_{ji}, z_{ij}, z_{ji})
\]

or by solving the corresponding 3N equations:

\[
\langle s_i^{\pi} \rangle = \sum_{i \neq j} \frac{(x_{ij} y_{ij})(1 - x_{ij} y_{ij})}{(1 - x_{ij} y_{ij})(1 - x_{ij} y_{ij})} = s_i^{\pi} (W^*) \quad \forall i \quad \text{(26)}
\]

\[
\langle s_i^{\pi} \rangle = \sum_{i \neq j} \frac{(x_{ij} y_{ij})(1 - x_{ij} y_{ij})}{(1 - x_{ij} y_{ij})} = s_i^{\pi} (W^*) \quad \forall i \quad \text{(27)}
\]

\[
\langle s_i^{\pi} \rangle = \sum_{i \neq j} \frac{z_{ij} z_{ij}}{(1 - z_{ij} z_{ij})} = s_i^{\pi} (W^*) \quad \forall i \quad \text{(28)}
\]

Equation (24) shows that pairs of nodes are independent, and that the probability that the nodes \( i \) and \( j \) are connected via a combination of weighted edges of the form \( (w_{ij}^{\pi}, w_{ij}^{\pi}, w_{ij}^{\pi}) \) is

\[
\left[ (x_{ij} y_{ij})^{w_{ij}^{\pi}} (z_{ij} z_{ij})^{w_{ij}^{\pi}} / Z_{ij}(x_{ij}, x_{ji}, y_{ij}, y_{ji}, z_{ij}, z_{ji}) \right]
\]

(29)

The existence of a non-reciprocated weight from \( j \) to \( i \) \((w_{ij}^{\pi} = 0, w_{ij}^{\pi} > 0)\) with probability \((x_{ij} y_{ij})(1 - x_{ij} y_{ij})\), we establish

Third, if a non-reciprocated connection has been established (i.e. if its weight \( w \) is at least one) we then focus on the positive weight to be assigned to it (i.e. on the ‘extra weight’ \( w - 1 \)). If \( w_{ij}^{\pi} > 0 \), we draw the weight \( w_{ij}^{\pi} \) from a geometric distribution \((x_{ij} y_{ij})^{w_{ij}^{\pi} - 1}(1 - x_{ij} y_{ij})\) (shifted to strictly positive integer values of \( w_{ij}^{\pi} \); note the rescaled exponent), while if \( w_{ij}^{\pi} > 0 \) we draw the weight \( w_{ij}^{\pi} \) from the distribution \((x_{ij} y_{ij})^{w_{ij}^{\pi} - 1}(1 - x_{ij} y_{ij})\) (shifted to positive integer values of \( w_{ij}^{\pi} \)).

The recipe described above is still of complexity \( O(N^2) \) and allows us to sample the canonical ensemble of the RWCM in an unbiased and efficient way. It should be noted that no microcanonical analogue of this algorithm has been proposed so far. As for the DWCM, we show no explicit application, even if the entire algorithm is available in our code (see Appendix).

### G. Weighted undirected networks with given strengths and degrees

We finally consider a ‘mixed’ null model of weighted networks constraining both binary (degree sequence \( \{k_i\}_{i=1}^N \)) and weighted (strength sequence \( \{s_i\}_{i=1}^N \)) quantities (we only consider undirected networks for simplicity, but the extension to the directed case is straightforward). The ensemble of weighted undirected networks with given strengths and degrees has been recently introduced as the (Undirected) Enhanced Configuration Model (UECM) \([26, 27]\).

This model, which is based on analytical results derived in \([25]\), is of great importance for the problem of network reconstruction from partial node-specific information \([28]\). As we have also illustrated in fig\(4\), the knowledge of the strength sequence alone is in general not enough in order to reproduce the higher-order properties of a real-world weighted network \([20, 21]\). Usually, this is due to the fact that the expected topology is much denser than the observed one (often the expected network is almost fully connected). By contrast, it turns out that the simultaneous specification of strengths and degrees, by constraining the local connectivity to be consistent with the observed one, allows a dramatically improved reconstruction of the higher-order structure of the original weighted network \([26, 27]\).

This very promising result calls for an efficient implementation of the UECM. We now describe an appropriate sampling procedure. The probability distribution characterizing the UECM is halfway between a Bernoulli (Fermi-like) and a geometric (Bose-like) distribution \([25]\), and reads

\[
P(W|\tilde{x}, \tilde{y}) = \prod_{i<j} \left[ (x_{ij} y_{ij})^{\Theta(w_{ij})} (y_{ij} y_{ij})^{w_{ij}} (1 - y_{ij}) \right]
\]

(29)
FIG. 5. Sampling weighted undirected networks with given degree and strength sequences (Undirected Enhanced Configuration Model). The example shown is the weighted World Trade Web (N = 162) [36]. The top panels show the convergence of the sample average $a_{ij}$ of each entry of the adjacency matrix to its exact canonical expectation $\langle a_{ij} \rangle$, for 100 (left) and 1000 (right) sampled matrices. The bottom panels show the convergence of the sample average $w_{ij}$ of each entry of the weight matrix to its exact canonical expectation $\langle w_{ij} \rangle$, for 100 (left) and 1000 (right) sampled matrices. The identity line is shown in red.

As usual, the $2N$ unknown parameters must be determined either by maximizing the log-likelihood function

$$\lambda(\vec{x}, \vec{y}) \equiv \ln P(\mathbf{W}^* | \vec{x}, \vec{y}) = \sum_i [k_i(W^*) \ln x_i + s_i(W^*) \ln y_i]$$

$$+ \sum_{i<j} \ln \frac{(1 - y_i y_j)}{(1 - y_i y_j + x_i x_j y_i y_j)}$$ \hspace{1cm} (30)

or by solving the $2N$ equations [26]:

$$\langle k_i \rangle = \sum_{i \neq j} p_{ij} = k_i(W^*) \quad \forall i$$ \hspace{1cm} (31)

$$\langle s_i \rangle = \sum_{i \neq j} \frac{p_{ij}}{1 - y_i y_j} = s_i(W^*) \quad \forall i$$ \hspace{1cm} (32)

where $p_{ij} \equiv \frac{x_i x_j y_i y_j}{1 - y_i y_j + x_i x_j y_i y_j}$. In order to define an unbiased sampling scheme, we note that eq. [29] highlights the two key ingredients of the UECM, respectively controlling for the probability that a link of any weight exists and, if so, that a specific positive weight is there. In more detail, the probability to observe a weight $w_{ij} \equiv w$ between the nodes $i$ and $j$ is

$$q_{ij}(w) = \left\{ \begin{array}{ll} 1 - p_{ij} & \text{if } w = 0 \\ p_{ij} (y_i y_j)^{w-1} (1 - y_i y_j) & \text{if } w > 0 \end{array} \right.$$  

The above expression identifies two steps, similar to one of the properties of the RWCM discussed above: the model is equivalent to one where the ‘first link’ (of unit weight) is extracted from a Bernoulli distribution with probability $p_{ij}$ and where the extra weight ($w_{ij} - 1$) is extracted from a geometric distribution (shifted to the strictly positive integers) with parameter $y_i y_j$. As all the other examples discussed so far, this algorithm can be easily implemented.

In fig. 5 we provide an application of this method to the World Trade Web [19, 20, 36]. We show the convergence of the sample averages ($\bar{a}_{ij}$ and $\bar{w}_{ij}$) of the entries of both binary and weighted adjacency matrices to their exact canonical expectations ($\langle a_{ij} \rangle$ and $\langle w_{ij} \rangle$ respectively). As in the previous cases, generating 1000 matrices is enough to guarantee a tight convergence of the sample averages to their exact values (in any case, this accuracy can be quantified and improved by sampling more matrices).

For this sample of 1000 matrices, in the top plots (two in this case) of fig. 5 we confirm that both the binary and weighted constraints are well reproduced by the sample averages. When we use this null model to check for higher-order patterns in this network, we find that two important topological quantities of interest (ANND and ANNS, bottom panels of fig. 5) are well replicated by
FIG. 6. Sampling weighted undirected networks with given degree and strength sequences (Undirected Enhanced Configuration Model). The example shown is the weighted World Trade Web (N = 162) [36]. The four panels show, for each node in the network, the comparison between the observed value and the sample average of the (constrained) degree (top left), the (constrained) strength (top right), the (unconstrained) ANND (bottom left) and the (unconstrained) ANNS (bottom right), for 1000 sampled matrices. The 95% confidence intervals of the distribution of the sampled quantities is shown in pink for each node.

the model. These results are consistent with what is obtained analytically by using the same canonical null model on the same network [27]. Moreover, in this case we can calculate confidence intervals besides expected values (for instance, in fig. 6 we can clearly identify outliers that are otherwise undetected), and do this for any desired topological property, not only those whose expected value is analytically computable. Our method therefore represents an improved algorithm for the unbiased reconstruction of weighted networks from strengths and degrees [26].

III. MICROCANONICAL CONSIDERATIONS

We stress once more that the accurate reproduction of the observed constraints defining the chosen model (shown in the top panels of figs. 2, 4 and 6), while representing a fundamental check of our method’s correctness, is completely unrelated to whether the higher-order patterns are reproduced or not (bottom panels of the same figures). For instance, while the confidence intervals of the enforced constraints (top panels of figs. 2, 4 and 6) would disappear if our algorithm was a microcanonical one (since all the constraints would be matched exactly in all realizations), those of higher-order properties (bottom panels) would persist even in the microcanonical case.

Elaborating more on this point, we note that there are no a priori reasons why one should prefer the microcanonical to the canonical ensemble. As already discussed in ref. [5], the microcanonical ensemble is much less robust to noise in the original data than the canonical one. For instance, if the ‘real’ network had a few links that are missing in the data, the microcanonical ensemble generated from the data themselves would not contain the real network, so that the initial error would strongly propagate to the entire analysis. By contrast, the canonical ensemble would always contain the real network and assign it just a slightly smaller probability than the maximum one assigned to the ‘incomplete’ microcanonical configurations. Unavoidably, this would still bias the analysis, but in a much weaker way.

Moreover, while most microcanonical algorithms require as input the entire adjacency matrix of the observed graph (see sec. 1A), our canonical approach requires only the empirical values of the constraints (e.g. the degree sequence). At a theoretical level, this desirable property restores the expectation that such constraints should be the sufficient statistics of the problem. At a practical level, it enormously simplifies the data requirements.
of the sampling process. For instance, if the sampling is needed in order to reconstruct an unknown network from partial node-specific information (e.g. to generate a collection of likely graphs consistent with an observed degree of strength sequence), then most microcanonical algorithms cannot be applied, while canonical ones can reconstruct the network to a high degree of accuracy\cite{1}. The above considerations show that there are no clear reasons why one should prefer the microcanonical ensemble. In fact, given its simplicity and elegance, as well as its ability to solve the problem of biased sampling, we believe that the use of the canonical ensemble should be preferred to that of the microcanonical one.

Nonetheless, it is important to note that, at least in principle, our canonical method can also be used to provide microcanonical expectations, if needed. In fact, if the sampled configurations that do not satisfy the chosen constraints exactly are discarded, what remains is precisely an unbiased (uniform) sample of the microcanonical ensemble of networks defined by the same constraints (now enforced sharply). The sample is uniform because all the microcanonical configurations have the same probability of occurrence in the canonical ensemble (since all probabilities, as we have shown, depend only on the value of the realized constraints). The same kind of analysis presented in this paper can then be repeated to obtain the microcanonical expectations. However, to be feasible, a microcanonical sampling based on our method requires that the number $R_c$ of canonical realizations to be sampled (among which a number $R_m$ of microcanonical ones will be selected) is not too large, especially because for each canonical realization one must (in the worst-case scenario) do $O(N)$ checks to ensure that each constraint matches the observed value exactly (the actual number is smaller, since all the checks after the first unsuccessful one can be aborted).

We first discuss the relation between $R_c$ and $R_m$. Let $G$ denote a generic graph (either binary or weighted) in the canonical ensemble, and $G^*$ the observed network that needs to be randomized. Let $\tilde{C}$ formally denote a generic vector of chosen constraints, and let $\tilde{C}^* = \tilde{C}(G^*)$ indicate the observed values of such constraints. Similarly, let $\tilde{\theta}$ denote the generic vector of Lagrange multipliers (hidden variables) associated with $\tilde{C}$, and let $\tilde{\theta}^*$ indicate the vector of their likelihood-maximizing values enforcing the constraints $\tilde{C}^*$. On average, out of $R_c$ canonical realizations, we will be left with a number

$$ R_m = Q(\tilde{C}^*)R_c \quad (33) $$

of microcanonical realizations, where $Q(\tilde{C}^*)$ is the probability to pick a graph in the canonical ensemble that matches the constraints $\tilde{C}^*$ exactly. This probability reads

$$ Q(\tilde{C}^*) = \sum_{G/\tilde{C}(G)=\tilde{C}^*} P(G|\tilde{\theta}^*) = N_m(\tilde{C}^*)P(G^*|\tilde{\theta}^*) \quad (34) $$

where $P(G|\tilde{\theta}^*)$ is the probability of graph $G$ in the canonical ensemble, and $N_m(\tilde{C}^*)$ is the number of microcanonical networks matching the constraints $\tilde{C}^*$ exactly (i.e. the number of graphs with given $\tilde{C}^*$). Inserting eq. (34) into eq. (33) and inverting, we find that the value of $R_c$ required to distill $R_m$ microcanonical graphs is

$$ R_c = \frac{R_m}{N_m(\tilde{C}^*)P(G^*|\tilde{\theta}^*)} \quad (35) $$

Note that $P(G^*|\tilde{\theta}^*)$ is nothing but the maximized likelihood of the observed network, which is directly accessible to our method. This is typically an extremely small number: for the networks in our analysis, it ranges between $3.8 \cdot 10^{-36468}$ (World Trade Web) and $4.9 \cdot 10^{-3499}$ (binary interbank network). On the other hand, the number $N_m(\tilde{C}^*)$ is very large (compensating the small value of the likelihood) but unknown in the general case: enumerating all graphs with given (sharp) properties is an open problem in combinatorics, and asymptotic estimates are available only under certain assumptions. This means that it is difficult to get a general estimate of the minimum number $R_c$ of canonical realizations required to distill a desired number $R_m$ of microcanonical graphs. However, if for large networks the two ensembles converge, we then expect that the canonical probability $P(G^*|\tilde{\theta}^*)$ approaches the corresponding microcanonical probability $1/N_m(\tilde{C}^*)$. If this is the case, then $R_c \approx R_m$, i.e. a large percentage of the canonical realizations are also microcanonical.

Another criterion can be obtained by estimating the number $R_c$ of canonical realizations such that the microcanonical subset samples a desired fraction $f_m$ (rather than a desired number $R_m$) of all the $N_m(\tilde{C}^*)$ microcanonical graphs. In this case, the knowledge of $N_m(\tilde{C}^*)$ becomes unnecessary: from the definition of $f_m$ we get

$$ f_m = \frac{R_m}{N_m(\tilde{C}^*)} = \frac{Q(\tilde{C}^*)R_c}{N_m(\tilde{C}^*)} = P(G^*|\tilde{\theta}^*)R_c \quad (36) $$

The above formula shows that, if we want to sample a number $R_m$ of microcanonical realizations that span a fraction $f_m$ of the microcanonical ensemble, we need to sample a number

$$ R_c = \frac{f_m}{P(G^*|\tilde{\theta}^*)} \quad (37) $$

of canonical realizations and discard all the non-microcanonical ones. This number can be extremely large, since $P(G^*|\tilde{\theta}^*)$ is very small, as we have already noticed. On the other hand, $f_m$ can be chosen to be very small as well. To see this, let us for instance compare $f_m$ with the corresponding fraction

$$ f_c = \frac{R_c}{N_c(\tilde{C}^*)} \quad (38) $$

of canonical configurations sampled by $R_c$ realizations, where $N_c(\tilde{C}^*) \gg N_m(\tilde{C}^*)$ is the number of graphs in
the canonical ensemble. For all networks we considered in this paper, we showed that $R_c = 1000$ realizations were enough to generate a good sample. This however corresponds to an extremely small value of $f_c$. For instance, for the binary interbank network we have $f_c = 1000/2^{2N(N−1)/2} ≈ 1.4 \cdot 10^{-6920}$. We might therefore be tempted to choose the same small value also for $f_m$, and find the required number $R_c$ from eq. (37). However, the result is a value $R_c ≪ 1$ (in the mentioned example, $R_c = 2.8 \cdot 10^{-3422}$), which clearly indicates that setting $f_m ≡ f_c$ (where $f_c$ is an acceptable canonical fraction) is inappropriate. In general, $f_m$ should be much larger than $f_c$.

Importantly, we can show that, given a value $R_c ≫ 1$ that generates a good canonical sample, the subset of the $R_m$ microcanonical relations contained in the $R_c$ canonical ones spans a fraction $f_m$ of the microcanonical ensemble that is indeed much larger than $f_c$. To see this, note that $P(G^*|\vec{\theta}^*)$, being obtained with the introduction of the constraints $C^*$, is necessarily much larger than the completely uniform probability $1/N_c(C^*)$ over the canonical ensemble (corresponding to the absence of constraints). This inequality implies that, if we compare $f_c$ with $f_m$ (both obtained with the same value of $R_c$), we find that

$$f_m = P(G^*|\vec{\theta}^*)R_c ≫ \frac{R_c}{N_c(C^*)} = f_c$$

(39)

The above expression shows that, even if only $R_m$ out of the (many more) $R_c$ canonical realizations belong to the microcanonical ensemble, the resulting microcanonical sampled fraction $f_m$ is always much larger than the corresponding canonical fraction $f_c$. This non-obvious result implies that, in order to sample a microcanonical fraction that is much larger than the canonical fraction obtained with a given value of $R_c$, one does not need to increase the number of canonical realizations beyond $R_c$. Although not a proof of the fact that $R_c ≫ 1$ implies $R_m ≫ 1$ (as would be desired), the above argument is consistent with the aforementioned expectation following from the assumption of equivalence of the two ensembles for large networks.

The above considerations suggest that, under appropriate conditions, using our ‘Max & Sam’ method to sample the microcanonical ensemble might be competitive with the available microcanonical algorithms. It should be noted that the value of $R_c$ affects neither the preliminary search for the hidden variables $\vec{\theta}$, nor the calculation of the microcanonical averages over the $R_m$ final networks. However, it does affect the number of checks one has to make on the constraints to select the microcanonical networks. The worst-case total number of checks is $O(R_c N)$, and performing such operation in a non-optimized way might slow down the algorithm considerably. A good optimization would be achieved by using $P(G|\vec{\theta}^*)$ to identify the vertices for which it is more unlikely that the local constraint is matched exactly, and check these vertices first. This would allow one to identify, for each of the $R_c$ canonical realizations, the constraint-violating nodes at the earliest possible stage, and thus to abort the following checks for that particular network. Implementing such an optimized microcanonical algorithm is however beyond the scope of this paper.

IV. CONCLUSIONS

The definition and correct implementation of null models is a crucial issue in network analysis. When applied to real-world networks (that are generally strongly heterogeneous), the existing algorithms to enforce simple constraints on binary graphs become biased or time-consuming, and in any case difficult to extend to networks of different type (e.g. weighted or directed) and to more complicated constraints. We have proposed a fast and unbiased ‘Max & Sam’ method to sample several canonical ensembles of networks with various constraints. While canonical ensembles are known to represent a mathematically tractable counterpart of microcanonical ones, they have not been used so far as a tool to sample networks with soft constraints, mainly because of the use of approximated expressions that result in ill-defined sampling probabilities. Here, we have shown that it is indeed possible to use the exact expressions to correctly sample a number of canonical ensembles, from the standard ones of binary graphs with given degree sequence to the more challenging ones of directed and weighted graphs with given reciprocity structure or joint strength-degree sequence.

Our algorithms are unbiased and efficient: their computational complexity is $O(N^2)$ even for strongly heterogeneous networks. Canonical sampling algorithms may therefore represent an unbiased, fast, and more flexible alternative to their microcanonical counterparts. Moreover, we have also illustrated the possibility to obtain an unbiased microcanonical method by discarding the realizations that do not match the constraints exactly. In our opinion, these findings might suggest new possibilities of exploitation of canonical ensembles as a solution to the problem of biased sampling in many other fields besides network science.

APPENDIX: THE ‘MAX & SAM’ CODE

A Matlab code, available here [37], has been written to implement the aforementioned procedure for all the seven null models described in sec. III. The routine can be implemented by typing a command having the typical form of a Matlab function, taking a number of different parameters as input. A detailed explanation accompanies the code in the form of a “Read me” file [37]. Here we briefly mention the main features.

The output of the algorithm is the numerical value of the hidden variables, i.e. the vectors $\vec{x}$, $\vec{y}$ and $\vec{z}$ (where applicable) maximizing the likelihood of the desired null
model (see sec. II), plus a specifiable number of sampled matrices. The hidden variables alone allow the user to numerically compute the expected values of the adjacency matrix entries \( \langle a_{ij} \rangle \equiv p_{ij} \) and \( \langle w_{ij} \rangle \), as well as the expected value of the constraints (as a check of its consistency with the observed value), according to the specific definition of each model. Moreover, the user can obtain as output any number of matrices (networks) sampled from the desired ensemble. These matrices are sampled in an unbiased way from the canonical ensemble corresponding to the chosen null model, using the relevant random variables as described in sec. II.

The command to be typed is the following (more details can be found in the “Read_me” file [37]):

\[
\text{output} = \text{MAXandSAM(method, Matrix, Par, List, eps, sam, x0new)}
\]

The first parameter (method) can be entered by typing the acronym associated with the selected null model:

- **UBCM** for the Undirected Binary Configuration Model, preserving the degree sequence \( \{k_i\}_{i=1}^N \) of an undirected binary network \( A^* \) (see sec. II A);
- **DBCM** for the Directed Binary Configuration Model, preserving the in- and out-degree sequences \( \{k^\text{in}_i\}_{i=1}^N \) and \( \{k^\text{out}_i\}_{i=1}^N \) of a directed binary network \( A^* \) (see sec. II B);
- **RBCM** for the Reciprocal Binary Configuration Model, preserving the reciprocated, incoming non-reciprocated and outgoing non-reciprocated degree sequences \( \{k^\leftrightarrow_i\}_{i=1}^N \), \( \{k^\text{in}_i\}_{i=1}^N \) and \( \{k^\text{out}_i\}_{i=1}^N \) of a directed binary network \( A^* \) (see sec. II C);
- **UWCM** for the Undirected Weighted Configuration Model, preserving the strength sequence \( \{s_i\}_{i=1}^N \) of an undirected weighted network \( W^* \) (see sec. II D);
- **DWCM** for the Directed Weighted Configuration Model, preserving the the strength sequences \( \{s^\text{in}_i\}_{i=1}^N \) and \( \{s^\text{out}_i\}_{i=1}^N \) of a directed weighted network \( W^* \) (see sec. II E);
- **RWC** for the Reciprocal Weighted Configuration Model, preserving the the reciprocated, incoming non-reciprocated and outgoing non-reciprocated strength sequences \( \{s^\leftrightarrow_i\}_{i=1}^N \), \( \{s^\text{in}_i\}_{i=1}^N \) and \( \{s^\text{out}_i\}_{i=1}^N \) of a directed weighted network \( W^* \) (see sec. II E);
- **UECM** for the Undirected Enhancened Configuration Model, preserving both the degree and strength sequences \( \{k_i\}_{i=1}^N \) and \( \{s_i\}_{i=1}^N \) of an undirected weighted network \( W^* \) (see sec. II F).

The second, third and fourth parameters (Matrix, Par and List respectively) specify the format of the input data (i.e. of \( A^* \) or \( W^* \)). Different data formats can be taken as input:

- **Matrix** for a (binary or weighted) matrix representation of the data, i.e. if the entire adjacency matrix is available;
- **List** for an edge-list representation of the data, i.e. a \( L \times 3 \) matrix (\( L \) being the number of links) with the first column listing the starting node, the second column listing the ending node and the third column listing the weight (if available) of the corresponding link;
- **Par** when only the constraints’ sequences (degrees, strengths, etc.) are available.

In any case, the two options that are not selected are left empty, i.e. their value should be “[ ]”. We stress that the likelihood maximization procedure (or the solution of the corresponding system of equations making the gradient of the likelihood vanish), which is the core of the algorithm, only needs the observed values of the chosen constraints to be implemented. However, since different representations of the system are available, we have chosen to exploit them all and to let the user choose the most appropriate to the specific case. For instance, in network reconstruction problems [25] one generally has empirical access only to the local properties (degree and/or strength) of each node, and the full adjacency matrix is unknown.

The fifth parameter (eps) controls for the maximum allowed relative error between the observed and the expected value of the constraints. According to this parameter, the routine solves the entropy-maximization problem by either just maximizing the likelihood function or also improving this first outcome solution by further solving the associated system. Even if this choice might strongly depend on the observed data, the value \( \epsilon = 10^{-6} \) works satisfactorily in most cases.

The sixth parameter (sam) is a boolean variable allowing the user to extract the desired number of matrices from the chosen ensemble (using the probabilities \( p_{ij} \)). The value “0” corresponds to no sampling: with this choice, the routine gives only the hidden variables as output. If the user enters “1” as input value, the algorithm will ask him/her to enter the number of desired matrices (after the hidden variables have been found). In this case, the routine outputs both the hidden variables and the sampled matrices, the latter in a .mat file called **Sampling.mat**.

The seventh parameter (x0new) is optional and has been introduced to further refine the solution of the UECM [26] in the very specific case of networks having, at the same time, big outliers in the strength distribution and a narrow degree distribution. In this case, the optional argument x0new can be inputed with the previously obtained output: in so doing, the routine will solve the system again, by using the previous solution as initial point. This procedure can be iterated until the desired
precision is reached. Note that, since $x_{\text{new}}$ is an optional parameter, it is not required to enter “[ ]” when the user does not need it (differently e.g. from the data format case).

ACKNOWLEDGMENTS

DG acknowledges support from the Dutch Econophysics Foundation (Stichting Econophysics, Leiden, the Netherlands) with funds from beneficiaries of Duyfken Trading Knowledge BV, Amsterdam, the Netherlands. This work was also supported by the EU project MULTIPLEX (contract 317532) and the Netherlands Organization for Scientific Research (NWO/OCW).

[1] M.E.J. Newman, “Networks: an introduction”, Oxford University Press (2010).
[2] V. Colizza, A. Barrat, M. Barthlemy, A. Vespignani, Proceedings of the National Academy of Sciences 103(7), 2015-2020 (2006).
[3] T. Squartini, I. van Lelyveld, D. Garlaschelli, Sci. Rep. 3(3357) (2013).
[4] A. Barrat, M. Barthlemy, A. Vespignani, “Dynamical processes on complex networks”, Cambridge University Press (2008).
[5] T. Squartini, D. Garlaschelli, New. J. Phys. 13, 083001 (2011).
[6] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, U. Alon, Science 298(5594), 824-827 (2002).
[7] S. Fortunato, Phys. Rep. 486(3), 75-174 (2010).
[8] M.E.J. Newman, S.H. Strogatz, D.J. Watts, Phys. Rev. E 64(2), 026118 (2001).
[9] S. Maslov and K. Sneppen, Science 296, 910 (2002).
[10] A.C.C. Coolen, A. De Martino, A. Annibale, J. Stat. Phys. 136, 1035-1067 (2009).
[11] E.S. Roberts, A.C.C. Coolen, Phys. Rev. E 85, 046103 (2012).
[12] Y. Artzy-Randrup, L. Stone, Phys. Rev. E 72(5), 056708 (2005).
[13] C.I. Del Genio, H. Kim, Z. Toroczkai, K.E. Bassler, PLos One 5(4), e10012 (2010).
[14] H. Kim, C.I. Del Genio, K.E. Bassler, Z. Toroczkai, New J. Phys. 14, 023012 (2012).
[15] J. Blitzstein, P. Diaconis, Internet Mathematics 6(4), 489-522 (2011).
[16] J. Park, M.E.J. Newman, Phys. Rev. E 70, 066117 (2004).
[17] G. Bianconi, Europhys. Lett. 81(2), 28005 (2007).
[18] A. Fronczak, P. Fronczak, J.A. Holyst, Phys. Rev. E 73, 016108 (2006).
[19] T. Squartini, G. Fagiolo, D. Garlaschelli, Phys. Rev. E 84, 046117 (2011).
[20] T. Squartini, G. Fagiolo, D. Garlaschelli, Phys. Rev. E 84, 046118 (2011).
[21] G. Fagiolo, T. Squartini, D. Garlaschelli, J. Econ. Interac. Coord., 8(1), 75-107 (2013).
[22] D. Garlaschelli and M.I. Loffredo, Phys. Rev. E 73, 015101(R) (2006).
[23] T. Squartini, D. Garlaschelli, Lec. Notes Comp. Sci. 7166, 24 (2012).
[24] T. Squartini, F. Picciolo, F. Ruzzenenti, D. Garlaschelli, Sci. Rep. 3(2729) (2013).
[25] D. Garlaschelli, M.I. Loffredo, Phys. Rev. Lett. 102, 038701 (2009).
[26] D. Garlaschelli, M.I. Loffredo, Phys. Rev. Lett. 102, 038701 (2009).
[27] R. Mastrandrea, T. Squartini, G. Fagiolo, D. Garlaschelli, New J. Phys. 16, 043022 (2014).
[28] R. Mastrandrea, T. Squartini, G. Fagiolo, D. Garlaschelli, http://arxiv.org/abs/1402.4171 (2014).
[29] R. Milo, N. Kashtan, S. Itzkovitz, M.E.J. Newman, U. Alon, http://arxiv.org/abs/cond-mat/0312028 (2003).
[30] P. Erdős, T. Gallai, Mat Lapok 11, 477 (1960).
[31] F. Chung, L. Lu, Proceedings of the National Academy of Sciences 99(25), 15879-15882 (2002).
[32] F. Chung, L. Lu, Annals of Combinatorics 6(2), 125-145 (2002).
[33] M. Boguña, R. Pastor-Satorras, A. Vespignani, The European Physical Journal B-Condensed Matter and Complex Systems 38(2), 205-209 (2004).
[34] M. Catanzaro, M. Boguña, R. Pastor-Satorras, Phys. Rev. E 71(2), 027103 (2005).
[35] D. Garlaschelli, New J. Phys. 11, 073005 (2009).
[36] G. De Masi, G. Iori, G. Caldarelli, Phys. Rev. E 74, 066112 (2006).
[37] UNCOMTRADE database: http://comtrade.un.org/
[38] http://www.mathworks.it/matlabcentral/fileexchange/46912-max-sam-package-zip