Model-based clustering of multiple networks with a hierarchical algorithm

Tabea Rebafka

manuscript No.
(will be inserted by the editor)

Model-based clustering of multiple networks with a hierarchical algorithm

Tabea Rebafka

Received: date / Accepted: date

Abstract The paper tackles the problem of clustering multiple networks, directed or not, that do not share the same set of vertices, into groups of networks with similar topology. A statistical model-based approach based on a finite mixture of stochastic block models is proposed. A clustering is obtained by maximizing the integrated classification likelihood criterion. This is done by a hierarchical agglomerative algorithm, that starts from singleton clusters and successively merges clusters of networks. As such, a sequence of nested clusterings is computed that can be represented by a dendrogram providing valuable insights on the collection of networks. Using a Bayesian framework, model selection is performed in an automated way since the algorithm stops when the best number of clusters is attained. The algorithm is computationally efficient, when carefully implemented. The aggregation of clusters requires a means to overcome the label-switching problem of the stochastic block model and to match the block labels of the networks. To address this problem, a new tool is proposed based on a comparison of the graphons of the associated stochastic block models. The clustering approach is assessed on synthetic data. An application to a set of ecological networks illustrates the interpretability of the obtained results.

Keywords Graph clustering, multiple networks, stochastic block model, agglomerative algorithm, graphon distance, integrated classification likelihood.

Tabea Rebafka
Sorbonne Université, Université Paris Cité, CNRS, Laboratoire de Probabilités, Statistique et Modélisation (LPSM), Paris, France
INRAE, MaIAGE, Jouy-en-Josas, France E-mail: tabea.rebafka@sorbonne-universite.fr

1 Introduction

Networks are key objects for describing interactions between individuals or entities in complex systems. Today, entire collections of networks emerge in more and more fields of application. To list a few examples, in social sciences face-to-face contacts among individuals at different time periods are represented as a set of behavioral networks (Isella et al., 2011). In medical research, a brain connectome is a network describing a patient's brain activity (Donnat and Holmes, 2018). In biology, metabolic networks for hundreds of different bacteria are available (Weber-Zendrera et al., 2021). In ecology, foodwebs represent the interactions of species in different ecosystems (Poisot et al., 2016).

When analyzing multiple networks, most questions are related to graph comparison. We may wish to quantify the (dis)similarity between networks, detect outliers or some temporal evolution of networks. In general, it is informative to reduce the dimension of the data by finding groups of networks sharing similar characteristics. For instance, we may want to automatically group together patients with the same brain state, or identify bacteria with roughly the same metabolism, or in the context of climate change find ecological networks with similar overall organization. The focus of this work is on clustering of networks, which are directed or undirected and that may not share the same set of vertices and may vary in size . We seek a method that partitions the networks according to their topology.

1.1 Graph comparison

The clustering task requires some notion of graph similarity. However, networks have complex structure, and

so graph comparison is not trivial and similarity or graph distances can be defined in many ways. A widespread approach is based on graph embeddings. A graph embedding is a low-dimensional vector representation of a network encoding structural information about the network. Traditional embeddings are hand-crafted and composed of local or global network summary statistics like the edge density, node degrees and clustering coefficients. Then, graph similarity is defined using the distance between the embedding vectors, and graph clustering is easily performed using off-the-shelf machine learning algorithms as k-means or spectral clustering. Clearly, the clustering result heavily depends on the chosen embedding.

The machine learning literature proposes many alternative graph embeddings, as for instance graph kernel methods (Gärtner, 2003; Shervashidze et al., 2009), graph Laplacian methods (Shimada et al., 2016), extensions of node embeddings (Hamilton et al., 2017), graph neural networks (Xu et al., 2019; Wu et al., 2021) and data-driven methods based on graphlets (le Gorrec et al., 2022). However, in practice it is far from evident how to choose the most suitable embedding (Botella et al., 2022).

An alternative to graph embeddings are model-based approaches. Here a statistical model is introduced and networks forming a cluster are assumed to be generated independently from a common probabilistic model. To put it differently, data are modeled by a finite mixture model of random graph models and mixture components correspond to clusters of networks. The problem of graph comparison is thus recast as a problem of estimating and comparing the probabilistic models that generated the observed networks (Stanley et al., 2016; Sabanayagam et al., 2022).

A major advantage of model-based approaches over graph embeddings is the possibility to quantify uncertainty of the results. For instance, one may compute the posterior probability for a network to belong to a given cluster or compare the likelihoods of two clusterings. Furthermore, it provides a natural framework for model selection, that is, the automated choice of the best number of clusters.

For graph comparison, two general settings must be distinguished. In the first case, all networks are defined on the same set of vertices, as for networks with a temporal dynamic or brain connectomes, where a vertex always refers to the same brain region. Then graph distances can be based on local features comparing structures of node neighborhoods. In the second case, the sets of vertices are completely different from one network to another, without any correspondence among the nodes of the different networks. This is the setting we are interested in and which is far less explored in the literature. The mangal database (Poisot et al., 2016), for example, provides hundreds of foodwebs from all over the globe, where each foodweb describes an ecosystem coming with its own set of species. To compare such networks, local features are useless, and only the overall topology of the networks is meaningful.

1.2 Mixture models for sets of networks

Using finite mixtures to perform clustering has a longstanding tradition (Titterington et al., 1985; McLachlan and Peel, 2000), but only recently, this approach has been explored for graph clustering. To define a mixture model, a random graph model for the mixture components has to be chosen. For networks with constant node sets, the stochastic block model and generealized linear models may be used (Stanley et al., 2016; Signorelli and Wit, 2019), or extensions of measurement error models, where networks are considered to be perturbations of some ground-truth graph (Mantziou et al., 2023; Young et al., 2022). Mukherjee et al. (2017) and Sabanayagam et al. (2022) propose nonparametric models, where the distribution of the mixture components is estimated by a graphon estimate. Shortcomings of the latter approach include the restriction to undirected graphs and the lack of interpretation, since analyzing graphons is not convenient.

In this paper, a new mixture model is proposed. As we desire an interpretable model, we choose the popular stochastic block model (SBM) (Nowicki and Snijders, 2001) for the mixture components. The SBM is a highly flexible model, which accommodates a large variety of heterogeneous graph topologies as often encountered in applications. A further advantage is the interpretability of the parameters of a SBM. Many model variants exist (see Matias and Robin (2014) for a review), which underlines the relevance of SBM. In particular, extensions of the SBM for sets of networks include repeated measurements of a ground-truth SBM network (Le et al., 2018), a mixture of SBMs with fixed nodes (Stanley et al., 2016), and networks that are generated by SBMs with varying parameters (Chabert-Liddell et al., 2022).

The SBM is a discrete latent variable model and parameter estimation is challenging due to its involved dependence structure. Several inference algorithms have been proposed like variational EM-algorithms (Daudin et al., 2008), MCMC methods (Nowicki and Snijders, 2001; Peixoto, 2014), a pseudo-likelihood approach (Amini et al., 2013), a Bayesian approach based on the integrated classification likelihood (ICL) (Côme and Latouche, 2015), spectral clustering (Rohe et al., 2011)

and, more recently, a variational autoencoder using neural networks (Mehta et al., 2019). None of them is perfect, some are time-consuming and not scalable to large networks, others are fast, but provide unstable results.

1.3 Graph clustering algorithms

A simple clustering approach is based on graph distances. That is, one computes a similarity matrix for the pairwise comparison of the networks and then a clustering is derived via spectral clustering (Mukherjee et al., 2017; Sabanayagam et al., 2022). This approach does not account for the uncertainty of estimates and lacks a natural model selection device.

In a mixture model the clustering task becomes an inference problem, since cluster labels correspond to latent variables of the model. In general model-based clustering, EM-type algorithms (McLachlan and Krishnan, 2008), MCMC (Liu, 2008) and hierarchical agglomerative algorithms (Fraley and Raftery, 2002) are traditionally used to jointly infer cluster labels and model parameters. In the case of graph clustering, for mixtures of networks with a constant node set, EM algorithms are developed (Stanley et al., 2016; Signorelli and Wit, 2019) as well as Gibbs samplers (Young et al., 2022; Mantziou et al., 2023). Among these methods only the one by Mantziou et al. (2023) includes the inference of the number of clusters in the algorithm by using a sparse finite mixture in a Bayesian framework (Frühwirth-Schnatter and Malsiner-Walli, 2019). All other methods have the disadvantage that they require the specification of the number of clusters. Then model selection is performed in an exploratory way by running the algorithm with different numbers of clusters and then comparing the solutions with an appropriate criterion.

In the present work, we explore the development of a hierarchical agglomerative algorithm. Starting from an oversegmented clustering with singleton clusters, clusters are successively merged to larger clusters while optimizing some criterion. Interestingly, the algorithm provides a whole cluster hierarchy that can be visualized by a dendrogram and intermediate clusterings are easily inspected. If the criterion includes a penalization of the number of clusters, the algorithm automatically stops when any further cluster aggregation results in a deterioration of the objective. Thus, model selection is performed automatically. Such penalized criterions are naturally obtained by using Bayes factors (Robert, 2007).

For our mixture model of SBMs, we follow the line of research initiated by Côme and Latouche (2015) that consists in choosing the integrated classification likelihood (ICL) as the objective for the hierarchical agglomerative algorithm. We show that the algorithm can be implemented efficiently and assess its performance by numerical experiments.

1.4 Block-label matching

In our algorithm an interesting issue is encountered during the aggregation of two clusters. Indeed, merging clusters amounts to combine the corresponding SBMs. However, due to the label-switching problem in the SBM, this is not simple. Using the graphon functions (Lovász and Szegedy, 2006) of the SBMs, we propose a new tool to match block labels in a computationally efficient way. This tool should also be of interest beyond our clustering algorithm, whenever two SBMs are compared and the problem of label-switching occurs.

1.5 Contributions

The contributions of the paper are as follows.

- A finite mixture model of SBMs is introduced for sets of networks that do not share the same vertices and not even the same number of vertices, applying to both directed and undirected graphs (Section 2).
- A hierarchical agglomerative algorithm to cluster networks and estimate model parameters is developed (Section 3 and 4).
- We propose a new tool to match block labels of two SBMs (Section 5).
- A numerical study assesses the performance of the algorithm and illustrates its utility on a collection of foodwebs (Section 6).

2 Mixture of stochastic block models

In this section we first recall the definition of the classical SBM for a single network. Then we introduce the mixture of SBMs for a collection of networks without vertex correspondence. Throughout the paper we consider directed binary networks without self-loops, but extensions to other types of networks are straightforward.

2.1 Stochastic block model for a single network

Consider a network with n vertices. Denote $(\boldsymbol{\pi}, \boldsymbol{\gamma})$ the parameters of a SBM with K blocks, where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K) \in (0,1)^K$ are the block proportions with $\sum_{k \in [\![K]\!]} \pi_k = 1$

and $\boldsymbol{\gamma} = (\gamma_{k,l})_{k,l} \in (0,1)^{K\times K}$ the connectivity matrix. Let $\mathbf{Z} = (Z_1, \ldots, Z_n) \in [\![K]\!]^n$ be a vector of independent discrete latent variables for the nodes, with $\mathbb{P}(Z_i = k) = \pi_k$ for all $k \in [\![K]\!]$ and $i \in [\![n]\!]$. Conditionally on the node labels $\mathbf{Z}$, the observed adjacency matrix $A = (A_{i,j})_{1\le i,j\le n} \in \{0,1\}^{n\times n}$ verifies

$$A|\mathbf{Z} = \bigotimes_{i\neq j} A_{i,j}|Z_i, Z_j = \bigotimes_{i\neq j} \mathcal{B}\left(\gamma_{Z_i,Z_j}\right),$$

where $\mathcal{B}(\gamma)$ is the Bernoulli distribution. We denote the distribution of A by $\mathcal{SBM}_n(\boldsymbol{\pi}, \boldsymbol{\gamma})$.

2.2 Mixture of SBMs for a collection of networks

Now we consider a collection of networks modeled by a finite mixture model, where each mixture component is a SBM. That is, networks belonging to the same cluster are independent realizations of the same SBM.

Formally, let $\mathcal{A} = \{A^{(m)}, m \in [\![M]\!]\}$ be a collection of M networks, where $A^{(m)} = (A^{(m)}_{i,j})_{1\le i,j\le n^{(m)}} \in \{0,1\}^{n^{(m)}\times n^{(m)}}$ denotes the adjacency matrix of the m-th network. Networks may have different numbers $n^{(m)}$ of vertices and no correspondence among the nodes is assumed. We introduce independent discrete latent variables $\mathcal{U} = (U^{(1)}, \ldots, U^{(M)}) \in [\![C]\!]^M$ defining a partitioning of the M networks into $C \ge 1$ clusters. Denote $p_c = \mathbb{P}(U^{(m)} = c), c \in [\![C]\!]$ the cluster proportions and $\mathbf{p} = (p_1, \ldots, p_C) \in (0,1)^C$. Now, let $(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)}), c \in [\![C]\!]$ be parameters of C different SBMs. The associated numbers of blocks, say K_c, are not constrained to be equal. We assume that all networks in cluster c are independent realizations of the SBM with parameter $(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)})$, that is, conditionally on $\mathcal{U}$,

$$\begin{aligned}\mathcal{A}|\mathcal{U} &= \bigotimes_{m=1}^{M} A^{(m)}|U^{(m)} \\ &= \bigotimes_{m=1}^{M} \mathcal{SBM}_{n^{(m)}}\left(\boldsymbol{\pi}^{(U^{(m)})}, \boldsymbol{\gamma}^{(U^{(m)})}\right).\end{aligned}$$

Denote $\theta = \left(\mathbf{p}, \{(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)}), c \in [\![C]\!]\}\right)$ the parameters of the mixture model, and note that θ is identifiable only up to label switching. That is, switching cluster labels always results in the same probability distribution of $\mathcal{A}$. In addition, in every SBM, the node labels are also identifiable only up to label switching. We adapt the notation of the node labels by adding superscript $^{(m)}$, that is, $\mathbf{Z}^{(m)} = (Z^{(m)}_1, \ldots, Z^{(m)}_{n^{(m)}})$, and also denote $\mathcal{Z} = \{\mathbf{Z}^{(m)}, m \in [\![M]\!]\}$.

3 Clustering and estimation using the ICL criterion

In a mixture of SBMs, graph clustering becomes the recovery of the latent variables $\mathcal{U}$ from the data $\mathcal{A}$. We develop a clustering algorithm by maximizing the so-called integrated classification likelihood criterion (ICL), defined as the log-likelihood function of the complete data, that is, the observations and the latent variables. Traditionally, this criterion has been used for model selection in various latent variable models, often in connection with the EM algorithm (Biernacki et al., 2000). More recently, Côme and Latouche (2015) showed that the ICL can also be used for directly estimating the latent variables. Compared to alternative approaches like EM, an unequivocal advantage is that model selection is performed automatically. Here we adapt the approach to mixtures of SBMs. In this section, the ICL is first introduced for a single cluster, then defined for our mixture model.

3.1 ICL criterion for a single cluster

In this subsection $\mathcal{A}$ is assumed to be a collection of i.i.d. networks of a SBM with K blocks and parameters $(\boldsymbol{\pi}, \boldsymbol{\gamma})$. Considering a Bayesian framework, let $p(\boldsymbol{\pi}, \boldsymbol{\gamma})$ be a prior distribution on the SBM parameters and define the ICL criterion as

$$\begin{aligned}\mathrm{ICL}^{\mathrm{sbm}}(\mathcal{A}, \mathcal{Z}) &= \log(p(\mathcal{A}, \mathcal{Z})) \\ &= \log\left(\int p(\mathcal{A}, \mathcal{Z}|\boldsymbol{\pi}, \boldsymbol{\gamma}) p(\boldsymbol{\pi}, \boldsymbol{\gamma}) \mathrm{d}(\boldsymbol{\pi}, \boldsymbol{\gamma})\right).\end{aligned}$$

Interestingly, by integrating out the model parameters, the criterion only depends on the observations $\mathcal{A}$ and the latent nodel labels $\mathcal{Z}$. The value of $\mathcal{Z}$ optimizing the ICL, that is,

$$\hat{\mathcal{Z}} = \arg\max_{\mathcal{Z}} \mathrm{ICL}^{\mathrm{sbm}}(\mathcal{A}, \mathcal{Z}),$$

corresponds to the node labels maximizing the posterior distribution of $\mathcal{Z}$ and hence is a natural estimate of the latent variables. Using the following prior

$$\begin{aligned}p(\boldsymbol{\pi}, \boldsymbol{\gamma}) &= p(\boldsymbol{\pi}) \times \prod_{k,l\in[\![K]\!]^2} p(\gamma_{k,l}) \\ &= \mathrm{Dir}(\boldsymbol{\pi}; \alpha_1, \ldots, \alpha_K) \times \prod_{k,l\in[\![K]\!]^2} \mathrm{Beta}(\gamma_{k,l}; \eta_{k,l}, \zeta_{k,l}),\end{aligned}$$

where $\alpha_1, \ldots, \alpha_K, \eta_{k,l}, \zeta_{k,l}$ are hyperparameters of the Dirichlet and the Beta distributions, the $\mathrm{ICL}^{\mathrm{sbm}}$ has closed-form expression. For simplicity, hyperparameters for all priors are set to identical values, that is, $\alpha = \alpha_k$, $\eta = \eta_{k,l}$ and $\zeta = \zeta_{k,l}$ for $(k,l) \in [\![K]\!]^2$. To state the

$\mathrm{ICL}^{\mathrm{sbm}}$, we use the one-hot encoding for node labels $Z_i^{(m)} = (Z_{i,1}^{(m)}, \ldots, Z_{i,K}^{(m)}) \in \{0,1\}^K$ and the following count statistics for the m-th network

$$s_k^{(m)} = \sum_{i \in [\![n]\!]} Z_{i,k}^{(m)}, \qquad a_{k,l}^{(m)} = \sum_{i \neq j} Z_{i,k}^{(m)} Z_{j,l}^{(m)} A_{i,j}^{(m)},$$
$$b_{k,l}^{(m)} = \sum_{i \neq j} Z_{i,k}^{(m)} Z_{j,l}^{(m)} (1 - A_{i,j}^{(m)}),$$

where $s_k^{(m)}$ is the number of vertices assigned to block k, $a_{k,l}^{(m)}$ the number of edges that link a vertex of block k with a vertex in block l and $b_{k,l}^{(m)}$ is the number of pairs with a vertex of block k and a vertex in block l that are not connected. Moreover, denote

$$\mathbf{s}_k = \sum_{m \in [\![M]\!]} s_k^{(m)}, \mathbf{a}_{k,l} = \sum_{m \in [\![M]\!]} a_{k,l}^{(m)}, \mathbf{b}_{k,l} = \sum_{m \in [\![M]\!]} b_{k,l}^{(m)}.$$

With these notations at hand, the ICL is given by

$$\begin{aligned}
&\mathrm{ICL}^{\mathrm{sbm}}(\mathcal{A}, \mathcal{Z}) \\
&= \sum_{(k,l) \in \in [\![K]\!]^2} \log\left(\frac{\Gamma(\eta + \mathbf{a}_{k,l})\Gamma(\zeta + \mathbf{b}_{k,l})}{\Gamma(\eta + \zeta + \mathbf{a}_{k,l} + \mathbf{b}_{k,l})}\right) \\
&\quad + \sum_{k \in [\![K]\!]} \log\left(\Gamma(\alpha + \mathbf{s}_k)\right) + K^2 \log\left(\frac{\Gamma(\eta + \zeta)}{\Gamma(\eta)\Gamma(\zeta)}\right) \\
&\quad + \log\left(\frac{\Gamma(K\alpha)}{\Gamma\left(K\alpha + \sum_m n^{(m)}\right)}\right) - K \log\left(\Gamma(\alpha)\right).
\end{aligned}$$

3.2 ICL criterion for a mixture of SBMs

In a mixture of SBMs, there are two types of latent variables, namely the clustering $\mathcal{U}$ of the networks and the node labels $\mathcal{Z}$. The ICL is then defined as

$$\begin{aligned}
\mathrm{ICL}^{\mathrm{mix}}(\mathcal{A}, \mathcal{U}, \mathcal{Z}) &= \log(p(\mathcal{A}, \mathcal{U}, \mathcal{Z})) \\
&= \log\left(\int p(\mathcal{A}, \mathcal{U}, \mathcal{Z} | \theta) p(\theta) \mathrm{d}\theta\right),
\end{aligned}$$

where $p(\theta)$ is a prior on the model parameters. The values $(\hat{\mathcal{U}}, \hat{\mathcal{Z}})$ that maximize the ICL are convenient estimates of the graph clustering and the node labels. They are defined as

$$(\hat{\mathcal{U}}, \hat{\mathcal{Z}}) = \arg\max_{\mathcal{U}, \mathcal{Z}} \mathrm{ICL}^{\mathrm{mix}}(\mathcal{A}, \mathcal{U}, \mathcal{Z}). \tag{1}$$

Again we consider classical independent conjugate priors given by

$$\begin{aligned}
p(\theta) &= p(\mathbf{p}) \prod_{c \in [\![C]\!]} p(\boldsymbol{\pi}^{(c)}) p(\boldsymbol{\gamma}^{(c)}) \\
&= \mathrm{Dir}(\mathbf{p}; \lambda_1, \ldots, \lambda_C) \prod_{c \in [\![C]\!]} \mathrm{Dir}(\boldsymbol{\pi}^{(c)}; \alpha_1, \ldots, \alpha_{K_c}) \\
&\quad \times \prod_{(k,l) \in [\![K_c]\!]^2} \mathrm{Beta}(\gamma_{k,l}^{(c)}; \eta_{k,l}, \zeta_{k,l}),
\end{aligned}$$

where $\lambda_c, \alpha_k, \eta_{k,l}, \zeta_{k,l}$ are hyperparameters. Let I_c be the set of indices of networks belonging to cluster c, that is, $I_c = \{m \in [\![M]\!] : U^{(m)} = c\}$ for $c \in [\![C]\!]$, and denote $\mathcal{A}^{(c)} = \{A^{(m)}, m \in I_c\}$ and $\mathcal{Z}^{(c)} = \{\mathbf{Z}^{(m)}, m \in I_c\}$. Then, one can show that the ICL can be rewritten as

$$\begin{aligned}
&\mathrm{ICL}^{\mathrm{mix}}(\mathcal{A}, \mathcal{U}, \mathcal{Z}) \\
&= \sum_{c \in [\![C]\!]} \mathrm{ICL}^{\mathrm{sbm}}(\mathcal{A}^{(c)}, \mathcal{Z}^{(c)}) + \log\left(\int p(\mathcal{U}|\mathbf{p}) p(\mathbf{p}) \mathrm{d}\mathbf{p}\right).
\end{aligned}$$

The last term on the right-hand side has closed form given by

$$\begin{aligned}
&\log\left(\int p(\mathcal{U}|\mathbf{p}) p(\mathbf{p}) \mathrm{d}\mathbf{p}\right) \\
&= \log\left(\frac{\Gamma(C\lambda)}{(\Gamma(\lambda))^C \Gamma(C\lambda + M)}\right) + \sum_{c \in [\![C]\!]} \log\left(\Gamma(\lambda + |I_c|)\right).
\end{aligned}$$

The ICL criterion is not exactly a similarity measure that compares clusters of networks, but it is a model-based likelihood criterion that defines what the best clustering is.

4 Hierarchical clustering algorithm

First the general structure of the new clustering algorithm is presented. Then we give more details on some parts of the algorithm.

4.1 General structure of the algorithm

To solve the discrete optimization problem given by (1), we propose a greedy hill-climbing algorithm. The algorithm is initialized by a mixture of M SBMs by setting $U^{(m)} = m$ for $m \in [\![M]\!]$, that is, every network forms a cluster on its own. Then, at every iteration, two clusters are combined to a single larger cluster. More precisely, for any pair of clusters $(c, c') \in [\![C]\!]^2$, the ICL variation $\Delta_{c,c'}$ is evaluated defined as

$$\Delta_{c,c'} = \mathrm{ICL}^{\mathrm{mix}}(\mathcal{A}, \mathcal{U}_{c \cup c'}, \mathcal{Z}_{c \cup c'}) - \mathrm{ICL}^{\mathrm{mix}}(\mathcal{A}, \mathcal{U}, \mathcal{Z}),$$

where $\mathcal{U}$ and $\mathcal{Z}$ are the current latent variables and $\mathcal{U}_{c \cup c'}$ and $\mathcal{Z}_{c \cup c'}$ the ones obtained by merging the clusters c and c'. Finally, the cluster aggregation yielding the largest ICL increase is actually performed. The algorithm stops automatically when the ICL would decrease if any further clusters are merged. The granularity of the final clustering $\hat{\mathcal{U}}$ depends on the data and on the hyperparameters λ_c, see Section 6.2 for a discussion of this point.

The algorithm also requires initial values for the latent node labels $\mathcal{Z}$. We propose to adjust a simple SBM

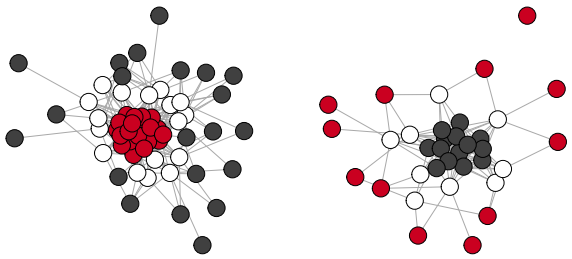


Fig. 1: Illustration of the non-identifiability of block labels in the SBM on two networks with similar topology. Colors indicate the block labels $\mathbf{Z}^{(m)}$.

to each network $A^{(m)}$ yielding an estimate $(\boldsymbol{\pi}^{(m)}, \boldsymbol{\gamma}^{(m)})$ of the SBM parameters as well as node labels $\mathbf{Z}^{(m)}$. Our implementation uses the variational EM algorithm of the R package `blockmodels` (Leger, 2016), which performs an automatic selection of the number of SBM blocks by running the algorithm several times with different numbers of blocks and comparing those solutions using a classical ICL-type criterion.

The aggregation of two clusters raises an issue related to the non-identifiability of the block labels in a SBM. In fact, it occurs that node labels in the two clusters do not refer to the same type of blocks as illustrated in Figure 1. Here, once the community is in red, once in black indicating that different block labels are used for the same type of block. However, in our algorithm, for a given cluster, node labels must designate the same SBM block in every network. If this is not the case, it is necessary to relabel the nodes before merging the clusters. In Section 5 we develop a new tool to find the best correspondence of the block labels of two SBMs.

After merging two clusters, the current node labels can be further improved by searching the maximum of $\mathrm{ICL}^{\mathrm{mix}}$ in $\mathcal{Z}$, while keeping the clustering $\mathcal{U}$ fixed. This amounts to maximize the term $\mathrm{ICL}^{\mathrm{sbm}}$ for the newly created cluster. We propose an adaptation of the procedure by Côme and Latouche (2015) to fit a SBM to a single network. Roughly, for every node we test if changing its node label increases the ICL or not. See Section 4.2 for details.

Algorithm 1 summarizes the entire clustering algorithm. It provides the best clustering $\hat{\mathcal{U}}$, node labels $\hat{\mathcal{Z}}$ and also parameter estimates for the SBM of every cluster.

Algorithm 1 Agglomerative algorithm for graph clustering

Input: Collection of networks $\mathcal{A}$.
Set $U^{(m)} = m$ for $m \in [\![M]\!]$ and set $C = M$.
for $m \in [\![M]\!]$ **do**
  Fit a SBM to $A^{(m)}$ yielding parameters $(\boldsymbol{\pi}^{(m)}, \boldsymbol{\gamma}^{(m)})$ and node labels $\mathbf{Z}^{(m)}$.
end for
Set $\mathcal{Z} = \{\mathbf{Z}^{(m)}, m \in [\![M]\!]\}$ and $\boldsymbol{\theta} = \{(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)}), c \in [\![C]\!]\}$.
while $C > 1$ **do**
  for $(c, c') \in [\![C]\!]^2$ **do**
    Compute $\Delta_{c,c'}$ according to Section 4.3.
  end for
  Choose (c_1, c_2) such that $\Delta_{c_1,c_2} = \max_{c,c'} \Delta_{c,c'}$.
  if $\Delta_{c_1,c_2} > 0$ **then**
    Set $U^{(m)} = \min\{c_1, c_2\}$ for all $m \in I_c \cup I_{c'}$.
    Update $\mathcal{Z}$ and $\boldsymbol{\theta}$ according to Algorithm 3.
    Set $C = C - 1$.
  else
    exit **while**
  end if
end while
Output: Clustering $\mathcal{U} = \{U^{(m)}, m \in [\![M]\!]\}$, node labels $\mathcal{Z}$, SBM parameters $\{(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)}), c \in [\![C]\!]\}$.

4.2 Update of node labels

After aggregating two clusters and relabeling the nodes, we can further improve node labels $\mathcal{Z}^{(c)}$ of the new cluster c by maximizing the associated ICL criterion $\mathrm{ICL}^{\mathrm{sbm}}$. We propose an adaptation of the algorithm by Côme and Latouche (2015), that fits a SBM to a single network, to multiple networks. Indeed, the proposed procedure is an algorithm to adjust one SBM to a collection of i.i.d. networks. The idea is to randomly choose a vertex and search its best block assignment in terms of the ICL. So, one by one, node labels are changed until no other swap would further improve the ICL. In the context of graph clustering, the convergence of this procedure is fast, since the current node labels are very good initial points.

For notational convenience, we drop superscript $^{(c)}$ of $\mathcal{A}^{(c)}$ and $\mathcal{Z}^{(c)}$ and simply write $\mathcal{A}$ and $\mathcal{Z}$, as all computations in this section only involve quantities related to the cluster under consideration. Now, an iteration of the procedure consists of the following steps. First, select a network indice, say $m^* \in [\![M]\!]$, and one of its vertices, say $i^* \in [\![n^{(m)}]\!]$. Denote $g = Z_{i^*}^{(m^*)}$ the current block assignment of i^*. For any block $h \in [\![K]\!]$ compute the impact on the ICL of moving node i^* to block h, that is,

$$\Delta_{m^*,i^*}^{\to h} = \mathrm{ICL}^{\mathrm{sbm}}(\mathcal{A}, \mathcal{Z}_{m^*,i^*}^{\to h}) - \mathrm{ICL}^{\mathrm{sbm}}(\mathcal{A}, \mathcal{Z}),$$

where $\mathcal{Z}$ denotes the current node labels with $Z_{i^*}^{(m^*)} = g$, and $\mathcal{Z}_{m^*,i^*}^{\to h}$ the labels after moving node i^* to block h, that is, $Z_{i^*}^{(m^*)} = h$. Finally, we choose the best block

assignment as

$$h^* = \arg\max_{h \in [\![K]\!]} \Delta^{\to h}_{m^*,i^*},$$

and set $Z_{i^*}^{(m^*)} = h^*$.

For the efficient computation of the ICL changes $\Delta^{\to h}_{m^*,i^*}$, two cases have to be distinguished: moving node i^* to block h (i) does not empty block g; (ii) does empty block g and so the number of blocks K diminishes.

First case: K does not change. First, a look on the evolution of the count statistics $s_k^{(m^*)}$, $a_{k,l}^{(m^*)}$ and $b_{k,l}^{(m^*)}$ induced by the swap shows that some of them only change by a simple additive term and the others remain identical. In particular, the count statistics that are affected by the swap can be efficiently updated from their current values. Likewise, only a small part of the terms of the criterion $\mathrm{ICL}^{\mathrm{sbm}}$ are affected leading to a formula with few terms for a fast evaluation of the ICL variation $\Delta^{\to h}_{m^*,i^*}$. See the Appendix for all details.

Second case: K changes. Moving the last vertex i^* to another block, diminishes the number K of blocks by one. Before giving the formula of $\Delta^{\to h}_{m^*,i^*}$ in this case, we have a closer look on the ICL criterion to better understand its dependency on the model size K. Let us compare the value of the ICL for a SBM with K blocks containing an empty block to the ICL value of the same data, but with the SBM where the empty block is deleted, that is, a SBM with $K-1$ blocks. The relation is given by

$$\mathrm{ICL}^{\mathrm{sbm}}(K) = \mathrm{ICL}^{\mathrm{sbm}}(K-1) + \log \frac{\Gamma(K\alpha)}{\Gamma((K-1)\alpha)} + \log \frac{\Gamma((K-1)\alpha + \sum_m n^{(m)})}{\Gamma(K\alpha + \sum_m n^{(m)})}. \quad (2)$$

The second and third term on the right-hand side are a penalty or the price to pay for using a larger model containing an empty block. Thus, by maximizing the ICL, parsimonious models are automatically favored. Now, the change of the ICL $\Delta^{\to h}_{m^*,i^*}$ is exactly the same term as in the first case, where K does not change, given in (6) in the Appendix, plus the penalty term given in (2).

The whole procedure to update node labels is summarized in Algorithm 2. In the implementation in the `graphclust` package, iterations are grouped together to epochs, where during one epoch all nodes of all networks are visited exactly once in a random order. The algorithm stops when a given maximal number of epochs is attained, or when during one epoch no node changed the block.

Algorithm 2 ICL maximization algorithm for fitting one SBM to multiple networks

Input: Set of networks $\mathcal{A}$, initial node labels $\mathcal{Z}$.
while not converged **do**
  Select a network $m^* \in [\![M]\!]$ and one of its vertices $i^* \in [\![n^{(m^*)}]\!]$.
  for $h \in [\![K]\!]$ **do**
    Compute the impact $\Delta^{\to h}_{m^*,i^*}$ on the ICL of moving node i^* to block h.
  end for
  Determine the best block assignment $h^* = \arg\max_{h \in [\![K]\!]} \Delta^{\to h}_{m^*,i^*}$.
  Set $Z_{i^*}^{(m^*)} = h^*$.
end while
Output: Updated node labels $\mathcal{Z}$.

4.3 Efficient computation of $\Delta_{c,c'}$

In view of the computing time, it is important that the evaluation of ICL variations $\Delta_{c,c'}$ is fast, as it is done at every iteration for every pair of clusters (c, c'). An inspection of the above expression reveals that only the last two terms depend on the current number of clusters C. In addition, the other terms do not change from one iteration to another if both c and c' have not been changed in the previous iteration, that is, if none of them is the result of the latest cluster aggregation. Hence, for those clusters the new value of $\Delta_{c,c'}$ is the previous value plus constant κ_C defined as

$$\kappa_C = -\beta(C\lambda, \lambda) - \log\left(\frac{\Gamma((C+1)\lambda + M)}{\Gamma(C\lambda + M)}\right) + \beta((C-1)\lambda, \lambda) + \log\left(\frac{\Gamma(C\lambda + M)}{\Gamma((C-1)\lambda + M)}\right),$$

where $\beta(x, y) = \log\left(\frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}\right)$ is the logarithm of the Beta function of x and y and C is the number of clusters that has diminished by 1 compared to the previous iteration. In short, for all pairs of clusters (c, c') where both clusters have remained unchanged in the previous iteration, the update is simply

$$\Delta^{\mathrm{new}}_{c,c'} = \Delta^{\mathrm{old}}_{c,c'} + \kappa_C. \quad (3)$$

However, for all pairs (c, c'), where one of the clusters has been obtained by the last cluster aggregation, $\Delta_{c,c'}$ is computed according to equation (7) in the Appendix. Moreover, we can avoid the computation of the statistics $s_k^{(m)}, a_{k,l}^{(m)}, b_{k,l}^{(m)}$ for all m at every iteration by storing them during the entire algorithm and only performing local updates when necessary.

To summarize, at the beginning of the clustering algorithm all sufficient statistics $s_k^{(m)}, a_{k,l}^{(m)}, b_{k,l}^{(m)}$ are evaluated on the data. Then, for the first iteration $\Delta_{c,c'}$ is evaluated by equation (7) for all $M(M-1)/2$ pairs of

initial clusters, which can be time-consuming, but may be parallelized. During the algorithm, when the current number of clusters is C and a total of $C(C-1)/2$ terms $\Delta_{c,c'}$ must be computed, only $C-3$ of these terms are obtained via (7), while all other terms are very quickly updated via (3).

5 Matching of SBM node labels

Given the node labels, say $\mathcal{Z}^{(c)}$ and $\mathcal{Z}^{(c')}$, and the SBM parameters of two clusters of networks, the goal is to find the best match of the block labels of the two SBMs. A naive strategy consists in ordering one part of the SBM parameters, for instance, the block proportions $\pi_1, \dots, \pi_K$ or the diagonal elements of the connectivity matrix $\boldsymbol{\gamma}$ in a monotone order. However, as none of the parts of the parameter contains all relevant information, there are always cases where such an approach fails. To take into account both parts of the parameter $(\boldsymbol{\pi}, \boldsymbol{\gamma})$, we propose to use the graphon of the SBM as shown in this section.

5.1 Graphon of a SBM parameter

The graphon, introduced by Lovász and Szegedy (2006), is a function $g : [0,1]^2 \to [0,1]$ that can be used as a generative model for exchangeable random graphs including SBM. First, generate independent random variables $U_i \sim U[0,1]$ for the vertices $i \in [\![n]\!]$. Then, conditionally on U_i and U_j, draw an edge $A_{i,j} \sim \mathcal{B}(g(U_i, U_j))$. The graphon of the $\mathcal{SBM}(\boldsymbol{\pi}, \boldsymbol{\gamma})$ is given by

$$g_{(\boldsymbol{\pi},\boldsymbol{\gamma})}(u,v) = \gamma_{k,l} \quad \text{for } (u,v) \in R_{k,l},$$

where $R_{k,l} = (q_{k-1}, q_k] \times (q_{l-1}, q_l]$ and $q_k = \sum_{s \in [\![k]\!]} \pi_s$, $k \in [\![K]\!]$, $q_0 = 0$. Indeed, when $U_i \in (q_{k-1}, q_k]$, then $Z_i = k$. The graphon $g_{(\boldsymbol{\pi},\boldsymbol{\gamma})}$ is a piecewise constant function depending on the entire SBM parameter. Clearly, it also depends on the order of the block labels. Changing the block labels implies the permutation of the piecewise constant parts of the graphon as illustrated in Figure 2.

5.2 Label-dependent distance measure for two SBM parameters

To compare SBMs with parameters $(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)})$ and $(\boldsymbol{\pi}^{(c')}, \boldsymbol{\gamma}^{(c')})$, consider the L^2-distance of their graphons.

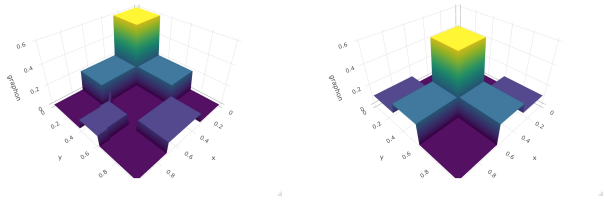


Fig. 2: Graphons of a SBM with two different orders of the block labels.

By the piecewise constant character, the square distance is a finite sum given by

$$\begin{aligned}
&\|g_{(\boldsymbol{\pi}^{(c)},\boldsymbol{\gamma}^{(c)})} - g_{(\boldsymbol{\pi}^{(c')},\boldsymbol{\gamma}^{(c')})}\|_2^2 \\
&= \int_{[0,1]^2} (g_{(\boldsymbol{\pi}^{(c)},\boldsymbol{\gamma}^{(c)})}(u,v) - g_{(\boldsymbol{\pi}^{(c')},\boldsymbol{\gamma}^{(c')})}(u,v))^2 \mathrm{d}(u,v) \\
&= \sum_{k,l,k',l'} \left(\gamma_{k,l}^{(c)} - \gamma_{k',l'}^{(c')}\right)^2 |R_{k,l,k',l'}|, \qquad (4)
\end{aligned}$$

where $|R_{k,l,k',l'}|$ denotes the area of $R_{k,l,k',l'}$ defined as

$$\begin{aligned}
R_{k,l,k',l'} = &\left\{ \left(q_{k-1}^{(c)}, q_k^{(c)}\right] \cap \left(q_{k'-1}^{(c')}, q_{k'}^{(c')}\right] \right\} \times \\
&\left\{ \left(q_{l-1}^{(c)}, q_l^{(c)}\right] \cap \left(q_{l'-1}^{(c')}, q_{l'}^{(c')}\right] \right\},
\end{aligned}$$

This distance is zero if and only if parameter values are identical ($(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)}) = (\boldsymbol{\pi}^{(c')}, \boldsymbol{\gamma}^{(c')})$) as well as the order of the blocks. Thus, it is a label-dependent distance measure. Furthermore, the graphon distance is well-defined even when the number of blocks of the two models differ.

5.3 Matching SBM blocks

Our tool to match block labels of two SBM parameters consists in finding the permutations yielding the smallest graphon distance. More precisely, let K_c and $K_{c'}$ be the number of blocks in $(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)})$ and $(\boldsymbol{\pi}^{(c')}, \boldsymbol{\gamma}^{(c')})$, resp. Denote by $\mathfrak{S}_K$ the set of all permutations of $[\![K]\!]$ and a parameter with permuted blocks by

$$\sigma(\boldsymbol{\pi}, \boldsymbol{\gamma}) = \left((\pi_{\sigma(1)}, \dots, \pi_{\sigma(K)}), (\gamma_{\sigma(k),\sigma(l)})_{k,l}\right).$$

We define permutations $\hat{\sigma}_c$ and $\hat{\sigma}_{c'}$ as the solutions of the minimization

$$\min_{\sigma_1 \in \mathfrak{S}_{K^{(c)}}, \sigma_2 \in \mathfrak{S}_{K^{(c')}}} \|g_{\sigma_1(\boldsymbol{\pi}^{(c)},\boldsymbol{\gamma}^{(c)})} - g_{\sigma_2(\boldsymbol{\pi}^{(c')},\boldsymbol{\gamma}^{(c')})}\|_2.$$

The solution is not unique, as for any $\tau \in \mathfrak{S}_{K^{(c)}}$ the minimum is also attained with the permutations $\tau \circ \hat{\sigma}_c$ and $\tau \circ \hat{\sigma}_{c'}$.

Algorithm 3 Graph cluster aggregation

Input: Two sets of networks $\mathcal{A}^{(c)}$ and $\mathcal{A}^{(c')}$ with associated node labels $\mathcal{Z}^{(c)}$ and $\mathcal{Z}^{(c')}$ and SBM parameters $(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)})$ and $(\boldsymbol{\pi}^{(c')}, \boldsymbol{\gamma}^{(c')})$.
Step 1 Find the permutations $\hat{\sigma}_c$ and $\hat{\sigma}_{c'}$ as described in Section 5.4 giving the best match of blocks of $(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)})$ and $(\boldsymbol{\pi}^{(c')}, \boldsymbol{\gamma}^{(c')})$.
Step 2 Reorder node labels: $\mathcal{Z}^{(c)} \leftarrow \hat{\sigma}_c(\mathcal{Z}^{(c)})$ and $\mathcal{Z}^{(c')} \leftarrow \hat{\sigma}_{c'}(\mathcal{Z}^{(c')})$.
Step 3 Update the node labels $\mathcal{Z}_{c\cup c'}$ by the ICL maximization Algorithm 2.
Step 4 Compute the SBM parameter $(\boldsymbol{\pi}^{(c\cup c')}, \boldsymbol{\gamma}^{(c\cup c')})$ associated with $\mathcal{A}_{c\cup c'}$ and $\mathcal{Z}_{c\cup c'}$ according to (3).
Output: Node labels $\mathcal{Z}_{c\cup c'}$ and parameter $(\boldsymbol{\pi}^{(c\cup c')}, \boldsymbol{\gamma}^{(c\cup c')})$ for the new cluster.

For the practical computation of $\hat{\sigma}_c$ and $\hat{\sigma}_{c'}$, an exhaustive exploration of all permutations $\mathfrak{S}_{K^{(c)}}$ and $\mathfrak{S}_{K^{(c')}}$ is feasible when the number of blocks K_c and $K_{c'}$ are not too large. However, we propose a general simplification based on an identifiability property of graphons (Bickel and Chen, 2009) which states that if an undirected random graph model admits a graphon such that its marginal $\bar{g} = \int g(u,v)\mathrm{d}v$ is strictly monotone, then the graphon is identifiable. As the SBM graphon is piecewise constant, strict monotonicity does not hold. Nevertheless, we introduce the canonical graphon denoted by g^{can} as the permutation of the SBM parameters such that its marginal $\bar{g}$ is monotone decreasing. Hence, instead of exploring all possible permutations of the block labels, we choose as $\hat{\sigma}_c$ and $\hat{\sigma}_{c'}$ the permutations providing the canonical representation of the graphons. In the directed case, where the marginals $\bar{g}(u) = \int g(u,v)\mathrm{d}v$ and $\bar{\bar{g}}(v) = \int g(u,v)\mathrm{d}u$ are not the same, a reasonable adaptation is to first order blocks according one marginal, say $\bar{g}$. Then, if $\bar{g}$ is constant over two SBM blocks, order these two blocks such that the other marginal $\bar{\bar{g}}$ is decreasing over these two blocks.

5.4 Relabeling nodes during cluster aggregation

Let us summarize all steps to relabel nodes when merging two clusters. First, estimate the SBM parameters for both clusters by the maximum a posterior estimator defined by

$$(\hat{\boldsymbol{\pi}}^{(c)}, \hat{\boldsymbol{\gamma}}^{(c)}) = \arg\max_{(\boldsymbol{\pi},\boldsymbol{\gamma})} p((\boldsymbol{\pi},\boldsymbol{\gamma})|\mathcal{A}^{(c)}, \mathcal{Z}^{(c)}),$$

with simple closed-form expressions given by

$$\hat{\pi}_k^{(c)} = \frac{\sum_{m\in I_c} s_k^{(m)} + \alpha - 1}{\sum_{m\in I_c} n_k^{(m)} + K(\alpha-1)}, \tag{5}$$

$$\hat{\gamma}_{k,\ell}^{(c)} = \frac{\sum_{m\in I_c} a_{k,\ell}^{(m)} + \eta - 1}{\sum_{m\in I_c} (a_{k,\ell}^{(m)} + b_{k,\ell}^{(m)}) + \eta + \zeta - 2}, \quad k,\ell \in [\![K_c]\!].$$

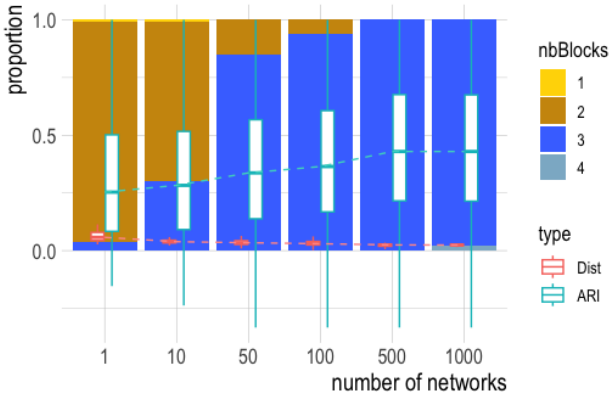


Fig. 3: Proportion of data sets on which the number of SBM blocks is estimated to be 3 (blue) or not. Boxplots of the graphon distances of the estimated SBM and the true one (red) and boxplots of the ARI of the node labels (cyan). All results on 100 data sets.

Next, the permutations $\hat{\sigma}_c$ and $\hat{\sigma}_{c'}$ to obtain the canonical representations of the associated graphons are determined and the node labels $\mathcal{Z}^{(c)}$ and $\mathcal{Z}^{(c')}$ are updated accordingly by

$$\mathcal{Z}^{(\ell)}_{\text{update}} = (\hat{\sigma}_\ell(\mathbf{Z}^{(j)}), j \in I_\ell), \quad \text{with}$$
$$\hat{\sigma}_\ell(\mathbf{Z}^{(j)}) = (\mathbf{Z}^{(j)}_{\hat{\sigma}_\ell(1)}, \ldots, \mathbf{Z}^{(j)}_{\hat{\sigma}_\ell(n^{(j)})}), \quad \ell \in \{c, c'\}, j \in I_\ell.$$

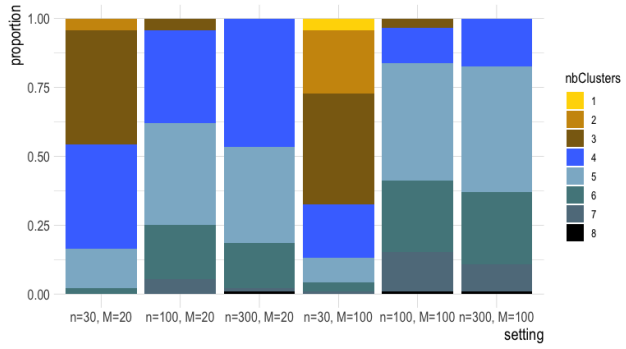
Finally, Algorithm 2 is applied to further improve the node labels of the newly created cluster. All steps of aggregating two clusters are given in Alogirhtm 3.

6 Numerical Study

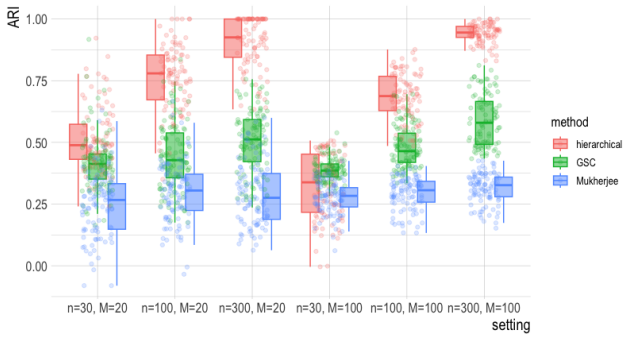
We conduct numerical experiments to assess the performance of our clustering algorithm, which is by the way available on CRAN via the R package `graphclust`.

6.1 Estimation accuracy

Before studying the cluster performance of our algorithm, we first investigate the estimation accuracy of model parameters and latent variables. This is done in an asymptotic setting, where the size of the collection increases. Here, data come from a mixture with a single component, that is, networks are i.i.d. realizations from the same SBM. Concretely, we consider a SBM with 3 blocks, block proportions $\boldsymbol{\pi} = (0.3, 0.3, 0.4)$ and



(a) Estimated number of clusters



(b) ARI of network clusters

Fig. 4: Monte-Carlo simulation results for our hierarchical algorithm, GCS and Mukherjee's method for varying number of nodes n and different collection sizes M.

connectivity matrix given by

$$\boldsymbol{\gamma} = \begin{pmatrix} 0.1 & 0.3 & 0.5 \\ 0.1 & 0.5 & 0.1 \\ 0.1 & 0.5 & 0.6 \end{pmatrix}.$$

From the connectivity probabilities $\boldsymbol{\gamma}$ it is clear that the nodes in block 2 are difficult to distinguish from those in block 3. Only small networks with 8 to 13 vertices are simulated, such that it is probable that a single network does provide enough evidence for the presence of 3 distinct blocks. Indeed, fitting a SBM to each of the networks by a standard estimation algorithm with an automatic selection of the number of blocks, as implemented in the R package `blockmodels`, mainly yields SBM estimates with 2 blocks only. This can be seen from the results in Figure 3 for collections containing only one network ($M = 1$).

Now, for collections of different sizes (M between 1 and 1000), we apply a variant of our hierarchical algorithm that has no stopping criterion, merging all networks to a single cluster. Contrary to a one-by-one analysis of the networks, we observe that our approach that typically starts with initial SBMs with 2 blocks is able to discover the richer true SBM with 3 blocks by progressively aggregating clusters. That is, our method is able to combine and exploit information coming from several networks in order to improve parameter estimation. More precisely, Figure 3 displays the proportion of 100 simulated data sets for every collection size M on which the procedure correctly selects a SBM with 3 blocks at the end of the algorithm (blue). Obviously, this proportion increases with M, and for collections with 500 networks the 3 SBM blocks are always correctly identified.

A finer evaluation of the estimation accuracy is given by the distance of the graphon of the estimated SBM parameters and the true SBM, as defined in Section 5.4. This is a valid comparison even when the number of blocks are not the same, which is the case for small sample sizes. We see that the graphon distance (red boxplots in Figure 3) steadily decreases when providing more and more data to the algorithm, meaning that the estimation accuracy is improved.

Finally, the estimated node labels $\hat{\mathcal{Z}}$ can be compared to the true ones by the adjusted Rand index (ARI) (Hubert and Arabie, 1985). The ARI (cyan boxplots) is strictly increasing in the sample size M indicating that the fit gets better and better. However, interestingly, the ARI does not tend to 1, that is, adding networks to the collection does not necessarily improve the estimation of the block labels. Indeed, this is expected, since even with the knowledge of the true SBM parameter, a small network may not provide enough information for an accurate block assignment of all its nodes. For a consistent estimation of the node labels, it would be necessary that the number of nodes in each network increases.

6.2 Graph clustering

Now we assess the performance of the new clustering algorithm on data from a 4-component SBM mixture. We consider two sample sizes $M \in \{20, 100\}$ and three different mean numbers of vertices $n_{\text{mean}} \in \{30, 100, 300\}$ per network, with large variations of the sizes $n^{(m)}$ of the individual networks around n_{mean}.

First, let us have a look on the estimated number of clusters on 100 simulated data sets displayed in Figure 4 a). When networks are small ($n_{\text{mean}} = 30$), the cluster number is often underestimated (for both, $M \in \{20, 100\}$), that is lower than 4. Increasing the network size generally leads to more estimated clusters. We also see that increasing the collection size M has not the same effect as increasing the number of nodes n_{mean}.

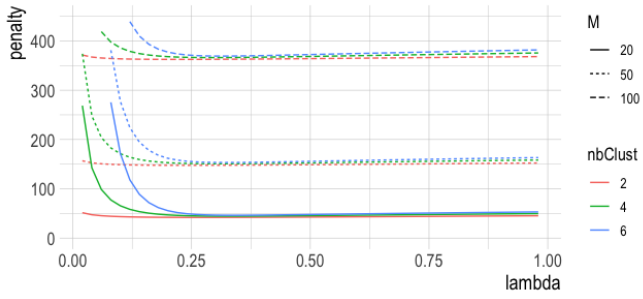


Fig. 5: Penalty term as a function of hyperparameter λ for different collection sizes M and number of clusters C.

However, on large collections with many nodes per network, the method tends to overestimate the number of clusters. This indicates that, in some sens, the model selection may not be optimal and the term in the ICL that penalizes models with large numbers of clusters may not be large enough. This penalty term is a function of hyperparameter λ and a closer look on the term helps to understand that the phenomenon is not simply due to a badly chosen value of λ, which is set to 0.5 in the simulations. Indeed, Figure 5 shows that the penalty term $\log\left(\Gamma(C\lambda)/[(\Gamma(\lambda))^C\Gamma(C\lambda+M)]\right)$ is nearly constant on the interval $[0.2, 1]$ for any collection size M and any number of clusters C, whereas close to 0, the penalty term increases exponentially fast. Thus, the calibration of hyperparameter λ is very difficult. More precisely, on the flat part all values yield virtually the same number of clusters, while on the steep part the slightest variation in λ leads to very different numbers of clusters. This has also been confirmed by additional simulations. We conclude that the model selection device is not exact, but still gives a rough idea of the right number of clusters. An improvement of the criterion (or a more convenient choice of the priors) to obtain a consistent estimate of the number of clusters is left for future work.

For a finer analysis of the clustering result we consider the ARI of the obtained clustering in comparison to the true cluster labels $\mathcal{U}$ in the SBM mixture model. Figure 4 b) illustrates that the clustering obtained with the hierarchical method (red boxplots) gets consistently better when more data are presented to the algorithm. On large collections and/or large network sizes, the clustering tends to be perfect. This indicates that, although the number of clusters may be overestimated, some of the mixture components may simply be split into smaller components.

Finally, a comparison to alternative graph-distance methods is in order. The first alternative is the clustering approach by Mukherjee et al. (2017) based on graph moments (blue boxplots). For the second method, we use an approach based on our graphon distance. More precisely, first a SBM is fit to every single network, then a similarity matrix with the graphon distance for all pairs of networks is computed, to which finally a spectral clustering algorithm is applied to derive a clustering. SBM parameters are obtained by the package `blockmodels` (which are also the initial values of the new hierarchical procedure). We refer to this method as the graphon spectral clustering (GSC) approach (green boxplots). Both GSC and Mukherjee's method require the specification of the number of clusters, which is set to 4 here. Recall that in the literature alternative clustering algorithms are rare for the setting of directed networks without node correspondence.

According to Figure 4 b) the hierarchical method clearly outperforms the others in all settings but one, which is the setting of a large collection of small networks. Moreover, the model-based hierarchical approach benefits the most of presenting more data to the algorithm by an important increase of the ARI, while the other methods only do slightly better. For the alternative methods it is not even certain whether their ARI will converge to 1 or they will saturate before then. This is in accordance with our understanding of distance-based approaches, where the estimation uncertainty is not taken into account and networks are only analyzed separately. We conclude that model-based approaches as ours, where a likelihood criterion is considered and a common descriptor of each cluster is computed using all data associated with one cluster, have a real advantage over graph-distance methods.

6.3 Robustness to model assumptions

In practice, model assumptions are never completely satisfied. Here, robustness is investigated in two settings: in the first, data come from small-world models rather than from a mixture of SBMs, in the second a collection of graphs containing a substantial part of outliers is considered.

For a small-world model, we consider the directed preferential attachment model (Bollobás et al., 2003) and generate 100 networks from a three-component mixture. Parameters are such that networks of all components have between 32 and 35 vertices and edge densities range from 0.22 to 0.28. The main difference between the three component models resides in different network topologies, since in one model nodes with more out-going than in-coming edges are added, in another it

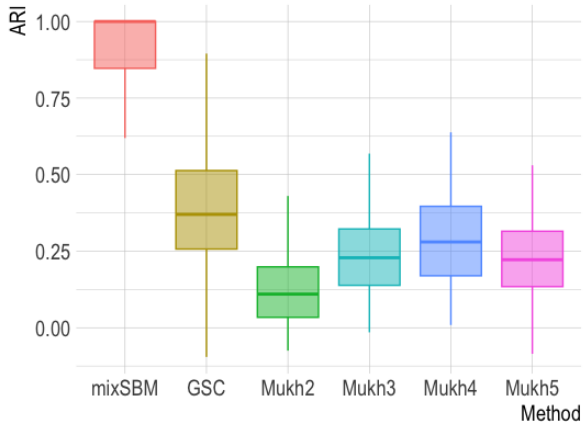


Fig. 6: Mixture of small-world models. Comparison of different approaches in terms of the ARI.

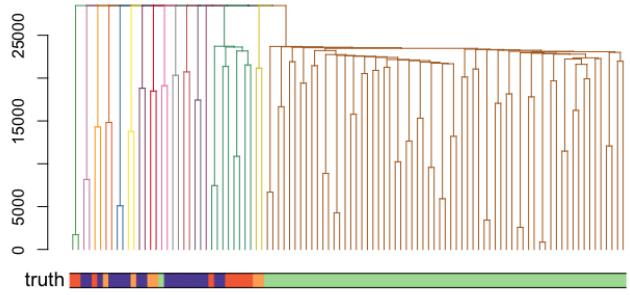


Fig. 7: Dendrogram of the clustering by our hierarchical algorithm. In the bar below, green is the dominant cluster, red and orange the networks in the two intermediate clusters and all outliers are purple.

is the inverse and in the third model as many out-going than in-coming edges are added in average.

On these data our hierarchical clustering algorithm is compared to the GSC approach and to the graph moment clustering method of Mukherjee et al. (2017) with 2, 3, 4 and 5 graph moments, respectively. The number of clusters is set to three for all methods except ours, where the number of clusters is selected automatically. In more than 50% of the data sets, our method correctly estimates the number of clusters to be 3. The ARI of all methods are displayed in Figure 6 and show that our SBM mixture approach outperforms the other methods by far. We also see that GSC does better than all graph moment algorithms.

Next, we consider data containing a substantial part of outliers. A large part of the networks are drawn from a 3-component SBM mixture, consisting of a dominant cluster and two clusters of intermediate size. Outlier networks are simulated by first generating a SBM parameter at random and then drawing one network from this SBM. In other words, every outlier has an individual SBM parameter. Finally, the simulated collection of 100 networks, each with 50 nodes, contains 19 outliers.

Figure 7 shows the dendrogram of the clustering obtained with our procedure. The hierarchical algorithm detects 16 clusters. The largest cluster (65 networks) contains only networks generated from the dominant mixture component. Furthermore, 88% of the networks generated by the 3-component SBM mixture (i.e. 71 networks) belong to clusters that are almost pure (more than 90% of the networks from one mixture component). Furthermore, 68% of the outliers (13 networks) are in clusters that do not contain any data from the mixture model. Thus, the algorithm is able to make a distinction between data from the mixture model and most of the outliers.

In terms of the ARI, our algorithm attains a value of 0.95, which is considerably larger than the ARI of 0.065 for the GSC procedure with the same number of clusters, that is 16. Varying the number of clusters, the highest ARI for GSC is achieved with 4 clusters and reaches the value 0.72, which is still far below the ARI of the model-based approach.

We conclude that our approach gives very satisfying results when the data contains outliers or noisy observations. By the way, this scenario is inspired by the model estimated on the collection of foodwebs analyzed in Section 6.4 and thus supports the validity of the results obtained for this application.

6.4 Application to ecological networks

The mangal database (Poisot et al., 2016) provides a huge collection of ecological networks available via the R package `rmangal`. We extract the 187 networks, where interactions among different taxa (vertices) are of the type predation. The median number of vertices per foodweb is 19 (ranging from 5 to 708) and the median number of edges 32 (ranging from 4 edges to 27,745). Our goal is the identification of foodwebs that have the same network structure regardless of the taxa or the size of the foodwebs. Is there any kind of universal topology of foodwebs? How many different organization forms of an ecosystem exist, and how can they be described and compared?

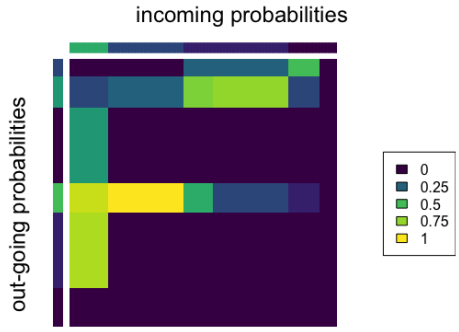


Fig. 8: Graphon of SBM parameter of the dominant cluster with in-coming and out-going probabilities on the sidebars.

Our agglomerative cluster algorithm applied to these foodwebs discovers 17 clusters. There is a dominant cluster containing 115 networks (61%), 5 clusters of intermediate size (6 to 14 networks) and none of the remaining 11 clusters contains more than 4 networks.

Figure 8 represents the SBM parameter associated with the dominant cluster. It contains six blocks, block proportions are in the range $[0.06, 0.28]$, half of the connectivity parameters $\gamma_{k,l}$ are lower than 0.01 and the largest connectivity parameters is 0.87. To interpret the different blocks, we consider the probabilities of in-coming and out-going edges for a node in block $k \in [\![K]\!]$ defined as

$$d_k^{\text{in}} = \sum_{l \in [\![K]\!]} \pi_l \gamma_{l,k}, \qquad d_k^{\text{out}} = \sum_{l \in [\![K]\!]} \pi_l \gamma_{k,l}.$$

A large value of d_k^{in} indicates that the species in block k are often eaten by other species, while a large d_k^{out} represents species that often eat other species. We define a vegetarian behavior by a low probability to eat others (say $d_k^{\text{out}} \leq 0.05$) and a significant probability of being victim ($d_k^{\text{in}} \geq 0.05$). Our model contains two vegetarian blocks representing 43% of the species. Likewise, we define predators by a significant probability to eat others ($d_k^{\text{out}} \geq 0.05$) and few chance to be eaten ($d_k^{\text{in}} \leq 0.05$). Then 18% of the species (two blocks) are predators. The remaining 39% are somewhere in the middle of the food pyramid with both good chances to be eaten and to eat others ($d_k^{\text{in}} \geq 0.05$, $d_k^{\text{out}} \geq 0.05$). So this is the typical structure of most foodwebs in the database.

To compare this topology to others, consider, for instance, the cluster containing the largest network with 708 nodes. The adjusted SBM has 29 blocks, which is explained by the very large network size. The question is whether this SBM is a kind of finer version of the SBM of the dominant cluster or whether there is a significant difference. Here block proportions lie in $[0.004, 0.14]$, two third of the connectivity parameters $\gamma_{k,l}$ are lower than 0.01 and the maximal value is 0.91. Furthermore, 53% of the species are vegetarians, 24% are predators and 7% are in-between. The remaining 16% are networks with very few interactions ($d_k^{\text{in}} \leq 0.015$, $d_k^{\text{out}} \leq 0.015$) and such inactive species are absent in the dominant cluster. Thus, it is clear that this network structure is very different from the dominant cluster.

Clusters can also be compared in terms of the graphon distance among the associated SBM parameters. Figure 10 displays the values of the observed graphon distances for all pairs of clusters in the model. The mean value is 0.23, corresponding to a significant difference between the SBM parameters, since the maximal graphon distance is 1 (the maximal values is the distance between the graphons constant to 0 and 1, respectively).

It is instructive to represent the clustering in connecection with the geographic location of the foodwebs (Figure 9). Foodwebs of the dominant cluster (lightblue circles) are present all over the globe and correspond indeed to some global or universal structure of ecosystems. Interestingly, also the intermediate clusters are all spread over several continents. This means that different types of graph topology are not related to a particular geographic region. We conclude that the results of our algorithm provide many insights on the structure of foodwebs and raise new questions in ecology.

Finally, the clustering may be compared to the one obtained, for instance, by Mukherjee's graph moments method. The ARI of -0.03 indicates that the two clusterings are completely different. A closer look reveals that the Mukherjee clustering is also composed of a dominant cluster containing 153 networks, but only 92 of them are common to the dominant cluster of the SBM mixture. Moreover, there are 3 intermediate clusters with 5, 6 and 8 networks, respectively, which are almost completely included in the dominant SBM cluster. All other clusters contain only 1 or 2 networks, that can be considered as outliers. A visualisation of the Mukherjee clustering on a map show that there are no geographic cluster either, but the geographic distribution of the clusters is not the same as for the SBM mixture (see Figure 11 in the Appendix).

7 Conclusion

We have developed an approach to cluster networks according to their graph topologies. To the best of our knowledge, this is the first parametric mixture model for networks that do not share the same set of vertices neither the same number of vertices and that applies to both directed and undirected graphs. We illustrated

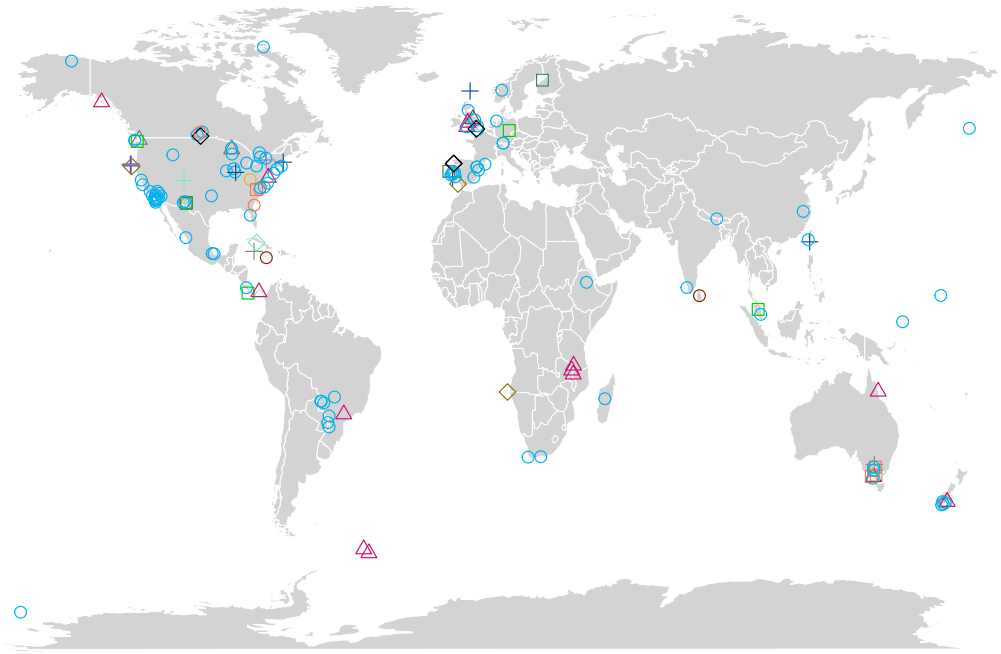


Fig. 9: Geographical representation of the clustering of the foodwebs.

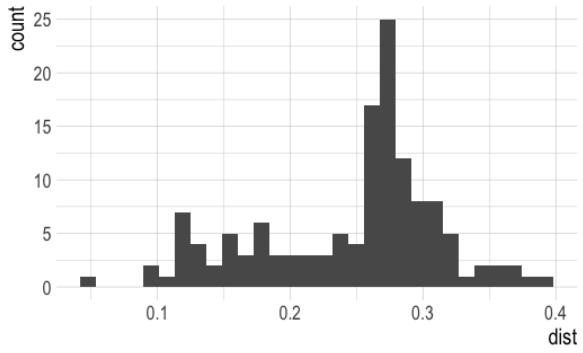


Fig. 10: Graphon distances between SBMs of all pairs of clusters for the foodweb mixture model.

that a model-based approach, where a description of each cluster is computed, outperforms clustering methods based on a graph distance between networks, since our model inherently takes into account the estimation uncertainty. Another advantage of our hierarchical algorithm is the automated selection of the number of clusters, which is done in a single run of the algorithm contrary to EM-type algorithms, where different numbers of clusters must be explored separately. Moreover, a finite mixture of SBMs is a highly interpretable model, which is important in practical applications as illustrated for ecological networks. Finally, we propose a new tool to match the block labels of two SBMs, which may be useful in other contexts.

In future work, to accommodate a wider spectrum of applications, this model may be extended to mixtures of SBMs with degree correction or including covariates. This requires a modification of the ICL criterion, namely the choice of appropriate prior distributions such that the ICL criterion has closed-form expression and estimation remains feasible.

As in our experiments the number of clusters tends to be overestimated on huge datasets, another important issue, which is out of the scope of this paper, is a general analysis of the ICL approach and its validity as a model selection device.

Acknowledgements Work partly supported by the grant ANR-18-CE02-0010 of the French National Research Agency ANR (project EcoNet).

References

Amini AA, Chen A, Bickel PJ, Levina E (2013) Pseudo-likelihood methods for community detection in large

sparse networks. The Annals of Statistics 41(4):2097–2122

Bickel PJ, Chen A (2009) A nonparametric view of network models and newman–girvan and other modularities. Proceedings of the National Academy of Sciences 106(50):21068–21073

Biernacki C, Celeux G, Govaert G (2000) Assessing a mixture model for clustering with the integrated completed likelihood. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(7):719–725

Bollobás B, Borgs C, Chayes J, Riordan O (2003) Directed scale-free graphs. In: SODA '03 Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms, pp 132–139

Botella C, Dray S, Matias C, Miele V, Thuiller W (2022) An appraisal of graph embeddings for comparing trophic network architectures. Methods in Ecology and Evolution 13(1):203–216

Chabert-Liddell SC, Barbillon P, Donnet S (2022) Learning common structures in a collection of networks. an application to food webs

Côme E, Latouche P (2015) Model selection and clustering in stochastic block models based on the exact integrated complete data likelihood. Statistical Modelling 15(6):564–589

Daudin JJ, Picard F, Robin S (2008) A mixture model for random graphs. Statistics and Computing 18(2):173–183

Donnat C, Holmes S (2018) Tracking network dynamics: A survey using graph distances. The Annals of Applied Statistics 12(2):971 – 1012

Fraley C, Raftery AE (2002) Model-based clustering, discriminant analysis, and density estimation. Journal of the American Statistical Association 97(458):611–631

Frühwirth-Schnatter S, Malsiner-Walli G (2019) From here to infinity: sparse finite versus Dirichlet process mixtures in model-based clustering. Advances in Data Analysis and Classification 13:33–64

Gärtner T (2003) A survey of kernels for structured data. ACM SIGKDD Explorations Newsletter 5(1):49–58

le Gorrec L, Knight PA, Caen A (2022) Learning network embeddings using small graphlets. Social Network Analysis and Mining 12(20)

Hamilton WL, Ying R, Leskovec J (2017) Representation learning on graphs: Methods and applications. IEEE Data Engineering Bulletin 40(3):52–74

Hubert L, Arabie P (1985) Comparing partitions. J Classif 2:193–218

Isella L, Stehlé J, Barrat A, Cattuto C, Pinton JF, den Broeck WV (2011) What's in a crowd? Analysis of face-to-face behavioral networks. Journal of Theoretical Biology 271(1):166–180

Le CM, Levin K, Levina E (2018) Estimating a network from multiple noisy realizations. Electronic Journal of Statistics 12(2):4697 – 4740

Leger JB (2016) Blockmodels: A R-package for estimating in latent block model and stochastic block model, with various probability functions, with or without covariates

Liu J (2008) Monte Carlo strategies in scientific computing. Springer Verlag, New York, Berlin, Heidelberg

Lovász L, Szegedy B (2006) Limits of dense graph sequences. Journal of Combinatorial Theory, Series B 96(6):933–957

Mantziou A, Lunagomez S, Mitra R (2023) Bayesian model-based clustering for multiple network data

Matias C, Robin S (2014) Modeling heterogeneity in random graphs through latent space models: a selective review. Esaim Proc & Surveys 47:55–74

McLachlan G, Krishnan T (2008) The EM algorithm and extensions, 2nd edn. Wiley series in probability and statistics, Wiley

McLachlan G, Peel D (2000) Finite Mixture Models. Wiley Series in Probability and Statistics, Wiley-Interscience

Mehta N, Duke LC, Rai P (2019) Stochastic blockmodels meet graph neural networks. In: Proceedings of the 36th International Conference on Machine Learning, vol 97, pp 4466–4474

Mukherjee SS, Sarkar P, Lin L (2017) On clustering network-valued data. In: Advances in Neural Information Processing Systems, vol 30

Nowicki K, Snijders TAB (2001) Estimation and prediction for stochastic blockstructures. Journal of the American Statistical Association 96(455):1077–1087

Peixoto T (2014) Efficient Monte Carlo and greedy heuristic for the inference of stochastic block models. Physical Review E 89(1)

Poisot T, Baiser B, Dunne JA, Kéfi S, Massol Fc, Mouquet N, Romanuk TN, Stouffer DB, Wood SA, Gravel D (2016) mangal – making ecological network analysis simple. Ecography 39(4):384–390

Robert CP (2007) The Bayesian choice: a decision-theoretic motivation. Springer, New York, (2nd ed.)

Rohe K, Chatterjee S, Yu B (2011) Spectral clustering and the high-dimensional stochastic blockmodel. Annals of Statistics 39(4):1878–1915

Sabanayagam M, Vankadara LC, Ghoshdastidar D (2022) Graphon based clustering and testing of networks: Algorithms and theory. In: The Tenth International Conference on Learning Representations

Shervashidze N, Vishwanathan S, Petri T, Mehlhorn K, Borgwardt K (2009) Efficient graphlet kernels for

large graph comparison. In: JMLR Workshop and Conference Proceedings: AISTATS, pp 488–495
Shimada Y, Hirata Y, Ikeguchi1 T, Aihara K (2016) A survey of kernels for structured data. Scientific Reports 6:34944
Signorelli M, Wit EC (2019) Model-based clustering for populations of networks. Statistical Modelling 20(1):9–29
Stanley N, Shai S, Taylor D, Mucha PJ (2016) Clustering network layers with the strata multilayer stochastic block model. IEEE Transactions on Network Science and Engineering 3(2):95–105
Titterington D, Smith A, Makov U (1985) Statistical Analysis of Finite Mixture Distributions. Wiley, New York
Weber-Zendrera A, Sokolovska N, Soula HA (2021) Functional prediction of environmental variables using metabolic networks. Scientific Reports 11:12192
Wu Z, Pan S, Chen F, Long G, Zhang C, Yu PS (2021) A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems 32(1):4–24
Xu K, Hu W, Leskovec J, Jegelka S (2019) How powerful are graph neural networks? In: International Conference on Learning Representations
Young JG, Kirkley A, Newman MEJ (2022) Clustering of heterogeneous populations of networks. Physical Review E 105(1)

8 Appendix

8.1 Details on the update of the node labels

Here we present the details on the efficient computation of the ICL changes $\Delta^{\to h}_{m^*,i^*}$, in the case when moving node i^* to block h does not empty block g.

Changes in the statistics. Let $s_k^{(m^*)}$ be the count statistic before the swap and $\vec{s}_k^{(m^*)}$ its value after the swap. We use the same notation for all other statistics. Clearly, $\vec{s}_g^{(m^*)} = s_g^{(m^*)} - 1$ and $\vec{s}_h^{(m^*)} = s_h^{(m^*)} + 1$, while the other terms remain unchanged. Define

$$\delta_{k,\cdot i^*} = \sum_{i \neq i^*} Z_{i,k}^{(m^*)} A_{i,i^*}^{(m^*)}, \qquad \delta_{\ell,i^*\cdot} = \sum_{j \neq i^*} Z_{j,\ell}^{(m^*)} A_{i^*,j}^{(m^*)}.$$

Then, for any $k, \ell \in [\![K]\!]$,

$$\vec{a}_{k,\ell}^{(m^*)} = a_{k,\ell}^{(m^*)} - \mathbb{1}_{k=g}\delta_{\ell,i^*\cdot} + \mathbb{1}_{k=h}\delta_{\ell,i^*\cdot} - \mathbb{1}_{\ell=g}\delta_{k,\cdot i^*} + \mathbb{1}_{\ell=h}\delta_{k,\cdot i^*}.$$

When considering the matrix $(a_{k,\ell}^{(m^*)})_{k,\ell}$, only the g-th and h-th row and the g-th and h-th column change when moving i^* from g to h. We introduce the number of possible dyads from nodes in block k to nodes in block ℓ in graph m defined as

$$r_{k,\ell}^{(m)} = \sum_{i \neq j} Z_{i,k}^{(m)} Z_{j,\ell}^{(m)} = \begin{cases} s_k^{(m)} s_\ell^{(m)} & \text{if } k \neq \ell \\ s_k^{(m)}(s_k^{(m)} - 1) & \text{if } k = \ell \end{cases}$$

Then $b_{k,\ell}^{(m)} = r_{k,\ell}^{(m)} - a_{k,\ell}^{(m)}$ and

$$\vec{r}_{k,\ell}^{(m^*)} = r_{k,\ell}^{(m^*)} - s_\ell^{(m^*)}\mathbb{1}_{k=g} + s_\ell^{(m^*)}\mathbb{1}_{k=h} - s_k^{(m^*)}\mathbb{1}_{\ell=g} + s_k^{(m^*)}\mathbb{1}_{\ell=h} + 2\mathbb{1}_{k=g,\ell=g} - \mathbb{1}_{k=g,\ell=h} - \mathbb{1}_{k=h,\ell=g}.$$

and $\vec{b}_{k,l}^{(m^*)} = \vec{r}_{k,l}^{(m^*)} - \vec{a}_{k,l}^{(m^*)}$. For any $m \neq m^*$, the statistics remain unchanged, that is, $\vec{a}_{k,l}^{(m)} = a_{k,l}^{(m)}$, $\vec{b}_{k,l}^{(m)} = b_{k,l}^{(m)}$ and $\vec{r}_{k,l}^{(m)} = r_{k,l}^{(m)}$. Finally, we define function $\Psi : \mathbb{R}_+ \times \mathbb{Z} \to \mathbb{R}$ as

$$\Psi(a, z) = \log\left(\frac{\Gamma(a+z)}{\Gamma(a)}\right)\mathbb{1}\{a + z > 0\}.$$

First case: K does not change. Suppose that i^* is not the last vertex in block g, that is, $\sum_m \sum_i Z_{i,g}^{(m)} > 1$. Then, moving node i^* to another block h does not empty block g and the number of blocks K remains unchanged. In this case, the ICL variation is given by

$$\begin{aligned}
&\Delta^{\to h}_{m^*,i^*} \\
&= \sum_{(k,\ell)\in I_{g,h}} \left\{ \log\left(\frac{\Gamma(\eta + \sum_m \vec{a}_{k,l}^{(m)})\Gamma(\zeta + \sum_m \vec{b}_{k,l}^{(m)})}{\Gamma(\eta + \zeta + \sum_m \vec{r}_{k,l}^{(m)})}\right) \right. \\
&\quad \left. - \log\left(\frac{\Gamma(\eta + \sum_m a_{k,l}^{(m)})\Gamma(\zeta + \sum_m b_{k,l}^{(m)})}{\Gamma(\eta + \zeta + \sum_m r_{k,l}^{(m)})}\right)\right\} \\
&\quad + \sum_{k\in\{g,h\}} \left\{ \log\left(\Gamma(\alpha + \sum_m \vec{s}_k^{(m)})\right) \right. \\
&\quad \left. - \log\left(\Gamma(\alpha + \sum_m s_k^{(m)})\right)\right\} \\
&= \sum_{(k,\ell)\in I_{g,h}} \left\{ \Psi\left(\eta + \sum_m a_{k,l}^{(m)}, \vec{a}_{k,l}^{(m^*)} - a_{k,l}^{(m^*)}\right) \right. \\
&\quad + \Psi\left(\zeta + \sum_m b_{k,l}^{(m)}, \vec{b}_{k,l}^{(m^*)} - b_{k,l}^{(m^*)}\right) \\
&\quad \left. - \Psi\left(\eta + \zeta + \sum_m r_{k,l}^{(m)}, \vec{r}_{k,l}^{(m^*)} - r_{k,l}^{(m^*)}\right)\right\} \\
&\quad + \log\left(\frac{\alpha + \sum_m s_h^{(m)}}{\alpha + \sum_m s_g^{(m)} - 1}\right),
\end{aligned} \tag{6}$$

where $I_{g,h} = \left\{(k,\ell) \in [\![K]\!]^2, k \in \{g,h\} \text{ or } \ell \in \{g,h\}\right\}$.

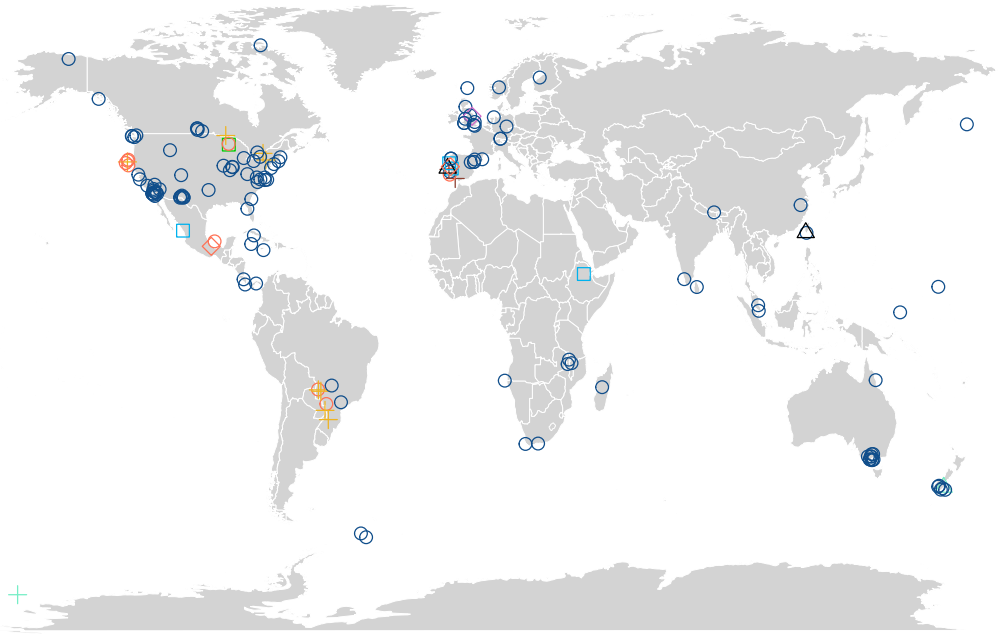


Fig. 11: Geographical representation of the clustering of the foodwebs obtained with Mukherjee's method.

8.2 Details on the efficient computation of $\Delta_{c,c'}$

Here it is shown how to evaluate $\Delta_{c,c'}$ efficiently. Denote $\mathcal{U}_{c\cup c'}$ the cluster labels afte merging clusters c and c', that is, $U^{(m)}_{c\cup c'} = \min\{c,c'\}$ if $m \in I_c \cup I_{c'}$ and $U^{(m)}_{c\cup c'} = U^{(m)}$ otherwise. Likewise, denote $\mathcal{Z}_{c\cup c'}$ the node labels after aggregation and relabeling with $\mathcal{Z}^{(\ell)}_{c\cup c'} = \{\hat{\sigma}_\ell(\mathbf{Z}^{(j)}), j \in I_\ell\}$ for $\ell \in \{c,c'\}$, where $\hat{\sigma}_\ell$ are the permutations that match the block labels. For convenience, denote by $\beta(x,y) = \log\left(\frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}\right)$ the logarithm of the Beta function of x and y. Moreover, for any $c \in [\![C]\!], (k,l) \in [\![K_c]\!]$, denote

$$\mathbf{s}^{(c)}_k = \sum_{m\in I_c} s^{(m)}_k, \quad \mathbf{a}^{(c)}_{k,l} = \sum_{m\in I_c} a^{(m)}_{k,l}, \quad \mathbf{b}^{(c)}_{k,l} = \sum_{m\in I_c} b^{(m)}_{k,l}.$$

Then $\Delta_{c,c'} = \mathrm{ICL}^{\mathrm{mix}}(\mathcal{A},\mathcal{U}_{c\cup c'},\mathcal{Z}_{c\cup c'}) - \mathrm{ICL}^{\mathrm{mix}}(\mathcal{A},\mathcal{U},\mathcal{Z})$ is given by

$$\begin{aligned}
\Delta_{c,c'} = &\sum_{(k,\ell)} \beta\left(\eta + \mathbf{a}^{(c)}_{\hat{\sigma}_c^{-1}(k),\hat{\sigma}_c^{-1}(l)} + \mathbf{a}^{(c')}_{\hat{\sigma}_{c'}^{-1}(k),\hat{\sigma}_{c'}^{-1}(l)}\right. \\
&\left. + \mathbf{b}^{(c)}_{\hat{\sigma}_c^{-1}(k),\hat{\sigma}_c^{-1}(l)} + \mathbf{b}^{(c')}_{\hat{\sigma}_{c'}^{-1}(k),\hat{\sigma}_{c'}^{-1}(l)}\right) \\
&- \sum_{(k,\ell)} \beta\left(\eta + \mathbf{a}^{(c)}_{k,l}, \zeta + \mathbf{b}^{(c)}_{k,l}\right) - \sum_{(k,\ell)} \beta\left(\eta + \mathbf{a}^{(c')}_{k,l}, \zeta + \mathbf{b}^{(c')}_{k,l}\right) \\
&+ \sum_k \log\left(\Gamma(\alpha + \mathbf{s}^{(c)}_{\hat{\sigma}_c^{-1}(k)} + \mathbf{s}^{(c')}_{\hat{\sigma}_{c'}^{-1}(k)})\right) - \log\left(\Gamma(\alpha + \mathbf{s}^{(c)}_k)\right) \\
&\quad - \log\left(\Gamma(\alpha + \mathbf{s}^{(c')}_k)\right) + \log\left(\frac{\Gamma(\lambda + |I_c| + |I_{c'}|)}{\Gamma(\lambda + |I_c|)\Gamma(\lambda + |I_{c'}|)}\right). \\
&+ \log\left(\frac{\Gamma(K_c\alpha + \sum_{m\in I_c} n^{(m)})\Gamma(K_{c'}\alpha + \sum_{m\in I_{c'}} n^{(m)})}{\Gamma\left(K_{\max}\alpha + \sum_{m\in I_c\cup I_{c'}} n^{(m)}\right)}\right) \\
&+ K^2_{\min}\beta(\eta,\zeta) + K_{\min}\log\left(\Gamma(\alpha)\right) \\
&+ \beta\left((C-1)\lambda,\lambda\right) + \log\left(\frac{\Gamma(C\lambda + M)}{\Gamma((C-1)\lambda + M)}\right),
\end{aligned} \tag{7}$$

where $K_{\max} = \max\{K_c, K_{c'}\}$ and $K_{\min} = \min\{K_c, K_{c'}\}$ are the maximal and minimal number of blocks in the clusters c and c'.

8.3 Supplement to the analysis of ecological networks

Figure 11 illustrates the clustering of the foodwebs obtained with the alternative graph moments method by Mukherjee et al. (2017). The obtained clustering is virtually very different from the one obtained by our graph clustering procedure.