Unbiased sampling of network ensembles

2015 New J. Phys. 17 023052

(http://iopscience.iop.org/1367-2630/17/2/023052)

New Journal of Physics
The open access journal at the forefront of physics

Deutsche Physikalische Gesellschaft DPG
IOP Institute of Physics

Published in partnership with: Deutsche Physikalische Gesellschaft and the Institute of Physics

OPEN ACCESS

RECEIVED
16 July 2014

REVISED
8 January 2015

ACCEPTED FOR PUBLICATION
26 January 2015

PUBLISHED
18 February 2015

PAPER

Unbiased sampling of network ensembles

Tiziano Squartini[1,2], **Rossana Mastrandrea**[3,4] **and Diego Garlaschelli**[1]

[1] Instituut-Lorentz for Theoretical Physics, University of Leiden, Niels Bohrweg 2, 2333 CA Leiden, The Netherlands
[2] Institute for Complex Systems UOS Sapienza, 'Sapienza' University of Rome, P.le Aldo Moro 5, I-00185 Rome, Italy
[3] Institute of Economics and LEM, Scuola Superiore Sant'Anna, I-56127 Pisa, Italy
[4] Aix Marseille Université, Université de Toulon, CNRS, CPT, UMR 7332, F-13288 Marseille, France

E-mail: garlaschelli@lorentz.leidenuniv.nl

Keywords: complex networks, maximum entropy principle, null models of graphs, sampling network ensembles, ensemble nonequivalence

Abstract

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. An important requirement is that the sampling process is unbiased and efficient. The main approaches are microcanonical, i.e. they sample graphs that match the enforced constraints exactly. Unfortunately, when applied to strongly heterogeneous networks (like most real-world examples), 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' for short) method to correctly sample ensembles of networks where the constraints are 'soft', i.e. realized as ensemble averages. Our method is based on exact maximum-entropy distributions and is therefore 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-strength sequences and full reciprocity structure, for which no alternative method exists. Our canonical approach can in principle be turned into an unbiased microcanonical one, via a restriction to the relevant subset. Importantly, the analysis of the fluctuations of the constraints suggests that the microcanonical and canonical versions of all the ensembles considered here are not equivalent. We show various real-world applications and provide a code implementing all our algorithms.

1. Introduction

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 additional factors.

To achieve this goal, one requires (a family of) randomized benchmarks, i.e. ensembles of graphs where the local heterogeneity is the same as in the real network, 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 theoretical expectations of the null model [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 alternatively a 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 [8–10]. 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 other null models, defined e.g. for directed or weighted graphs and specified by more general constraints.

Several approaches to the problem have been proposed and can be roughly divided in two large classes: *microcanonical* and *canonical* methods. Microcanonical approaches [11–17] 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. As we discuss in this paper, 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, 18–28] the constraints are 'soft', i.e. they can be violated by individual graphs in the ensemble, even if the ensemble average of each constraint still matches 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. Unfortunately, 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. This implies that the computational use of canonical null models has not been implemented systematically so far.

In this paper, by combining an exact maximum-likelihood approach with an efficient computational sampling scheme, we define a rigorously unbiased method to sample ensembles of various types of networks (i.e. directed, undirected, weighted, binary) with many possible constraints (degree sequence, strength sequence, reciprocity structure, mixed binary and weighted properties, etc). We make use of a series of recent analytical results that generate the exact probabilities in all these cases of interest [5, 21–29] and consider various examples illustrating the usefulness of our method when applied to real-world networks.

We also analyse the canonical fluctuations of the constraints in each model. Previous theoretical analyses of fluctuations in some network ensembles have been carried out, for instance, in [37] for graphs with given degree sequence and in [38] for graphs with given community structure. Also, a comparison between some microcanonical and canonical network ensembles has been carried out in [39]. In this paper, we provide a complete analytical characterization of the fluctuations of each constraint for all the ensembles under study. For the majority of these ensembles, the exact analytical expressions characterizing the fluctuations are derived here for the first time. Moreover, in our maximum-likelihood approach the knowledge of the hidden variables allows us to calculate, for the first time, the exact value of the fluctuations explicitly for each node in the empirical networks considered. Our results suggest that, unlike in most physical systems, the microcanonical and canonical versions of the graph ensembles considered here are surprisingly *not* equivalent (see [40] for a recent mathematical proof of ensemble nonequivalence in the UBCM).

In any case, our canonical method can in principle be converted into an unbiased microcanonical one, if we discard all the sampled networks that violate the sharp constraints. At the end of the paper, we discuss the advantages and disadvantages of this procedure explicitly, and clarify that canonical ensembles are more appropriate in presence of missing entries or errors in the data.

Finally, we include an appendix with a description of a algorithm that we have explicitly coded in various ways [43–45]. The algorithm allows the users to sample all the graph ensembles described in this paper, given an empirically observed network (or even only the values of the constraints).

2. 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 limitations that call for an improved solution. 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 and other types of networks as well.

2.1. 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 [10]. 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 [11]. 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 [11]. This 'local rewiring algorithm' (LRA) starts from the original network, rather than from scratch, and randomizes the topology through the iteration of an elementary move that preserves the degrees of all nodes. While this algorithm always produces random networks, it is very time consuming since many iterations of the fundamental move are needed in order to produce just one randomized variant, and this entire operation has to be repeated several times (the mixing time being still unknown [30]) in order to produce many variants.

Besides these practical problems, the main conceptual limitation 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 [12–14]. For undirected networks, uniformity has been shown to hold, at least approximately, only when the degree sequence is such that [12]

$$k_{\max} \cdot \overline{k^2} \Big/ \left(\bar{k}\right)^2 \ll N, \tag{1}$$

where $k_{\max}$ is the largest degree in the network, $\bar{k}$ is the average degree, $\overline{k^2}$ is the second moment, and N is the number of vertices. 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 equation (1) is required to avoid the generation of bias [13]. Again, this condition is strongly violated by most real-world networks. Moreover, the directed version of the LRA is also non-ergodic, i.e. it is in general not able to explore the entire ensemble of networks [13].

It has been shown that ergodicity can be restored by introducing an additional triangular move inverting the direction of closed loops of three vertices [13]. However, in order to restore uniformity (for both directed and undirected graphs) one needs to introduce an appropriate acceptance probability for the rewiring move [12–14]. Unfortunately, the acceptance probability depends on some nontrivial property of the current network configuration. Since this property must be recalculated at each step, the resulting algorithm is significantly time consuming. Quantifying the bias generated by the LRA when equation (1) (or its directed counterpart) is violated is difficult, mainly because an exact mathematical characterization of microcanonical graph ensembles valid in such regime is still lacking. Yet, the proof of the existence of bias provided in [12, 13] is an obvious warning against the use of the LRA on strongly heterogeneous networks. The reader is referred to those papers for a discussion.

Other recent alternatives [15–17] rely on theorems, such as the Erdős–Gallai [31] 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 [15–17]. Del Genio *et al* [15] 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

$$k_{\max} < \sqrt{N} \tag{2}$$

and to $O(N^3)$ if $k_{\max} > \sqrt{N}$. 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 equation (2) is another indication that, for strongly heterogeneous networks, the problem of sampling becomes more complicated. Unfortunately, most real networks violate equation (2) strongly.

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

2.2. 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 proability distribution over the enlarged ensemble has maximum entropy [5, 18].

For binary graphs, since any topological property X is a function $X(\mathbf{A})$ of the adjacency matrix $\mathbf{A}$ 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 that of finding a mathematical expression for the probability $P(\mathbf{A})$ of occurrence of each graph. This allows to compute the expected value of X as $\sum_{\mathbf{A}} P(\mathbf{A}) X(\mathbf{A})$. Importantly, for canonical ensembles with local constraints $P(\mathbf{A})$ factorizes to a product over pairs of nodes, where each term in the product involves the probability p_{ij} that the vertices i and j are connected in the ensemble. Determining the mathematical form of p_{ij} is the main goal of canonical approaches. Note that, by contrast, in the microcanonical ensemble all links are dependent on each other (the degree sequence must be reproduced exactly in each realization), which implies that the probability of the entire graph does not factorize to node-pair probabilities.

For binary undirected networks (BUNs), the most popular specification for p_{ij} is the factorized one [1, 32, 33]:

$$p_{ij} = \frac{k_i k_j}{k_{\text{tot}}}, \tag{3}$$

(where k_i is the degree of node i and k_{tot} is the total degree over all nodes). For weighted undirected networks (WUNs), where each link can have a non-negative weight w_{ij} and each vertex i is characterized by a given strength s_i (the total weight of the links of node i), the corresponding assumption is that the expected weight of the link connecting the vertices i and j is

$$\langle w_{ij} \rangle = \frac{s_i s_j}{s_{\text{tot}}}, \tag{4}$$

(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 frequent use of these expressions is perhaps encountered in the empirical analysis of *communities*, i.e. relatively denser modules of vertices in large networks [7]. Most community detection algorithms compare different partitions of vertices into communities (each partition being parametrized by a matrix $\mathbf{C}$ such that $c_{ij} = 1$ if the vertices i and j belong to the same community, and $c_{ij} = 0$ otherwise) and search for the optimal partition. The latter is the one that maximizes the modularity function which, for binary networks, is defined as

$$Q(\mathbf{C}) \equiv \frac{1}{k_{\text{tot}}} \sum_{i,j} \left[a_{ij} - \frac{k_i k_j}{k_{\text{tot}}} \right] c_{ij}, \tag{5}$$

where equation (3) appears explicitly as a null model for a_{ij}. For weighted networks, a similar expression involving equation (4) applies. Other important examples where equation (3) is used are the characterization of the connected components of networks [33], the average distance among vertices [32], 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 that these equations play in many applications, it is remarkable that the literature puts very little emphasis on the fact that equations (3) and (4) are valid only under strict conditions that, for most real networks, are strongly violated. It is evident that equation (3) represents a probability only if the largest degree $k_{\max}$ in the network does not exceed the so-called 'structural cut-off' $k_c \equiv \sqrt{k_{\text{tot}}}$ [34], i.e. if

$$k_{\max} < \sqrt{k_{\text{tot}}}. \tag{6}$$

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. Unfortunately, as we discuss below, it has been shown that $k_{\max}$ strongly exceeds k_c in most real-world networks, making equation (3) ill-defined.

It should be noted that in principle the knowledge of p_{ij} allows one to sample networks from the canonical ensemble very easily, by running over all pairs of nodes and connecting them with the appropriate probability. However, the fact that $p_{ij} \gg 1$ when $k_{\max} \gg k_c$ makes such probability useless for sampling purposes. This is why, despite their conceptual simplicity, general algorithms to sample canonical ensembles of networks have not been implemented so far, and the emphasis has remained on microcanonical approaches.

2.3. The 'strong heterogeneity regime' challenging most algorithms

Equations (1), (2) and (6), along with our discussion above, show that most methods run into problems when the heterogeneity of the network is too pronounced: strongly heterogeneous networks elude most microcanonical and canonical approaches proposed so far. Unfortunately, networks in this extreme regime are known 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 [11], 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 ten-fold. Consequently, if equation (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 equation (5), this value of p_{ij} would produce, in the summation, a single term 40 times larger than any other 'regular' (i.e. of order unity) term, 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 results are found in the majority of real networks, making the problem entirely general. To see this, it is enough to exploit the fact 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 $\overline{k} = k_{\text{tot}}/N$ is finite but the second moment $\overline{k^2}$ diverges. Therefore the structural cut-off scales as $k_c \sim N^{1/2}$ [34], which means that equations (2) and (6) coincide. By contrast, extreme value theory shows that the largest degree scales as $k_{\max} \sim N^{1/(\gamma-1)}$ [34]. 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 naively 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 normally circumvented by enforcing the condition $k_{\max} < k_c$ explicitly, e.g. by considering a truncated power-law distribution. This procedure is usually justified with the expectation that the inequality $k_{\max} < k_c$ should hold for sparse networks where the average degree does not grow with N, as in most real networks [10, 35]. This interpretation of the role of sparsity is however misleading, since in real scale-free networks with $2 < \gamma < 3$ the average degree is finite irrespective of the presence of the cut-off. This makes those networks sparse even without assuming a truncation in the degree distribution. As a matter of fact, as clear from the example above, real networks systematically violate the cut-off value, and are therefore 'strongly heterogeneous', even if sparse. By the way, the fact that a high density is not the origin of the breakdown of the available approaches should be clear by considering that dense but homogeneous networks (including the densest of all, i.e. the complete graph) are such that $k_{\max} < k_c$ and are therefore correctly described by equation (3), just like sparse homogeneous networks. This confirms that the problem is in fact due to *strong heterogeneity* and not to high density.

The above arguments can be extended 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.

3. The 'Max & Sam' method

In what follows, building on a series of recent results characterizing several canonical ensembles of networks [5, 24, 26–28], we introduce 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 [18] (thus ensuring that the sampling is unbiased), and the numerical coefficients of this

probability are derived by maximizing the probability (i.e. the likelihood) itself [5]. Since this double maximization is the core of our approach, we call our method the 'Maximize and Sample' ('Max & Sam' for short) method. We also provide a code implementing all our sampling algorithms (see appendix).

We will consider canonical ensembles of binary graphs with given degree sequence (both undirected [5, 21] and directed [5, 21, 23]), of weighted networks with given strength sequence (both undirected [5, 22] and directed [5, 22, 23, 26]), of directed networks with given reciprocity structure (both binary [24, 25] and weighted [26]), and of weighted networks with given combined strength sequence and degree sequence [27–29]. In all these cases, that have been treated only separately so far, we implement an explicit sampling protocol based on the exact result that the probability of the entire network always factorizes as a product of dyadic probabilities over pairs of nodes. This ensures that the computational complexity of our sampling method is always $O(N^2)$ in all cases considered here, irrespective of the level of heterogeneity of the real-world network being randomized. Therefore our method does not suffer from the limitations of the other methods discussed in section 2: it is efficient and unbiased even for strongly heterogeneous networks.

It should be noted that, while most microcanonical algorithms require as input the entire adjacency matrix of the observed graph (see section 2.1), 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 and/or strength sequence), then most microcanonical algorithms cannot be applied, while canonical ones can reconstruct the network to a high degree of accuracy [28].

3.1. Binary undirected graphs with given degree sequence

Let us start by considering BUNs. A generic BUN is uniquely specified by its binary adjacency matrix $\mathbf{A}$. The particular matrix corresponding to the observed graph that we want to randomize will be denoted by $\mathbf{A}^{\star}$. 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 UBCM.

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, 18] is that the graph probability factorizes as

$$P(\mathbf{A}|\mathbf{x}) = \prod_i \prod_{j<i} p_{ij}^{a_{ij}} (1 - p_{ij})^{1-a_{ij}}, \tag{7}$$

where $p_{ij} \equiv \frac{x_i x_j}{1 + x_i x_j}$. The vector $\mathbf{x}$ of N unknown parameters (or 'hidden variables') is to be determined either by maximizing the log-likelihood function

$$\begin{aligned} \lambda(\mathbf{x}) &\equiv \ln P(\mathbf{A}^{\star} \mid \mathbf{x}) = \\ &= \sum_i k_i\left(\mathbf{A}^{\star}\right) \ln x_i - \sum_i \sum_{j<i} \ln\left(1 + x_i x_j\right) \end{aligned} \tag{8}$$

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

$$\langle k_i \rangle = \sum_{j \neq i} \frac{x_i x_j}{1 + x_i x_j} = k_i\left(\mathbf{A}^{\star}\right) \quad \forall\, i, \tag{9}$$

where $k_i(\mathbf{A}^{\star})$ is the observed degree of vertex i and $\langle k_i \rangle$ indicates its ensemble average. In both cases, the parameters $\mathbf{x}$ vary in the region defined by $x_i \geqslant 0$ for all i [5].

From equation (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 $\mathbf{x}$, which in turn fix the value of all the probabilities $\{p_{ij}\}$). 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 applies to all the models described in this paper and implemented in the code.

Note that the above form of p_{ij} represents the exact expression that should be used in place of equation (3). This reveals the highly nonlinear 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 equation (3). Only when the

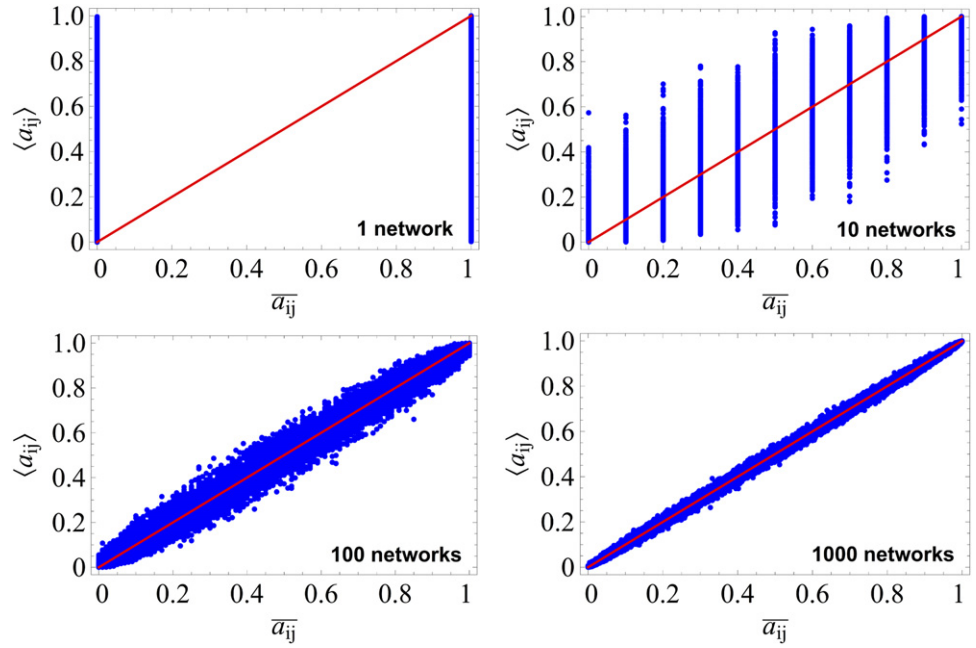

Figure 1. Sampling binary undirected networks with given degree sequence (undirected binary configuration model). The example shown is the binary network of liquidity reserves exchanges between Italian banks in 1999 [41] ($N = 215$). The four panels show the convergence of the sample average $\overline{a_{ij}}$ 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 networks. The identity line is shown in red.

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 [18], while, as we discussed in section 2.3, what defines this limit is a low level of heterogeneity, and not sparsity.

Unlike equation (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 equation (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 equation (7) to obtain the probability to correctly sample any graph $\mathbf{A}$ 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 $\mathbf{x}$ 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 section 2.1.

In figure 1 we show an application of this procedure to the network of liquidity reserves exchanges between Italian banks in 1999 [41]. For an increasing number of sampled graphs, we show the convergence of the sample average $\overline{a_{ij}}$ 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 networks (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 networks and confirm (see the top panel of figure 2) that the imposed constraints (the observed degrees of all nodes) are very well reproduced by the sample average, and that the confidence intervals are narrowly spread around the identity line. This is an important 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

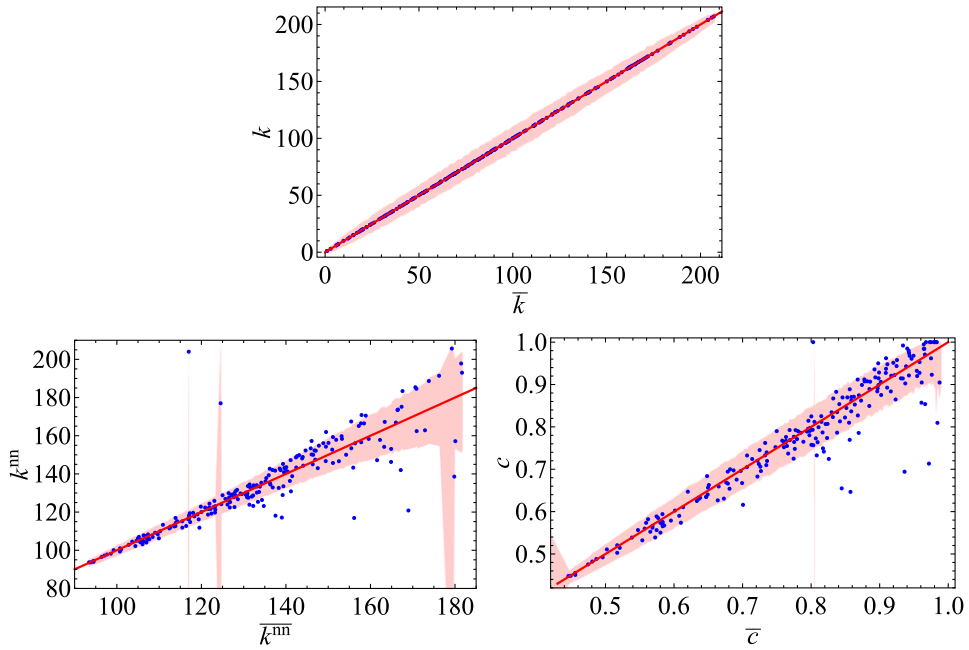

Figure 2. Sampling binary undirected networks with given degree sequence (undirected binary configuration model). The example shown is the binary network of liquidity reserves exchanges between Italian banks in 1999 [41] ($N = 215$). The three panels show, for each node in the network, the comparison between the observed value and the sample average of the (constrained) degree (top), the (unconstrained) ANND (bottom left) and the (unconstrained) clustering coefficient (bottom right), for 1000 sampled networks. The 95% confidence intervals of the distribution of the sampled quantities is shown in pink for each node.

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 figure 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), $k_i^{nn} = \frac{\sum_j a_{ij}k_j}{k_i}$, and the clustering coefficient, $c_i = \frac{\sum_{j,k} a_{ij}a_{jk}a_{ki}}{k_i(k_i-1)}$ of each vertex.

Note that, since our sampling method is unbiased, the arithmetic mean over the sample automatically weighs the configurations according to their correct probability. In this particular case, we find that the null model reproduces the observed network very 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, 21, 22], but not true in general for other networks or other constraints, as we show later on. From the bottom panels of figure 2 we also note that the confidence intervals highlight a non-obvious feature: the fact that the few points further away from the identity line turn out to be actually within (or at the border of) the chosen confidence intervals, while several points 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 derived from the analysis of the expected values alone, e.g. using expressions like equation (3) or similar.

We now calculate the fluctuations of the constraints explicitly. We start by calculating the ensemble variance of each degree k_i, defined as $\sigma^2[k_i] \equiv \langle k_i^2 \rangle - \langle k_i \rangle^2$. In the microcanonical ensemble, one obviously has $\sigma^2[k_i] = 0$. In the canonical ensemble, the independence of pairs of nodes implies that the variance of the sum $\sum_{j\neq i} a_{ij}$ coincides with the sum of the variances of its terms, i.e.

$$\sigma^2\left[k_i\right] = \sum_{j\neq i}\sigma^2[a_{ij}] = \sum_{j\neq i}\left(\left\langle a_{ij}^2\right\rangle - \left\langle a_{ij}\right\rangle^2\right)$$
$$= \sum_{j\neq i} p_{ij}\left(1-p_{ij}\right) = k_i - \sum_{j\neq i} p_{ij}^2. \qquad (10)$$

Then, the canonical relative fluctuations can be measured in terms of the so-called *coefficient of variation*, which we conveniently express in the form

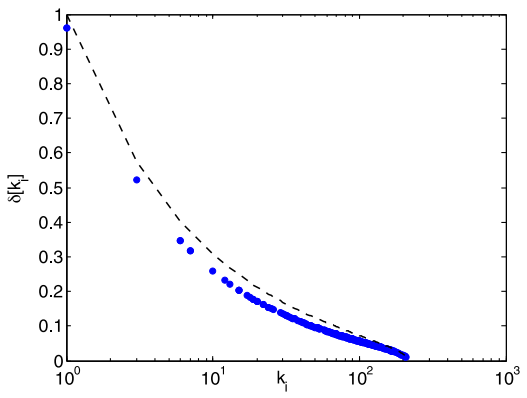

Figure 3. Coefficient of variation $\delta[k_i]$ as a function of the degree k_i for each node of the binary network of liquidity reserves exchanges between Italian banks in 1999 [41] ($N = 215$). The blue points are the exact values in equation (11), while the dashed curve is the upper bound in equation (12). The lower bound is the abscissa $\delta[k_i] = 0$.

$$\delta\left[k_i\right] \equiv \frac{\sigma\left[k_i\right]}{k_i} = \sqrt{\frac{1}{k_i} - \frac{\sum_{j\neq i} p_{ij}^2}{\left(\sum_{j\neq i} p_{ij}\right)^2}}, \tag{11}$$

where we have restricted ourselves to the case $k_i > 0$[5]. A plot of $\delta[k_i]$ as a function of k_i for the interbank network considered above is shown in figure 3. We find that the relative fluctuations vanish for vertices with large degree, while they are very large for vertices with moderate and small degree. In particular, $\delta[k_i] \approx 1$ when $k_i = 1$.

In general, we note that the term $\sum_{j\neq i} p_{ij}^2 \Big/ \left(\sum_{j\neq i} p_{ij}\right)^2$ in equation (11) is a *participation ratio*[6], measuring the inverse of the effective number of equally important terms in the sum $\sum_{j\neq i} p_{ij}$: in particular, it equals 1 if and only if there is only one nonzero term (complete concentration), while it equals $(N-1)^{-1}$ if and only if there are $N-1$ identical terms (complete homogeneity), i.e. $p_{ij} = k_i/(N-1)$ for all $j \neq i$. Since these are the two extreme bounds for a participation ratio, and since in the case of complete concentration we also have $k_i = 1$, we conclude that the bounds for $\delta[k_i]$ are

$$0 \leqslant \delta\left[k_i\right] \leqslant \sqrt{\frac{1}{k_i} - \frac{1}{N-1}}. \tag{12}$$

The resulting allowed region for $\delta[k_i]$ is the one comprised between the abscissa and the dashed line in figure 3. We find that the realized trend is close to the upper bound. This suggests that the maximum-entropy nature of our algorithm produces almost maximally homogeneous terms in the sum $\sum_{j\neq i} p_{ij}$, i.e. no particular subset of vertices is preferred as canditate partners for i, the only preference being obviously given (as a consequence of the explicit form of p_{ij} in terms of x_i and x_j) to vertices with larger degree.

Since the degree distribution of most real-world networks is such that the average degree remains finite even when the size of the network becomes very large, the above results suggest that, unlike most physical systems, the microcanonical and canonical ensembles defined by the UBCM are *not* equivalent in the 'thermodynamic' limit $N \to \infty$. While equation (12) shows that values closer to the lower bound $\delta[k_i] = 0$ can be in principle achieved, the maximization of the entropy appears to push the ensemble towards the opposite upper bound where the equivalence of the microcanonical and canonical ensembles is maximally violated. On the other hand, one might in principle construct synthetic networks with sufficiently large degrees, such that the canonical fluctuations are arbitrarily small and the two ensembles arbitrarily close.

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

For binary directed networks (BDNs), the adjacency matrix $\mathbf{A}$ is (in general) not symmetric, and each node 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), which is 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 $\{k_i^{\text{in}}\}_{i=1}^N$.

[5] The case $k_i = 0$ also implies $\sigma[k_i] = 0$ and leads to an indeterminate form for $\delta[k_i]$. However this case is uninteresting since each isolated node i remains isolated across the entire ensemble ($p_{ij} = 0\ \forall\ j$) and can be safely removed without loss of generality.

[6] Strictly speaking, it is the inverse of a so-called *inverse participation ratio*, but we avoid the use of 'inverse' twice.

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 '$j < i$' replaced by '$j \neq i$' and $p_{ij} = \frac{x_i y_j}{1 + x_i y_j}$, where the $2N$ parameters $\mathbf{x}$ and $\mathbf{y}$ are determined by either maximizing the log-likelihood function [5]

$$\begin{aligned}\lambda(\mathbf{x}, \mathbf{y}) &\equiv \ln P\left(\mathbf{A}^{\star} \mid \mathbf{x}, \mathbf{y}\right) \\ &= \sum_i \left[k_i^{\text{out}}\left(\mathbf{A}^{\star}\right) \ln x_i + k_i^{\text{in}}\left(\mathbf{A}^{\star}\right) \ln y_i\right] \\ &\quad - \sum_i \sum_{j \neq i} \ln\left(1 + x_i y_j\right),\end{aligned} \tag{13}$$

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

$$\langle k_i^{\text{out}} \rangle = \sum_{j \neq i} \frac{x_i y_j}{1 + x_i y_j} = k_i^{\text{out}}\left(\mathbf{A}^{\star}\right) \quad \forall i, \tag{14}$$

$$\langle k_i^{\text{in}} \rangle = \sum_{j \neq i} \frac{x_j y_i}{1 + x_j y_i} = k_i^{\text{in}}\left(\mathbf{A}^{\star}\right) \quad \forall i. \tag{15}$$

The parameters $\mathbf{x}$ and $\mathbf{y}$ vary in the region defined by $x_i \geqslant 0$ and $y_i \geqslant 0$ for all i respectively [5].

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).

We conclude the discussion of this ensemble with the calculation of the canonical fluctuations. In analogy with equations (10) and (11), the variances of k_i^{out} and k_i^{in} are given by

$$\sigma^2\left[k_i^{\text{out}}\right] = \sum_{j \neq i} p_{ij}\left(1 - p_{ij}\right) = k_i^{\text{out}} - \sum_{j \neq i} p_{ij}^2, \tag{16}$$

$$\sigma^2\left[k_i^{\text{in}}\right] = \sum_{j \neq i} p_{ji}\left(1 - p_{ji}\right) = k_i^{\text{in}} - \sum_{j \neq i} p_{ji}^2. \tag{17}$$

For $k_i^{\text{out}} > 0$ and $k_i^{\text{in}} > 0$, the relative fluctuations are

$$\delta\left[k_i^{\text{out}}\right] \equiv \frac{\sigma\left[k_i^{\text{out}}\right]}{k_i^{\text{out}}} = \sqrt{\frac{1}{k_i^{\text{out}}} - \frac{\sum_{j \neq i} p_{ij}^2}{\left(\sum_{j \neq i} p_{ij}\right)^2}}, \tag{18}$$

$$\delta\left[k_i^{\text{in}}\right] \equiv \frac{\sigma\left[k_i^{\text{in}}\right]}{k_i^{\text{in}}} = \sqrt{\frac{1}{k_i^{\text{in}}} - \frac{\sum_{j \neq i} p_{ji}^2}{\left(\sum_{j \neq i} p_{ji}\right)^2}}. \tag{19}$$

The above quantities still involve participation ratios defined by the connection probabilities. For the bounds of $\delta[k_i^{\text{out}}]$ and $\delta[k_i^{\text{in}}]$, considerations similar to those leading to equation (12) apply here.

3.3. 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, 24, 25]. This is equivalent to the specification of the three degree sequences defined as the vector of the numbers of non-reciprocated outgoing links, $\{k_i^{\rightarrow}\}_{i=1}^N$, the vector of the numbers of non-reciprocated incoming links, $\{k_i^{\leftarrow}\}_{i=1}^N$, and the vector of the numbers of reciprocated links, $\{k_i^{\leftrightarrow}\}_{i=1}^N$ [5, 24, 25]. These numbers are defined as $k_i^{\rightarrow} \equiv \sum_j a_{ij}(1 - a_{ji})$, $k_i^{\leftarrow} \equiv \sum_j a_{ji}(1 - a_{ij})$, and $k_i^{\leftrightarrow} \equiv \sum_j a_{ij} a_{ji}$ respectively [24, 25].

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, 25], 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 equation (3) exists, therefore the null model is usually implemented microcanonically using a generalization of the LRA that we have discussed in section 2.1. 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 [12–14]. To our knowledge, in this case no correction analogous to that proposed in [13] has been developed in order to restore uniformity.

In our 'Max & Sam' approach, we exploit known analytical results [5, 24, 25] showing that the probability of each graph $\mathbf{A}$ in the RBCM is

$$P(\mathbf{A}|\mathbf{x},\mathbf{y},\mathbf{z}) = \prod_i \prod_{j<i} \left(p_{ij}^{\rightarrow}\right)^{a_{ij}^{\rightarrow}} \left(p_{ij}^{\leftarrow}\right)^{a_{ij}^{\leftarrow}} \left(p_{ij}^{\leftrightarrow}\right)^{a_{ij}^{\leftrightarrow}} \left(p_{ij}^{\not\leftrightarrow}\right)^{a_{ij}^{\not\leftrightarrow}}, \tag{20}$$

where $p_{ij}^{\rightarrow} \equiv \frac{x_i y_j}{1+x_i y_j + x_j y_i + z_i z_j}$, $p_{ij}^{\leftarrow} \equiv \frac{x_j y_i}{1+x_i y_j + x_j y_i + z_i z_j}$, $p_{ij}^{\leftrightarrow} \equiv \frac{z_i z_j}{1+x_i y_j + x_j y_i + z_i z_j}$ and $p_{ij}^{\not\leftrightarrow} \equiv \frac{1}{1+x_i y_j + x_j y_i + z_i z_j}$ 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, $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$, must be determined by either maximizing the log-likelihood [5]

$$\begin{aligned}\lambda(\mathbf{x},\mathbf{y},\mathbf{z}) &\equiv \ln P\left(\mathbf{A}^{\star}\middle|\mathbf{x},\mathbf{y},\mathbf{z}\right) \\ &= \sum_i \left[k_i^{\rightarrow}\left(\mathbf{A}^{\star}\right)\ln x_i + k_i^{\leftarrow}\left(\mathbf{A}^{\star}\right)\ln y_i \right. \\ &\quad \left. + k_i^{\leftrightarrow}\left(\mathbf{A}^{\star}\right)\ln z_i\right] - \sum_i \sum_{j<i} \ln\left(1+x_i y_j + x_j y_i + z_i z_j\right)\end{aligned} \tag{21}$$

or, equivalently, solving the $3N$ coupled equations [5, 24, 25]:

$$\langle k_i^{\rightarrow}\rangle = \sum_{j\neq i} \frac{x_i y_j}{1+x_i y_j + x_j y_i + z_i z_j} = k_i^{\rightarrow}\left(\mathbf{A}^{\star}\right) \quad \forall i, \tag{22}$$

$$\langle k_i^{\leftarrow}\rangle = \sum_{j\neq i} \frac{x_j y_i}{1+x_i y_j + x_j y_i + z_i z_j} = k_i^{\leftarrow}\left(\mathbf{A}^{\star}\right) \quad \forall i, \tag{23}$$

$$\langle k_i^{\leftrightarrow}\rangle = \sum_{j\neq i} \frac{z_i z_j}{1+x_i y_j + x_j y_i + z_i z_j} = k_i^{\leftrightarrow}\left(\mathbf{A}^{\star}\right) \quad \forall i. \tag{24}$$

The parameters $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ vary in the region defined by $x_i \geqslant 0$, $y_i \geqslant 0$ and $z_i \geqslant 0$ for all i respectively [5].

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 $p_{ij}^{\rightarrow}$, draw a single link directed from j to i with probability $p_{ij}^{\leftarrow}$, draw two mutual links with probability $p_{ij}^{\leftrightarrow}$, or draw no link at all with probability $p_{ij}^{\not\leftrightarrow}$. 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).

Coming to the canonical fluctuations, in this ensemble equations (16) and (17) generalize to

$$\sigma^2\left[k_i^{\rightarrow}\right] = \sum_{j\neq i} p_{ij}^{\rightarrow}\left(1-p_{ij}^{\rightarrow}\right) = k_i^{\rightarrow} - \sum_{j\neq i}\left(p_{ij}^{\rightarrow}\right)^2, \tag{25}$$

$$\sigma^2\left[k_i^{\leftarrow}\right] = \sum_{j\neq i} p_{ij}^{\leftarrow}\left(1-p_{ij}^{\leftarrow}\right) = k_i^{\leftarrow} - \sum_{j\neq i}\left(p_{ij}^{\leftarrow}\right)^2. \tag{26}$$

$$\sigma^2\left[k_i^{\leftrightarrow}\right] = \sum_{j\neq i} p_{ij}^{\leftrightarrow}\left(1-p_{ij}^{\leftrightarrow}\right) = k_i^{\leftrightarrow} - \sum_{j\neq i}\left(p_{ij}^{\leftrightarrow}\right)^2. \tag{27}$$

For $k_i^{\rightarrow} > 0, k_i^{\leftarrow} > 0$ and $k_i^{\leftrightarrow} > 0$, the relative fluctuations are

$$\delta\left[k_i^{\rightarrow}\right] \equiv \frac{\sigma\left[k_i^{\rightarrow}\right]}{k_i^{\rightarrow}} = \sqrt{\frac{1}{k_i^{\rightarrow}} - \frac{\sum_{j\neq i}\left(p_{ij}^{\rightarrow}\right)^2}{\left(\sum_{j\neq i} p_{ij}^{\rightarrow}\right)^2}}, \tag{28}$$

$$\delta\left[k_i^{\leftarrow}\right] \equiv \frac{\sigma\left[k_i^{\leftarrow}\right]}{k_i^{\leftarrow}} = \sqrt{\frac{1}{k_i^{\leftarrow}} - \frac{\sum_{j\neq i}\left(p_{ji}^{\leftarrow}\right)^2}{\left(\sum_{j\neq i}p_{ji}^{\leftarrow}\right)^2}}, \tag{29}$$

$$\delta\left[k_i^{\leftrightarrow}\right] \equiv \frac{\sigma\left[k_i^{\leftrightarrow}\right]}{k_i^{\leftrightarrow}} = \sqrt{\frac{1}{k_i^{\leftrightarrow}} - \frac{\sum_{j\neq i}\left(p_{ji}^{\leftrightarrow}\right)^2}{\left(\sum_{j\neq i}p_{ji}^{\leftrightarrow}\right)^2}}. \tag{30}$$

Thus, in all the ensembles considered so far (which are defined in terms of *purely binary* constraints), the squared relative fluctuation of each constraint always takes the form of the inverse of the value of the constraint itself, *minus* the participation ratio of the corresponding probabilities.

3.4. WUNs with given strength sequence

Let us now consider 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 [1] incorrectly assume that this model is characterized by equation (4), which is instead only a highly simplified expression [5].

In the canonical ensemble, the probability of each weighted network $\mathbf{W}$ is [5]

$$P(\mathbf{W}|\mathbf{x}) = \prod_i \prod_{j<i} p_{ij}^{w_{ij}}\left(1 - p_{ij}\right), \tag{31}$$

where now $p_{ij} \equiv x_i x_j$, showing that the weights are drawn from geometric distributions [36]. As usual, the numerical values of the unknown parameters $\mathbf{x}$ are found by either maximizing the log-likelihood function

$$\begin{aligned}\lambda(\mathbf{x}) &\equiv \ln P\left(\mathbf{W}^* \mid \mathbf{x}\right) \\ &= \sum_i s_i\left(\mathbf{W}^*\right)\ln x_i + \sum_i\sum_{j<i}\ln\left(1 - x_i x_j\right)\end{aligned} \tag{32}$$

or solving the system of N equations:

$$\langle s_i\rangle = \sum_{j\neq i}\frac{x_i x_j}{1 - x_i x_j} = s_i\left(\mathbf{W}^*\right) \quad \forall i. \tag{33}$$

In both approaches, now the parameters $\mathbf{x}$ vary in the region defined by the constraint $0 \leqslant x_i x_j < 1$ for all i, j [5].

In this model, after finding the unknown parameters we can sample the canonical ensemble by drawing, for each pair of vertices i and j, a link of weight w with geometrically distributed probability $p_{ij}^w(1 - p_{ij})$. Note that this correctly includes the case $w_{ij} = 0$, occurring with probability $1 - p_{ij}$, corresponding to the absence of a link. Alternatively, using a procedure similar to that discussed in [36], one can 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 until the desired number of networks is sampled.

In figure 4 we show an application of this method to the same interbank network considered previously in figures 1 and 2, but now using its weighted representation [41]. In this case we plot, for increasing numbers of sampled networks, the convergence of the sample average $\overline{w_{ij}}$ 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

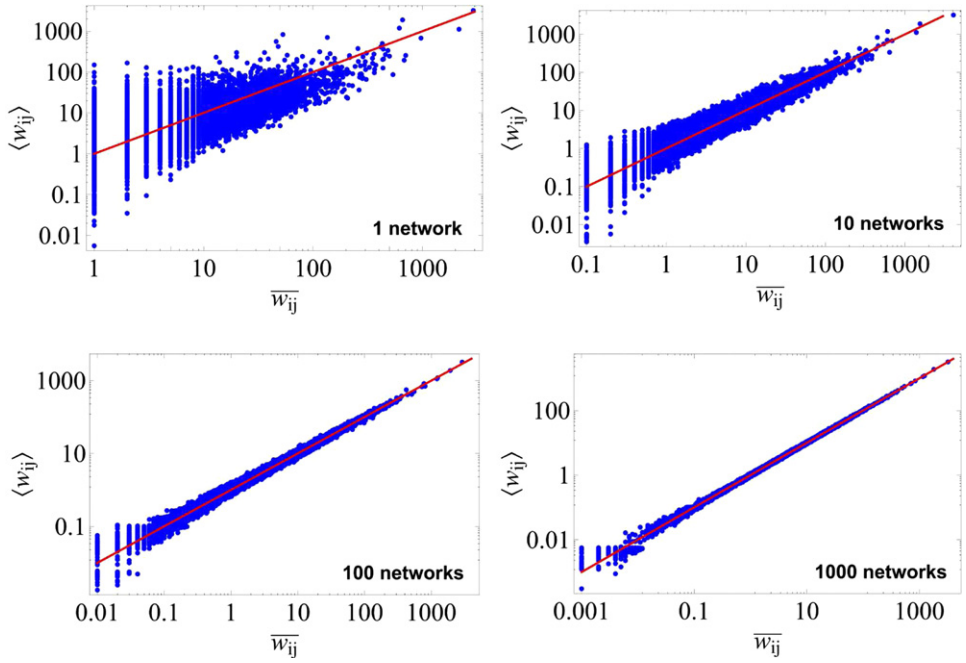

Figure 4. Sampling weighted undirected networks with given strength sequence (undirected weighted configuration model). The example shown is the weighted network of liquidity reserves exchanges between Italian banks in 1999 [41] ($N = 215$). The four panels show the convergence of the sample average $\overline{w_{ij}}$ 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 networks. The identity line is shown in red.

out to be enough to obtain a high level of accuracy for this network. This important check is impossible in microcanonical approaches, where there is no knowledge of the exact value of the expected weights.

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 figure 5 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 figure 5 we show the average nearest neighbor strength (ANNS), $s_i^{nn} = \frac{\sum_j a_{ij} s_j}{k_i}$, 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 of different networks [5, 21–23, 28], 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).

We now come to the analysis of the canonical fluctuations. The ensemble variance of each strength s_i is defined as $\sigma^2[s_i] \equiv \langle s_i^2 \rangle - \langle s_i \rangle^2$, and the independence of pairs of nodes implies

$$\begin{aligned}\sigma^2[s_i] &= \sum_{j \neq i} \sigma^2[w_{ij}] = \sum_{j \neq i} \left(\left\langle w_{ij}^2 \right\rangle - \left\langle w_{ij} \right\rangle^2 \right) \\ &= \sum_{j \neq i} \frac{p_{ij}}{\left(1 - p_{ij}\right)^2} = \sum_{j \neq i} \left\langle w_{ij} \right\rangle \left(1 + \left\langle w_{ij} \right\rangle\right) \\ &= s_i + \sum_{j \neq i} \left\langle w_{ij} \right\rangle^2. \end{aligned} \tag{34}$$

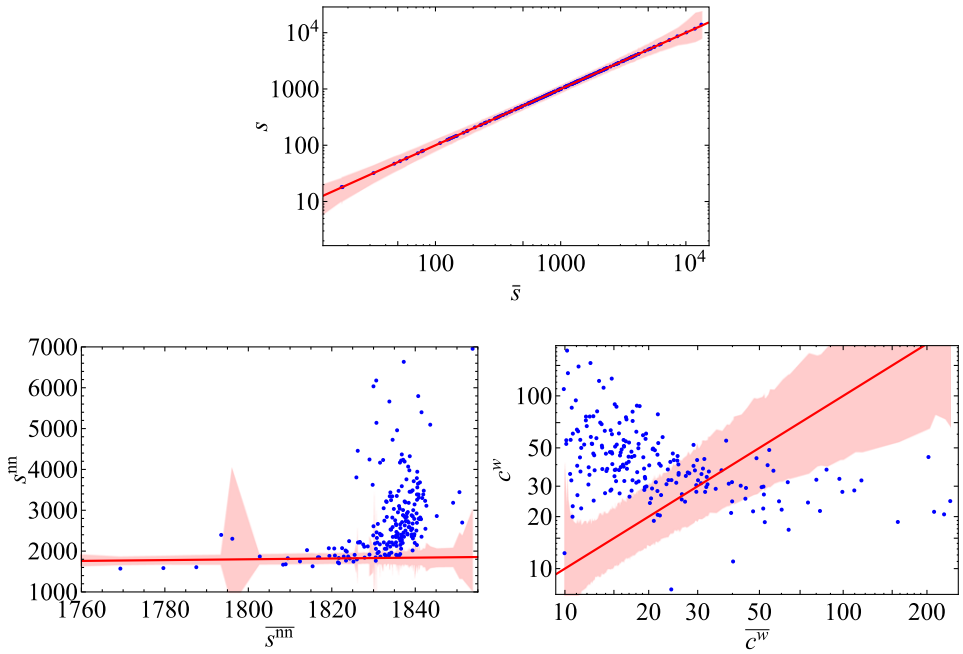

Figure 5. Sampling weighted undirected networks with given strength sequence (undirected weighted configuration model). The example shown is the weighted network of liquidity reserves exchanges between Italian banks in 1999 [41] ($N = 215$). The three panels show, for each node in the network, the comparison 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 networks. The 95% confidence intervals of the distribution of the sampled quantities is shown in pink for each node.

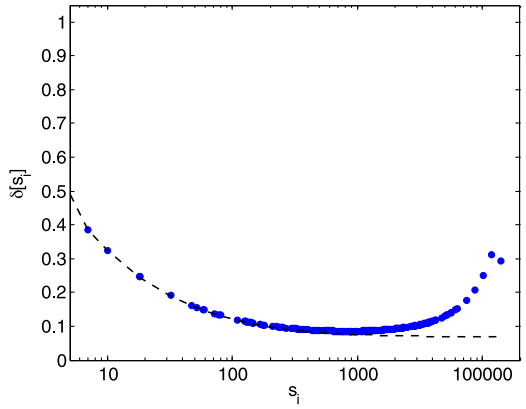

Figure 6. Coefficient of variation $\delta[s_i]$ as a function of the strength s_i for each node of the binary network of liquidity reserves exchanges between Italian banks in 1999 [41] ($N = 215$). The blue points are the exact values in equation (35), while the dashed curve is the lower bound in equation (36). The upper bound exceeds 1 and extends beyond the region shown.

Therefore the relative fluctuations take the form

$$\delta[s_i] \equiv \frac{\sigma[s_i]}{s_i} = \sqrt{\frac{1}{s_i} + \frac{\sum_{j\neq i}\langle w_{ij}\rangle^2}{\left(\sum_{j\neq i}\langle w_{ij}\rangle\right)^2}} \tag{35}$$

for $s_i > 0$. A plot of $\delta[s_i]$ as a function of s_i for the interbank network is shown in figure 6. Unlike in the UBCM, here the relative fluctuations are found to be smaller for intermediate values of the strength.

When comparing equation (35) with equation (11), it is interesting to notice that the term $\sum_{j\neq i}\langle w_{ij}\rangle^2 \Big/ \left(\sum_{j\neq i}\langle w_{ij}\rangle\right)^2$, while still being a participation ratio[7], is now predeced by a *positive* sign. This implies

[7] In this case, the participatio ratio measures the inverse of the effective number of equally important terms in the sum $\sum_{j\neq i}\langle w_{ij}\rangle$. It equals 1 if and only if there is only one nonzero term (complete concentration, which still implies $\langle k_i\rangle = 1$ but *not* $s_i = 1$), while it equals $(N-1)^{-1}$ if and only if there are $N-1$ identical terms (complete homogeneity), i.e. $\langle w_{ij}\rangle = s_i/(N-1)$ for all $j \neq i$.

that the bounds for $\delta[s_i]$ are quite different from those for $\delta[k_i]$ shown in equation (12):

$$\sqrt{\frac{1}{s_i}+\frac{1}{N-1}} \leqslant \delta[s_i] \leqslant \sqrt{\frac{1}{s_i}+1}\,. \tag{36}$$

The allowed region for $\delta[s_i]$ is the one above the dashed line in figure 6, and extends beyond 1. We now find that the realized trend is very close to the *lower* bound for small and intermediate values of the strength (again suggesting that in this regime our maximum-entropy method produces almost maximally homogeneous terms in the sum $\sum_{j\neq i}\langle w_{ij}\rangle$), while it exceeds the lower bound significantly for large values of the strength. In any case, since equation (36) implies that $\delta[s_i]$ cannot vanish for any value of s_i, we find evidence of the fact that for this model the microcanonical and canonical ensembles are *always* not equivalent.

3.5. Weighted directed networks (WDNs) with given in-strength and out-strength sequences

We now consider WDNs, defined by a weight matrix $\mathbf{W}$ which is in general not symmetric. Each node is now characterized by two strengths, the out-strength $s_i^{\rm out}\equiv\sum_j w_{ij}$ and the in-strength $s_i^{\rm in}\equiv\sum_j w_{ji}$. The *directed weighted configuration model* (DWCM), the directed version of the UWCM, enforces the out- and in-strength sequences, $\{s_i^{\rm out}\}_{i=1}^N$ and $\{s_i^{\rm in}\}_{i=1}^N$, of a real-world network $\mathbf{W}^*$ [5, 22, 23]. The model is widely used to detect modules and communities in real WDNs [1].

In its canonical version, the DWCM is still characterized by equation (31) where '$j<i$' is replaced by '$j\neq i$' and now $p_{ij}\equiv x_i y_j$. The $2N$ unknown parameters $\mathbf{x}$ and $\mathbf{y}$ can be fixed by either maximizing the log-likelihood function [5]

$$\begin{aligned}\lambda(\mathbf{x},\mathbf{y}) &\equiv \ln P\left(\mathbf{W}^*\mid \mathbf{x},\mathbf{y}\right)\\ &=\sum_i\left[s_i^{\rm out}\left(\mathbf{W}^*\right)\ln x_i+s_i^{\rm in}\left(\mathbf{W}^*\right)\ln y_i\right]\\ &\quad+\sum_i\sum_{j\neq i}\ln\left(1-x_iy_j\right)\end{aligned} \tag{37}$$

or solving the the $2N$ equations [5]

$$\langle s_i^{\rm out}\rangle=\sum_{j\neq i}\frac{x_iy_j}{1-x_iy_j}=s_i^{\rm out}\left(\mathbf{W}^*\right)\quad\forall i, \tag{38}$$

$$\langle s_i^{\rm in}\rangle=\sum_{j\neq i}\frac{x_jy_i}{1-x_jy_i}=s_i^{\rm in}\left(\mathbf{W}^*\right)\quad\forall i, \tag{39}$$

where in both cases the parameters $\mathbf{x}$ and $\mathbf{y}$ vary in the region defined by $0\leqslant x_iy_j<1$ for all i, j [26].

Once the unknown variables are found, we can implement an efficient and unbiased sampling scheme in the same way as for the UWCM, but now running over each pair of vertices *twice* (i.e. in both directions). One can establish the weight of a link from vertex i to vertex j using the geometric distribution $p_{ij}^w(1-p_{ij})$, and the 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}. Since this directed generalization of the undirected case is straightforward, we do not consider any explicit application. However, we have explicitly included the DWCM model in the code (see appendix).

We now come to the canonical fluctuations. In analogy with equation (34), it is easy to show that the variances of $s_i^{\rm out}$ and $s_i^{\rm in}$ are given by

$$\sigma^2\left[s_i^{\rm out}\right]=\sum_{j\neq i}\left\langle w_{ij}\right\rangle\left(1+\left\langle w_{ij}\right\rangle\right)=s_i^{\rm out}+\sum_{j\neq i}\left\langle w_{ij}\right\rangle^2, \tag{40}$$

$$\sigma^2\left[s_i^{\rm in}\right]=\sum_{j\neq i}\left\langle w_{ji}\right\rangle\left(1+\left\langle w_{ji}\right\rangle\right)=s_i^{\rm in}+\sum_{j\neq i}\left\langle w_{ji}\right\rangle^2. \tag{41}$$

For $s_i^{\rm out}>0$ and $s_i^{\rm in}>0$, the relative fluctuations are

$$\delta\left[s_i^{\rm out}\right]\equiv\frac{\sigma\left[s_i^{\rm out}\right]}{s_i^{\rm out}}=\sqrt{\frac{1}{s_i^{\rm out}}+\frac{\sum_{j\neq i}\left\langle w_{ij}\right\rangle^2}{\left(\sum_{j\neq i}\left\langle w_{ij}\right\rangle\right)^2}}, \tag{42}$$

$$\delta\left[s_i^{\text{in}}\right] \equiv \frac{\sigma\left[s_i^{\text{in}}\right]}{s_i^{\text{in}}} = \sqrt{\frac{1}{s_i^{\text{in}}} + \frac{\sum_{j\neq i}\left\langle w_{ji}\right\rangle^2}{\left(\sum_{j\neq i}\left\langle w_{ji}\right\rangle\right)^2}}. \tag{43}$$

For the bounds of the above quantities, expressions similar to equation (36) apply, suggesting that the microcanonical and canonical versions of this ensemble are also not equivalent.

3.6. WDNs 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 constrain the reciprocity structure in WDNs [26]. The RWCM enforces three strengths for each node: the non-reciprocated incoming strength, $s_i^{\leftarrow} \equiv \sum_j w_{ij}^{\leftarrow}$, the non-reciprocated outgoing strength, $s_i^{\rightarrow} \equiv \sum_j w_{ij}^{\rightarrow}$, and the reciprocated strength, $s_i^{\leftrightarrow} \equiv \sum_j w_{ij}^{\leftrightarrow}$ [26]. Such quantities are defined by means of three pair-specific variables: $w_{ij}^{\leftrightarrow} \equiv \min\left[w_{ij}, w_{ji}\right]$ (reciprocated weight), $w_{ij}^{\rightarrow} \equiv w_{ij} - w_{ij}^{\leftrightarrow}$ and $w_{ij}^{\leftarrow} \equiv w_{ji} - w_{ij}^{\leftrightarrow}$ (non-reciprocated weights).

Despite its complexity, the RWCM is analytically solvable [26] and the graph probability factorizes as:

$$P\left(\mathbf{W}|\mathbf{x},\mathbf{y},\mathbf{z}\right) = \prod_i\prod_{j<i}\left[\frac{\left(x_iy_j\right)^{w_{ij}^{\rightarrow}}\left(x_jy_i\right)^{w_{ij}^{\leftarrow}}\left(z_iz_j\right)^{w_{ij}^{\leftrightarrow}}}{Z_{ij}\left(x_i,x_j,y_i,y_j,z_i,z_j\right)}\right], \tag{44}$$

where $Z_{ij}\left(x_i,x_j,y_i,y_j,z_i,z_j\right) \equiv \frac{\left(1-x_ix_jy_iy_j\right)}{\left(1-x_iy_j\right)\left(1-x_jy_i\right)\left(1-z_iz_j\right)}$ is the node-pair partition function. The $3N$ unknown parameters $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ must be determined either by maximizing the log-likelihood function

$$\begin{aligned}\lambda\left(\mathbf{x},\mathbf{y},\mathbf{z}\right) &\equiv \ln P\left(\mathbf{W}^{\star}\mid\mathbf{x},\mathbf{y},\mathbf{z}\right)\\ &= \sum_i\Big[s_i^{\rightarrow}\left(\mathbf{W}^{\star}\right)\ln x_i + s_i^{\leftarrow}\left(\mathbf{W}^{\star}\right)\ln y_i\\ &\quad + s_i^{\leftrightarrow}\left(\mathbf{W}^{\star}\right)\ln z_i\Big] - \sum_i\sum_{j<i}\ln Z_{ij}\left(x_i,x_j,y_i,y_j,z_i,z_j\right)\end{aligned} \tag{45}$$

or by solving the $3N$ equations:

$$\left\langle s_i^{\rightarrow}\right\rangle = \sum_{j\neq i}\frac{x_iy_j\left(1-x_jy_i\right)}{\left(1-x_iy_j\right)\left(1-x_ix_jy_iy_j\right)} = s_i^{\rightarrow}\left(\mathbf{W}^{\star}\right) \quad \forall i, \tag{46}$$

$$\left\langle s_i^{\leftarrow}\right\rangle = \sum_{j\neq i}\frac{x_jy_i\left(1-x_iy_j\right)}{\left(1-x_jy_i\right)\left(1-x_ix_jy_iy_j\right)} = s_i^{\leftarrow}\left(\mathbf{W}^{\star}\right) \quad \forall i, \tag{47}$$

$$\left\langle s_i^{\leftrightarrow}\right\rangle = \sum_{j\neq i}\frac{z_iz_j}{1-z_iz_j} = s_i^{\leftrightarrow}\left(\mathbf{W}^{\star}\right) \quad \forall i. \tag{48}$$

Here, the parameters $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ vary in the region defined by $0 \leqslant x_iy_j < 1$ and $0 \leqslant z_iz_j < 1$ for all i, j [26].

Equation (44) 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 $\left(w_{ij}^{\leftarrow}, w_{ij}^{\rightarrow}, w_{ij}^{\leftrightarrow}\right)$ is $\left[\frac{\left(x_iy_j\right)^{w_{ij}^{\rightarrow}}\left(x_jy_i\right)^{w_{ij}^{\leftarrow}}\left(z_iz_j\right)^{w_{ij}^{\leftrightarrow}}}{Z_{ij}\left(x_i,x_j,y_i,y_j,z_i,z_j\right)}\right]$ (where, as usual, all the parameters are intended to be the ones maximizing the likelihood). Also, note that $w_{ij}^{\leftarrow}$ and $w_{ij}^{\rightarrow}$ cannot be both nonzero, but they are independent of $w_{ij}^{\leftrightarrow}$ (the joint distribution of these three quantities shown above is not simply a multivariate geometric distribution).

The above observations allow us to define an unbiased sampling scheme, even if more complicated than the ones described so far. For each pair of nodes i, j, we define a procedure in three steps. First, we draw the reciprocal weight $w_{ij}^{\leftrightarrow}$ from the geometric distribution $\left(z_iz_j\right)^{w_{ij}^{\leftrightarrow}}\left(1-z_iz_j\right)$ (or equivalently, from the composition of Bernoulli distributions as discussed for the UWCM). Second, we focus on the *mere existence* of non-reciprocated weights (irrespective of their magnitude). We randomly select one of these three (mutually excluding) events: we establish the absence of any non-reciprocated weight between i and j ($w_{ij}^{\rightarrow}=0, w_{ij}^{\leftarrow}=0$) with probability $\frac{\left(1-x_iy_j\right)\left(1-x_jy_i\right)}{1-x_ix_jy_iy_j}$, we establish the existence of a non-reciprocated weight from i to j ($w_{ij}^{\rightarrow}>0$, $w_{ij}^{\leftarrow}=0$) with probability $\frac{x_iy_j\left(1-x_jy_i\right)}{1-x_ix_jy_iy_j}$, we establish the existence of a non-reciprocated weight from j to i ($w_{ij}^{\rightarrow}=0$,

$w_{ij}^{\leftarrow} > 0$) with probability $\frac{x_j y_i (1 - x_i y_j)}{1 - x_i x_j y_i y_j}$. Third, if a non-reciprocated connection has been established (i.e. if its weight w is positive) we then focus on the value to be assigned to it (i.e. on the extra weight $w - 1$). If $w_{ij}^{\rightarrow} > 0$, we draw the weight $w_{ij}^{\rightarrow}$ from a geometric distribution $(x_i y_j)^{w_{ij}^{\rightarrow} - 1}(1 - x_i y_j)$ (shifted to strictly positive integer values of $w_{ij}^{\rightarrow}$ via the rescaled exponent), while if $w_{ij}^{\leftarrow} > 0$ we draw the weight $w_{ij}^{\leftarrow}$ from the distribution $(x_j y_i)^{w_{ij}^{\leftarrow} - 1}(1 - x_j y_i)$.

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 the microcanonical analogue of this algorithm has not 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).

In this model, the canonical fluctuations are somewhat more compicated than in the previous models. The variances of the constraints are

$$\sigma^2\left[s_i^{\rightarrow}\right] = \sum_{j \neq i} \frac{x_i y_j \left(1 - x_j y_i\right)\left(1 - x_i^2 x_j y_i y_j^2\right)}{\left(1 - x_i y_j\right)^2 \left(1 - x_i x_j y_i y_j\right)^2}, \tag{49}$$

$$\sigma^2\left[s_i^{\leftarrow}\right] = \sum_{j \neq i} \frac{x_j y_i \left(1 - x_i y_j\right)\left(1 - x_i x_j^2 y_i^2 y_j\right)}{\left(1 - x_j y_i\right)^2 \left(1 - x_i x_j y_i y_j\right)^2}, \tag{50}$$

$$\sigma^2\left[s_i^{\leftrightarrow}\right] = \sum_{j \neq i} \frac{z_i z_j}{\left(1 - z_i z_j\right)^2}. \tag{51}$$

While for the variance of the reciprocated weight we can still write $\sigma^2[w_{ij}^{\leftrightarrow}] = \langle w_{ij}^{\leftrightarrow}\rangle(1 + \langle w_{ij}^{\leftrightarrow}\rangle)$ in analogy with the UWCM and DWCM, similar relations do not hold for the non-reciprocated weights. However, since $x_i y_j < 1$ for all i, j, it is easy to show that $\sigma^2[w_{ij}^{\rightarrow}] > \langle w_{ij}^{\rightarrow}\rangle(1 + \langle w_{ij}^{\rightarrow}\rangle)$ and $\sigma^2[w_{ij}^{\leftarrow}] > \langle w_{ij}^{\leftarrow}\rangle(1 + \langle w_{ij}^{\leftarrow}\rangle)$. This still allows us to obtain a lower bound for all quantities as in the other weighted models, by using

$$\sigma^2\left[s_i^{\rightarrow}\right] > \sum_{j \neq i}\left\langle w_{ij}^{\rightarrow}\right\rangle\left(1 + \left\langle w_{ij}^{\rightarrow}\right\rangle\right) = s_i^{\rightarrow} + \sum_{j \neq i}\left\langle w_{ij}^{\rightarrow}\right\rangle^2, \tag{52}$$

$$\sigma^2\left[s_i^{\leftarrow}\right] > \sum_{j \neq i}\left\langle w_{ij}^{\leftarrow}\right\rangle\left(1 + \left\langle w_{ij}^{\leftarrow}\right\rangle\right) = s_i^{\leftarrow} + \sum_{j \neq i}\left\langle w_{ij}^{\leftarrow}\right\rangle^2, \tag{53}$$

$$\sigma^2\left[s_i^{\leftrightarrow}\right] = \sum_{j \neq i}\left\langle w_{ij}^{\leftrightarrow}\right\rangle\left(1 + \left\langle w_{ij}^{\leftrightarrow}\right\rangle\right) = s_i^{\leftrightarrow} + \sum_{j \neq i}\left\langle w_{ij}^{\leftrightarrow}\right\rangle^2. \tag{54}$$

Then, for $s_i^{\rightarrow} > 0, s_i^{\leftarrow} > 0$ and $s_i^{\leftrightarrow} > 0$, we get

$$\delta\left[s_i^{\rightarrow}\right] \equiv \frac{\sigma\left[s_i^{\rightarrow}\right]}{s_i^{\rightarrow}} > \sqrt{\frac{1}{s_i^{\rightarrow}} + \frac{\sum_{j \neq i}\left\langle w_{ij}^{\rightarrow}\right\rangle^2}{\left(\sum_{j \neq i}\left\langle w_{ij}^{\rightarrow}\right\rangle\right)^2}}, \tag{55}$$

$$\delta\left[s_i^{\leftarrow}\right] \equiv \frac{\sigma\left[s_i^{\leftarrow}\right]}{s_i^{\leftarrow}} > \sqrt{\frac{1}{s_i^{\leftarrow}} + \frac{\sum_{j \neq i}\left\langle w_{ij}^{\leftarrow}\right\rangle^2}{\left(\sum_{j \neq i}\left\langle w_{ij}^{\leftarrow}\right\rangle\right)^2}}, \tag{56}$$

$$\delta\left[s_i^{\leftrightarrow}\right] \equiv \frac{\sigma\left[s_i^{\leftrightarrow}\right]}{s_i^{\leftrightarrow}} = \sqrt{\frac{1}{s_i^{\leftrightarrow}} + \frac{\sum_{j \neq i}\left\langle w_{ij}^{\leftrightarrow}\right\rangle^2}{\left(\sum_{j \neq i}\left\langle w_{ij}^{\leftrightarrow}\right\rangle\right)^2}}. \tag{57}$$

Therefore, for all three quantities, a lower bound of the form $\delta[s_i] \geqslant \sqrt{\frac{1}{s_i} + \frac{1}{N-1}}$ still applies, as in equation (36). This suggests that, for this model as well, the microcanonical and canonical ensembles are not equivalent.

3.7. WUNs with given strengths and degrees

We finally consider a 'mixed' null model of weighted networks with both binary (degree sequence $\{k_i\}_{i=1}^N$) and weighted (strength sequence $\{s_i\}_{i=1}^N$) constraints. We only consider undirected networks for simplicity, but the

extension to the directed case is straightforward. The ensemble of WUNs with given strengths and degrees has been recently introduced as the *(undirected) enhanced configuration model* (UECM) [28, 29].

This model, which is based on analytical results derived in [27], is of great importance for the problem of *network reconstruction* from partial node-specific information [28]. As we have also illustrated in figure 5, 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 [22, 23]. 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 [28, 29].

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 [27], and reads

$$P(\mathbf{W} \mid \mathbf{x}, \mathbf{y}) = \prod_i \prod_{j<i} \left[\frac{(x_i x_j)^{\Theta(w_{ij})} \left(y_i y_j\right)^{w_{ij}} \left(1 - y_i y_j\right)}{1 - y_i y_j + x_i x_j y_i y_j} \right]. \tag{58}$$

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

$$\begin{aligned} \lambda(\mathbf{x}, \mathbf{y}) &\equiv \ln P\left(\mathbf{W}^{\star} \middle| \mathbf{x}, \mathbf{y}\right) \\ &= \sum_i \left[k_i\left(\mathbf{W}^{\star}\right) \ln x_i + s_i\left(\mathbf{W}^{\star}\right) \ln y_i \right] \\ &\quad + \sum_i \sum_{j<i} \ln \frac{1 - y_i y_j}{\left(1 - y_i y_j + x_i x_j y_i y_j\right)} \end{aligned} \tag{59}$$

or by solving the $2N$ equations [28]:

$$\langle k_i \rangle = \sum_{j \neq i} p_{ij} = k_i\left(\mathbf{W}^{\star}\right) \quad \forall i, \tag{60}$$

$$\langle s_i \rangle = \sum_{j \neq i} \frac{p_{ij}}{1 - y_i y_j} = s_i\left(\mathbf{W}^{\star}\right) \quad \forall i, \tag{61}$$

where $p_{ij} \equiv \frac{x_i x_j y_i y_j}{1 - y_i y_j + x_i x_j y_i y_j}$. Here, the parameters $\mathbf{x}$ and $\mathbf{y}$ vary in the region $x_i \geqslant 0$ for all i and $0 \leqslant y_i y_j < 1$ for all i, j respectively [28].

In order to define an unbiased sampling scheme, we note that equation (58) 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. The probability to generate a link of weight w between the nodes i and j is

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

The above expression identifies two key steps: 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 figure 7 we provide an application of this method to the world trade web [21, 22, 42]. We show the convergence of the sample averages ($\overline{a_{ij}}$ and $\overline{w_{ij}}$) 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 figure 8 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 figure 8) are well replicated by the model. These results are consistent with what is obtained analytically by using the same canonical null model on the same network [29]. Moreover, in this case we can calculate confidence intervals besides expected values (for instance, in figure 8 we can clearly identify

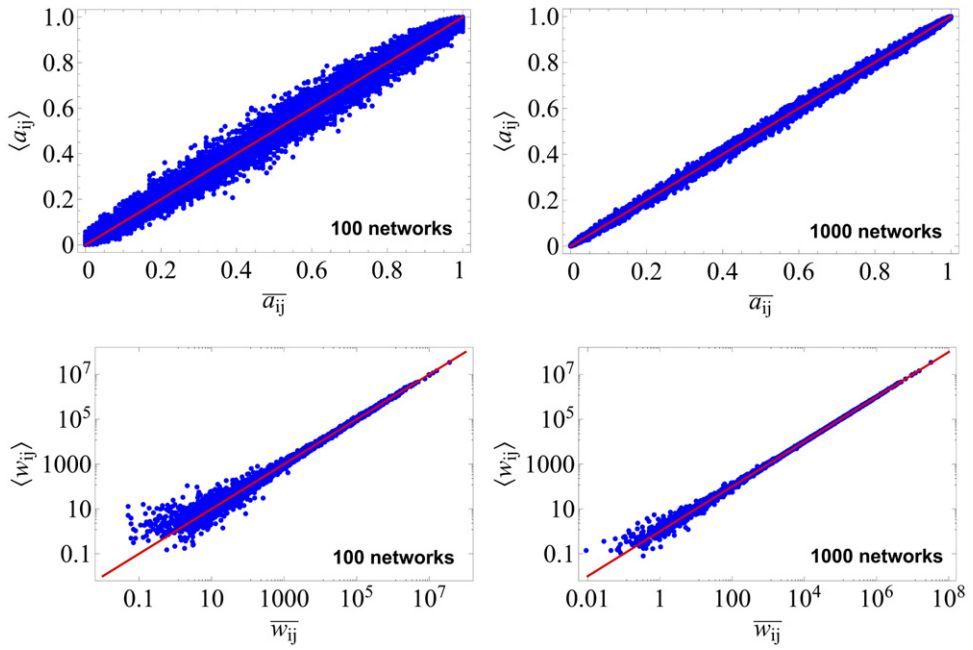

Figure 7. Sampling weighted undirected networks with given degree and strength sequences (undirected enhanced configuration model). The example shown is the weighted world trade web ($N = 162$) [42]. The top panels show the convergence of the sample average $\overline{a_{ij}}$ of each entry of the adjacency matrix to its exact canonical expectation $\langle p_{ij} \rangle$, for 100 (left) and 1000 (right) sampled matrices. The bottom 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 100 (left) and 1000 (right) sampled networks. The identity line is shown in red.

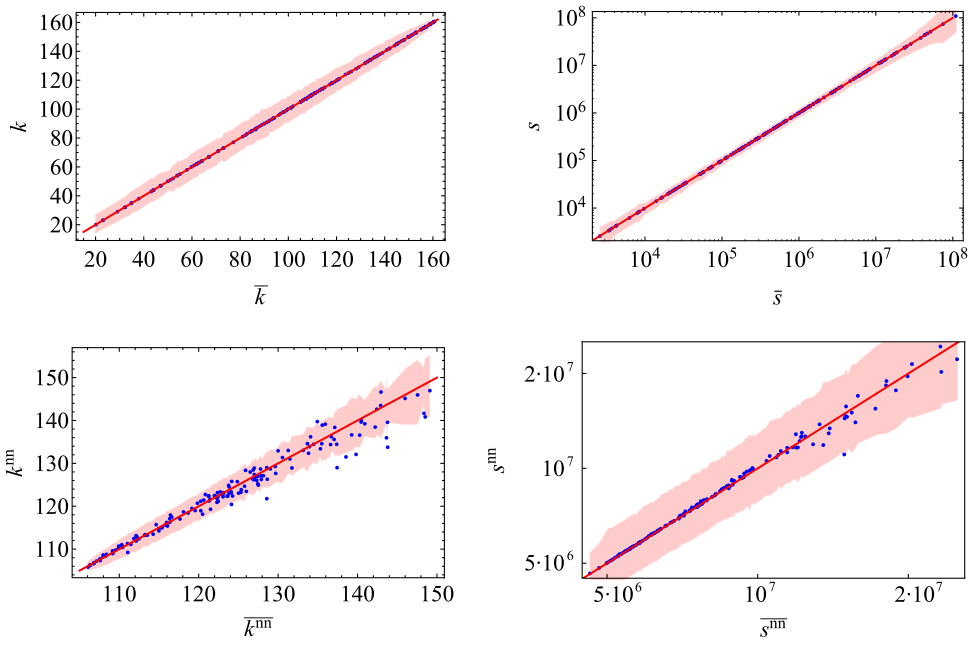

Figure 8. Sampling weighted undirected networks with given degree and strength sequences (undirected enhanced configuration model). The example shown is the weighted world trade web ($N = 162$) [42]. The four panels show, for each node in the network, the comparison between the observed value and the 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 networks. The 95% confidence intervals of the distribution of the sampled quantities is shown in pink for each node.

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 [28].

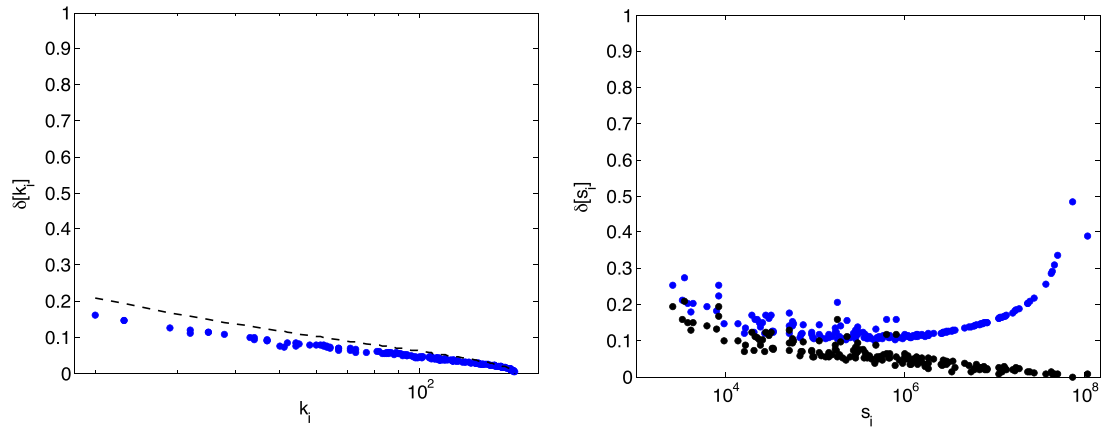

Figure 9. Coefficients of variation $\delta[k_i]$ (left) and $\delta[s_i]$ (right), plotted as a function of the degree k_i and the strength s_i respectively, for each node of the weighted world trade web ($N = 162$) [42]. The blue points indicate the exact values, while the dashed curve (left) and black points (right) indicate in both cases the value $\sqrt{\frac{1}{k_i} - \frac{1}{N-1}}$, which is a strict upper bound for $\delta[k_i]$ and a reference value, typically exceeded, for $\delta[s_i]$.

The canonical fluctuations in this ensemble can be also calculated analytically. For the variance of the degrees, we can still exploit the expression $\sigma^2[a_{ij}] = p_{ij}(1 - p_{ij})$. For the variance of the strengths, we can use the definition $\sigma^2[w_{ij}] = \langle w_{ij}^2 \rangle - \langle w_{ij} \rangle^2$, which however leads to a more complicated expression in this case. Using the relation $\langle w_{ij} \rangle = p_{ij}/(1 - y_i y_j)$, the end result can be expressed as follows:

$$\sigma^2\left[k_i\right] = \sum_{j \neq i} p_{ij}\left(1 - p_{ij}\right), \tag{62}$$

$$\sigma^2[s_i] = \sum_{j \neq i} \frac{p_{ij}\left(1 + y_i y_j - p_{ij}\right)}{\left(1 - y_i y_j\right)^2} = \sum_{j \neq i} \left\langle w_{ij} \right\rangle \left(\frac{1 + y_i y_j}{1 - y_i y_j} - \left\langle w_{ij} \right\rangle \right). \tag{63}$$

Since $(1 + y_i y_j)/(1 - y_i y_j) \geqslant 1$, we can obtain the following relations for the relative fluctuations:

$$\delta\left[k_i\right] \equiv \frac{\sigma\left[k_i\right]}{k_i} = \sqrt{\frac{1}{k_i} - \frac{\sum_{j \neq i} p_{ij}^2}{\left(\sum_{j \neq i} p_{ij}\right)^2}}, \tag{64}$$

$$\delta[s_i] \equiv \frac{\sigma[s_i]}{s_i} \geqslant \sqrt{\frac{1}{s_i} - \frac{\sum_{j \neq i} \left\langle w_{ij} \right\rangle^2}{\left(\sum_{j \neq i} \left\langle w_{ij} \right\rangle\right)^2}}. \tag{65}$$

So $\delta[k_i]$ retains the same expression valid for the UBCM and all the other ensembles of binary graphs considered previously, which in turn leads to the same bounds as in equation (12). This is confirmed in figure 9. By contrast, $\delta[s_i]$ has a more complicated form, which differs from that valid for the UWCM and does not lead to simple expressions for the upper and lower bounds. Also note the presence of a *minus* sign in equation (65). What can be concluded relatively easily is that, in the ideal limit $y_i \to 0$ (corresponding to very small values of s_i), we have $\langle w_{ij} \rangle \to p_{ij}$ which implies $s_i \to k_i$ and $\delta[s_i] \to \delta[k_i]$. This means that, in this extreme (and typically unrealized) limit, $\delta[s_i]$ behaves as $\delta[k_i]$, so it has the same upper bound $\sqrt{\frac{1}{k_i} - \frac{1}{N-1}}$. However, since y_i is typically larger than zero, this bound is systematically exceeded, especially for large values of s_i. This is also confirmed in figure 9. As in the other models, the non-vanishing of the fluctuations suggests that the microcanonical and canonical ensembles are not equivalent.

4. Microcanonical considerations

In this section we come back to the difference between canonical and microcanonical approaches to the sampling of network ensembles and discuss how, at least in principle, our method can be turned into an unbiased microcanonical one.

We provided evidence that, for all the models considered in this paper, the canonical and microcanonical ensembles are *not* equivalent (see also [40] for a recent mathematical proof of nonequivalence for the UBCM). This result implies that choosing between microcanonical and canonical approaches to the sampling of network ensembles is not only a matter of (computational) convenience, but also a theoretical issue that should be addressed more formally.

To this end, we recall that microcanonical ensembles describe isolated systems that do not interact with an external 'heat bath' or 'reservoir'. In ordinary statistical physics, this means that there is no exchange of energy with the external world. In our setting, this means that microcanonical approaches do not contemplate the possibility that the network interacts with some external 'source of error', i.e. that the value of the enforced constraints might be affected by errors or missing entries in the data. When present, such errors (e.g. a missing link, implying a wrong value of the degree of two nodes) are propagated to the entire collection of randomized networks, with the result that the 'correct' network is not included in the microcanonical collection of graphs on which inference is being made.

By contrast, besides being unbiased and mathematically tractable, our canonical approach is also the most appropriate choice if one wants to account for possible errors in the data, since canonical ensembles appropriately describe systems in contact with an external reservoir (source of errors) affecting the value of the constraints. While in presence of even small errors microcanonical methods assign zero probability to the 'uncorrupted' configuration and to all the configurations with the same value of the constraints, our method assigns these configurations a probability which is only slightly smaller than the (maximum) probability assigned to the set of configurations consistent with the observed ('corrupted') one. These considerations suggest that, given its simplicity, elegance, and ability to deal with potential errors in the data, 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 unbiased microcanonical expectations, if theoretical considerations suggest that the microcanonical ensemble is more appropriate in some specific cases. 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. In the rest of this section, we discuss some advantages and limitations of this approach.

As a guiding principle, one should bear in mind that, to be feasible, a microcanonical sampling based on our method requires that the number R_c of canonical realizations to be sampled (among which only a number $R_m < R_c$ 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 $\mathbf{G}$ denote a generic graph (either binary or weighted) in the canonical ensemble, and $\mathbf{G}^\star$ the observed network that needs to be randomized. Let $\mathbf{h}$ formally denote a generic vector of chosen constraints, and let $\mathbf{h}^\star \equiv \mathbf{h}(\mathbf{G}^\star)$ indicate the observed values of such constraints. Similarly, let θ denote the generic vector of Lagrange multipliers (hidden variables) associated with $\mathbf{h}$, and let $\theta^\star$ indicate the vector of their likelihood-maximizing values enforcing the constraints $\mathbf{h}^\star$. On average, out of R_c canonical realizations, we will be left with a number

$$R_m = Q\left(\mathbf{h}^\star\right) R_c \tag{66}$$

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

$$Q\left(\mathbf{h}^\star\right) = \sum_{\mathbf{G}/\mathbf{h}(\mathbf{G})=\mathbf{h}^\star} P\left(\mathbf{G} \mid \boldsymbol{\theta}^\star\right) = N_m\left(\mathbf{h}^\star\right) P\left(\mathbf{G}^\star \mid \boldsymbol{\theta}^\star\right), \tag{67}$$

where $P(\mathbf{G}|\boldsymbol{\theta}^{\star})$ is the probability of graph $\mathbf{G}$ in the canonical ensemble, and $N_m(\mathbf{h}^{\star})$ is the number of microcanonical networks matching the constraints $\mathbf{h}^{\star}$ exactly (i.e. the number of graphs with given $\mathbf{h}^{\star}$). Inserting equation (67) into equation (66) and inverting, we find that the value of R_c required to distill R_m microcanonical graphs is

$$R_c = \frac{R_m}{N_m\left(\mathbf{h}^{\star}\right)P\left(\mathbf{G}^{\star}\,|\boldsymbol{\theta}^{\star}\right)}. \tag{68}$$

Note that $P(\mathbf{G}^{\star}\mid\boldsymbol{\theta}^{\star})$ is nothing but the maximized likelihood of the observed network, which is automatically measured in our method. This is typically an extremely small number: for the networks in our analysis, it ranges between 3.8×10^{-36468} (world trade web) and 4.9×10^{-3499} (binary interbank network). On the other hand, the number $N_m(\mathbf{h}^{\star})$ 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.

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 $N_m(\mathbf{h}^{\star})$ microcanonical graphs. In this case, the knowledge of $N_m(\mathbf{h}^{\star})$ becomes unnecessary: from the definition of f_m we get

$$f_m \equiv \frac{R_m}{N_m\left(\mathbf{h}^{\star}\right)} = \frac{Q\left(\mathbf{h}^{\star}\right)R_c}{N_m\left(\mathbf{h}^{\star}\right)} = P\left(\mathbf{G}^{\star}\mid\boldsymbol{\theta}^{\star}\right)R_c. \tag{69}$$

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

$$R_c = \frac{f_m}{P\left(\mathbf{G}^{\star}\,|\boldsymbol{\theta}^{\star}\right)} \tag{70}$$

of canonical realizations and discard all the non-microcanonical ones. This number can be extremely large, since $P(\mathbf{G}^{\star}\mid\boldsymbol{\theta}^{\star})$ 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

$$f_c \equiv \frac{R_c}{N_c\left(\mathbf{h}^{\star}\right)} \tag{71}$$

of *canonical* configurations sampled by R_c realizations, where $N_c(\mathbf{h}^{\star}) \gg N_m(\mathbf{h}^{\star})$ 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 $f_c = 1000/2^{N(N-1)/2} \approx 1.4\times10^{-6920}$. We might therefore be tempted to choose the same small value also for f_m, and find the required number R_c from equation (70). However, the result is a value $R_c \ll 1$ (in the mentioned example, $R_c = 2.8\times10^{-3422}$), which clearly indicates that setting $f_m \equiv f_c$ (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 $R_c \gg 1$ 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 $P(\mathbf{G}^{\star}|\boldsymbol{\theta}^{\star})$, being obtained with the introduction of the constraints $\mathbf{h}^{\star}$, is necessarily much larger than the completely uniform probability $1/N_c(\mathbf{h}^{\star})$ 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

$$f_m = P\left(\mathbf{G}^{\star}\mid\boldsymbol{\theta}^{\star}\right)R_c \gg \frac{R_c}{N_c\left(\mathbf{h}^{\star}\right)} = f_c. \tag{72}$$

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 still 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.

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 $\boldsymbol{\theta}^{\star}$, 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 strategy would be that of exploiting our analysis of the canonical fluctuations 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.

5. 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 general 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 believed 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 exact expressions to correctly sample a number of canonical ensembles, from the standard case of binary graphs with given degree sequence to the more challenging models of directed and weighted graphs with given reciprocity structure or joint strength-degree sequence. Moreover, we have provided evidence that microcanonical and canonical ensembles of graphs with local constraints are not equivalent, and suggested that canonical ones can account for possible errors or missing entries in the data, while microcanonical ones do not.

Our algorithms are unbiased and efficient, as 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. 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.

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). TS acknowledges support from the Italian PNR project CRISIS-Lab.

Appendix: The 'Max & Sam' code

An algorithm has been coded in various ways [43–45] in order to implement our sampling procedure for all the seven null models described in section 3. In what follows, we describe the Matlab implementation [43]. A more detailed explanation accompanies the code in the form of a 'Read_me' file [43]. Here we briefly mention the main features.

The code can be implemented by typing a command having the typical form of a Matlab function, taking a number of different parameters as input. The output of the algorithm is the numerical value of the *hidden variables*, i.e. the vectors $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ (where applicable) maximizing the likelihood of the desired null model (see section 3), 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 section 3.

The command to be typed is the following (more details can be found in the 'Read_me' file [43]):

```
output = 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 $\mathbf{A}^{\star}$ (see section 3.1);
- **DBCM** for the directed binary configuration model, preserving the in- and out-degree sequences ($\{k_i^{\rm in}\}_{i=1}^{N}$ and $\{k_i^{\rm out}\}_{i=1}^{N}$) of a directed binary network $\mathbf{A}^{\star}$ (see section 3.2);
- **RBCM** for the reciprocal binary configuration model, preserving the reciprocated, incoming non-reciprocated and outgoing non-reciprocated degree sequences ($\{k_i^{\leftrightarrow}\}_{i=1}^{N}$, $\{k_i^{\leftarrow}\}_{i=1}^{N}$ and $\{k_i^{\rightarrow}\}_{i=1}^{N}$) of a directed binary network $\mathbf{A}^{\star}$ (see section 3.3);
- **UWCM** for the undirected weighted configuration model, preserving the strength sequence ($\{s_i\}_{i=1}^{N}$) of an undirected weighted network $\mathbf{W}^{\star}$ (see section 3.4);
- **DWCM** for the directed weighted configuration model, preserving the in- and out-strength sequences ($\{s_i^{\rm in}\}_{i=1}^{N}$ and $\{s_i^{\rm out}\}_{i=1}^{N}$) of a directed weighted network $\mathbf{W}^{\star}$ (see section 3.5);
- **RWCM** for the reciprocal weighted configuration model, preserving the the reciprocated, incoming non-reciprocated and outgoing non-reciprocated strength sequences ($\{s_i^{\leftrightarrow}\}_{i=1}^{N}$, $\{s_i^{\leftarrow}\}_{i=1}^{N}$ and $\{s_i^{\rightarrow}\}_{i=1}^{N}$) of a directed weighted network $\mathbf{W}^{\star}$ (see section 3.6);
- **UECM** for the undirected enhnaced 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 $\mathbf{W}^{\star}$ (see section 3.7).

The second, third and fourth parameters (`Matrix`, `Par` and `List` respectively) specify the format of the input data (i.e. of $\mathbf{A}^{\star}$ or $\mathbf{W}^{\star}$). 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 [28] 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 code 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 code 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 code 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 [28] 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 code 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 `x0new` 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).

References

[1] Newman M E J 2010 *Networks: An Introduction* (Oxford: Oxford University Press)
[2] Colizza V, Barrat A, Barthlemy M and Vespignani A 2006 *Proc. Natl Acad. Sci.* **103** 2015–20
[3] Squartini T, van Lelyveld I and Garlaschelli D 2013 *Sci. Rep.* **3** 3357
[4] Barrat A, Barthlemy M and Vespignani A 2008 *Dynamical Processes on Complex Networks* (Cambridge: Cambridge University Press)
[5] Squartini T and Garlaschelli D 2011 *New. J. Phys.* **13** 083001
[6] Milo R, Shen-Orr S, Itzkovitz S, Kashtan N, Chklovskii D and Alon U 2002 *Science* **298** 824–7
[7] Fortunato S 2010 *Phys. Rep.* **486** 75–174
[8] Bollobás B 1980 *Eur. J. Comb.* **1** 311–6
[9] Molloy M and Reed B 1995 *Random Struct. Algorithms* **6** 161–80
[10] Newman M E J, Strogatz S H and Watts D J 2001 *Phys. Rev.* E **64** 026118
[11] Maslov S and Sneppen K 2002 *Science* **296** 910
[12] Coolen A C C, De Martino A and Annibale A 2009 *J. Stat. Phys.* B **136** 1035–67
[13] Roberts E S and Coolen A C C 2012 *Phys. Rev.* E **85** 046103
[14] Artzy-Randrup Y and Stone L 2005 *Phys. Rev.* E **72** 056708
[15] Del Genio C I, Kim H, Toroczkai Z and Bassler K E 2010 *PLoS One* **5** e10012
[16] Kim H, Del Genio C I, Bassler K E and Toroczkai Z 2012 *New J. Phys.* **14** 023012
[17] Blitzstein J and Diaconis P 2011 *Internet Math.* **6** 489–522
[18] Park J and Newman M E J 2004 *Phys. Rev.* E **70** 066117
[19] Bianconi G 2007 *Europhys. Lett.* **81** 28005
[20] Fronczak A, Fronczak P and Holyst J A 2006 *Phys. Rev.* E **73** 016108
[21] Squartini T, Fagiolo G and Garlaschelli D 2011 *Phys. Rev.* E **84** 046117
[22] Squartini T, Fagiolo G and Garlaschelli D 2011 *Phys. Rev.* E **84** 046118
[23] Fagiolo G, Squartini T and Garlaschelli D 2013 *J. Econ. Interact. Coord.* **8** 75–107
[24] Garlaschelli D and Loffredo M I 2006 *Phys. Rev.* E **73** 015101(R)
[25] Squartini T and Garlaschelli D 2012 *Lec. Notes Comput. Sci.* **7166** 2435
[26] Squartini T, Picciolo F, Ruzzenenti F and Garlaschelli D 2013 *Sci. Rep.* **3** 2729
[27] Garlaschelli D and Loffredo M I 2009 *Phys. Rev. Lett.* **102** 038701
[28] Mastrandrea R, Squartini T, Fagiolo G and Garlaschelli D 2014 *New J. Phys.* **16** 043022
[29] Mastrandrea R, Squartini T, Fagiolo G and Garlaschelli D 2014 *Phys. Rev.* E **90** 062804
[30] Milo R, Kashtan N, Itzkovitz S, Newman M E J and Alon U 2003 arXiv:0312028
[31] Erdős P and Gallai T 1960 *Mat Lapok* **11** 264–74
[32] Chung F and Lu L 2002 *Proc. Natl Acad. Sci.* **99** 15879–82
[33] Chung F and Lu L 2002 *Ann. Comb.* **6** 125–45
[34] Boguña M, Pastor-Satorras R and Vespignani A 2004 *Eur. Phys. J. B-Condens. Matter Complex Syst.* **38** 205–9
[35] Catanzaro M, Boguña M and Pastor-Satorras R 2005 *Phys. Rev.* E **71** 027103
[36] Garlaschelli D 2009 *New J. Phys.* **11** 073005
[37] Bianconi G, Coolen A C C and Perez Vicente C J 2008 *Physical Review* E **78** 016114
[38] Anand K and Bianconi G 2010 *Phys. Rev.* E **82** 011116
[39] Anand K and Bianconi G 2009 *Physical Review* E **80** 045102
[40] Squartini T, De Mol J, Den Hollander F and Garlaschelli D 2015 arXiv:1501.00388
[41] De Masi G, Iori G and Caldarelli G 2006 *Phys. Rev.* E **74** 066112
[42] UNCOMTRADE database: http://comtrade.un.org/
[43] http://mathworks.it/matlabcentral/fileexchange/46912-max-sam-package-zip
[44] https://drive.google.com/open?id=0B_rBKSwFTur3M0tvd0w4dW45aE0&authuser=0
[45] http://lorentz.leidenuniv.nl/~garlaschelli/