



HAL
open science

Model-based clustering of multiple networks with a hierarchical algorithm

Tabea Rebafka

► **To cite this version:**

Tabea Rebafka. Model-based clustering of multiple networks with a hierarchical algorithm. 2023. hal-03837505v2

HAL Id: hal-03837505

<https://hal.science/hal-03837505v2>

Preprint submitted on 13 Jan 2023 (v2), last revised 5 Nov 2023 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Model-based clustering of multiple networks with a hierarchical algorithm

Tabea Rebafka*

Laboratoire de Probabilités, Statistique et Modélisation
Sorbonne Université, Université Paris Cité, CNRS
Paris, France

January 13, 2023

Abstract

The paper tackles the problem of clustering multiple networks, 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 groups of networks requires a means to overcome the label-switching problem of the stochastic block model and to match the block labels of the graphs. 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 collection 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.

*The author gratefully acknowledges funding by the ANR project EcoNet (ANR-18-CE02-0010-01).

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 that may not share the same set of vertices and may vary in size, and 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, nodes 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 likelihood 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 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. 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 long-standing tradition (Titterton 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 generalized 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., 2021; 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 collections 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 (Mantziou et al., 2021; Young et al., 2022). They all have the disadvantage that the number of clusters must be set by the user.

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 merge of groups results in a deterioration of the objective. Thus, model selection is performed automatically. Such penalized criteria 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 collections of networks (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).
- Details for an efficient implementation of the algorithm are provided in the Appendix.

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 $\boldsymbol{\pi} = (\pi_1, \dots, \pi_K) \in (0, 1)^K$ are the block proportions with $\sum_{k \in \llbracket K \rrbracket} \pi_k = 1$ and $\boldsymbol{\gamma} = (\gamma_{k,l})_{k,l} \in (0, 1)^{K \times K}$ the connectivity matrix. Let $\mathbf{Z} = (Z_1, \dots, Z_n) \in \llbracket K \rrbracket^n$ be a vector of independent discrete latent variables for the nodes, with $\mathbb{P}(Z_i = k) = \pi_k$ for all $k \in \llbracket K \rrbracket$ and $i \in \llbracket n \rrbracket$. Conditionally on the node labels $\mathcal{Z} = (Z)_{m \in \llbracket M \rrbracket}$, the observed adjacency matrix $A = (A_{i,j})_{1 \leq i,j \leq n} \in \{0, 1\}^{n \times n}$ verifies

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

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

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 $\mathcal{A} = \{A^{(m)}, m \in \llbracket M \rrbracket\}$ be a collection of M networks, where $A^{(m)} = (A_{i,j}^{(m)})_{1 \leq i,j \leq n^{(m)}} \in \{0, 1\}^{n^{(m)} \times n^{(m)}}$ 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 $\mathcal{U} = (U^{(1)}, \dots, U^{(M)}) \in \llbracket C \rrbracket^M$ defining a partitioning of the M networks into $C \geq 1$ clusters. Denote $p_c = \mathbb{P}(U^{(m)} = c), c \in \llbracket C \rrbracket$ the cluster proportions and $\mathbf{p} = (p_1, \dots, p_C) \in (0, 1)^C$. Now, let $(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)}), c \in \llbracket C \rrbracket$ 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 in-

dependent realizations of the SBM with parameter $(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)})$, that is, conditionally on \mathcal{U} ,

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

Denote $\theta = (\mathbf{p}, \{(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)}), c \in \llbracket C \rrbracket\})$ 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, $\mathbf{Z}^{(m)} = (Z_1^{(m)}, \dots, Z_{n^{(m)}}^{(m)})$, and also denote $\mathcal{Z} = \{\mathbf{Z}^{(m)}, m \in \llbracket M \rrbracket\}$.

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

$$\text{ICL}^{\text{sbm}}(\mathcal{A}, \mathcal{Z}) = \log(p(\mathcal{A}, \mathcal{Z})) = \log \left(\int p(\mathcal{A}, \mathcal{Z} | \boldsymbol{\pi}, \boldsymbol{\gamma}) p(\boldsymbol{\pi}, \boldsymbol{\gamma}) d(\boldsymbol{\pi}, \boldsymbol{\gamma}) \right). \quad (1)$$

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

$$\hat{\mathcal{Z}} = \arg \max_{\mathcal{Z}} \text{ICL}^{\text{sbm}}(\mathcal{A}, \mathcal{Z}), \quad (2)$$

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

$$p(\boldsymbol{\pi}, \boldsymbol{\gamma}) = p(\boldsymbol{\pi}) \times \prod_{k,l \in \llbracket K \rrbracket^2} p(\gamma_{k,l}) = \text{Dirichlet}(\boldsymbol{\pi}; \alpha_1, \dots, \alpha_K) \times \prod_{k,l \in \llbracket K \rrbracket^2} \text{Beta}(\gamma_{k,l}; \eta_{k,l}, \zeta_{k,l}).$$

where $\alpha_1, \dots, \alpha_K, \eta_{k,l}, \zeta_{k,l}$ are hyperparameters, the ICL^{sbm} has closed-form expression, which is given in the Appendix.

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

$$\text{ICL}^{\text{mix}}(\mathcal{A}, \mathcal{U}, \mathcal{Z}) = \log(p(\mathcal{A}, \mathcal{U}, \mathcal{Z})) = \log \left(\int p(\mathcal{A}, \mathcal{U}, \mathcal{Z} | \theta) p(\theta) d\theta \right), \quad (3)$$

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

$$(\hat{\mathcal{U}}, \hat{\mathcal{Z}}) = \arg \max_{\mathcal{U}, \mathcal{Z}} \text{ICL}^{\text{mix}}(\mathcal{A}, \mathcal{U}, \mathcal{Z}). \quad (4)$$

Again we consider classical independent conjugate priors given by

$$\begin{aligned} p(\theta) &= p(\mathbf{p}) \prod_{c \in \llbracket C \rrbracket} p(\boldsymbol{\pi}^{(c)}) p(\boldsymbol{\gamma}^{(c)}) \\ &= \text{Dirichlet}(\mathbf{p}; \lambda_1, \dots, \lambda_C) \prod_{c \in \llbracket C \rrbracket} \text{Dirichlet}(\boldsymbol{\pi}^{(c)}; \alpha_1, \dots, \alpha_{K_c}) \prod_{(k,l) \in \llbracket K_c \rrbracket^2} \text{Beta}(\gamma_{k,l}^{(c)}; \eta_{k,l}, \zeta_{k,l}), \end{aligned}$$

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, $I_c = \{m \in \llbracket M \rrbracket : U^{(m)} = c\}$ for $c \in \llbracket C \rrbracket$, 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

$$\text{ICL}^{\text{mix}}(\mathcal{A}, \mathcal{U}, \mathcal{Z}) = \sum_{c \in \llbracket C \rrbracket} \text{ICL}^{\text{sbm}}(\mathcal{A}^{(c)}, \mathcal{Z}^{(c)}) + \log \left(\int p(\mathcal{U}|\mathbf{p})p(\mathbf{p})d\mathbf{p} \right).$$

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

$$\log \left(\int p(\mathcal{U}|\mathbf{p})p(\mathbf{p})d\mathbf{p} \right) = \log \left(\frac{\Gamma(C\lambda)}{(\Gamma(\lambda))^C \Gamma(C\lambda + M)} \right) + \sum_{c \in \llbracket C \rrbracket} \log(\Gamma(\lambda + |I_c|)).$$

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

4 Hierarchical clustering algorithm

To solve the discrete optimization problem given by (4), we propose a greedy hill-climbing algorithm. The algorithm is initialized by a mixture of M SBMs by setting $U^{(m)} = m$ for $m \in \llbracket M \rrbracket$, 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 $(c, c') \in \llbracket C \rrbracket^2$, the ICL variation $\Delta_{c,c'}$ is evaluated defined as

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

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 .

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

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. 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 ICL^{mix} in \mathcal{Z} , while keeping the clustering \mathcal{U} fixed. This amounts to maximize the term ICL^{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. This algorithm is presented in the Appendix.

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.

In view of the computing time, it is important that the evaluation of ICL variations $\Delta_{c,c'}$ is fast. The Appendix provides explicit and fast formulae for the computation of $\Delta_{c,c'}$.

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

Algorithm 1: Agglomerative algorithm for graph clustering

Input: Collection of networks \mathcal{A} .

Set $U^{(m)} = m$ for $m \in \llbracket M \rrbracket$ and set $C = M$.

for $m \in \llbracket M \rrbracket$ **do**

| Fit a SBM to $A^{(m)}$ yielding parameters $(\boldsymbol{\pi}^{(m)}, \boldsymbol{\gamma}^{(m)})$ and node labels $\mathbf{Z}^{(m)}$.

end

Set $\mathcal{Z} = \{\mathbf{Z}^{(m)}, m \in \llbracket M \rrbracket\}$ and $\boldsymbol{\theta} = \{(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)}), c \in \llbracket C \rrbracket\}$.

while $C > 0$ **do**

| **for** $(c, c') \in \llbracket C \rrbracket^2$ **do**

| | Compute $\Delta_{c,c'}$ according to Section 8.2.

| **end**

| 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 $m \in I_{c_1} \cup I_{c_2}$.

| | Update \mathcal{Z} and $\boldsymbol{\theta}$ according to Algorithm 3.

| | Set $C = C - 1$.

| **end**

| **else**

| | exit **while**

| **end**

end

Output: Clustering $\mathcal{U} = \{U^{(m)}, m \in \llbracket M \rrbracket\}$, node labels \mathcal{Z} , SBM parameters

$\{(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)}), c \in \llbracket C \rrbracket\}$.

naive strategy consists in ordering one part of the SBM parameters, for instance, the block proportions π_1, \dots, π_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,

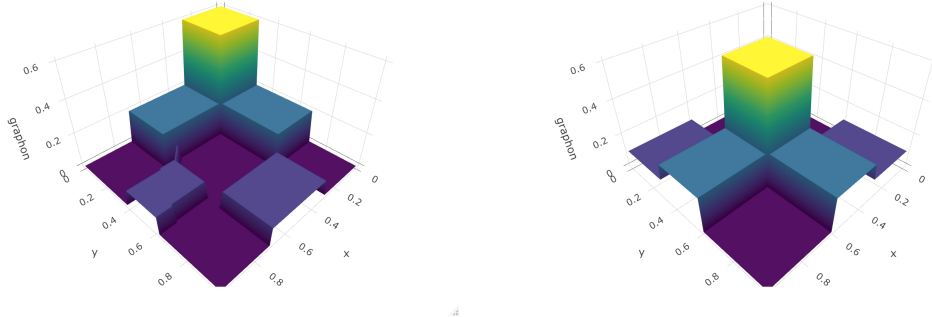


Figure 1: Graphons of a SBM with two different orders of the block labels.

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 $g : [0, 1]^2 \rightarrow [0, 1]$ 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 $i \in \llbracket n \rrbracket$. 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 a SBM $\mathcal{SBM}(\boldsymbol{\pi}, \boldsymbol{\gamma})$ is given by

$$g_{(\boldsymbol{\pi}, \boldsymbol{\gamma})}(u, v) = \gamma_{k,l} \quad \text{for every } (u, v) \in R_{k,l} = (q_{k-1}, q_k] \times (q_{l-1}, q_l], \quad (5)$$

where $q_k = \sum_{s \in \llbracket k \rrbracket} \pi_s$, $k \in \llbracket K \rrbracket$, $q_0 = 0$. 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 1.

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

$$\begin{aligned} \|g_{(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)})} - g_{(\boldsymbol{\pi}^{(c')}, \boldsymbol{\gamma}^{(c')})}\|_2 &= \left(\int_{[0,1]^2} (g_{(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)})}(u, v) - g_{(\boldsymbol{\pi}^{(c')}, \boldsymbol{\gamma}^{(c')})}(u, v))^2 d(u, v) \right)^{\frac{1}{2}} \\ &= \sum_{k,l,k',l'} \left(\gamma_{k,l}^{(c)} - \gamma_{k',l'}^{(c')} \right)^2 |R_{k,l,k',l'}|, \end{aligned} \quad (6)$$

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

$$R_{k,l,k',l'} = \left\{ \left(\pi_{k-1}^{(c)}, \pi_k^{(c)} \right] \cap \left(\pi_{k'-1}^{(c')}, \pi_{k'}^{(c')} \right] \right\} \times \left\{ \left(\pi_{l-1}^{(c)}, \pi_l^{(c)} \right] \cap \left(\pi_{l'-1}^{(c')}, \pi_{l'}^{(c')} \right] \right\},$$

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 $\llbracket K \rrbracket$ 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

$$(\hat{\sigma}_c, \hat{\sigma}_{c'}) \in \arg \min_{\sigma_1 \in \mathfrak{S}_{K^{(c)}}, \sigma_2 \in \mathfrak{S}_{K^{(c')}}} \|g_{\sigma_1(\boldsymbol{\pi}^{(c)}, \boldsymbol{\gamma}^{(c)})} - g_{\sigma_2(\boldsymbol{\pi}^{(c')}, \boldsymbol{\gamma}^{(c')})}\|_2. \quad (7)$$

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 $\tau \circ \hat{\sigma}_{c'}$. But for our purpose, any solution is convenient for matching block labels.

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). Defining the marginal of graphon g as $\bar{g}(u) = \int g(u, v)dv$, one can show for undirected networks that there exists a unique graphon g^{can} which defines the same distribution as g and whose marginal \bar{g}^{can} is monotone decreasing. Graphon g^{can} is called the canonical representation of g . 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. This is justified for undirected networks. In the directed case, where the marginals $\bar{g}(u) = \int g(u, v)dv$ and $\bar{\bar{g}}(v) = \int g(u, v)du$ 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

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

The estimator has simple closed-form expressions detailed in the Appendix. Next, the permutations $\hat{\sigma}_c$ and $\hat{\sigma}_{c'}$ to obtain the canonical representations of the associated graphons

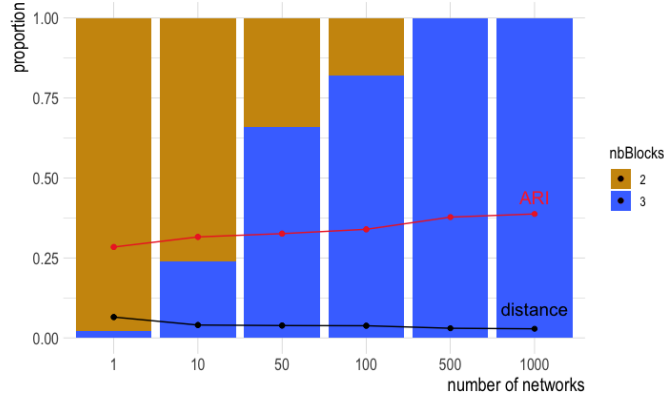


Figure 2: Proportion of collections of networks on which the number of SBM blocks is estimated to be 3 (blue) and 2 (yellow). The ARI (red) of the node labels and the graphon distance (black) of the estimated SBM compared to the true one.

are determined and the node labels $\mathcal{Z}^{(c)}$ and $\mathcal{Z}^{(c')}$ are updated accordingly by

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

6 Numerical Study

We conduct numerical experiments to assess the performance of our clustering algorithm.

The algorithm will be very soon available on CRAN as the R package `grupchclust`.

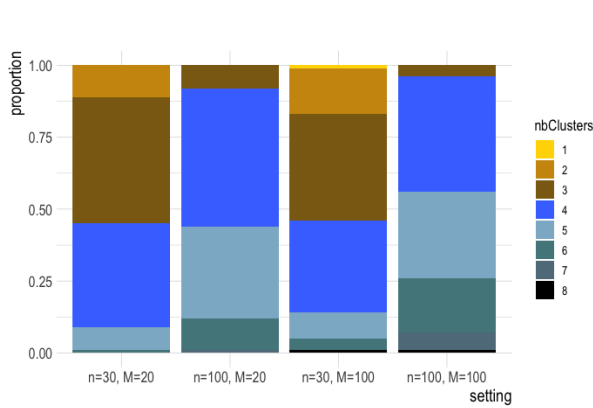
6.1 Estimation accuracy

In the single network setting it is well known that parameter estimates converge to the true SBM parameter when the number of nodes increases. In the multiple network framework a different question is the accuracy of the estimators as a function of the number M of networks, when the network size is bounded. To study this question we consider a mixture with a single component. Concretely, we simulate multiple networks with only

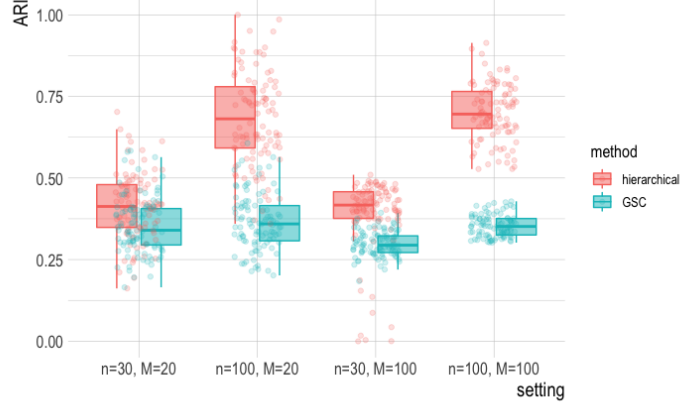
10 vertices from a SBM with 3 blocks. Fitting a SBM to a single small network by a standard estimation algorithm, as done at the initialisation step of our algorithm, yields SBM estimates with 2 blocks only, since data do not provide enough evidence to estimate the parameters of a more complex model. We apply to these synthetic data a variant of our hierarchical agglomerative algorithm with 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 jointly analyzes all networks, allows to discover the richer true SBM with 3 blocks, even when all networks are small. More precisely, Figure 2 displays the proportion of 100 simulated data sets on which the procedure correctly selects a SBM with 3 blocks (blue). Obviously, this proportion increases with M , and for 500 networks and more, no errors are made. We also compare the estimated node labels $\hat{\mathcal{Z}}$ to the true ones by the adjusted Rand index (ARI) (Hubert and Arabie, 1985). The ARI (red line) is strictly increasing in the sample size M indicating that the fit gets better and better. Finally, we evaluate the quality of the estimated SBM by comparing the associated graphon with the one of the true SBM as described 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 from Figure 2 that the graphon distance (black line) steadily decreases when providing more and more data to the algorithm, meaning that the estimation accuracy improves.

6.2 Graph clustering

To study the impact of both the number M of networks and the sizes $n^{(m)}$ of the individual networks on the performance of the clustering algorithm, data from a mixture of four SBMs with equal block proportions are simulated. We consider two sample sizes $M \in \{20, 100\}$ and networks with two different mean numbers of vertices, say n_{mean} . For $n_{\text{mean}} = 30$, the



(a) Estimated number of clusters



(b) ARI

Figure 3: Estimated number of clusters for our hierarchical algorithm, and ARI for both our hierarchical algorithm and GCS. 100 datasets in each of the four settings.

numbers of nodes $n^{(m)}$ vary from 15 to 100. For $n_{\text{mean}} = 100$, all $n^{(m)}$ are chosen in $[20, 200]$. For each of the four settings, Figure 3 a) displays the estimated number of clusters on 100 simulated collections of networks. When networks are small ($n_{\text{mean}} = 30$), the cluster number is often underestimated (for both, $M \in \{20, 100\}$). Increasing the network size to $n_{\text{mean}} = 100$ should increase the accuracy of the estimation of the underlying SBMs as seen in the previous section, and indeed we observe in Figure 3 a) that the number of clusters is significantly more often correctly estimated. Moreover, Figure 3 b) illustrates the ARI of the obtained graph clusterings (in red) compared to the truth. Again, we see that increasing the number of vertices n_{mean} per network has not the same effect as increasing the number of networks M . This confirms that a good clustering result goes in hand with high estimation accuracy of the model parameters.

We can also compare our clustering algorithm to alternative methods. Most clustering procedures in the literature based on a graph distance consist in constructing a similarity

matrix with the graph distance for all pairs of networks to which a spectral clustering algorithm is applied to derive a clustering. Here we propose to consider the graphon distance defined in this work as a graph distance evaluated on the SBM parameter estimated on a single network. For the spectral clustering algorithm we specify the correct number of clusters to be found. We refer to this method as the graphon spectral clustering (GSC) approach. The procedure is applied to the same data as our model-based algorithm and the corresponding ARIs are also represented in Figure 3 b) (in blue). Obviously, in all settings our graph clustering approach has much better performance although we provided the correct number of clusters to the GCS method. Moreover, no substantial improvement of GSC is observed when more data are available. This is in accordance with our understanding of such 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 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 assumption

In practice, model assumptions are never satisfied. To test robustness, we consider data containing a substantial part of outliers or noisy observations. Here we simulate an outlier network 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. The rest of the data are drawn from a SBM mixture with 3 components. More precisely, the simulated collection of 100 networks (each with 50 nodes) is composed of a dominant cluster (66 networks) and two intermediate clusters (6 and 9 networks) and the remaining 19 networks are outliers.

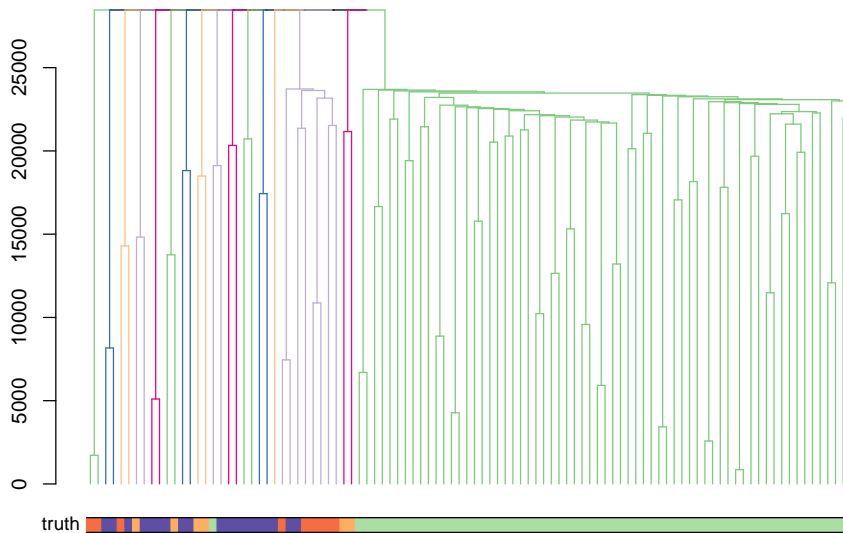


Figure 4: Dendrogram of clustering obtained by the algorithm. In the bar below spurious networks are represented in purple.

Figure 4 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 three-component the 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 three-component mixture model. Thus, the algorithm is able to make a distinction between data from the mixture model and most of the noisy observations.

In terms of the ARI, our algorithm attains a value of 0.95, which is considerably larger than the ARI of 0.065 obtained with 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

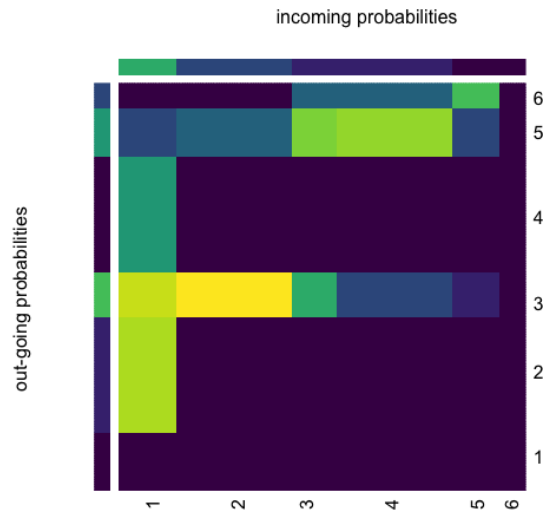


Figure 5: Graphon of SBM parameter of the dominant cluster with in-coming and out-coming probabilities on the sidebars.

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 also very satisfying results when the data contains outliers or noisy observations.

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 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?

Our agglomerative cluster algorithm applied to these foodwebs discovers 17 clusters.

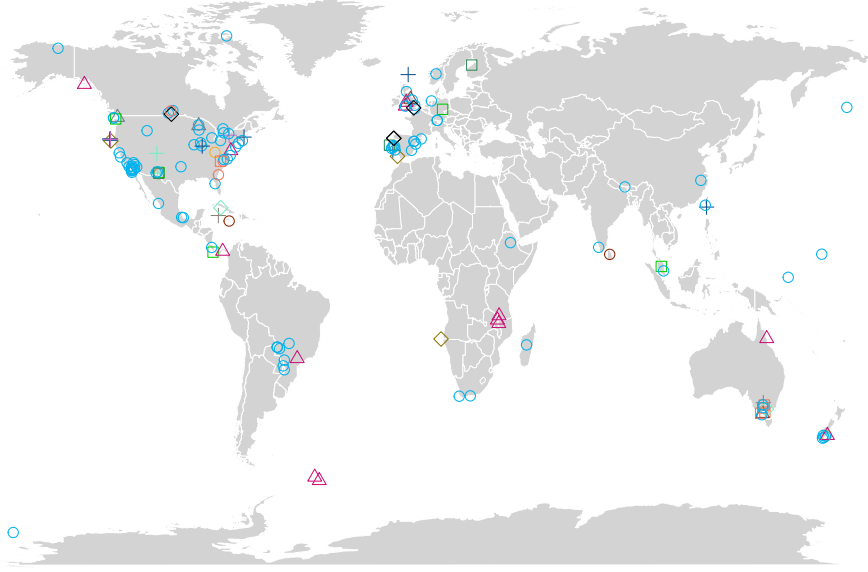


Figure 6: Geographical representation of the clustering of the foodwebs.

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 6 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 $k \in \llbracket K \rrbracket$ defined as

$$d_k^{\text{in}} = \sum_{l=1}^K \pi_l \gamma_{l,k}, \quad d_k^{\text{out}} = \sum_{l=1}^K \pi_l \gamma_{k,l}.$$

A large values 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 with 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. It is clear that this network structure is very different from the dominant cluster.

It is instructive to represent the clustering in connection with the geographic location of the foodwebs (Figure 6). 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.

7 Conclusion

We have developed an approach to cluster networks according to their graph topologies. We illustrated that a model-based approach, where a description of each cluster is computed,

outperforms methods based only on pairwise distances between networks, as we inherently take 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. Finally, we propose a new tool to match the block labels of two SBMs, which may be useful in other contexts.

8 Appendix

8.1 Expression of ICL^{sbm}

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 $(k,l) \in \llbracket K \rrbracket^2$. Using the one-hot encoding for node labels $Z_i^{(m)} = (Z_{i,1}^{(m)}, \dots, Z_{i,K}^{(m)}) \in \{0, 1\}^K$, useful count statistics for the m -th network are

$$s_k^{(m)} = \sum_{i \in \llbracket n \rrbracket} Z_{i,k}^{(m)}, \quad a_{k,l}^{(m)} = \sum_{i \neq j} Z_{i,k}^{(m)} Z_{j,l}^{(m)} A_{i,j}^{(m)}, \quad b_{k,l}^{(m)} = \sum_{i \neq j} Z_{i,k}^{(m)} Z_{j,l}^{(m)} (1 - A_{i,j}^{(m)}), \quad (8)$$

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

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

With these notations at hand, the ICL is given by

$$\begin{aligned} \text{ICL}^{\text{sbm}}(\mathcal{A}, \mathcal{Z}) = & \sum_{(k,l) \in \llbracket K \rrbracket^2} \log \left(\frac{\Gamma(\eta + \mathbf{a}_{k,l}) \Gamma(\zeta + \mathbf{b}_{k,l})}{\Gamma(\eta + \zeta + \mathbf{a}_{k,l} + \mathbf{b}_{k,l})} \right) + \sum_{k \in \llbracket K \rrbracket} \log(\Gamma(\alpha + \mathbf{s}_k)) \quad (9) \\ & + K^2 \log \left(\frac{\Gamma(\eta + \zeta)}{\Gamma(\eta) \Gamma(\zeta)} \right) + \log \left(\frac{\Gamma(K\alpha)}{\Gamma(K\alpha + \sum_m n^{(m)})} \right) - K \log(\Gamma(\alpha)). \end{aligned}$$

8.2 Computation of $\Delta_{c,c'}$

In this section it is shown how to evaluate $\Delta_{c,c'}$ efficiently. Denote $\mathcal{U}_{c \cup c'}$ the cluster labels after merging clusters c and c' , that is, $U_{c \cup c'}^{(m)} = \min\{c, c'\}$ if $m \in I_c \cup I_{c'}$ and $U_{c \cup c'}^{(m)} = U^{(m)}$ otherwise. Likewise, denote $\mathcal{Z}_{c \cup c'}$ the node labels after aggregation and relabeling with $\mathcal{Z}_{c \cup c'}^{(\ell)} = \{\hat{\sigma}_\ell(\mathbf{Z}^{(j)}), j \in I_\ell\}$ 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 $c \in \llbracket C \rrbracket$, $(k, l) \in \llbracket K_c \rrbracket$, denote

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

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

$$\begin{aligned} \Delta_{c,c'} &= \sum_{(k,\ell)} \beta \left(\eta + \mathbf{a}_{\hat{\sigma}_c^{-1}(k), \hat{\sigma}_c^{-1}(\ell)}^{(c)} + \mathbf{a}_{\hat{\sigma}_{c'}^{-1}(k), \hat{\sigma}_{c'}^{-1}(\ell)}^{(c')} + \mathbf{b}_{\hat{\sigma}_c^{-1}(k), \hat{\sigma}_c^{-1}(\ell)}^{(c)} + \mathbf{b}_{\hat{\sigma}_{c'}^{-1}(k), \hat{\sigma}_{c'}^{-1}(\ell)}^{(c')} \right) \\ &\quad - \sum_{(k,\ell)} \beta \left(\eta + \mathbf{a}_{k,l}^{(c)}, \zeta + \mathbf{b}_{k,l}^{(c)} \right) - \sum_{(k,\ell)} \beta \left(\eta + \mathbf{a}_{k,l}^{(c')}, \zeta + \mathbf{b}_{k,l}^{(c')} \right) \\ &\quad + \sum_k \log \left(\Gamma(\alpha + \mathbf{s}_{\hat{\sigma}_c^{-1}(k)}^{(c)} + \mathbf{s}_{\hat{\sigma}_{c'}^{-1}(k)}^{(c')}) \right) - \log \left(\Gamma(\alpha + \mathbf{s}_k^{(c)}) \right) - \log \left(\Gamma(\alpha + \mathbf{s}_k^{(c')}) \right) \\ &\quad + \log \left(\frac{\Gamma(K_c \alpha + \sum_{m \in I_c} n^{(m)}) \Gamma(K_{c'} \alpha + \sum_{m \in I_{c'}} n^{(m)})}{\Gamma(K_{\max} \alpha + \sum_{m \in I_c \cup I_{c'}} n^{(m)})} \right) + \log \left(\frac{\Gamma(\lambda + |I_c| + |I_{c'}|)}{\Gamma(\lambda + |I_c|) \Gamma(\lambda + |I_{c'}|)} \right) \\ &\quad + K_{\min}^2 \beta(\eta, \zeta) + K_{\min} \log(\Gamma(\alpha)) \\ &\quad + \beta((C-1)\lambda, \lambda) + \log \left(\frac{\Gamma(C\lambda + M)}{\Gamma((C-1)\lambda + M)} \right), \end{aligned} \tag{10}$$

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' . 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 we use that the number C of clusters 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

$$\Delta_{c,c'}^{\text{new}} = \Delta_{c,c'}^{\text{old}} + \kappa_C.$$

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 (10). 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.

8.3 Parameter estimates

The MAP estimate of the SBM parameters associated with cluster c is given by

$$\hat{\pi}_k^{(c)} = \frac{\sum_{m \in I_c} s_k^{(m)} + \alpha - 1}{\sum_{m \in I_c} n_k^{(m)} + K(\alpha - 1)}, \hat{\gamma}_{k,\ell}^{(c)} = \frac{\sum_{m \in I_c} a_{k,\ell}^{(m)} + \eta - 1}{\sum_{m \in I_c} (a_{k,\ell}^{(m)} + b_{k,\ell}^{(m)}) + \eta + \zeta - 2}, k, \ell \in \llbracket K_c \rrbracket. \quad (11)$$

8.4 Update of nodes labels

After the aggregation of two clusters and relabeling the node, we can further improve node labels $\mathcal{Z}^{(c)}$ of the new cluster c by maximizing the associated ICL criterion ICL^{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} . 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 $m^* \in \llbracket M \rrbracket$, and one of its vertices, say $i^* \in \llbracket n^{(m)} \rrbracket$. Denote $g = Z_{i^*}^{(m^*)}$ the current block assignment of i^* . For any block $h \in \llbracket K \rrbracket$ compute the impact on the ICL of moving node i^* to block h , that is,

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

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

$$h^* = \arg \max_{h \in \llbracket K \rrbracket} \Delta_{m^*, i^*}^{\rightarrow h},$$

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

For the efficient computation of the ICL changes $\Delta_{m^*, i^*}^{\rightarrow h}$, 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. But first, let us have 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.

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

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

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 $m^* \in \llbracket M \rrbracket$ and one of its vertices $i^* \in \llbracket n^{(m^*)} \rrbracket$.

for $h \in \llbracket K \rrbracket$ **do**

| Compute the impact $\Delta_{m^*, i^*}^{\rightarrow h}$ on the ICL of moving node i^* to block h .

end

Determine the best block assignment $h^* = \arg \max_{h \in \llbracket K \rrbracket} \Delta_{m^*, i^*}^{\rightarrow h}$.

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

end

Output: Updated node labels \mathcal{Z} .

Then, for any $k, \ell \in \llbracket K \rrbracket$,

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

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 $\vec{b}_{k,l}^{(m^*)} = r_{k,l}^{(m^*)} - \vec{a}_{k,l}^{(m^*)}$. 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 $\Psi : \mathbb{R}_+ \times \mathbb{Z} \rightarrow \mathbb{R}$ as

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

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 (11).

Output: Node labels $\mathcal{Z}_{c \cup c'}$ and parameter $(\boldsymbol{\pi}^{(c \cup c')}, \boldsymbol{\gamma}^{(c \cup c')})$ for the new cluster.

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

$$\begin{aligned}
 \Delta_{m^*, i^*}^{\rightarrow h} &= \sum_{(k, \ell) \in I_{g, h}} \left\{ \log \left(\frac{\Gamma(\eta + \sum_m \bar{a}_{k, \ell}^{(m)}) \Gamma(\zeta + \sum_m \bar{b}_{k, \ell}^{(m)})}{\Gamma(\eta + \zeta + \sum_m \bar{r}_{k, \ell}^{(m)})} \right) \right. \\
 &\quad \left. - \log \left(\frac{\Gamma(\eta + \sum_m a_{k, \ell}^{(m)}) \Gamma(\zeta + \sum_m b_{k, \ell}^{(m)})}{\Gamma(\eta + \zeta + \sum_m r_{k, \ell}^{(m)})} \right) \right\} \\
 &\quad + \sum_{k \in \{g, h\}} \left\{ \log \left(\Gamma(\alpha + \sum_m \bar{s}_k^{(m)}) \right) - \log \left(\Gamma(\alpha + \sum_m s_k^{(m)}) \right) \right\} \\
 &= \sum_{(k, \ell) \in I_{g, h}} \left\{ \Psi \left(\eta + \sum_m a_{k, \ell}^{(m)}, \bar{a}_{k, \ell}^{(m^*)} - a_{k, \ell}^{(m^*)} \right) + \Psi \left(\zeta + \sum_m b_{k, \ell}^{(m)}, \bar{b}_{k, \ell}^{(m^*)} - b_{k, \ell}^{(m^*)} \right) \right. \\
 &\quad \left. - \Psi \left(\eta + \zeta + \sum_m r_{k, \ell}^{(m)}, \bar{r}_{k, \ell}^{(m^*)} - r_{k, \ell}^{(m^*)} \right) \right\} + \log \left(\frac{\alpha + \sum_m s_h^{(m)}}{\alpha + \sum_m s_g^{(m)} - 1} \right), \quad (12)
 \end{aligned}$$

where $I_{g, h} = \{(k, \ell) \in \llbracket K \rrbracket^2, k \in \{g, h\} \text{ or } \ell \in \{g, h\}\}$.

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 $\Delta_{m^*, i^*}^{\rightarrow h}$ 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

$$\text{ICL}^{\text{sbm}}(K) = \text{ICL}^{\text{sbm}}(K - 1) + \log \frac{\Gamma(K\alpha)}{\Gamma((K - 1)\alpha)} + \log \frac{\Gamma((K - 1)\alpha + \sum_m n^{(m)})}{\Gamma(K\alpha + \sum_m n^{(m)})}. \quad (13)$$

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 $\Delta_{m^*, i^*}^{\rightarrow h}$ is exactly the same term as in (12) plus the penalty term given in (13).

The whole procedure to update node labels is summarized in Algorithm 2. Moreover, Algorithm 3 presents all steps of the aggregation of two clusters.

References

- Amini, A. A., A. Chen, P. J. Bickel, and E. Levina (2013). Pseudo-likelihood methods for community detection in large sparse networks. *The Annals of Statistics* 41(4), 2097–2122.
- Bickel, P. J. and A. Chen (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., G. Celeux, and G. Govaert (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(7), 719–725.

- Botella, C., S. Dray, C. Matias, V. Miele, and W. Thuiller (2022). An appraisal of graph embeddings for comparing trophic network architectures. *Methods in Ecology and Evolution* 13(1), 203–216.
- Chabert-Liddell, S.-C., P. Barbillon, and S. Donnet (2022). Learning common structures in a collection of networks. an application to food webs.
- Côme, E. and P. Latouche (2015). Model selection and clustering in stochastic block models based on the exact integrated complete data likelihood. *Statistical Modelling* 15(6), 564–589.
- Daudin, J.-J., F. Picard, and S. Robin (2008). A mixture model for random graphs. *Statistics and Computing* 18(2), 173–183.
- Donnat, C. and S. Holmes (2018). Tracking network dynamics: A survey using graph distances. *The Annals of Applied Statistics* 12(2), 971 – 1012.
- Fraley, C. and A. E. Raftery (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association* 97(458), 611–631.
- Gärtner, T. (2003). A survey of kernels for structured data. *ACM SIGKDD Explorations Newsletter* 5(1), 49–58.
- Hamilton, W. L., R. Ying, and J. Leskovec (2017). Representation learning on graphs: Methods and applications. *IEEE Data Engineering Bulletin* 40(3), 52–74.
- Hubert, L. and P. Arabie (1985). Comparing partitions. *J. Classif.* 2, 193–218.
- Isella, L., J. Stehlé, A. Barrat, C. Cattuto, J.-F. Pinton, and W. V. den Broeck (2011). What's in a crowd? analysis of face-to-face behavioral networks. *Journal of Theoretical Biology* 271(1), 166–180.

- Le, C. M., K. Levin, and E. Levina (2018). Estimating a network from multiple noisy realizations. *Electronic Journal of Statistics* 12(2), 4697 – 4740.
- le Gorrec, L., P. A. Knight, and A. Caen (2022). Learning network embeddings using small graphlets. *Social Network Analysis and Mining* 12(20).
- Leger, J.-B. (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*. New York, Berlin, Heidelberg: Springer Verlag.
- Lovász, L. and B. Szegedy (2006). Limits of dense graph sequences. *Journal of Combinatorial Theory, Series B* 96(6), 933–957.
- Mantziou, A., S. Lunagomez, and R. Mitra (2021). Bayesian model-based clustering for multiple network data.
- Matias, C. and S. Robin (2014). Modeling heterogeneity in random graphs through latent space models: a selective review. *Esaim Proc. & Surveys* 47, 55–74.
- McLachlan, G. and T. Krishnan (2008). *The EM algorithm and extensions* (2. ed ed.). Wiley series in probability and statistics. Wiley.
- McLachlan, G. and D. Peel (2000). *Finite Mixture Models*. Wiley Series in Probability and Statistics. Wiley-Interscience.
- Mehta, N., L. C. Duke, and P. Rai (2019, 09–15 Jun). Stochastic blockmodels meet graph neural networks. In K. Chaudhuri and R. Salakhutdinov (Eds.), *Proceedings of the 36th International Conference on Machine Learning*, Volume 97 of *Proceedings of Machine Learning Research*, pp. 4466–4474. PMLR.

- Mukherjee, S. S., P. Sarkar, and L. Lin (2017). On clustering network-valued data. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, Volume 30. Curran Associates, Inc.
- Nowicki, K. and T. A. B. Snijders (2001). Estimation and prediction for stochastic block-structures. *Journal of the American Statistical Association* 96(455), 1077–1087.
- Peixoto, T. (2014, October). Efficient Monte Carlo and greedy heuristic for the inference of stochastic block models. *Physical Review E* 89(1).
- Poisot, T., B. Baiser, J. A. Dunne, S. Kéfi, F. c. Massol, N. Mouquet, T. N. Romanuk, D. B. Stouffer, S. A. Wood, and D. Gravel (2016). mangal – making ecological network analysis simple. *Ecography* 39(4), 384–390.
- Robert, C. P. (2007). *The Bayesian choice: a decision-theoretic motivation*. New York: Springer. (2nd ed.).
- Rohe, K., S. Chatterjee, and B. Yu (2011). Spectral clustering and the high-dimensional stochastic blockmodel. *Annals of Statistics* 39(4), 1878–1915.
- Sabanayagam, M., L. C. Vankadara, and D. Ghoshdastidar (2022). Graphon based clustering and testing of networks: Algorithms and theory. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.
- Shervashidze, N., S. Vishwanathan, T. Petri, K. Mehlhorn, and K. Borgwardt (2009, April). Efficient graphlet kernels for large graph comparison. In *JMLR Workshop and Conference Proceedings Volume 5: AISTATS 2009*, Cambridge, MA, USA, pp. 488–495. Max-Planck-Gesellschaft: MIT Press.

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