Casiraghi *Applied Network Science* (2019) 4:123
https://doi.org/10.1007/s41109-019-0241-1

Applied Network Science

RESEARCH Open Access

The block-constrained configuration model

Giona Casiraghi

Correspondence:
gcasiraghi@ethz.ch
Chair of Systems Design, ETH Zürich, Weinbergstrasse 56/58, 8092 Zürich, Switzerland

Abstract

We provide a novel family of generative block-models for random graphs that naturally incorporates degree distributions: the block-constrained configuration model. Block-constrained configuration models build on the generalized hypergeometric ensemble of random graphs and extend the well-known configuration model by enforcing block-constraints on the edge-generating process. The resulting models are practical to fit even to large networks. These models provide a new, flexible tool for the study of community structure and for network science in general, where modeling networks with heterogeneous degree distributions is of central importance.

Keywords: Block model, Community structure, Random graphs, Configuration model, Network analysis, gHypEG

Introduction

Stochastic block-models (SBMs) are random models for graphs characterized by group, communities, or block structures. They are a generalization of the classical $G(n,p)$ Erdős-Rènyi model (1959), where vertices are separated into B different blocks, and different probabilities to create edges are then assigned to each block. This way, higher probabilities correspond to more densely connected groups of vertices, capturing the structure of clustered graphs (Fienberg et al. 1985; Holland et al. 1983; Peixoto 2012).

SBMs are specified by defining a $B \times B$ block-matrix of probabilities $\mathbf{B}$ such that each of its elements $\omega_{b_i b_j}$ is the probability of observing an edge between vertices i and j, where b_i denotes the block to which vertex i belongs. Most commonly, block-matrices are used to encode community structures. This is achieved by defining a diagonal block-matrix, with the inclusion of small off-diagonal elements.

Thanks to its simple formulation, the edge generating process of the standard SBM can retain the block structure of the graph that needs to be modeled (Karrer and Newman 2011). However, it fails to reproduce empirical degree sequences. The reason for this is that in the $G(n,p)$ model and its extensions, edges are sampled independently from each other with fixed probabilities, generating homogeneous degree-sequences across blocks. This issue impairs the applicability of the standard SBM to most real-world graphs. Because of the lack of control on the degree distributions generated by the model, SBMs are not able to reproduce the complex structures of empirical graphs, resulting in poorly fitted models (Karrer and Newman 2011).

Different strategies have been formulated to overcome this issue. Among others, one approach is that of using exponential random graph models (Krivitsky 2012). These models are very flexible in terms of the kind of patterns they can incorporate. However, as soon as their complexity increases, they lose the analytical tractability that characterizes the standard SBM. This is due to the need for computing the partition function that defines the underlying probability distribution (Park and Newman 2004). Another, more prominent, approach taken to address the issue of uniform degree-sequences in SBMs are degree-corrected block models (DC-SBM) (e.g. Peixoto (2014); Newman and Peixoto (2015); Karrer and Newman (2011); Peixoto (2015)). Degree-corrected block-models address this problem by extending standard SBMs with degree corrections, which serve the purpose of enforcing a given expected degree-sequence within the block structures. Moreover, this is achieved without hampering the simplicity of the standard SBM. For this reason, DC-SBMs are widely used for community detection tasks (Newman and Reinert 2016; Peixoto 2015). Recently, they have further been extended to a Bayesian framework, allowing non-parametric model estimation (Peixoto 2017; Peixoto 2018).

One of the main assumptions of G(n,p) models, SBMs, and DC-SBMs as well, is that the probability of creating edges for each pair of vertices are independent of each others (Karrer and Newman 2011). While such a modeling assumption allows defining distributions whose parameters are in general easy to estimate, for many real-world graphs, this is a strong assumption that should be verified, and which is possibly unrealistic (Squartini et al. 2015). Many social phenomena studied through empirical graphs, such as triadic closure (Granovetter 1973), or balance theory (Newcomb and Heider 1958), are based on the assumption that edges between vertices are not independent. Similarly, for graphs arising from the observation of constrained systems, like financial and economic networks, it is unreasonable to assume that edge probabilities are independent of each other. This is because the observed edges in the graph, which are the representation of interactions between actors in a system, are driven by optimization processes characterized by limited resources and budget constraints, which introduce correlations among different edge probabilities (Caldarelli et al. 2013; Nanumyan et al. 2015).

Moreover, one of the consequences of the assumption of independence of edge probabilities is the fact that the total number of edges of the modelled graph is preserved only in expectation. In the case of SBMs and DC-SBMs, the total number of edges is assumed to follow a Poisson distribution. For a Poisson process to be the appropriate model for an empirical graph, the underlying edge generating process needs to meet the following conditions (Consul and Jain 1973): (i) the observation of one edge should not affect the probability that a second edge will be observed, i.e., edges occur independently; (ii) the rate at which edges are observed has to be constant; (iii) two edges cannot be observed at precisely the same instant. However, it is often hard to evaluate whether these conditions are verified because the edge generating process may not be known, or these conditions are not met altogether.

Melnik et al. (2014) have proposed an alternative approach to the problem of preserving degree distributions and the independence of edges. Such an approach is a generalisation of the configuration model that allows constructing modular random graphs, characterised by heterogeneous degree-degree correlations between each block. The model, in particular, relies on specifying different values $P_{k,k'}^{b_i,b_{i'}}$ for the probability that a randomly chosen edge connects a degree-k node from block b_i to a degree-k' node from block

$b_{i'}$. The so-called $P^{i,i'}_{k,k'}$-model, though, only considers *unweighted* and *undirected* graphs (Melnik et al. 2014).

Similarly to the approach discussed in Melnik et al. (2014), we address the problem of incorporating degree distributions generalising the configuration model. Doing so, we propose a family of block-models that preserves the number of edges exactly, instead of in expectation. This circumvents the issue of assuming a given model for the number of edges in the graph, treating it merely as an observed datum. The configuration model of random graphs (Chung and Lu 2002a; 2002b; Bender and Canfield 1978; Molloy and Reed 1995) is, in fact, the simplest random model that can reproduce heterogeneous degree distributions given a fixed number of edges. It achieves this by randomly rewiring edges between vertices and thus preserving the degree-sequence of the original graph. Doing so, it keeps the number of edges in the graph fixed.

Differently from what proposed by Melnik et al. (2014), though, we extend the standard configuration model to reproduce arbitrary block structures by introducing block constraints on its rewiring process by means of the formalism provided by the generalised hypergeometric ensemble of random graphs. While this approach is not as general as the one proposed by Melnik et al. (2014) in terms of how degree-degree correlations can be incorporated, it allows us to deal with *multi-edge, directed graphs*. We refer to the resulting model as *block-constrained configuration model* (BCCM). Significant advantages of our approach are (i) the natural degree-correction provided by BCCM, and (ii) the preservation of the exact number of edges.

Generalised hypergeometric ensembles of random graphs (gHypEG)

Our approach builds on the generalized hypergeometric ensemble of random graphs (gHypEG) (Casiraghi et al. 2016; Casiraghi and Nanumyan 2018). This class of models extends the configuration model (CM) (Molloy and Reed 1995; 1998) by encoding complex topological patterns, while at the same time preserving degree distributions. Block constraints fall into the larger class of patterns that can be encoded by means of gHypEG. For this reason, before introducing the formulation of the block-constrained configuration model, we provide a brief overview of gHypEG. More details, together with a more formal presentation, are given in Casiraghi et al. (2016); Casiraghi and Nanumyan (2018).

In the configuration model of random graphs, the probability of connecting two vertices depends only on their (out- and in-) degrees. In its most common formulation, the configuration model assigns to each vertex as many out-stubs (or half-edges) as its out-degree, and as many in-stubs as its in-degree. It then proceeds connecting random pairs of vertices joining out- and in-stubs. This is done by sampling uniformly at random one out- and one in-stub from the pool of all out- and in-stubs respectively and then connecting them, until all stubs are connected. The left side of Fig. 1 illustrates the case from the perspective of a vertex A. The probability of connecting vertex A with one of the vertices B, C or D depends only on the abundance of stubs, and hence on the in-degree of the vertices themselves. The higher the in-degree, the higher the number of in-stubs of the vertex. Hence, the higher the probability to randomly sample a stub belonging to the vertex.

Generalized hypergeometric ensembles of random graphs provide an expression for the probability distribution underlying this process, where the degrees of the vertices are preserved in expectations. This result is achieved by mapping the process described above

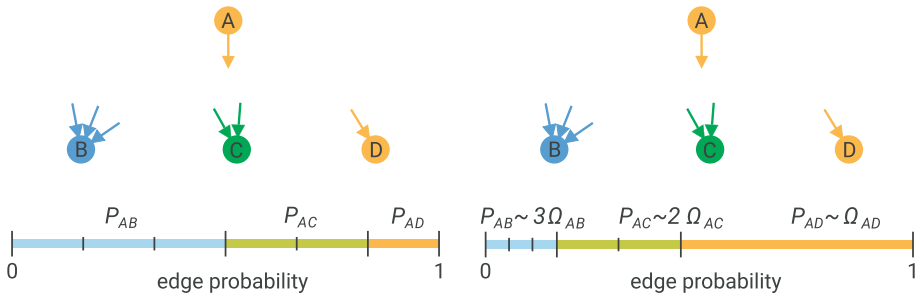

Fig. 1 Probabilities of connecting different stubs. Graphical illustration of the probability of connecting two vertices as a function of degrees (left figure), and degree and propensities (right figure)

to an urn problem. Edges are represented by balls in an urn, and sampling from the configuration model is described by sampling balls (i.e., edges) from an urn appropriately constructed. For each pair of vertices (i, j), we can denote with k_i^{out} and k_j^{in} their respective out- and in-degrees. The number of combinations of out-stubs of i with in-stubs of j which could be connected to create an edge is then given by $k_i^{\text{out}} k_j^{\text{in}}$. To map this process to an urn, for each dyad (i, j) we should place exactly $k_i^{\text{out}} k_j^{\text{in}}$ balls of a given colour in the urn (Casiraghi and Nanumyan 2018). The process of sampling m edges from the configuration model is hence described by sampling m balls from this urn, and the probability distribution of observing a graph $\mathcal{G}$ under the model is given by the multivariate hypergeometric distribution with parameters $\Xi = \{k_i^{\text{out}} k_j^{\text{in}}\}_{i,j}$:

$$\Pr(\mathcal{G}|\Xi) = \binom{\sum_{ij} \Xi_{ij}}{m}^{-1} \prod_{i,j \in V} \binom{\Xi_{ij}}{A_{ij}}, \tag{1}$$

where A_{ij} denotes the element ij of the adjacency matrix of $\mathcal{G}$, and the probability of observing $\mathcal{G}$ is non-zero only if $\sum_{ij} A_{ij} = m$.

Generalized hypergeometric ensembles of random graphs further extend this formulation. In gHypEG, the probability of connecting two vertices depends not only on the degree (i.e., number of stubs) of the two vertices but also on an independent propensity of the two vertices to be connected, which captures non-degree related effects. Doing so allows constraining the configuration model such that given edges are more likely than others, independently of the degrees of the respective vertices. The right side of Fig. 1 illustrates this case, where A is most likely to connect with vertex D, belonging to the same group, even though D has only one available stub.

In generalized hypergeometric ensembles the distribution over multi-graphs (denoted $\mathcal{G}$) is formulated such that it depends on two sets of parameters: the combinatorial matrix Ξ, and a propensity matrix Ω that captures the propensity each pair of vertices to be connected. Each of these two matrices has dimensions $n \times n$ where n is the number of vertices in $\mathcal{G}$. The contributions of the two matrices to the model are as follows. The combinatorial matrix Ξ encodes the configuration model as described above. The propensity matrix Ω encodes dyadic propensities of vertices that go beyond what prescribed by the combinatorial matrix Ξ. The ratio between any two elements Ω_{ij} and Ω_{kl} of the propensity matrix is the odds-ratio of observing an edge between vertices i and j instead k and l, independently of the degrees of the vertices.

As for the case of the configuration model, this process can be seen as sampling edges from an urn. Moreover, specifying a propensity matrix Ω allows to bias the sampling in specified ways, so that some edges are more likely to be sampled than others. The probability distribution over a graph $\mathcal{G}$ given Ξ and Ω is then described by the multivariate Wallenius' noncentral hypergeometric distribution (Wallenius 1963; Chesson 1978).

We further denote with $\mathbf{A}$ the adjacency matrix of the multi-graph $\mathcal{G}$ and with V its set of vertices, the probability distribution underlying a gHypEG $\mathbb{X}(\Xi, \Omega, m)$ with parameters Ξ, Ω, and with m edges is defined as follows:

$$\Pr(\mathcal{G}|\Xi,\Omega) = \left[\prod_{i,j\in V}\binom{\Xi_{ij}}{A_{ij}}\right]\int_0^1 \prod_{i,j\in V}\left(1-z^{\frac{\Omega_{ij}}{S_\Omega}}\right)^{A_{ij}} dz \tag{2}$$

with

$$S_\Omega = \sum_{i,j\in V}\Omega_{ij}(\Xi_{ij}-A_{ij}). \tag{3}$$

In Eq. 2, the first term on the right-hand side represents combinatorial effects encoding degrees, inherited from the configuration model. The second term, constituted by the integral, encodes the biases that need to be enforced on top of the process defined by the configuration model. Note that, if $\Omega_{ij} = c$ for all i,j and for any constant c, i.e., if no biases are enforced on the configuration model, Eq. 2 corresponds to Eq. 1 (Casiraghi and Nanumyan 2018). The probability distribution for undirected graphs and graphs without self-loops are defined similarly: by excluding the lower triangular entries of the adjacency matrix or by excluding its diagonal entries respectively (we refer to Casiraghi and Nanumyan (2018) for more details).

In the case of large graphs, sampling from an urn without replacement can be approximated by sampling with replacement from the same urn. Under this assumption, the approximation allows to estimate the probability given in Eq. 2 by means of a multinomial distribution with parameters $p_{ij} = \Xi_{ij}\Omega_{ij}/\sum_{kl}\Xi_{kl}\Omega_{kl}$.

Block-constrained configuration model

The main modelling assumption that differentiate gHypEGs from SBMs is in the dependence/independence of edge probabilities. In particular, while SBMs assume independent edge probabilities, and specifies a Poisson process for the edge generating process, gHypEG fixes the total number of edges m in the model and removes the assumption of independence between edge probabilities. This assumption has the conceptual advantage of not assuming an arbitrary edge generating process, such as the Poisson process considered by DC-SBMs.

We hence define the block-constrained configuration model (BCCM) building on the framework provided by generalized hypergeometric ensembles of random graphs. We achieve so by utilizing a particular form of the propensity matrix Ω. Specifically, we need to encode the block structure that we observe in the propensity matrix Ω. We do so by specifying a block propensity matrix $\Omega^{(B)}$ where each of its elements ${\Omega^{(B)}}_{ij} = \omega_{b_i}$ if the vertices i and j are in the same block b_i, and ${\Omega^{(B)}}_{ij} = \omega_{b_ib_j}$ if the vertices i and j are in different blocks b_i and b_j respectively. Figure 2 shows a block-propensity matrix characterised by three blocks. Similarly to SBMs, in the presence of B blocks, we can specify a $B \times B$ block-matrix $\mathbf{B}$ that captures the block structure through its parameters $\omega_{b_ib_j}$.

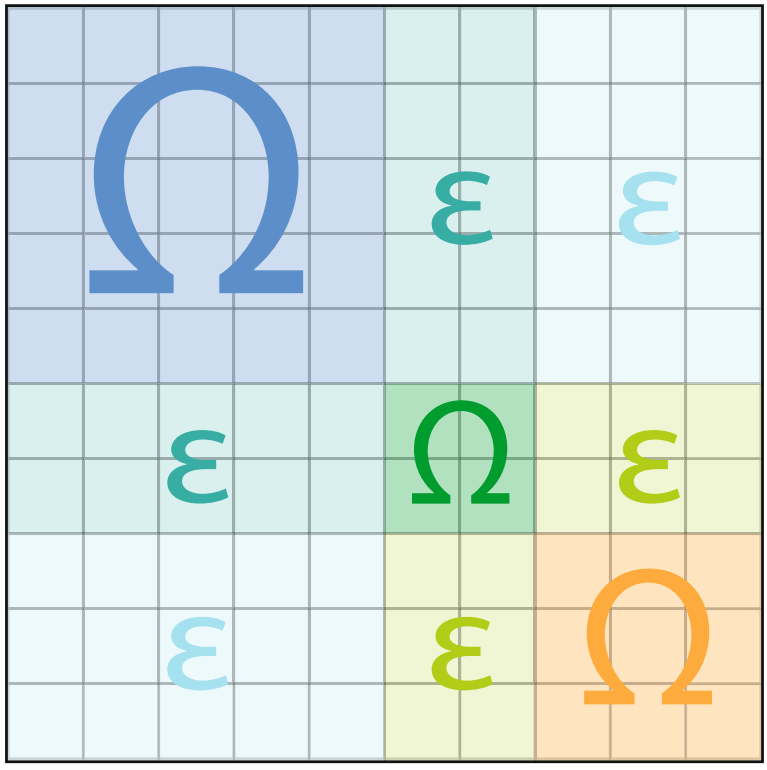

Fig. 2 Block-matrix. Structure of a block propensity matrix with 3 different blocks (blue, green, yellow). The entries along the diagonal capture the within-block propensities, those away from the diagonal capture the between-block propensities

However, in the case of a BCCM, the entries $\omega_{b_i b_j}$ capture the deviations in terms of edge propensities from the configuration model defined by the matrix Ξ, constraining edges into blocks.

The block-matrix $\mathbf{B}$ can be specified to generate various structures, extending those naturally produced by degrees only, such as a diagonal block-matrix can model graphs with disconnected components. The inclusion of small off-diagonal elements gives rise to standard community structures, with densely connected clusters of vertices. By specifying different types of block-matrices, it is also possible to model core-periphery, hierarchical, or multipartite structures.

The block-constrained configuration model $\mathbb{X}(\Xi, \mathbf{B}, m)$ with m edges is thus completely defined by the combinatorial matrix Ξ, and by the block-matrix $\mathbf{B}$ generating the propensity matrix $\Omega^{(B)}$. We can then rewrite the general probability for a gHypEG given in Eq. 2 for BCCM:

$$\Pr(\mathcal{G}|\Xi, \mathbf{B}) = \left[\prod_{i,j\in V}\binom{\Xi_{ij}}{A_{ij}}\right]\int_0^1 \prod_{i,j\in V}\left(1 - z^{\frac{\omega_{b_i b_j}}{S_{\mathbf{B}}}}\right)^{A_{ij}} dz \tag{4}$$

with

$$S_{\mathbf{B}} = \sum_{i,j\in V}\omega_{b_i b_j}(\Xi_{ij} - A_{ij}). \tag{5}$$

Table 1 summarises the differences between the distributions underlying the DC-SBM and the BCCM.

Table 1 Comparison of the properties of DC-SBMs and BCCMs

Model	**DC-SBM**	**BCCM**
Multi-edges distribution	n^2 independent Poisson	n^2-variate Wallenius
Number of edges m	preserved in expectation	fixed value
Distribution of m	Poisson(m)	fixed value
Degrees	preserved in expectation	preserved in expectation
Distribution of degrees	n independent Poisson	n-variate Wallenius

Despite its complicated expression, the probability distribution in Eq. 4 allows computing probabilities for large graphs, without the need to resort to Monte-Carlo simulations (Fog 2008a). This permits the study of large graphs and provides simple model selection methods based on the comparison of likelihoods, such as likelihood-ratio tests, or those based on information criteria. In this article, we will consider model selection based on the comparison of information criteria.

We will adopt the two most commonly used ones: Akaike information criterion (AIC) (Akaike 1974), and Schwarz or Bayesian information criterion (BIC) (Schwarz and et al 1978). Both criteria depend on the likelihood function of the models to be compared and penalize for the number of parameters estimated by the model. The model with the lowest score is the preferred one, as it best fits the data without overfitting it. In particular, it is not the absolute size of the score, but it is the difference between values that matters for model selection.

Information-theoretic methods considered here provide a simple way to select the best-approximating model from a candidate set of models. The concept of information criterion has allowed major practical and theoretical advances in model selection and the analysis of complex data sets (Stone 1982; Bozdogan 1987; DeLeeuw 1992). In particular, AIC and BIC allow performing model selection without the need of simulations, nor the assumption of specific asymptotic behaviors of the probability distribution of the model (although BIC assumes that *the priors* for the parameters estimated are asymptotically normal). Moreover, the aim of model selection by means AIC and BIC is not to identify exactly the 'true model,' i.e., the actual process generating the data, but to propose simpler models that are good approximations of it (Kuha 2004). They only allow the selection of the best model among those within a specified set. This means that, if all models in the set are very poor, information criteria will select the best model, but even that relatively best model might be poor in the absolute sense (Burnham and Anderson 2004).

The Akaike information criterion for a model $\mathbb{X}$ given a graph $\mathcal{G}$ is formulated as follows:

$$\mathrm{AIC}(\mathbb{X}|\mathcal{G}) = 2k - 2\log\left[\hat{L}(\mathbb{X}|\mathcal{G})\right], \tag{6}$$

where k is the number of parameters estimated by $\mathbf{X}$ and $\hat{L}(\mathbb{X}|\mathcal{G})$ is the likelihood of model $\mathbb{X}$ given the graph $\mathcal{G}$. AIC gives an estimate of the expected, relative Kullback-Leibler distance between the fitted model and the unknown true mechanism generating the observed data. Hence, the best model among a set of models is the one that has the minimal distance from the true process, and thus the one that minimizes AIC.

The Bayesian information criterion for a model $\mathbb{X}$ given a graph $\mathcal{G}$ is given by:

$$\mathrm{BIC}(\mathbb{X}|\mathcal{G}) = \log(m)k - 2\log\left[\hat{L}(\mathbb{X}|\mathcal{G})\right], \tag{7}$$

where k is the number of parameters estimated by $\mathbb{X}$, m is the number of observations, i.e., edges, and $\hat{L}(\mathbb{X}|\mathcal{G})$ is the likelihood of model $\mathbb{X}$ given the graph $\mathcal{G}$. Similarly to AIC, the best model in a set according to BIC is the one which minimizes the criterion. Because of the presence of a higher penalty for model size, BIC tends to select models with lower parameters compared to AIC.

As mentioned above, what matters for model selection is the difference between the value of AICs or BICs and not their absolute values. For this reason, it is helpful to rank models in terms of their differences from the model which minimizes a given criterion. Suppose that there are R models, and we want to find the best one according to either AIC or BIC. Let AIC_{min} be the model which minimizes AIC for a given dataset. Then we can define the AIC differences Δ_i^{AIC} as the difference $\text{AIC}_i - \text{AIC}_{\text{min}}$ of the AIC score for model $i \in \{1, \ldots, R\}$, and the model which minimizes AIC. BIC differences are defined in a similar manner. While AIC and BIC differences are useful in ranking models, it is possible to quantify the plausibility of each model by defining relative likelihoods for the models. Specifically, the quantity $e^{-1/2\Delta_i}$ defines the relative likelihood of model i given the data (Burnham and Anderson 2004). To better interpret relative likelihoods, statisticians usually normalize relative likelihoods to be a set of positive weights w_i defined as

$$w_i := \frac{e^{-1/2\Delta_i}}{\sum_{r=1}^{R} e^{-1/2\Delta_r}}. \tag{8}$$

In the case of AIC, such model weights are usually referred to as Akaike weights and are considered to be the weight of evidence in favor of model i being the best model. In the case of BIC, instead, the weights define the posterior model probabilities. The bigger Δ_i is, the smaller w_i and the less plausible is model i as being the actual best model based on the design and sample size used. These weights provide an effective way to scale and interpret the Δ_i values and hence select the best model (Burnham and Anderson 2004).

In the next sections, we describe how BCCM can be used to generate graphs and how to fit its parameters to an observed graph. Because the absolute values of AIC and BIC are not important, and only relative Δ_is matter, in the following we will usually report only the value of the relative differences.

Generating realizations from the BCCM. BCCM is a practical generative model that allows the creation of synthetic graphs with complex structures by drawing realizations from the multivariate Wallenius non-central hypergeometric distribution. The process of generating synthetic graphs can be divided into two tasks. First, it is needed to specify the degree sequences for the vertices. It can be accomplished by, e.g., sampling the degree sequences from a power-law or exponential distributions. From the degree sequences we can generate the combinatorial matrix Ξ, specifying its elements $\Xi_{ij} = k_i^{\text{out}} k_j^{\text{in}}$, where k_i^{out} is the out-degree of vertex i. Second, we need to define a block-matrix $\mathbf{B}$, whose elements specify the propensities of observing edges between vertices, between and within the different blocks.

The block-matrix $\mathbf{B}$ takes the form given in Eq. 9:

$$\mathbf{B} = \begin{bmatrix} \omega_1 & \ldots & \omega_{1B} \\ & \vdots & \\ \omega_{B1} & \ldots & \omega_B \end{bmatrix}. \tag{9}$$

Elements ω_{kl}, with $k, l \in \{1, \ldots, B\}$, should be specified such that the ratio between any two elements corresponds to the chosen odds-ratios of observing an edge in the block corresponding to the first element instead of the block corresponding to the second element, given the degrees of the corresponding vertices were the same. For example, ω_1/ω_{32} corresponds to the odds-ratio of observing an edge between vertices in block 1 compared to an edge between block 2 and block 3. Note that in the case of an undirected graph, $\omega_{kl} = \omega_{lk}\ \forall k, l \in \{1, \ldots, B\}$. On the other hand, in the case of a directed graph, blocks may have a preferred directionality, i.e., edges between blocks may be more likely in one direction. In this case, we may choose $\omega_{kl} \neq \omega_{lk}$.

Once the parameters of the model are defined, we sample graphs with m edges from the BCCM $\mathbb{X}(\Xi, \Omega_B, m)$ defined by the combinatorial matrix Ξ, and the block-propensity matrix Ω_B defined by $\mathbf{B}$. As described in the previous section, sampling a graph from $\mathbb{X}(\Xi, \Omega_B, m)$ corresponds to sample m edges according to the multivariate Wallenius non-central hypergeometric distribution.

Examples We can specify different types of clustered graphs using this construction. As a demonstrative example, we define a block-matrix with five blocks connected in a ring. Each block is as dense as the others, and blocks are weakly connected with only their closest neighbors. The block-matrix quantifying these specification is given as

$$\mathbf{B} = \begin{bmatrix} 1 & 0.1 & 0 & 0 & 0 \\ 0 & 1 & 0.1 & 0 & 0 \\ 0 & 0 & 1 & 0.1 & 0 \\ 0 & 0 & 0 & 1 & 0.1 \\ 0.1 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{10}$$

According to the choice made in Eq. 10, edges within diagonal blocks are 10 times more likely than edges within off-diagonal blocks.

After fixing this block-matrix, we can define different degree sequences for the vertices. We highlight here the results obtained when fixing three different options in a directed graph without self-loops, with $n = 50$ vertices and $m = 500$ edges. We generate realizations by specifying the combinatorial matrix Ξ and the block propensity matrix and exploiting the random number generator provided by Fog (2008b) in the `R` library `BiasedUrn`.

The first degree-sequence we can set is the most straightforward option, corresponding to the standard non-degree-corrected stochastic block-model. This model corresponds to setting each entry in the combinatorial matrix Ξ equal to $m^2/(n(n-1))$ (Casiraghi et al. 2016). If we assign the same number of vertices to each block, we expect the model to generate graphs with homogeneous blocks. Figure 3a shows a realisation from this model. The second degree-sequence we can set is defined such that the degrees of the vertices of each block are drawn from a power-law distribution. We expect that each block shows the same structure, with few vertices with high degrees, and many with low degrees. Because of this, we expect that most blocks are connected with directed edges starting from high-degree vertices. Figure 3b shows a realization from this model where this is visible. Finally, we set a degree sequence where the degrees of all vertices are drawn from a power-law distribution. Figure 3c shows a realization from this model. The combinatorial matrices corresponding to Fig. 3b and c are included with this article as additional files.

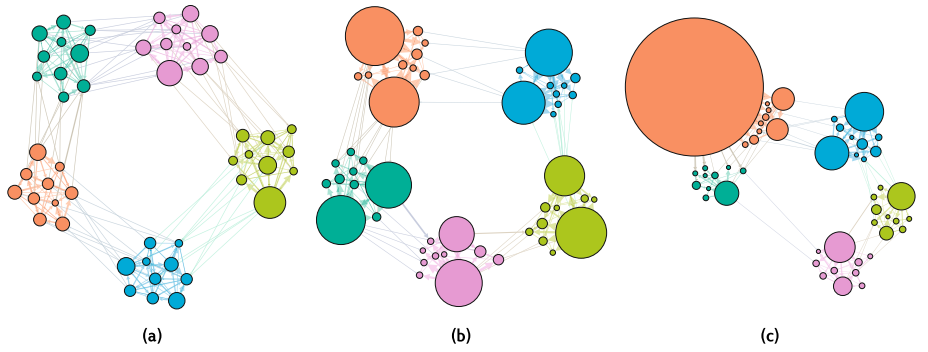

Fig. 3 Realisations from a block-constrained configuration model obtained by fixing the block-matrix **B** and varying the out-degree distribution. Each realisation is obtained from a BCCM with $N = 50$ vertices and $m = 500$ directed edges. The vertices are separated into 5 equally sized blocks and the block-matrix **B** is given by Eq. 10. On left side, **a** is a realisation from a BCCM where the degree distributions are uniform. It corresponds to a realisation from a standard SBM. In the center, **b** is a realisation obtained by drawing the out-degree distribution of the vertices in each block from a power-law distribution with parameter $\alpha = 1.8$. On the right side, **c** is a realisation obtained by drawing the out-degree distribution of all vertices from the same power-law. All graphs are visualised using the force-atlas2 layout with weighted edges. Out-degrees determine vertex sizes, and edge widths the edge counts

Instead of varying the degree sequences of the underlying configuration model, we can as well alter the strength of the block structure, changing the block-matrix **B**. Similarly to what we did above, we show three different combinations of parameters. First, we set the within group parameters ω_{b_i} equal to the between group parameters $\omega_{b_i b_j}$ $\forall i, j$. Second, we set the parameters $\omega_{b_1} = 10$ so that the more edges are concentrated in the first block. Third, we set the parameter to reconstruct a hierarchical structure. We modify the parameters $\omega_{b_1 b_2} = \omega_{b_3 b_4} = \omega_{b_4 b_5} = 0.8$ to model graphs with two macro clusters weakly connected, where the one is split into two clusters strongly connected and the other into three clusters strongly connected. Realizations drawn from each of these three models are shown in Fig. 4.

Fitting the block-matrix. In DC-SBMs the number of edges between each pair (i, j) of vertices are assumed to be drawn from independent Poisson distributions, with parameters $\theta_i \theta_j \omega_{b_i b_j}$. Let $A_{b_\alpha b_\beta} = \sum_{i \in b_\alpha, j \in b_\beta} A_{ij}$ denote the number of edges between all vertices i that are in the block b_α and j in block b_β. We further denote b_i the partition of vertex i. Exploiting the independence of probabilities, the maximum likelihood estimates $\widehat{\theta_i}$ and $\widehat{\omega}_{b_i b_j}$ of the parameters θ_i and $\omega_{b_i b_j}$ are given by $\widehat{\theta_i} := k_i / \kappa_{b_i}$ and $\widehat{\omega}_{b_i b_j} := A_{b_i b_j}$ (Karrer and Newman 2011).

Because BCCM does not assume independence among edge probabilities, the parameter estimation is necessarily more complicated than that of DC-SBMs. However, the formulation of the block-constrained configuration model allows for the fast estimation of the parameters of the block-matrix. Similarly to what is done with SBMs, we fit the BCCM by preserving in expectation the observed number of edges between and within different blocks. To estimate the entries ω_b of the block-matrix $\hat{\mathbf{B}}$, we exploit the properties of the generalized hypergeometric ensemble of random graphs.

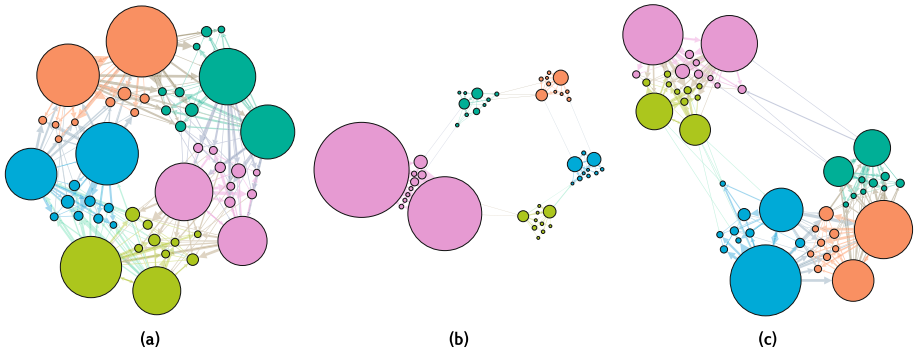

Fig. 4 Realisations from a block-constrained configuration model obtained by fixing the out-degree distribution and varying the parameters within the block-matrix **B**. Each realisation is obtained from a BCCM with $N = 50$ vertices and $m = 500$ directed edges. The out-degree distribution of the vertices in each block follows a power-law distribution with parameter $\alpha = 1.8$ The vertices are separated into 5 equally sized blocks and the structure of the block-matrix **B** is given by Eq. 10, but in each graph the values of some of the parameters $\omega_{b_i b_j}$ are changed. On left side, **a** is a realisation from a BCCM where the between-block parameters are increased to 1. In the center, **b** is a realisation obtained by increasing the parameter ω_{b_1} that controls for the internal cohesion of the first block. On the right side, **c** is a realisation obtained by increasing to 0.8 the between-block parameters $\omega_{b_1 b_2}$, $\omega_{b_3 b_4}$, and $\omega_{b_4 b_5}$, to create a hierarchical block structure where the first two blocks are part of a macro cluster, and the last three blocks are part of another. All graphs are visualised using the force-atlas2 layout with weighted edges. Out-degrees determine vertex sizes, and edge widths the edge counts

In gHypE, the entries of the expected adjacency matrix $\langle A_{ij} \rangle$ are obtained by solving the following system of equations (Casiraghi and Nanumyan 2018):

$$\left(1 - \frac{\langle A_{11} \rangle}{\Xi_{11}}\right)^{\frac{1}{\Omega_{11}}} = \left(1 - \frac{\langle A_{12} \rangle}{\Xi_{12}}\right)^{\frac{1}{\Omega_{12}}} = \dots \tag{11}$$

with the constraint $\sum_{i,j \in V} \langle A_{ij} \rangle = m$.

Because to estimate BCCM we need to fix the expectation of the number of edges between blocks and not between dyads, we proceed as described below. We denote with $\Xi_{b_\alpha} = \sum_{i,j \in b_\alpha} \Xi_{ij}$ the sum of all the elements of the matrix Ξ corresponding to those dyads. Then, we fix the expectations of the ensemble such that the number of edges between and within blocks is given by A_{b_α}s. Hence, in the case of the block-constrained configuration model with B blocks we estimate the $B \cdot (B+1)/2$ parameters $\omega_{b_\alpha b_\beta}$s constituting the block-matrix $\hat{\mathbf{B}}$ solving the following set of independent equations, defined up to an arbitrary constant k:

$$\begin{cases} \left(1 - \frac{A_{b_1}}{\Xi_{b_1}}\right)^{\frac{1}{\omega_{b_1}}} = k \\ \qquad\qquad \vdots \\ \left(1 - \frac{A_{b_B}}{\Xi_{b_B}}\right)^{\frac{1}{\omega_{b_B}}} = k\,. \end{cases} \tag{12}$$

Solving for $\omega_{b_\alpha b_\beta}$, we find that the entries of the block-matrix $\hat{\mathbf{B}}$ that preserve in expectation the observed number of edges between and within blocks are given by

$$\omega_{b_\alpha b_\beta} := -\log\left(1 - \frac{A_{b_\alpha b_\beta}}{\Xi_{b_\alpha b_\beta}}\right). \tag{13}$$

The estimation of the parameters scales quadratically only with the number of blocks.

When the parameters of the BCCM are estimated as described here, the block-constrained configuration model has the advantageous property of asymptotic consistency. It means that, if the method described here is applied to synthetic graphs generated from a BCCM, the technique introduced in this article can correctly recover the original model.

Estimating the Ξ matrix. In the case of the configuration model defined by Eq. 1, the elements Ξ_{ij} of the combinatorial matrix are defined as $k_i^{\text{in}} k_j^{\text{out}}$. This definition generates a model that preserves the degree sequences of the observed graph (Casiraghi and Nanumyan 2018). By generalizing the model according to Eq. 4, where the propensity matrix is estimated as in Eq. 13, we introduce constraints on the edge sampling process that allows preserving the observed number of edges in each block. The estimated parameters can hence be interpreted as the bias needed to modify the configuration model to reproduce block structures.

To preserve the degrees of the observed graph in the BCCM, we need to update the combinatorial matrix such that it defines the degree-sequences of the corresponding configuration model like there were no block constraints. We achieve this by redefining the combinatorial matrix elements as $\Xi_{ij} = k_i^{\text{in}} k_j^{\text{out}} \theta_i^{\text{in}} \theta_j^{\text{out}}$. The estimation of Ξ and Ω is then performed by an expectation-maximization algorithm that iteratively estimates Ξ and Ω such that degrees and blocks are preserved in expectation. A pseudo-code for the algorithm estimating the parameters of a BCCM model for directed graphs is provided in Algorithm 1. In the case of undirected graphs, the algorithm is adapted according to the fact that Ξ and Ω are upper-triangular matrices.

Algorithm 1 estimateBCCM($\mathcal{G}$, tolerance): *Adjust entries of* Ξ *to match expected degrees (based on* Ξ, Ω*) to observed degrees (based on* $\mathcal{G}$*), within 'tolerance' error, where the error is computed as the mean absolute error (MAE).*

Require: $\mathcal{G}$ observed graph, tolerance
Ensure: Ξ, Ω
$k_v^{\text{out}} \leftarrow \sum_x A_{v,x}$ {out-degrees}
$k_v^{\text{in}} \leftarrow \sum_x A_{x,v}$ {in-degrees}
$m \leftarrow \sum_v k_v^{\text{out}}$ {number of edges}
$\Xi_{vw} \leftarrow k_v^{\text{out}} \cdot k_w^{\text{in}}$ {initialize Ξ for all $v,w \in \mathcal{G}$}
$A_{b_\alpha b_\beta} \leftarrow \sum_{x \in b_\alpha, y \in b_\beta} A_{x,y}$ {number of edges in block $b_{\alpha\beta}$}
$\Xi_{b_\alpha b_\beta} \leftarrow \sum_{x \in b_\alpha, y \in b_\beta} \Xi_{x,y}$ {combinatorial matrix for block $b_{\alpha\beta}$}
$\omega_{b_\alpha b_\beta} := -\log(1 - \frac{A_{b_\alpha b_\beta}}{\Xi_{b_\alpha b_\beta}})$ {initialize Ω for all $v,w \in \mathcal{G}$}
repeat
  $\widehat{k}_v^{\text{in}} \leftarrow \sum_x \mathbb{E}(A_{x,v})$ {Expectation for in-degrees}
  $\Xi_{vw} \leftarrow \Xi_{vw} \cdot \frac{k_w^{\text{in}}}{\widehat{k}_w^{\text{in}}}$ {Correction for in-degrees}
  $\widehat{k}_v^{\text{out}} \leftarrow \sum_x \mathbb{E}(A_{v,x})$ {Expectation for out-degrees}
  $\Xi_{vw} \leftarrow \Xi_{vw} \cdot \frac{k_v^{\text{out}}}{\widehat{k}_v^{\text{out}}}$ {Correction for out-degrees}
  $\Xi_{b_\alpha b_\beta} \leftarrow \sum_{x \in b_\alpha, y \in b_\beta} \Xi_{x,y}$ {new combinatorial matrix for block $b_{\alpha\beta}$}
  $\omega_{b_v b_w} := -\log(1 - \frac{A_{b_v b_w}}{\Xi_{b_v b_w}})$ {ML estimation of $\omega_{b_v b_w}$ for all v, w}
until $\text{MAE}(k^{\text{out}}, \widehat{k}^{\text{out}}) + \text{MAE}(k^{\text{in}}, \widehat{k}^{\text{in}}) \leq$ tolerance
return Ξ, Ω

Case studies

We conclude the article with a case study analysis of synthetic and empirical graphs. We highlight the interpretability of the resulting block-constrained configuration models in terms of deviations from the classical configuration model. In particular, a weak community structure in a graph is reflected in a small contribution to the likelihood of the estimated block-matrix. On the other hand, a strong community structure is reflected in a substantial contribution to the likelihood of the estimated block-matrix. Here, we quantify this difference employing AIC or BIC. However, other information criteria may also be used. Moreover, studying the relative values of the estimated parameters in the block matrices quantifies how much the configuration model has to be biased towards a block structure to fit the observed graph optimally. The more different are the values of the parameters, the stronger is the block structure compared to what is expected from the configuration model.

We start by analyzing synthetic graphs generated according to different rules, and we show that fitting the block-constrained configuration model parameters allows selecting the correct, i.e., planted, partition of vertices, among a given set of different partitions. We perform three experiments with large directed graphs with clusters of different sizes. Finally, we conclude by employing the BCCM to compare how well different partitions obtained by different clustering algorithms fit popular real-world networks.

Analysis of synthetic graphs. We generate synthetic graphs incorporating 'activities' of vertices in a classical SBM, to be able to plant different out-degree sequences in the synthetic graphs. First, we need to assign the given activity to each vertex. Higher activity means that the vertex is more likely to have a higher degree. Second, we need to assign vertices to blocks and assign a probability of sampling edges to each block. Densely connected blocks have a higher probability than weakly connected blocks. The graph is then generated by a weighted sampling of edges with replacement from the list containing all dyads of the graph. The product between the activity corresponding to the from-vertex and the weight corresponding to the block to which the dyad belongs gives sampling-weights for each dyad. The probabilities of sampling edges correspond to the normalized weights so that their sum is 1.

For example, let us assume we want to generate a 3-vertices graph with two clusters. We can fix the block weights as follows: edges in block 1 or 2 have weight w_1 and w_2 respectively; edges between block 1 and block 2 have weight w_{12}. Table 2 shows the list of dyads from which to sample together with their weights, where the activity of vertices is fixed to (a_1, a_2, a_3), and the first two vertices belong to the first block. Note that if the activities of the vertices were all set to the same value, this process would correspond to the original SBM. In the following experiments, we generate different directed graphs with $N = 500$ vertices, $m = 40000$ edges, and different planted block structures and vertex activities.

In the first experiment, we show the difference between estimating the parameters for an SBM and the BCCM when the block structure is given. To do so, we first generate the activities of vertices from an exponential distribution with parameter $\lambda = N/m$ (such that the expected sum of all activities is equal to the number of edges m we want to sample). After sorting the activity vector in decreasing order, we assign it to the vertices. In this way, the first vertex has the highest activity, and hence the highest out-degree, and so

Table 2 Edge list with weights for the generation of synthetic graphs with given vertex activities and block structure

dyad	activity	block id	block weight	sampling weight
1 – 1	a_1	1	w_1	$a_1 w_1$
1 – 2	a_1	1	w_1	$a_1 w_1$
1 – 3	a_1	12	w_{12}	$a_1 w_{12}$
2 – 1	a_2	1	w_1	$a_2 w_1$
2 – 2	a_2	1	w_1	$a_2 w_1$
2 – 3	a_2	12	w_{12}	$a_2 w_{12}$
3 – 1	a_3	12	w_{12}	$a_3 w_{12}$
3 – 2	a_3	12	w_{12}	$a_3 w_{12}$
3 – 3	a_3	2	w_2	$a_3 w_2$

on. In the first experiment, we do not assign block weights so that the graphs obtained do not show any consistent cluster structure, and have a skewed out-degree distribution according to the fixed vertex activity (correlation ~ 1).

First, we assign the vertices randomly to two blocks. We proceed by estimating the parameters for an SBM and a BCCM, according to the blocks to which the vertex has been assigned. Since no block structure has been enforced and the vertex has been assigned randomly to blocks, we expect that the estimated parameters for the block matrices $\hat{\mathbf{B}}_{\mathrm{SBM}}$ and $\hat{\mathbf{B}}_{\mathrm{BCCM}}$ will all be close to 1 (when normalized by the maximum value), reflecting the absence of a block structure. The resulting estimated parameters for an exemplary realisation are reported in Eq. 14.

$$\hat{\mathbf{B}}_{\mathrm{SBM}} = \begin{bmatrix} 1.0000000 & 0.9992577 \\ 0.9992577 & 0.9603127 \end{bmatrix} \qquad \hat{\mathbf{B}}_{\mathrm{BCCM}} = \begin{bmatrix} 0.9808935 & 1.0000000 \\ 1.0000000 & 0.9805065 \end{bmatrix} \tag{14}$$

As expected, the estimated values for both models are close to 1.

After changing the way vertices are assigned to blocks, we repeat the estimation of the two models. Now, we separate the vertices into two blocks such that the first 250 vertices ordered by activity are assigned to the first block and the last 250 to the second one. We expect that the SBM will assign different parameters to the different blocks because now the first block contains all vertices with high degree, and the second block all vertices with low degree. Hence, most of the edges are found between vertices in the first block or between the two blocks. Differently, from the SBM, the BCCM corrects for the observed degrees. Hence, we expect that the parameters found for the block-matrix will be all close to 1 again, as no structure beyond that one generated by degrees is present. Thus the block assignment does not matter for the estimated parameter. The block matrices for the two models, estimated for the same realisation used above, are provided in Eq. 15.

$$\hat{\mathbf{B}}_{\mathrm{SBM}} = \begin{bmatrix} 1.000000 & 0.597866 \\ 0.597866 & 0.194896 \end{bmatrix} \qquad \hat{\mathbf{B}}_{\mathrm{BCCM}} = \begin{bmatrix} 0.997024 & 0.995108 \\ 0.995108 & 1.000000 \end{bmatrix} \tag{15}$$

We observe that the SBM assigns different values to each block, impairing the interpretability of the result. In particular, the parameters of $\hat{\mathbf{B}}_{\mathrm{SBM}}$ show the presence of a core-periphery structure which cannot be distinguished from what obtained naturally from skewed degree distributions. The estimation of $\hat{\mathbf{B}}_{\mathrm{BCCM}}$, on the contrary, highlights the absence of any block structure beyond that one generated by the degree sequence, and we can correctly conclude that the degree distributions entirely generate the core-periphery structure of the observed graph.

In the second synthetic experiment, we highlight the model selection features of the BCCM. Thanks to the fact that we are able to compute the likelihood of the model directly, we can efficiently compute information criteria such as AIC or BIC to perform model selection. We generate directed graphs with self-loops with $N = 500$ vertices, $m = 40000$ edges, and two equally sized clusters. Again, we generate vertex activities from an exponential distribution with rate $\lambda = N/m$. We fix the block weights to be $w_1 = 1$, $w_2 = 3$, and $w_{12} = 0.1$. Using this setup, we can generate synthetic graphs with two clusters, one of which is denser than the other. If we fit a BCCM to the synthetic graph with the correct assignment of vertices to blocks, we obtain the following block-matrix $\hat{\mathbf{B}}_{\mathrm{BCCM}}$ for an exemplary realization:

$$\hat{\mathbf{B}}_{\mathrm{BCCM}} = \begin{bmatrix} 1.1760878 & 0.1108463 \\ 0.1108463 & 3.0000000 \end{bmatrix} \tag{16}$$

We note that we approximately recover the original block weights used to generate the graph.

We can now compare the AIC obtained for the fitted BCCM model, $\mathrm{AIC}_{\mathrm{BCCM}} = 662060$, to that obtained from a simple configuration model (CM) with no block assignment, $\mathrm{AIC}_{\mathrm{CM}} = 693540$. The CM model is formulated in terms of a gHypEG where the propensity matrix $\Omega \equiv 1$. The AIC for the BCCM is considerably smaller, confirming that the model with block structure fits better the observed graph. In terms of AIC differences, $\Delta^{\mathrm{AIC}}_{\mathrm{BCCM}} = 0$ and $\Delta^{\mathrm{AIC}}_{\mathrm{CM}} = 31480$. This corresponds to model weights $w_{\mathrm{BCCM}} \sim 1$ and $w_{\mathrm{CM}} \sim 0$. That means that there is no evidence for model CM. As a benchmark, we compute the AIC for BCCM models where the vertices have been assigned randomly to the two blocks.

Table 3 reports the AIC differences obtained for 1000 random assignment of vertices to the blocks, computed on the same observed graph. We observe that this usually results in values close to that of the simple configuration model, as the block assignments do not reflect the structure of the graph. In a few cases, a small number of vertices are correctly assigned to blocks, showing a slight reduction in AIC, which is however far from that of the correct assignment.

BCCM also allows comparing models with a different number of blocks. To do so, we separate the vertices in one of the blocks of the model above into two new blocks. Because we add more degrees of freedom, we expect an increase in the likelihood of the new BCCM with three blocks, but this should not be enough to give a considerable decrease in AIC. Since the synthetic graph has been built planting two blocks, the AIC should allow us to select as an optimal model the BCCM with two blocks. The resulting block-matrix $\hat{\mathbf{B}}^{(3)}_{\mathrm{BCCM}}$ with three blocks is reported in Eq. 17.

$$\hat{\mathbf{B}}^{(3)}_{\mathrm{BCCM}} = \begin{bmatrix} 1.1739475 & 1.1797875 & 0.1088987 \\ 1.1797875 & 1.1706410 & 0.1129094 \\ 0.1088987 & 0.1129094 & 3.0000000 \end{bmatrix} \tag{17}$$

Table 3 Δ_l^{AIC} values from the model with the correct assignment vertices-blocks, obtained for 1000 random assignment of vertices to the blocks, computed on the same observed graph

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Δ_l^{AIC}	31471	31483	31484	31483	31484	31484

We see that the estimated model fits different parameter values for the two sub-blocks, since the added parameters can now accommodate for random variations generated by the edge sampling process. However, as expected, there is no (statistical) evidence to support the more complex model. In fact, comparing the AIC values we obtain $\mathrm{AIC}^{(3)}_{\mathrm{BCCM}} = 662065 > 662060 = \mathrm{AIC}_{\mathrm{BCCM}}$. This corresponds to $\Delta^{\mathrm{AIC}}_{\mathrm{BCCM}} = 0$ and $\Delta^{\mathrm{AIC}}_{\mathrm{BCCM}^{(3)}} = 5$. In terms of model weights, we get $w_{\mathrm{BCCM}} \sim 0.92$ and $w^{(3)}_{\mathrm{BCCM}} \sim 0.08$. That means that there is strong evidence against the more complex model, as the probability that the more complex model is closer to the real process is only 0.08, given the data used to estimate the model.

To provide more evidence in support of this selection procedure, we can repeat this experiment on 100 samples from the same model used before. The results provide median AIC differences of $\Delta_{\mathrm{BCCM}} = 0$ and $\Delta^{(3)}_{\mathrm{BCCM}} = 4.32$. Moreover, out of the 100 samples only 7 have $\mathrm{AIC}^{(3)}_{\mathrm{BCCM}} < \mathrm{AIC}_{\mathrm{BCCM}}$. This is aligned with the probability of 0.08 estimated employing model weights. We can thus successfully use BCCM to perform model selection, both when a different number of clusters or various vertex assignments are used.

In the third experiment, instead of two clusters, we plant three clusters of different sizes ($|B_1| = 250$, $|B_2| = 125$, $|B_3| = 125$). We choose the block parameters such that one of the smaller clusters is more densely connected with the bigger cluster, and the smaller cluster is relatively denser than the others. To do so we choose the block weights as follows: $w_1 = w_2 = 1$, $w_3 = 3$, $w_{13} = w_{23} = 0.1$, $w_{12} = 0.8$. As before, we draw vertex activities from an exponential distribution with parameter $\lambda = N/m$. One exemplary realisation is plotted in Fig. 5. The plot clearly shows the separation into three clusters, with

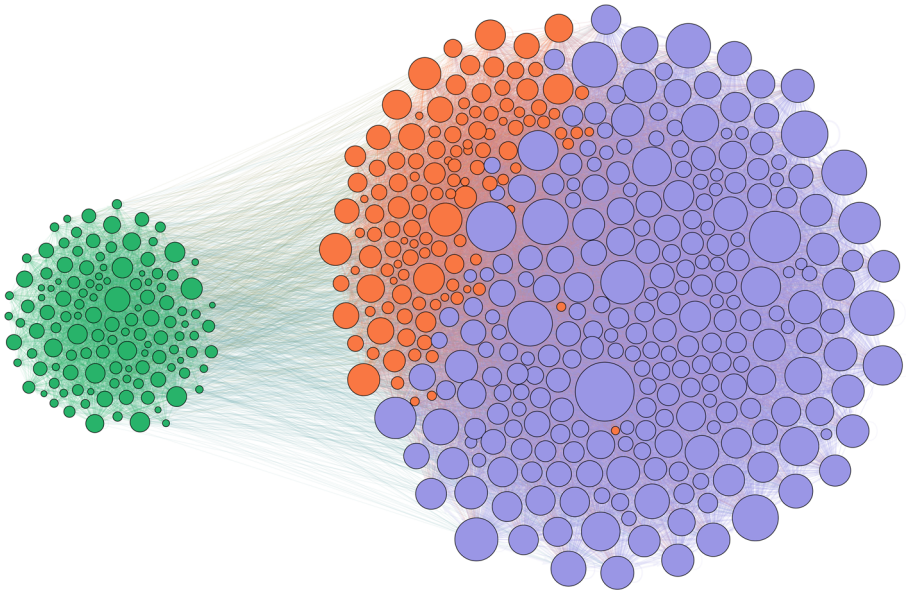

Fig. 5 Visualisation of a synthetic graph with $N = 500$ vertices and $m = 40000$ directed edges, obtained with the force-atlas2 layout. Vertices are separated into three blocks of different sizes, such that the largest block (250 vertices, in purple) is strongly connected with one of the smaller blocks (125 vertices, in orange). Both blocks are weakly connected to the third block, that is clearly separated (125 vertices, in green). The out-degree sequence of the graph follows an exponential distribution with parameter $\lambda = N/m$. The joint effects of the non-uniform degree sequence together with the asymmetric block structure makes the task of community detection on this graph particularly hard for standard algorithms

cluster 1 (purple) and 2 (orange) more densely connected to each other than to cluster 3 (green). Fitting the same BCCM as before allows comparing the AICs for the three-blocks BCCM to the 2-block BCCM. In this case, we expect that the model with three blocks will fit considerably better the graph. Results of the fitting for the realisation plotted in Fig. 5 give $\text{AIC}^{(3)}_{\text{BCCM}} = 673585 < 699765 = \text{AIC}^{(2)}_{\text{BCCM}}$, correctly selecting the more complex model. This corresponds to $\Delta^{\text{AIC}}_{\text{BCCM}^{(2)}} = 26180$ and $\Delta^{\text{AIC}}_{\text{BCCM}^{(3)}} = 0$. In terms of model weights, we get $w^{(2)}_{\text{BCCM}} \sim 0$ and $w^{(3)}_{\text{BCCM}} \sim 1$. That means that there is strong evidence against the simpler model.

It is known that AIC does not punish model complexity as much as BIC. For this reason, in this case, we also compare the values of BIC obtained for the two models. Also in this case, with $\text{BIC}^{(3)}_{\text{BCCM}} = 2822787 < 2848941 = \text{BIC}^{(2)}_{\text{BCCM}}$, the information criterion allows to correctly select the model with 3 blocks. Comparing posterior probabilities for the two models, we get again $w^{(2)}_{\text{BCCM}} \sim 0$ and $w^{(3)}_{\text{BCCM}} \sim 1$.

Finally, we can use AIC and BIC to evaluate and rank the goodness-of-fit different block assignments that are obtained from various community detection algorithms. This allows choosing the best block assignment in terms of deviations from the configuration model, i.e., which of the detected block assignment better captures the block structure that goes beyond that generated by the degree sequence of the observed graph. We compare the result obtained from 5 different algorithms run using their `igraph` implementation for `R`. In the following we use: `cluster_fast_greedy`, a greedy optimisation of modularity (Clauset et al. 2004); `cluster_infomap`, the implementation of `infomap` available through `igraph` (Rosvall and Bergstrom 2008); `cluster_label_prop`, label propagation algorithm (Raghavan et al. 2007); `cluster_spinglass`, find communities in graphs via a spin-glass model and simulated annealing (Reichardt and Bornholdt 2006); `cluster_louvain`, the Louvain multi-level modularity optimisation algorithm (Blondel et al. 2008). As the modularity maximization algorithms are implemented only for undirected graphs, we apply them to the undirected version of the observed graph. The results of the application of the 5 different algorithms on the realisation shown in Fig. 5 are reported in the table in Table 4.

The five different community detection algorithms find three different block structures. Three of them are not able to detect the third block, while the other two algorithms split the vertices into too many blocks. AIC ranks best `infomap` even though it detects one block too many. BIC punishes for the number of parameters more, so ranks best

Table 4 Comparison of the goodness-of-fit of 5 different block structures detected by five different community detection algorithms

	fast_greedy	infomap	label_prop	spinglass	louvain	original
B	2	4	2	7	2	3
Δ^{AIC}_i	4	0	4	40	4	-282
w_i	0.12	0.88	0.12	0	0.12	
B	2	4	2	7	2	3
Δ^{BIC}_i	0	57	0	251	0	-260
w_i	1	0	1	0	1	

The different partitions are compared in terms of the AIC and BIC obtained by the corresponding BCCM. Note that model weights are computed normalizing over three models because the three algorithms that detect two blocks detect the same partitioning. In the right-most column, we give the results corresponding to the ground-truth block partitioning. Because none of the algorithms was able to detect the original partitioning, we do not include the corresponding model when computing weights

the 2-blocks. These results are consistent when repeating the experiment with different synthetic graphs generated from the same model. It is worth noting that none of the community detection algorithms was able to detect the planted block structure correctly. However, both the AIC and BIC of the BCCM fitted with the correct block structure are lower than those found by the different algorithms. This shows that information criteria computed using BCCM have the potential to develop novel community detection algorithms that are particularly suited for applications where degree correction is crucial. However, the development of such algorithms is beyond the scope of this article and is left to future investigations.

Analysis of empirical graphs We conclude this article by providing a comparison of the BCCM obtained by fitting the block structures detected by the five community detection algorithms described above on five different real-world networks. The results show that different algorithm performs better for different graphs, highlighting the non-trivial effect that degrees have on block structure and community detection in general.

We study five well-known graphs with heterogeneous characteristics and sizes. All graphs are multi-edge, and are freely available as dataset within the `igraphdata` `R` package. The first graph analyzed is `rfid`: hospital encounter network data. It consists of 32424 undirected edges between 75 individuals (Vanhems et al. 2013). The second graph analyzed is `karate`: Zachary's Karate Club. It consists of 231 undirected edges between 34 vertices (Zachary 1977). The third graph analyzed is `UKfaculty`: Friendship network of a UK university faculty. It consists of 3730 directed edges between 81 vertices (Nepusz et al. 2008). The fourth graph is `USairports`: US airport network of December 2010. It consists of 23473 directed edges between 755 airports (Von Mering et al. 2002). It has self-loops. The graph is plotted in Fig. 6, using the force-atlas2 layout (Jacomy et al. 2014). The four different plots are colored according to the block structures detected by four of the five algorithms (`cluster_spinglass` cannot be applied as the graph is disconnected). They are ordered by increasing AIC. From the visualization, we can see that the best block structure is the one who can separate three different blocks within the largest cluster of vertices (top of the visualizations). In particular, it is essential to note that the largest cluster consists of high- and low-degree vertices. If these vertices are belonging to the same block, the configuration model predicts then high-degree vertices should be connected by many edges (similarly to the first synthetic experiment described above). However, we observe then some of these high-degree vertices are separated and mainly connected to low-degree vertices. For this reason, block structures that can separate these high-degree vertices into different blocks rank higher than others. The fifth graph analyzed is `enron`: Enron Email Network. It consists of 125409 directed edges between 184 individuals (Priebe et al. 2005). It has self-loops.

Each of these graphs has a clear block structure that could be detected. The different algorithms provide different results, both in the number of blocks detected and in the assignment of vertices. Ranking the different results employing the goodness-of-fit of BCCM fitted according to the different block partitions shows that the best results are not necessarily those with fewer or more blocks, nor those obtained from a specific algorithm, as the results change with the graph studied. The results of this analysis are provided in Table 5, where the smallest AICs and BICs for each graph are highlighted in bold, together with the algorithm that provides the smallest number

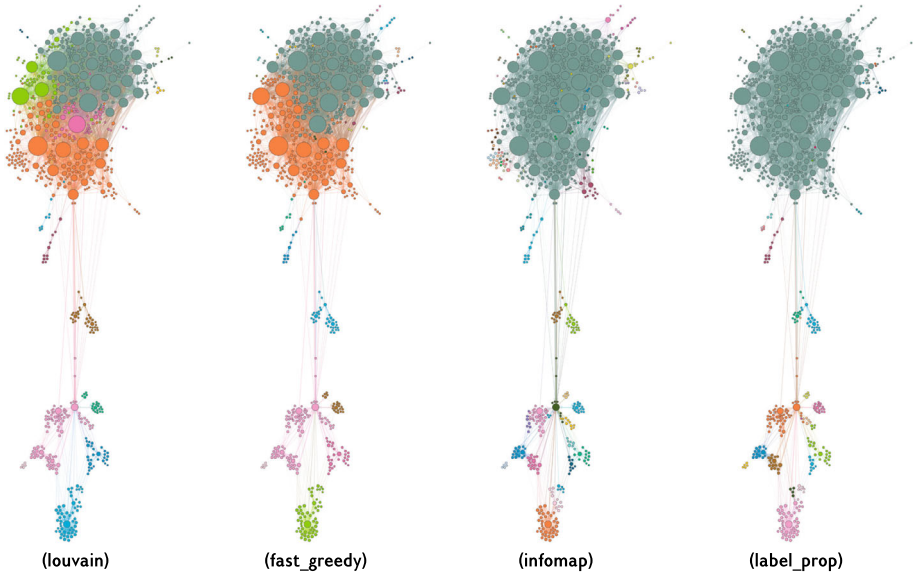

Fig. 6 USairports graph visualisation. `USairports` graph visualisation. The graph is plotted by means of the force-atlas2 layout with weighted edges, and the size of the vertices reflects their out-degrees. Only the largest connected component of the graph is shown. The visualisations clearly show the block structure that characterises this graph. The vertices in the four visualisations are coloured according to the labels detected applying four community detection algorithms, as described in Table 5. The visualisations are ordered from left to right according to the AIC of the BCCM fitted to observed graph according to the corresponding block structure. From left to right, we see the colours corresponding to the labels obtained from louvain, fast_greedy, infomap and label_propagation detection algorithms respectively. We highlight the fact that the ranking according to AIC corresponds approximately to the ability of the algorithms to detect the separation between high-degree (and low-degree) vertices within the largest cluster, at the top of the visualisations. The reason for this is that within the largest cluster there are clear deviations from what the configuration model predicts, i.e., high-degree vertices tend to connect to each other, and the best BCCMs captures more of these deviations

of blocks. The algorithm that provides the largest number of blocks is highlighted in italic.

Conclusion

In this article we have presented a novel generative model for clustered graphs: the block-constrained configuration model. It generalizes the standard configuration model of random graphs by constraining edges within blocks, preserving degree distributions. The BCCM builds on the generalized hypergeometric ensemble of random graphs, by giving the propensity matrix Ω a block structure. The framework provided by gHypEG allows for a fast estimation of the parameters of the model. Moreover, thanks to the fact that the probability distribution underlying gHypEG is known, it allows for the generation of random realizations, as well as to the effortless computation of likelihoods, and hence various kinds of information criteria and goodness-of-fit measures, such as AIC and BIC.

There are many advantages of the formulation highlighted above. Firstly, the proposed model seamlessly applies to directed and undirected graphs with or without self-loops. Moreover, the BCCM preserves exactly the number of edges in the graph, avoiding the need for assuming an arbitrary edge generating process. This also allows dropping the assumption of independence between edge probabilities, which characterizes degree-corrected stochastic block models. Finally, model selection, facilitated by the gHypE

Table 5 Results of the fitting of BCCM to five real-world graphs, with vertex blocks given obtained from five different community detection algorithms

Data specifications					
dataset	vertices	edges	directed	self-loops	
rfid	75	32424	False	False	
karate	34	231	False	False	
UKfaculty	81	3730	True	False	
USairports	755	23473	True	True	
enron	184	125409	True	True	
Number of clusters					
dataset	fast_greedy	infomap	label_prop	spinglass	louvain
rfid	6	4	**3**	*7*	6
karate	**3**	**3**	**3**	*4*	*4*
UKfaculty	**5**	*10*	7	7	**5**
USairports	28	*57*	40	NA	**21**
enron	11	*22*	20	NA	**10**
Δ_i^{AIC}					
dataset	fast_greedy	infomap	label_prop	spinglass	louvain
rfid	1856	12370	13523	**0**	1856
karate	28	28	28	4	**0**
UKfaculty	992	**0**	960	523	992
USairports	1903	2759	5133	NA	**0**
enron	**0**	9881	46945	NA	1956
Δ_i^{BIC}					
dataset	fast_greedy	infomap	label_prop	spinglass	louvain
rfid	1798	12219	13339	**0**	1798
karate	14	14	14	4	**0**
UKfaculty	743	**0**	792	355	743
USairports	3315	14227	9883	NA	**0**
enron	**0**	11702	48347	NA	1849

The first table reports information about the five different graphs used. The second table reports the number of clusters detected by each algorithm for each dataset. The algorithm detecting the smallest number of clusters is highlighted in bold, and the algorithm detecting the largest number of clusters is highlighted in italic. The third table reports AIC differences of the different models computed using the different vertex blocks. The fourth table reports BIC differences of the different models computed using the different vertex blocks. The best model, i.e., the one with the lowest AIC/BIC score, respectively, is highlighted in bold. Because the spin-glass algorithm is not suitable for disconnected graphs, no result is reported for this method for the last two real-world graphs

framework, provides a natural method to quantify the optimal number of blocks needed to model given real-world graph. The statistical significance of a block structure can be studied performing likelihood-ratio tests (Casiraghi et al. 2016), or comparing information criteria such as AIC, BIC, or the description length of the estimated models. Furthermore, within the framework of generalized hypergeometric ensembles block-constrained configuration models can be extended, including heterogeneous properties of vertices or edges (see Casiraghi (2017)).

The more complicated expression and estimation of BCCM compared to DC-SBMs arises from dropping the assumption of independence between edge probabilities. However, thanks to the formulation provided in this article, BCCM is still practicable and can be applied to empirical graphs of various sizes. BCCM opens new routes to develop

community detection algorithms suitable for applications where degree correction is particularly valuable, and where the assumption of an arbitrary edge generating process is not acceptable.

Abbreviations
AIC: Akaike information criterion; BCCM: Block-constrained configuration model; BIC: Bayesian information criterion; DC-SBM: Degree-corrected stochastic block model; gHypEG: Generalised hypergeometric ensemble of random graphs; SBM: Stochastic block model

Acknowledgements
The author thanks Frank Schweitzer for his support and valuable comments, and Laurence Brandenberger, Giacomo Vaccario and Vahan Nanumyan for useful discussions.

Authors' contributions
The author read and approved the final manuscript.

Funding
Not applicable.

Availability of data and materials
The datasets generated and/or analysed during the current study are available as a `gitHub` repository, https://github.com/gi0na/BCCM--Supporting-Material.git. The combinatorial matrices corresponding to Fig. 3b and c are included within the article (and its additional file(s)). A software implementation of the BCCM can be found as part of the `R` package `ghypernet` available at https://github.com/gi0na/ghypernet.git.

Competing interests
The authors declare that they have no competing interests.

Received: 23 July 2019 Accepted: 28 November 2019
Published online: 23 December 2019

References
Akaike H (1974) A new look at the statistical model identification. IEEE Trans Autom Control 19(6):716–723. https://doi.org/10.1109/TAC.1974.1100705
Bender EA, Canfield ER (1978) The asymptotic number of labeled graphs with given degree sequences. J Comb Theory Ser A 24(3):296–307. https://doi.org/10.1016/0097-3165(78)90059-6
Blondel VD, Guillaume J-L, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. J Stat Mech Theory Exp 2008(10):10008
Bozdogan H (1987) Model selection and Akaike's Information Criterion (AIC): The general theory and its analytical extensions. Psychometrika 52(3):345–370. https://doi.org/10.1007/BF02294361
(2004) Model Selection and Multimodel Inference(Burnham KP, Anderson DR, eds.). Springer, New York. https://doi.org/10.1007/b97636. http://link.springer.com/10.1007/b97636
Caldarelli G, Chessa A, Pammolli F, Gabrielli A, Puliga M (2013) Reconstructing a credit network. Nat Phys. https://doi.org/10.1038/nphys2580
Casiraghi G (2017) Multiplex Network Regression: How do relations drive interactions? arXiv preprint arXiv:1702.02048. 1702.02048
Casiraghi G, Nanumyan V (2018) Generalised hypergeometric ensembles of random graphs: the configuration model as an urn problem. 1810.06495
Casiraghi G, Nanumyan V, Scholtes I, Schweitzer F (2016) Generalized Hypergeometric Ensembles: Statistical Hypothesis Testing in Complex Networks. arXiv preprint arXiv:1607.02441. 1607.02441
Chesson J (1978) Measuring Preference in Selective Predation. Ecology 59(2):211–215
Chung F, Lu L (2002) Connected Components in Random Graphs with Given Expected Degree Sequences. Ann Comb 6(2):125–145. https://doi.org/10.1007/PL00012580
Chung F, Lu L (2002) The average distances in random graphs with given expected degrees. Proc Natl Acad Sci 99(25):15879–15882. https://doi.org/10.1073/pnas.252631999
Clauset A, Newman ME, Moore C (2004) Finding community structure in very large networks. Phys Rev E 70(6):066111
Consul PC, Jain GC (1973) A Generalization of the Poisson Distribution. Technometrics 15(4):791–799. https://doi.org/10.1080/00401706.1973.10489112
DeLeeuw J (1992) Introduction to Akaike 1973 Information Theory and an Extension of the Maximum Likelihood Principle:599–609. https://doi.org/10.1007/978-1-4612-0919-5_37. http://link.springer.com/10.1007/978-1-4612-0919-5_37
Erdös P, Rényi A (1959) On random graphs I. Publ Math Debrecen 6:290–297
Fienberg SE, Meyer MM, Wasserman SS (1985) Statistical Analysis of Multiple Sociometric Relations. J Am Stat Assoc 80(389):51–67. https://doi.org/10.1080/01621459.1985.10477129
Fog A (2008) Calculation Methods for Wallenius' Noncentral Hypergeometric Distribution. Commun Stat Sim Comput 37(2):258–273. https://doi.org/10.1080/03610910701790269
Fog A (2008) Sampling Methods for Wallenius' and Fisher's Noncentral Hypergeometric Distributions. Commun Stat Sim Comput 37(2):241–257. https://doi.org/10.1080/03610910701790236
Granovetter MS (1973) The Strength of Weak Ties. Am J Soc 78(6):1360–80. https://doi.org/10.1086/225469. NIHMS150003

Holland PW, Laskey KB, Leinhardt S (1983) Stochastic blockmodels: First steps. Social Networks 5(2):109–137. https://doi.org/10.1016/0378-8733(83)90021-7
Jacomy M, Venturini T, Heymann S, Bastian M (2014) Forceatlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software. PLoS ONE 9(6):98679
Karrer B, Newman MEJ (2011) Stochastic blockmodels and community structure in networks. Phys Rev E 83(1):16107. https://doi.org/10.1103/PhysRevE.83.016107
Krivitsky PN (2012) Exponential-family random graph models for valued networks. Electron J Stat 6:1100–1128. https://doi.org/10.1214/12-EJS696
Kuha J (2004) AIC and BIC. Soc Methods Res 33(2):188–229. https://doi.org/10.1177/0049124103262065
Melnik S, Porter MA, Mucha PJ, Gleeson JP (2014) Dynamics on modular networks with heterogeneous correlations. Chaos Interdiscip J Nonlinear Sci 24(2):023106
Molloy M, Reed B (1995) A critical point for random graphs with a given degree sequence. Random Struct Algoritms 6(2-3):161–180. https://doi.org/10.1002/rsa.3240060204
Molloy M, Reed B (1998) The Size of the Giant Component of a Random Graph with a Given Degree Sequence. Comb Probab Comput 7(3):295–305
Nanumyan V, Garas A, Schweitzer F (2015) The Network of Counterparty Risk: Analysing Correlations in OTC Derivatives. PLoS ONE 10(9):0136638. https://doi.org/10.1371/journal.pone.0136638. 1506.04663
Nepusz T, Petróczi A, Négyessy L, Bazsó F (2008) Fuzzy communities and the concept of bridgeness in complex networks. Phys Rev E 77(1):016107
Newcomb T, Heider F (1958) The Psychology of Interpersonal Relations. Am Soc Rev. https://doi.org/10.2307/2089062. arXiv:1011.1669v3
Newman MEJ, Peixoto TP (2015) Generalized Communities in Networks. Phys Rev Lett 115(8):088701. https://doi.org/10.1103/PhysRevLett.115.088701
Newman MEJ, Reinert G (2016) Estimating the Number of Communities in a Network. Phys Rev Lett 117(7):078301. https://doi.org/10.1103/PhysRevLett.117.078301
Park J, Newman MEJ (2004) Statistical mechanics of networks. Phys Rev E 70(6):066117. https://doi.org/10.1103/PhysRevE.70.066117
Peixoto TP (2012) Entropy of stochastic blockmodel ensembles. Phys Rev E 85(5):056122. https://doi.org/10.1103/PhysRevE.85.056122
Peixoto TP (2014) Hierarchical Block Structures and High-Resolution Model Selection in Large Networks. Phys Rev X 4(1):011047. https://doi.org/10.1103/PhysRevX.4.011047
Peixoto TP (2015) Model Selection and Hypothesis Testing for Large-Scale Network Models with Overlapping Groups. Phys Rev X 5(1):011033. https://doi.org/10.1103/PhysRevX.5.011033
Peixoto, TP (2017) Nonparametric Bayesian inference of the microcanonical stochastic block model. Phys Rev E 95(1):012317. https://doi.org/10.1103/PhysRevE.95.012317
Peixoto TP (2018) Reconstructing Networks with Unknown and Heterogeneous Errors. Phys Rev X 8(4):041011. https://doi.org/10.1103/PhysRevX.8.041011
Priebe CE, Conroy JM, Marchette DJ, Park Y (2005) Scan statistics on enron graphs. Comput Math Org Theory 11(3):229–247
Raghavan UN, Albert R, Kumara S (2007) Near linear time algorithm to detect community structures in large-scale networks. Phys Rev E 76(3):036106
Reichardt J, Bornholdt S (2006) Statistical mechanics of community detection. Phys Rev E 74(1):016110
Rosvall M, Bergstrom CT (2008) Maps of random walks on complex networks reveal community structure. Proc Natl Acad Sci 105(4):1118–1123
Schwarz G, et al (1978) Estimating the dimension of a model. Ann Stat 6(2):461–464
Squartini T, Mastrandrea R, Garlaschelli D (2015) Unbiased sampling of network ensembles. N J Phys. 10.1088/1367-2630/17/2/023052. 1406.1197
Stone CJ (1982) Local asymptotic admissibility of a generalization of Akaike's model selection rule. Ann Inst Stat Math 34(1):123–133. https://doi.org/10.1007/BF02481014
Vanhems P, Barrat A, Cattuto C, Pinton J-F, Khanafer N, Régis C, Kim B-a, Comte B, Voirin N (2013) Estimating potential infection transmission routes in hospital wards using wearable proximity sensors. PloS ONE 8(9):73970
Von Mering C, Krause R, Snel B, Cornell M, Oliver SG, Fields S, Bork P (2002) Comparative assessment of large-scale data sets of protein–protein interactions. Nature 417(6887):399
Wallenius KT (1963) Biased Sampling: the Noncentral Hypergeometric Probability Distribution. Ph.d. thesis. https://doi.org/10.21236/ad0426243
Zachary WW (1977) An Information Flow Model for Conflict and Fission in Small Groups. J Anthropol Res 33(4):452–473

Publisher's Note