Rapport de stage dans l'UMR MIA Paris-Saclay

Louis Lacoste

July 10, 2023

Contents

0.1 Remerciements

Je tiens à remercier en premier lieu Pierre Barbillon pour son encadrement remarquable, sa disponibilité, ses conseils avisés et sa gentillesse. Saint-Clair Chabert-Liddell pour son accompagnement, ses remarques, ses explications et le temps qu'il m'a consacré. Merci à Sophie Donnet, pour les cours et les idées qu'elle m'a donné

Chapter 1

L'UMR MIA Paris-Saclay

1.1 Présentation

L'UMR MIA Paris-Saclay est une entité de recherche qui regroupe des statisticiens et des informaticiens spécialisés dans la modélisation et l'apprentissage statistique et informatique appliqués à la biologie, l'écologie, l'environnement, l'agronomie et l'agro-alimentaire. Elle est affiliée à AgroParisTech, INRAE et l'Université Paris Saclay.

Les membres de cette unité possèdent des compétences variées en matière de méthodes d'inférence statistique, telles que les modèles complexes, les modèles à variables latentes, l'inférence bayésienne, l'apprentissage et la sélection de modèle. Ils sont également experts en algorithmique, notamment en généralisation, transfert de domaine et représentation des connaissances.

L'objectif de cette unité est de développer des méthodes statistiques et informatiques originales, à la fois génériques et motivées par des problématiques spécifiques dans le domaine des sciences du vivant. Les activités de recherche s'appuient sur une solide culture dans les disciplines cibles, telles que l'écologie, l'environnement, l'agro-alimentaire, la biologie moléculaire et la biologie des systèmes.

L'unité est structurée en deux équipes de recherche : SOLsTIS (Statistical mOdelling and Learning for environnemenT and lIfe Sciences) et EkINocs (Expert Knowledge, INteractive modellING and learnING for understandING and decisiOn makING in dINamic Complexe Systems).

Elle est rattachée au département MATHNUM d'INRAE et au département MMIP d'AgroParisTech.

Les responsables au sein de l'unité sont : Julien Chiquet en tant que Directeur d'unité, Sophie Donnet en tant que Directrice d'unité adjointe, Antoine Cornuéjols en tant que Responsable de l'équipe EkINocs, et Sophie Donnet et Pierre Barbillon en tant que Responsables de l'équipe SOLsTIS.
Source: Accueil | MIA Paris-Saclay *n.d.*
La figure 1.1 présente l'organigramme complet de l'unité.

1.2 Encadrement et vie en stage

Au cours de mon stage, j'étais encadré par Pierre Barbillon et fréquemment en discussion avec lui et Saint-Clair Chabert-Liddell dont j'ai poursuivi les travaux.

Le contexte de travail, au sein des ingénieurs d'études, des doctorants, des chercheurs et des maîtres de conférences, a été pour moi très enrichissant. Ce stage s'inscrit dans la construction de mon parcours professionnel en validant le désir que je présentais de faire de la recherche.

Par ailleurs, divers projets entrepris au sein du laboratoire ont permis de nouer des relations amicales en dehors des heures de travail. Par exemple, le projet de construction d'une borne d'arcade pour le laboratoire, impulsé par Julien Chiquet,

a été une expérience extrêmement agréable et captivante à laquelle prendre part.

J'ai particulièrement apprécié la disponibilité de toutes les personnes de l'unité qui n'ont jamais hésité à se rendre disponible pour répondre à mes questions. Les nombreux séminaires et le désir de partage de connaissances à travers des formations internes et de l'auto-formation m'a vraiment plu et m'a ouvert à de nouvelles problématiques passionnantes. De plus j'ai beaucoup progressé dans les domaines abordés pendant mon stage, et cela m'a rendu confiant dans le choix de faire le master *MathSV* pour l'année scolaire 2023-2024. Ce stage a donc été déterminant et confirme l'orientation de mon parcours professionnel.

Note La suite de ce rapport a été rédigée en anglais.

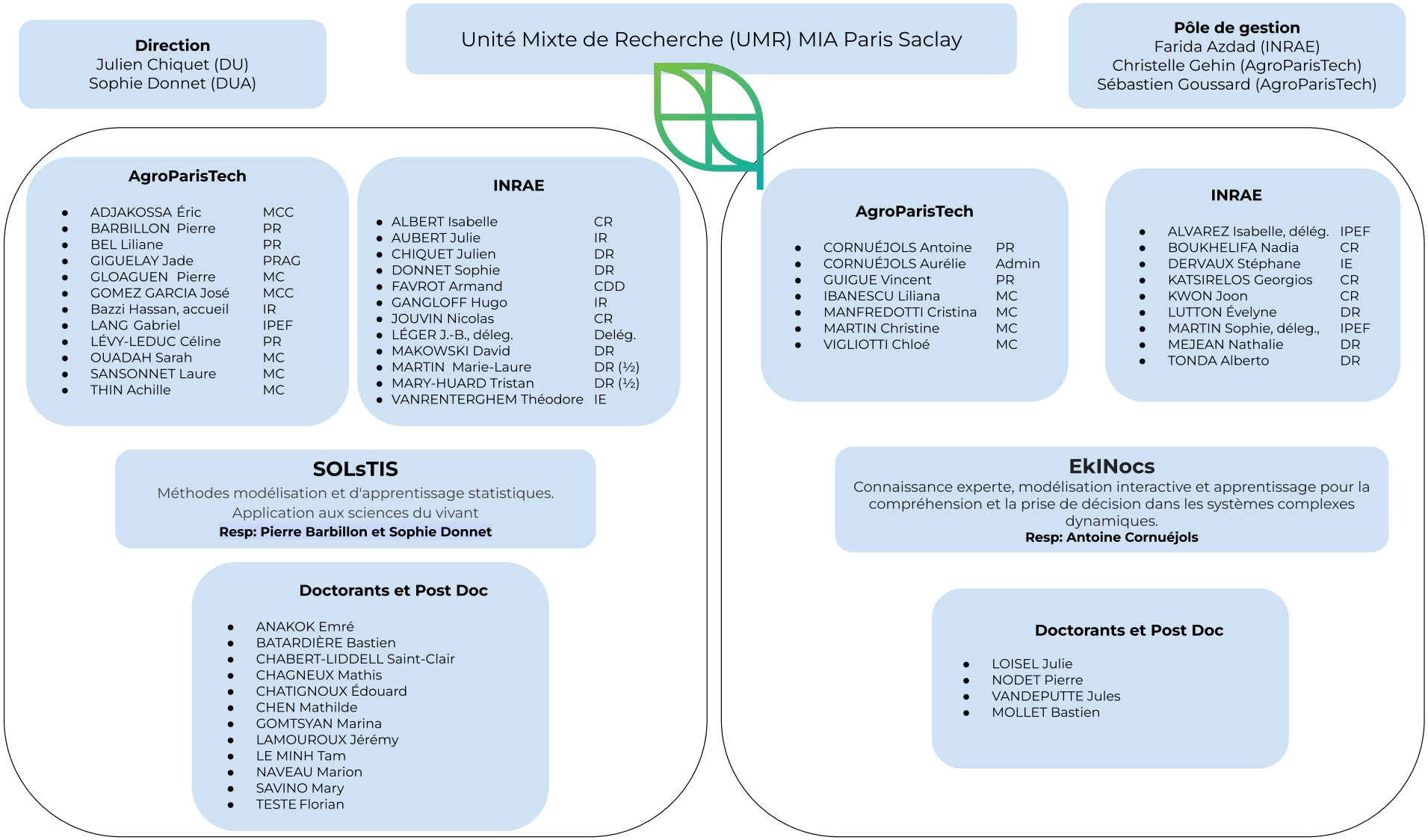


Figure 1.1: Organigramme de l'UMR

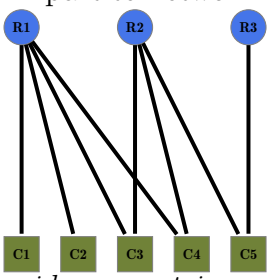
Chapter 2

Context of the study

2.1 Usage and importance of bipartite graphs

Bipartite graphs, denoted as $G = (U, V, E)$ with U and V two disjoint and independent sets of vertices and E the set of edges connecting U vertices to V vertices.

Bipartite network



Incidence matrix

$$X = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

X is the *incidence matrix* and is the mathematical object on which computations are performed. It is filled with the following rule:

$$\begin{cases} X_{ij} = 0 & \text{if no interaction is observed between species } i \text{ and } j \\ X_{ij} \neq 0 & \text{otherwise} \end{cases}$$

If the network represents binary observation (like presence-absence observation) then $X_{ij} \in \mathcal{K} = \{0, 1\}, \forall (i, j)$; if the interactions are weighted (like an abundance count), $X_{ij} \in \mathcal{K} = \mathbb{N}, \forall (i, j)$.

This representation can be used to represent various forms of interactions were two kinds of "actors" interact. Those interactions can be binary or valued and a numeric representation is the incidence matrix, in the above example X.

Among the use case of bipartite graphs one can find the Netflix Problem, which was a prize organized by Netflix to improve its Recommender system. The row nodes are the movies and the columns are the user, at the intersection the value is the review of the user j for the movie i.

Another use is the representation of ecological interactions like plant-pollinator (Ramos-Jiliberto et al. 2010), birds-seed dispersion, prey-predator or host-parasite (Kaszewska-Gilas et al. 2021). In those cases, the rows are pollinator species and the columns are plant species, and the intersection is a value, binary if it is a presence/absence or a value if it is an abundance count.

Bipartite graphs are widely used in biology, in various fields, among which the previously cited ecological networks, but also in medicine with biomedical networks, biomolecular networks or epidemiological networks. (Pavlopoulos et al. 2018)

Some interesting results can arise when applying a tool widely used on a particular kind of interactions is used on another kind of interactions. Companies like Netflix use recommender system, to recommend another product to consumers based on their previous interactions. In Desjardins-Proulx et al. 2017 the authors use the *K-nearest neighbour* (KNN) algorithm as a Recommender to predict missing preys for predators in a predator-prey network.

2.2 Latent Block Model

The Latent Block Model (LBM) introduced by Gérard Govaert and Mohamed Nadif 2010 adapts the Stochastic Block Model (SBM) (Holland et al. 1983; Snijders and Nowicki 1997) to bipartite graphs.

Please note that we prefer the term "BiSBM" and will use both LBM and BiSBM to designate the Stochastic Block model applied on bipartite networks.

This model supposes that:

- Row nodes are members of row blocks and column nodes are members of column blocks.
- The connectivity of two individuals is determined by their block memberships.
- An interaction can only occur between a row and a column node.

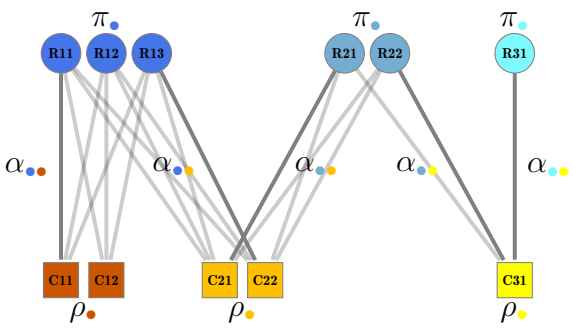


Figure 2.1: An LBM model visualization

- $Q_1 = |\{\bullet, \bullet, \bullet\}|$ *given* blocks in rows
- $Q_2 = |\{\bullet, \bullet, \bullet\}|$ *given* blocks in columns

Parameters

- $\pi_\bullet = \mathbb{P}(Z_i = \bullet)$ for rows and $\rho_\bullet = \mathbb{P}(W_j = \bullet)$ for columns
- $\alpha_{\bullet\bullet} = \mathbb{P}(X_{ij} = 1 | Z_i = \bullet, W_j = \bullet)$, probability of connectivity knowing node membership blocks.

On 2.1, $\boldsymbol{\pi}$ are the probabilities for a row node to belong to the row block of corresponding color, $\boldsymbol{\rho}$ are the probabilities for a column node to belong to the column block of corresponding color and $\boldsymbol{\alpha}$ are the connectivity parameters between the row and column blocks.

This model can be used to easily generate bipartite graphs with complex and very varied structures. But when trying to determine the structure of a given network we need to find those parameters and as the row and column block memberships are *latent* i.e., they are not known and must be inferred.

For this a common approach is to use a VEM algorithm (proposed for SBM in Daudin et al. 2008 and for LBM in G. Govaert and M. Nadif 2005) those groups and the required parameters can be inferred by maximizing a lower bound of the likelihood minus a penalty.

2.3 colSBM model, a joint model for a collection of networks

The *colSBM* model introduced by Chabert-Liddell et al. 2023, March 27 propose an extension of the SBM model to collections of SBMs. A collection is a set of networks which nodes are not common or linked between different networks, the interactions have the same valuations and are of the same type.

The model can retrieve the shared structure in a collection, indicate if networks should be grouped in a collection and in a large pool of networks, collections with common structures.

The next step after designing this collection model for unipartite was to adapt it to the bipartite case.

Chapter 3

Structure detection in a collection of bipartite networks : Adjustment of colSBM to the bipartite case

3.1 Definition of a collection

We define a collection of bipartite networks as $\boldsymbol{X} = (X^1, \ldots, X^M)$ the collection of incidence matrix. Moreover, all the networks in the collection have the same type of interaction (e.g., all interactions are binary).

3.2 Separate BiSBM (sep-BiSBM)

A first approach to deal with a collection of networks is to adjust separate BiSBM for each network of the collection.

For network m, let n_1^m (resp. n_2^m) be the number of nodes in row (resp. column) divided into Q_1^m row clusters (resp. Q_2^m column clusters).
Let $Z^m = (Z_i^m, \ldots, Z_{n_1^m}^m)$ and $W^m = (W_j^m, \ldots, W_{n_2^m}^m)$ be independent latent variables such that $Z_i^m = q$ if row node i of network m belongs to row cluster q ($q \in \{1, \ldots, Q_1^m\}$) and $W_j^m = r$ if column node j of network m belong to column block r ($r \in \{1, \ldots, Q_2^m\}$). And we have

$$\mathbb{P}(Z_i^m = q) = \pi_q^m, \qquad \mathbb{P}(W_j^m = r) = \rho_r^m \tag{3.1}$$

where $\pi_q^m > 0$, $\rho_r^m > 0$, $\sum_{q=1}^{Q_1^m} \pi_q^m = 1$ and $\sum_{r=1}^{Q_2^m} \rho_r^m = 1$. Given the latent variables Z^m, W^m, the X_{ij}^ms are assumed to be independent and distributed as

$$X_{ij}^m | Z_i^m = q, W_j^m = r \sim \mathcal{F}(.; \alpha_{qr}^m) \tag{3.2}$$

where $\mathcal{F}$ is referred to as the emission distribution. $\mathcal{F}$ is chosen to be the Bernoulli distribution for binary interactions, and the Poisson distribution for weighted

interactions such as counts. Let f be the density of the emission distribution, then:

$$\log f(X_{ij}^m; \alpha_{qr}^m) = \begin{cases} X_{ij}^m \log(\alpha_{qr}^m) + (1 - X_{ij}^m) \log(1 - \alpha_{qr}^m) & \text{for Bernoulli emission} \\ -\alpha_{qr}^m + X_{ij}^m \log(\alpha_{qr}^m) - \log(X_{ij}^m!) & \text{for Poisson emission} \end{cases} \tag{3.3}$$

Equations (3.1), (3.2) and (3.3) defines the BiSBM model and we will now use a short notation:

$$X^m \sim \mathcal{F}\text{-BiSBM}_{n_1^m, n_2^m}(Q_1^m, Q_2^m, \boldsymbol{\pi^m}, \boldsymbol{\rho^m}, \boldsymbol{\alpha^m}) \qquad (\textit{sep-BiSBM})$$

where $\mathcal{F}$ encodes the emission distribution, n_1^m, n_2^m are the row and column nodes, Q_1^m, Q_2^m are the number of row and column blocks in network m, $\boldsymbol{\pi}^m = (\pi_q^m)_{q=1,\dots,Q_1^m}$ and $\boldsymbol{\rho}^m = (\rho_r^m)_{r=1,\dots,Q_2^m}$ are the vectors of their proportions. The $Q_1^m \times Q_2^m$ matrix $\boldsymbol{\alpha}^m = (\alpha_{qr}^m)_{\substack{q=1,\dots,Q_1^m \\ r=1,\dots,Q_2^m}}$ are the connectivity parameters, the parameters of the emission distribution. $\alpha_{qr}^m \in \mathcal{A}_\mathcal{F}$ where, for the Bernoulli (resp. Poisson) emission distribution, $\mathcal{A}_\mathcal{F} = (0, 1)$ (resp. $\mathcal{A}_\mathcal{F} = \mathbb{R}^{*+}$). In this *sep-BiSBM* each network m is assumed to follow a *BiSBM* with its own parameters $(\boldsymbol{\pi}^m, \boldsymbol{\rho}^m, \boldsymbol{\alpha}^m)$.

3.3 Definition of the colBiSBM models

3.3.1 A collection of i.i.d bipartite SBM

As for *colSBM* this first model is the most constrained. It assumes that all the networks are the independent realizations of the same Q_1-Q_2-BiSBM with identical parameters. The *iid-colBiSBM* is defined as follows:

$$X^m \sim \mathcal{F} - BiSBM_{n_1^m, n_2^m}(Q_1, Q_2, \boldsymbol{\pi}, \boldsymbol{\rho}, \boldsymbol{\alpha}), \quad \forall m = 1, \dots M \qquad (\textit{iid-colBiSBM})$$

where $\forall (q, r) \in \{1, \dots, Q_1\} \times \{1, \dots, Q_2\}$, $\alpha_{qr} \in \mathcal{A}_\mathcal{F}$, $\pi_q \in (0, 1]$, $\sum_{q=1}^{Q_1} \pi_q = 1$ and $\rho_r \in (0, 1]$, $\sum_{r=1}^{Q_2} \rho_r = 1$. This model involves $(Q_1 - 1) + (Q_2 - 1) + Q_1 \times Q_2$ parameters, the two first terms corresponding to block proportions on the row and column dimensions and the third term to connectivity parameters.

But the assumption that block proportions are the same among the networks is a strong assumption. In plant-pollinator networks, the proportion of specialist species can differ between networks and thus the model may benefit from not having the same block proportions but sharing a common connectivity structure. The following models relaxes this assumption on either row, column or both.

3.3.2 A collection of bipartite SBM with varying block size on either rows or columns

π-colBiSBM model still assumes that the networks share a common connectivity structure represented by $\boldsymbol{\alpha}$ but that each network has its own row block proportions. For $m \in \{1, \dots, M\}$, the X^m are independent and

$$X^m \sim \mathcal{F} - BiSBM_{n_1^m, n_2^m}(Q_1, Q_2, \boldsymbol{\pi^m}, \boldsymbol{\rho}, \boldsymbol{\alpha}), \quad \forall m = 1, \dots, M \quad (\pi\text{-}colBiSBM)$$

where $\forall (q, r) \in \{1, \dots, Q_1\} \times \{1, \dots, Q_2\}$, $\alpha_{qr} \in \mathcal{A}_{\mathcal{F}}$, $\pi_q^m \in [0, 1]$, $\sum_{q=1}^{Q_1} \pi_q^m = 1, \forall m \in \{1, \dots, M\}$ and $\rho_r \in (0, 1]$, $\sum_{r=1}^{Q_2} \rho_r = 1$. This model is more flexible than the iid-colBiSBM as it allows some row block proportions to be null in certain networks ($\pi_q^m \in [0, 1]$): if $\pi_q^m = 0$ then the block q is not represented in the network m. The connectivity structure is thus a subset of a large connectivity structure common to all networks. We face the same problems as Chabert-Liddell et al. 2023, March 27 and adapt the support S they define for the π-colSBM to the bipartite case by having S^1 of size $M \times Q_1$ the support for the rows and S^2 of size $M \times Q_2$ the support for the columns. Thus $S^1_{mq} = \mathbb{1}_{\pi_q^m > 0}$ and $S^2_{mr} = \mathbb{1}_{\rho_r^m > 0}$. In this case, $S^2 = \mathbf{1}$, because there is no freedom on the column dimension.

For a given number of blocks Q_1, Q_2 and matrix S^1 (S^2 being in this case the matrix full of ones), the number of parameters is:

$$\text{NP}(\pi\text{-}colBiSBM) = \sum_{m=1}^{M} \left(\sum_{q=1}^{Q_1} S^1_{mq} - 1 \right) + (Q_2 - 1) + \sum_{\substack{q=1,\dots,Q_1 \\ r=1,\dots,Q_2}} \mathbb{1}_{(S^{1\prime} S^2)_{qr} > 0}$$

The first term corresponds to the non-null block proportions in each network. The third quantity accounts for the fact that some blocks may never be represented simultaneously in any network, so the corresponding connection parameters α_{qr} are not useful for defining the model.

ρ-colBiSBM model still assumes that the networks share a common connectivity structure represented by $\boldsymbol{\alpha}$ but that each network has its own column block proportions. For $m \in \{1, \dots, M\}$, the X^m are independent and

$$X^m \sim \mathcal{F} - BiSBM_{n_1^m, n_2^m}(Q_1, Q_2, \boldsymbol{\pi}, \boldsymbol{\rho^m}, \boldsymbol{\alpha}), \quad \forall m = 1, \dots, M \quad (\rho\text{-}colBiSBM)$$

where $\forall (q, r) \in \{1, \dots, Q_1\} \times \{1, \dots, Q_2\}$, $\alpha_{qr} \in \mathcal{A}_{\mathcal{F}}$, $\pi_q \in (0, 1]$, $\sum_{q=1}^{Q_1} \pi_q = 1$ and $\rho_r^m \in [0, 1]$, $\sum_{r=1}^{Q_2} \rho_r^m = 1$. This model is more flexible than the iid-colBiSBM as it allows some column block proportions to be null in certain networks ($\rho_r^m \in [0, 1]$): if $\rho_r^m = 0$ then the column block r is not represented in the network m. "Mirroring" the formulas for the $\pi\text{-}colBiSBM$ we relax the constraints on the column dimension.

For a given number of blocks Q_1, Q_2 and matrix S^2 (S^1 being in this case the matrix full of ones), the number of parameters is:

$$\text{NP}(\rho\text{-}colBiSBM) = (Q_1 - 1) + \sum_{m=1}^{M} \left(\sum_{r=1}^{Q_2} S^2_{mr} - 1 \right) + \sum_{\substack{q=1,\dots,Q_1 \\ r=1,\dots,Q_2}} \mathbb{1}_{(S^{1\prime} S^2)_{qr} > 0}$$

$\pi\rho$-colBiSBM model still assumes that the networks share a common connectivity structure represented by $\boldsymbol{\alpha}$ but that each network has its own row and column block proportions, it is the less constrained model. For $m \in \{1, \dots, M\}$, the X^m are independent and

$$X^m \sim \mathcal{F} - BiSBM_{n_1^m, n_2^m}(Q_1, Q_2, \boldsymbol{\pi^m}, \boldsymbol{\rho^m}, \boldsymbol{\alpha}), \qquad \forall m = 1, \dots, M \quad (\pi\rho\text{-}colBiSBM)$$

where $\forall (q,r) \in \{1,\dots,Q_1\} \times \{1,\dots,Q_2\}$, $\alpha_{qr} \in \mathcal{A}_{\mathcal{F}}$, $\pi_q^m \in [0,1]\,, \sum_{q=1}^{Q_1} \pi_q^m = 1, \forall m \in \{1,\dots,M\}$ and $\rho_r^m \in [0,1]\,, \sum_{r=1}^{Q_2} \rho_r^m = 1$.

For a given number of blocks Q_1, Q_2 and matrices S^1, S^2, the number of parameters is:

$$\text{NP}(\pi\rho\text{-}colBiSBM) = \sum_{m=1}^{M} \left(\sum_{q=1}^{Q_1} S^1_{mq} - 1 \right) + \sum_{m=1}^{M} \left(\sum_{r=1}^{Q_2} S^2_{mr} - 1 \right) + \sum_{\substack{q=1,\dots,Q_1 \\ r=1,\dots,Q_2}} \mathbb{1}_{(S^{1\prime} S^2)_{qr} > 0}$$

3.4 Variational estimation of the parameters

In practice, the estimation of the likelihood is not tractable. Following the classical approach defined in Daudin et al. 2008 we use a variatonal version of the Expectation Maximization (VEM) algorithm.

We maximize a variational lower bound of the log-likelihood of the observed data by approximating $p(\boldsymbol{Z}, \boldsymbol{W} | \boldsymbol{X}; \boldsymbol{\theta})$ with a distribution on $\boldsymbol{Z}$ and $\boldsymbol{W}$ named $\mathcal{R}$ defined as $\mathcal{R} = \otimes_{m=1}^{M} \mathcal{R}_m$.

The lower bound is defined as:

$$\mathcal{J}(\mathcal{R}; \boldsymbol{\theta}) \coloneqq \sum_{m=1}^{M} \left(\mathbb{E}_{\mathcal{R}_m}[\ell(X^m, Z^m, W^m; \boldsymbol{\theta})] + \mathcal{H}(\mathcal{R}_m) \right) \leq \ell(\boldsymbol{X}; \boldsymbol{\theta})$$

$\boldsymbol{Z}$ and $\boldsymbol{W}$ are redefined using the *one-hot encoded* conversion (i.e., $Z_i^m = q \rightarrow Z_{iq}^m = 1$ and $W_j^m = r \rightarrow W_{jr}^m = 1$).

When $\mathcal{R}_m$ is issued from the set of the factorizable distributions, we denote $\tau_{iq}^{1,m} = \mathbb{P}_{\mathcal{R}_m}(Z_{iq}^m = 1 | X_{ij}^m)$ and $\tau_{jr}^{2,m} = \mathbb{P}_{\mathcal{R}_m}(W_{jr}^m = 1 | X_{ij}^m)$, thus we have: $\mathbb{P}_{\mathcal{R}_m}(Z_{iq}^m = 1, W_{jr}^m = 1 | X_{ij}^m) = \mathbb{P}_{\mathcal{R}_m}(Z_{iq}^m = 1 | X_{ij}^m) \times \mathbb{P}_{\mathcal{R}_m}(W_{jr}^m = 1 | X_{ij}^m) = \tau_{iq}^{1,m} \times \tau_{jr}^{2,m}$.

The formula for the entropy per network is thus:

$$\mathcal{H}(\mathcal{R}_m) = -\sum_{i=1}^{n_1} \tau_{i,q}^{1,m} \log \tau_{i,q}^{1,m} - \sum_{j=1}^{n_2} \tau_{j,r}^{2,m} \log \tau_{j,r}^{2,m}$$

And the expectation of the completed log-likelihood under the $\mathcal{R}_m$ variational distribution for network m is:

$$\begin{aligned}\mathbb{E}_{\mathcal{R}_m}[\ell(X^m, Z^m, W^m; \boldsymbol{\theta})] = &\sum_{i=1}^{n_1^m} \sum_{j=1}^{n_2^m} \sum_{q \in \mathcal{Q}_{1,m}} \sum_{r \in \mathcal{Q}_{2,m}} \tau_{i,q}^{1,m} \tau_{j,r}^{2,m} \log f(X_{ij}^m; \alpha_{qr}) \\ &+ \sum_{i=1}^{n_1^m} \sum_{q \in \mathcal{Q}_{1,m}} \tau_{i,q}^{1,m} \log {\pi_q}^m + \sum_{j=1}^{n_2^m} \sum_{r \in \mathcal{Q}_{2,m}} \tau_{j,r}^{2,m} \log {\rho_r}^m\end{aligned}$$

And thus the lower bound becomes:

$$\begin{aligned}\mathcal{J}(\boldsymbol{\tau}; \boldsymbol{\theta}) \coloneqq \sum_{m=1}^{M} \Bigg(&\sum_{i=1}^{n_1^m} \sum_{j=1}^{n_2^m} \sum_{q \in \mathcal{Q}_{1,m}} \sum_{r \in \mathcal{Q}_{2,m}} \tau_{i,q}^{1,m} \tau_{j,r}^{2,m} \log f(X_{ij}^m; \alpha_{qr}) \\ &+ \sum_{i=1}^{n_1^m} \sum_{q \in \mathcal{Q}_{1,m}} \tau_{i,q}^{1,m} \log {\pi_q}^m + \sum_{j=1}^{n_2^m} \sum_{r \in \mathcal{Q}_{2,m}} \tau_{j,r}^{2,m} \log {\rho_r}^m \\ &- \sum_{i=1}^{n_1} \tau_{i,q}^{1,m} \log \tau_{i,q}^{1,m} - \sum_{j=1}^{n_2} \tau_{j,r}^{2,m} \log \tau_{j,r}^{2,m} \Bigg)\end{aligned}$$

where we identify the variational distribution $\mathcal{R}$ with its parameter $\boldsymbol{\tau}$.

The VEM algorithm alternates between two steps, the variational E step and the M step. The E steps consists in optimizing $\mathcal{J}(\boldsymbol{\tau}; \boldsymbol{\theta})$ for a current value of $\boldsymbol{\theta}$ with respect to $\boldsymbol{\tau}$. And the M step consists of maximizing $\mathcal{J}(\boldsymbol{\tau}; \boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ and for a given variational distribution $\boldsymbol{\tau}$.

3.4.1 Variational E step

At this step we maximize with respect to the variational distribution $\boldsymbol{\tau}$:

$$\widehat{\boldsymbol{\tau}}^{(t+1)} = \arg\max_{\boldsymbol{\tau}} \mathcal{J}(\boldsymbol{\tau}, \widehat{\boldsymbol{\theta}}^{(t)}).$$

And we obtain the following formulae for the $\boldsymbol{\tau^m}$:

$$\widehat{\tau}_{iq}^{1,m} \propto \widehat{\pi}_q^{m(t)} \prod_{j=1}^{n_2^m} \prod_{r \in \mathcal{Q}_2^m} f(X_{ij}^m; \widehat{\alpha}_{qr}^{(t)})^{\widehat{\tau}_{jr}^{2,m(t+1)}} \forall i = 1, \ldots, n_1^m, q \in \mathcal{Q}_1^m$$

$$\widehat{\tau}_{jr}^{2,m} \propto \widehat{\rho}_r^{m(t)} \prod_{i=1}^{n_1^m} \prod_{q \in \mathcal{Q}_1^m} f(X_{ij}^m; \widehat{\alpha}_{qr}^{(t)})^{\widehat{\tau}_{iq}^{1,m(t+1)}} \forall j = 1, \ldots, n_2^m, r \in \mathcal{Q}_2^m$$

which are used to update iteratively the values by a fixed point algorithm with only one step.

3.4.2 M step of the algorithm

At iteration (t) the M-step maximizes the variational bound with respect to the model parameters $\boldsymbol{\theta}$:

$$\widehat{\boldsymbol{\theta}}^{(t+1)} = \arg\max_{\boldsymbol{\theta}} \mathcal{J}(\widehat{\boldsymbol{\tau}}^{(t+1)}, \boldsymbol{\theta})$$

The following quantities are involved in the obtained formulae:

$$e_{qr}^m = \sum_{i=1}^{n_1^m} \sum_{j=1}^{n_2^m} \tau_{iq}^{1,m} \tau_{jr}^{2,m} X_{ij}^m, \quad n_{qr}^m = \sum_{i=1}^{n_1^m} \sum_{j=1}^{n_2^m} \tau_{iq}^{1,m} \tau_{jr}^{2,m}, \quad n_q^{1,m} = \sum_{i=1}^{n_1^m} \tau_{iq}^{1,m}, \quad n_r^{2,m} = \sum_{j=1}^{n_2^m} \tau_{jr}^{2,m}$$

The block proportions, in free mixture models, $(\pi_q^m)_{q \in \mathcal{Q}_1^m}, (\rho_r^m)_{r \in \mathcal{Q}_2^m}$ are estimated as

$$\widehat{\pi}_q^m = \frac{n_q^{1,m}}{n_1^m} \qquad \text{for } \pi\text{-}colBiSBM \text{ and } \pi\rho\text{-}colBiSBM$$

$$\widehat{\rho}_r^m = \frac{n_r^{2,m}}{n_2^m} \qquad \text{for } \rho\text{-}colBiSBM \text{ and } \pi\rho\text{-}colBiSBM$$

while on the other hand,

$$\widehat{\pi}_q = \frac{\sum_{m=1}^M n_q^{1,m}}{\sum_{m=1}^M n_1^m} \qquad \text{for } iid\text{-}colBiSBM \text{ and } \rho\text{-}colBiSBM$$

$$\widehat{\rho}_r = \frac{\sum_{m=1}^M n_r^{2,m}}{\sum_{m=1}^M n_2^m} \qquad \text{for } iid\text{-}colBiSBM \text{ and } \pi\text{-}colBiSBM$$

the parameters takes into account all the networks at the same time. The connectivity parameters α_{qr} for all models are estimated as the ratio of the number of interactions between row block q and column block r among all networks over the number of number of possible interactions:

$$\widehat{\alpha}_{qr} = \frac{\sum_{m=1}^M e_{qr}^m}{\sum_{m=1}^M n_{qr}^m}$$

3.5 Model selection

As discussed in Chabert-Liddell et al. 2023, March 27, the algorithmic aspect becomes complex when dealing with the bipartite case. Due to the size of the latent space being $\mathbb{N}^2$, conducting a complete exploration of the latent space is practically infeasible. Therefore, in addition to adapting the existing formulas, our contribution to addressing this challenge involved making significant choices, which are outlined below.

The below procedures are implemented in the *colSBM* package, available on https://github.com/Chabert-Liddell/colSBM.

3.5.1 The BIC-L criterion for model selection

The Integrated Classified Likelihood (ICL) is a well-established tool in the SBM and LBM domains for selecting the appropriate number of blocks. It was introduced by Biernacki et al. 2000; Daudin et al. 2008. The ICL is derived from an asymptotic approximation of the marginal complete likelihood. In this approach, the model parameters are integrated out using a prior distribution, resulting in a penalized likelihood criterion. By employing the ICL, one can effectively determine the optimal number of blocks for the given problem in a systematic manner. We obtain the following expression

$$\text{ICL} = \max_{\theta} \mathbb{E}_{\widehat{\mathcal{R}}}[\ell(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{W}; \boldsymbol{\theta})] - \frac{1}{2}\text{pen}$$

with pen the penalties.
Using the formula $\mathbb{E}_{\widehat{\mathcal{R}}}[\ell(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{W}; \boldsymbol{\theta})] \approx \ell(\boldsymbol{X}; \boldsymbol{\theta}) - \mathcal{H}(\widehat{\mathcal{R}})$, it becomes clearer, as highlighted in the existing literature, that the Integrated Classified Likelihood (ICL) gives preference to well-separated blocks by imposing a penalty on the entropy of node grouping. However, the objective of our study extends beyond grouping nodes into coherent blocks. We also aim to assess the similarity of connectivity patterns across different networks. Consequently, we aim to permit models that offer more flexible node grouping without penalizing entropy. This leads us to formulate a BIC-like criterion in the following manner:

$$\text{BIC-L} = \max_{\boldsymbol{\theta}} \mathbb{E}_{\widehat{\mathcal{R}}}[\ell(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{W}; \boldsymbol{\theta})] + \mathcal{H}(\widehat{\mathcal{R}}) - \frac{1}{2}\text{pen} = \max_{\boldsymbol{\theta}} \mathcal{J}(\widehat{\mathcal{R}}, \boldsymbol{\theta}) - \frac{1}{2}\text{pen}$$

We provide below the expression for the penalties for the 4 models that we propose.

iid-colBiSBM For the *iid-colBiSBM* the penalties were modified in the following way:

- For the πs and ρs:

$$\text{pen}_\pi(Q_1) = (Q_1 - 1)\log(\sum_{m=1}^{M} n_1^m)$$

$$\text{pen}_\rho(Q_2) = (Q_2 - 1)\log(\sum_{m=1}^{M} n_2^m)$$

- For the αs :

$$\text{pen}_\alpha(Q_1, Q_2) = Q_1 \times Q_2 \log(N_M)$$

with

$$N_M = \sum_{m=1}^{M} n_1^m \times n_2^m$$

And thus the BIC-L formula is now:

$$\text{BIC-L}(\boldsymbol{X}, Q_1, Q_2) = \max_\theta \mathcal{J}(\hat{\mathcal{R}}, \boldsymbol{\theta}) - \frac{1}{2}[\text{pen}_\pi(Q_1) + \text{pen}_\rho(Q_2) + \text{pen}_\alpha(Q_1, Q_2)]$$

$\rho\pi$-colBiSBM For the *$\rho\pi$-colBiSBM* the penalties are the following:

- The support penalties are:

$$\text{pen}_{S_1}(Q_1) = -2\log p_{Q_1}(S_1)$$

$$\text{pen}_{S_2}(Q_2) = -2\log p_{Q_2}(S_2)$$

with

$$\log p_{Q_1}(S_1) = -M\log(Q_1) - \sum_{m=1}^{M} \log \binom{Q_1}{Q_1^{(m)}}$$

$$\log p_{Q_2}(S_2) = -M\log(Q_2) - \sum_{m=1}^{M} \log \binom{Q_2}{Q_2^{(m)}}$$

- Penalties for the ρs and πs:

$$\text{pen}_\pi(Q_1, S_1) = \sum_{m=1}^{M} (Q_1^{(m)} - 1)\log n_1^m$$

$$\text{pen}_\rho(Q_2, S_2) = \sum_{m=1}^{M} (Q_2^{(m)} - 1)\log n_2^m$$

- Penalties for the αs:

$$\text{pen}_\alpha(Q_1, Q_2, S_1, S_2) = (\sum_{q=1}^{Q_1}\sum_{r=1}^{Q_2} \mathbb{1}_{(S_1)'S_2>0}) \log(N_M)$$

And the corresponding BIC-L formula:

$$\begin{aligned}\text{BIC-L}(\boldsymbol{X}, Q_1, Q_2) = \max_{S_1,S_2}[& \max_{\theta_{S_1,S_2}\in\Theta_{S_1,S_2}} \mathcal{J}(\hat{\mathcal{R}}, \theta_{S_1,S_2}) \\ & -\frac{1}{2}(\text{pen}_\pi(Q_1, S_1) + \text{pen}_\rho(Q_2, S_2) \\ & \quad + \text{pen}_\alpha(Q_1, Q_2, S_1, S_2) \\ & \quad + \text{pen}_{S_1}(Q_1) + \text{pen}_{S_2}(Q_2))]\end{aligned}$$

3.5.2 Initialization and pairing of the models

First to combine the information from the M networks we fit a collection model for each network at the two points $Q = (1, 2)$ and $Q = (2, 1)$. Using the previously described VEM algorithm we obtain for each network its parameters $(\boldsymbol{\rho}, \boldsymbol{\pi}, \boldsymbol{\alpha})$.

We then compute the marginal laws for each dimension, for each network. Then we order the network blocks by the probabilities obtained in decreasing order.

- For the memberships on the columns: $col\ order_m = order\,(\pi_m \times \alpha_m)$
- For the memberships on the rows: $row\ order_m = order\,(\rho_m \times\ {}^t(\alpha_m))$

Using this order we relabel the memberships for the M fitted collection of a single network. Then we use the M memberships to fit a collection containing the M networks.

3.5.3 Greedy exploration to find an estimation of the mode

Using the previously fitted models for $Q = (1, 2)$ and $Q = (2, 1)$ we choose to perform a greedy exploration to find a first mode.

Meaning that for a given $Q = (Q_1, Q_2)$ we will compute all the possible memberships for the points $Q \in \{(Q_1+1, Q_2), (Q_1, Q_2+1), (Q_1-1, Q_2), (Q_1, Q_2-1)\}$, fit the corresponding models and choose the one that maximizes the BIC-L as the next point from which to repeat the procedure. We repeat the procedure until the BIC-L stops increasing 2 times in a row.

Input : Fitted models for $Q = (1, 2)$ and $Q = (2, 1)$
Output: Estimation of the mode using greedy exploration

Initialize $Q = (1, 2)$ as the starting point Initialize BIC-$L_{\max}$ as the maximum achieved BIC-L value Initialize *consecutive_count* as 0

while *consecutive_count* < 2 **do**
  Compute possible memberships for $Q \in \{(Q_1 + 1, Q_2), (Q_1, Q_2 + 1), (Q_1 - 1, Q_2), (Q_1, Q_2 - 1)\}$;
  Fit models with the computed memberships Choose the model with the maximum BIC-L as the next point
  if *BIC-L* $>$ *BIC-*L_{max} **then**
    BIC-$L_{\max} \leftarrow$ BIC-L *consecutive_count* $\leftarrow 0$
  end
  else
    consecutive_count $\leftarrow$ *consecutive_count* $+ 1$
  end
  $Q \leftarrow$ Next selected point
end

Output: Estimation of the mode using greedy exploration

Algorithm 1: Greedy Exploration for Mode Estimation

When this first estimation of the BIC-L mode has been find we apply the moving window on it.

3.5.4 Moving window to update the block memberships and the BIC-L

The *moving window* is used to update the block memberships on rows and columns and fit new models with those changes. To define the window, we use a center point and a *depth*, giving us the bottom left corner $(Q_{1,center} - depth, Q_{2,center} - depth)$ and the top right corner of the window $(Q_{1,center} + depth, Q_{2,center} + depth)$. All the points in this square will be updated and contribute to the update of the others. This procedure is repeated until convergence of the BIC-L.

The figure 3.1 illustrates the procedure. It consists of two alternating steps:

- the *forward pass*: repeatedly computing the possible splits to fit the current model.
- the *backward pass*: computing the possible merges to fit the current model.

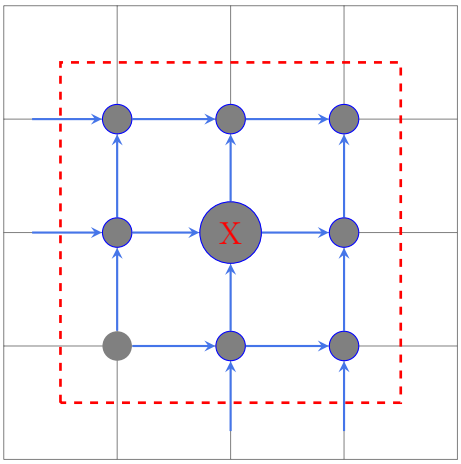
Input : Center point $(Q_{1,\text{center}}, Q_{2,\text{center}})$, depth
Output: Best model with maximum BIC-L in the window

Define bottom left corner $(Q_{1,\text{center}} - \text{depth}, Q_{2,\text{center}} - \text{depth})$
Define top right corner $(Q_{1,\text{center}} + \text{depth}, Q_{2,\text{center}} + \text{depth})$

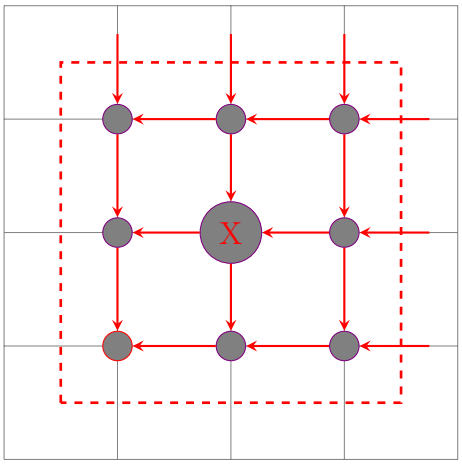
while *not converged* **do**
    Forward pass:
    for $Q_1 \in [Q_{1,center} - depth; Q_{1,center} + depth]$ **do**
        for $Q_2 \in [Q_{2,center} - depth; Q_{2,center} + depth]$ **do**
            Compute possible splits from predecessors $(Q_1 - 1, Q_2)$ and $(Q_1, Q_2 - 1)$ Fit models with the block membership changes Compare and keep the best model based on BIC-L
        end
    end
    Backward pass:
    for $Q_1 \in [Q_{1,center} + depth; Q_{1,center} - depth]$ **do**
        for $Q_2 \in [Q_{2,center} + depth; Q_{2,center} - depth]$ **do**
            Compute possible merges from predecessors $(Q_1 + 1, Q_2)$ and $(Q_1, Q_2 + 1)$ Fit models with the block membership changes Compare and keep the best model based on BIC-L
        end
    end
    Update the best model based on the maximum BIC-L
end

Output: Best model with maximum BIC-L in the window

Algorithm 2: Moving Window Procedure



(a) Visualisation of a forward pass of moving window



(b) Visualisation of a backward pass of moving window

Figure 3.1: Moving window procedure, the center node marked with an X is the mode of BIC-L

Forward pass The forward pass consists for a model at (Q_1, Q_2) to compute the possible splits from the block memberships of its "predecessors". The predecessors are the point at the left $(Q_1 - 1, Q_2)$ and below $(Q_1, Q_2 - 1)$ the current model (if they exist). To update the current model, we take its predecessors block memberships and try to split one of the blocks in two. Then the current model is fitted using this clustering as a starting clustering. Once all the possible splits are fitted, they are compared, keeping the best, in the sense of the BIC-L. If a model was already present it is also compared and the best is chosen as the model for this round at (Q_1, Q_2).
The procedure then repeats for the point at (Q_1+1, Q_2) until it reaches $(Q_{1,center}+depth, Q_2)$ from which it repeats from $(Q_{1,center}-depth, Q_2+1)$. This repeats until computing the best model for $(Q_{1,center} + depth, Q_{2,center} + depth)$. *Note on the initialization:* The forward pass starts from the point $(Q_{1,center}+depth, Q_{2,center}+depth)$, so this points needs to have at least a model fitted. In the best case, the greedy exploration will have visited this point. But if the point has not been visited, a model will be fitted from a spectral initialization (i.e the block memberships is computed by using a spectral clustering). From this point, the next model will have at least one predecessor and the procedure can iterate.

Backward pass The backward pass consists for a model at (Q_1, Q_2) to compute the possible merges from the block memberships of its "predecessors". The predecessors are the point at the right $(Q_1 + 1, Q_2)$ and on top $(Q_1, Q_2 + 1)$ of the current model (if the predecessors exist). To update the current model, we take its predecessors block memberships and try to merge two blocks in one. Then

the current model is fitted using this clustering as a starting clustering. Once all the possible merges are fitted, they are compared, keeping the best, in the sense of the BIC-L. If a model was already present it is also compared and the best is chosen as the model for this round at (Q_1, Q_2).
The procedure then repeats for the point at (Q_1-1, Q_2) until it reaches $(Q_{1,center}-depth, Q_2)$ from which it repeats from $(Q_{1,center}-depth, Q_2-1)$. This repeats until computing the best model for $(Q_{1,center}-depth, Q_{2,center}-depth)$. *Note on the initialization:* The backward pass starts from $(Q_{1,center}+depth, Q_{2,center}+depth)$, we know it was initialized at least by the forward pass, no special case here.

At the end of the moving window pass, the model of max BIC-L is the new best fit and the procedure can repeat until convergence.

3.6 Networks clustering

As in Chabert-Liddell et al. 2023, March 27 we use a recursive algorithm to determine the best clustering of the given networks. The procedure being the same, we will present it briefly and focus on adjustments.

When networks in a collection do not share the same mesoscale connectivity structure we want to be able to partition them correctly. For this we perform a clustering of networks.

The process of clustering a collection of networks involves discovering a partition $\mathcal{G} = (\mathcal{M}_g)_{g=1,\dots,G}$ of $\{1, \dots, M\}$. Given $\mathcal{G}$ we set the following model on $\boldsymbol{X}$:

$$\forall g \in \{1, \dots, G\}, \forall m \in \mathcal{M}_g, X^m \sim \mathcal{F}\text{-}BiSBM(Q_1^g, Q_2^g, \boldsymbol{\pi^m}, \boldsymbol{\rho^m}, \boldsymbol{\alpha}^g)$$

And we defined the score of a given partition $\mathcal{G}$:

$$Sc(\mathcal{G}) = \sum_{g=1}^{G} \max_{Q^g=1,\dots,Q_{\max}} \text{BIC-L}((X^m)_{m\in\mathcal{M}_g}, Q_1^g, Q_2^g)$$

Thus the score consists of the sum of the BIC-L of the sub-collections for the partition $\mathcal{G}$.

3.6.1 Dissimilarity between two networks

The parameters for the dissimilarity are defined as follow:

$$\widetilde{n}^m_{qr} = \sum_{i=1}^{n_1^m}\sum_{j=1}^{n_2^m} \widehat{\tau}^{1,m}_{iq}\widehat{\tau}^{2,m}_{jr}, \qquad \widetilde{\alpha}^m_{qr} = \frac{\sum_{i=1}^{n_1^m}\sum_{j=1}^{n_2^m} \widehat{\tau}^{1,m}_{iq}\widehat{\tau}^{2,m}_{jr} X^m_{ij}}{\widetilde{n}^m_{qr}},$$

$$\widetilde{\pi}^m_q = \frac{\sum_{i=1}^{n_1^m} \widehat{\tau}^{1,m}_{iq}}{n_1^m}, \qquad \widetilde{\rho}^m_r = \frac{\sum_{j=1}^{n_2^m} \widehat{\tau}^{2,m}_{jr}}{n_2^m}$$

And the dissimilarity between any pair of networks $(m, m') \in \mathcal{M}^2$ is then:

$$D_{\mathcal{M}}(m, m') = \sum_{q=1}^{Q_1}\sum_{r=1}^{Q_2} \max(\widetilde{\pi}^m_q, \widetilde{\pi}^{m'}_q)\left(\widetilde{\alpha}^m_{qr} - \widetilde{\alpha}^{m'}_{qr}\right)^2 \max(\widetilde{\rho}^m_r, \widetilde{\rho}^{m'}_r)$$

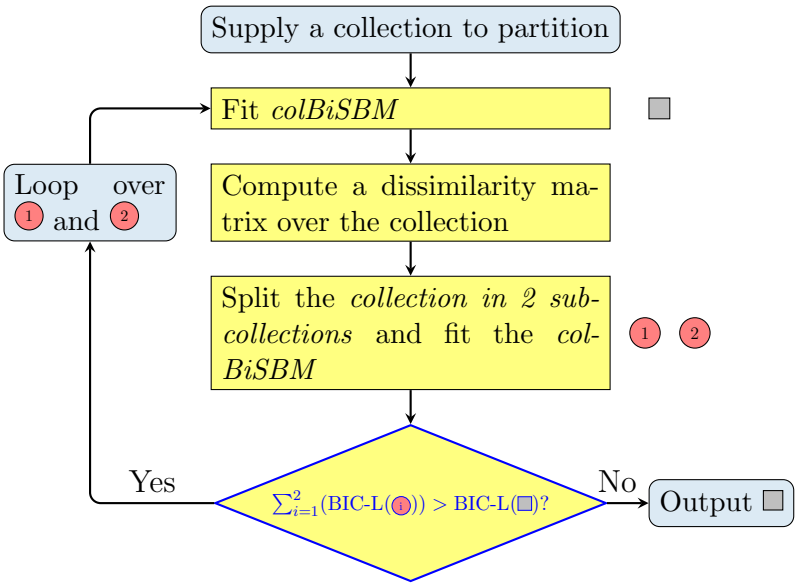


Figure 3.2: Network clustering procedure

The above figure (3.2) shows a condensed explanation of the network clustering algorithm.

The idea is to adjust the *colBiSBM* model over the full collection of M networks and then compute the dissimilarity matrix between all networks of the collection. We obtain the collection $\mathcal{G} = \{\mathcal{M}\}$ the trivial partition in a unique group.

Then using the *Kmeans* we split the collection in two sub-collections with the dissimilarity matrix. The two sub-collections are fitted and we compute the score of this new partition $\mathcal{G}^* = \{G_1, G_2\}$.

If $Sc(\mathcal{G}^*) > Sc(\mathcal{G})$ then we repeat the same procedure on G_1 and G_2. Else we return $\mathcal{G}$.

We illustrate our capacity to perform a partition of a collection for all col-BiSBM models in 4.3.

Chapter 4

Simulation studies

The below simulations are meant to test the capacities of our models. We assess the inference capacities of the algorithm and method, the model selection performances and the clustering capacities.

Reproducibility All the codes used to obtain data and to perform the analysis can be found on the report repository at https://gitea.polarolouis.fr/polarolouis/rapport-CEI-MIA-2023.

4.1 Efficiency of the inference

Simulation settings For this simulation the data is simulated with $M = 2, n_1^m = 120, n_2^m = 120, Q_1 = Q_2 = 4$, $\boldsymbol{\alpha}, \boldsymbol{\pi}$ and $\boldsymbol{\rho}$ are set as follows:

$$\boldsymbol{\alpha} = .25 + \begin{pmatrix} 3\epsilon_\alpha & 2\epsilon_\alpha & \epsilon_\alpha & -\epsilon_\alpha \\ 2\epsilon_\alpha & 2\epsilon_\alpha & -\epsilon_\alpha & \epsilon_\alpha \\ \epsilon_\alpha & -\epsilon_\alpha & \epsilon_\alpha & 2\epsilon_\alpha \\ -\epsilon_\alpha & \epsilon_\alpha & 2\epsilon_\alpha & 0 \end{pmatrix},$$

$$\boldsymbol{\pi}^1 = \sigma_1 \begin{pmatrix} 0.2 & 0.4 & 0.4 & 0 \end{pmatrix}, \qquad \boldsymbol{\pi}^2 = \begin{pmatrix} 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix},$$

$$\boldsymbol{\rho}^1 = \begin{pmatrix} 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}, \qquad \boldsymbol{\rho}^2 = \sigma_2 \begin{pmatrix} 0 & 0.33 & 0.33 & 0.33 \end{pmatrix},$$

with ϵ_α taking nine equally spaced values ranging from 0 to 0.24. For each value of ϵ_α, 108 datasets (X_1, X_2) are simulated, resulting in $9 \times 108 = 972$ datasets. More precisely, for each dataset, we pick uniformly at random two permutations of $\{1, \ldots, 4\}$ (σ_1, σ_2) with the constraint that $\sigma_1(4) \neq \sigma_2(1)$. This ensures that each of the two networks have a non-empty block that is empty in the other one. Then the networks are simulated with $\mathcal{B}\text{ern-}BiSBM_{120}(4, \boldsymbol{\alpha}, \boldsymbol{\pi}^m, \boldsymbol{\rho}^m)$ with the previous parameters. Each network has 2 blocks in common and their connectivity structures encompass a mix of core-periphery, assortative community and disassortative community structures, depending on which 3 of the 4 blocks are selected for each network. ϵ_α represents the strength of these structures, the larger, the easier it is to tell apart one block from another. The true model of all the simulation is a $\pi\rho$-$colBiSBM$.

Inference We want to measure the quality of the inference procedure, for this we use the inference described in the section 3.4.

Quality indicators To assess the quality of the inference, we will use the following indicators:

- First, for each dataset, we put in competition π-$colBiSBM$ with *sep*-$BiSBM$, *iid*-$colBiSBM$, ρ-$colBiSBM$, $\pi\rho$-$colBiSBM$ respectively. To do so, for each dataset, we compute the BIC-L of each model π-$colBiSBM$ is preferred to *sep*-$BiSBM$ (resp. *iid*-$colBiSBM$, ρ-$colBiSBM$, $\pi\rho$-$colBiSBM$) if its BIC-L is greater.
- When considering π-$colBiSBM$, ρ-$colBiSBM$, $\pi\rho$-$colBiSBM$ we compare $\widehat{Q_1}$, $\widehat{Q_2}$ to their true values. ($Q_1 = 4$ and $Q_2 = 4$)
- Finally, we assess the quality of the node grouping by computing the Adjusted Rand Index (Hubert and Arabie 1985, ARI = 0 for a random grouping, ARI = 1 for a perfect recovery). For each network, for the π-$colBiSBM$,

ρ-colBiSBM, *πρ-colBiSBM* we compare the inferred block memberships to the real ones by computing the mean of the ARI per axis over the two networks

$$\overline{\text{ARI}}_d = \frac{1}{2}\text{ARI}\big(\text{ARI}(\widehat{\boldsymbol{Z}_d^1}, \boldsymbol{Z}_d^1) + \text{ARI}(\widehat{\boldsymbol{Z}_d^2}, \boldsymbol{Z}_d^2)\big)$$

where d is the dimension or axis (i.e., rows, $d = 1$, or columns, $d = 2$) of the block memberships. And we compute the ARI of the whole set of nodes to account for block pairing between networks

$$\text{ARI}_d = \text{ARI}\big((\widehat{\boldsymbol{Z}_d^1}, \widehat{\boldsymbol{Z}_d^2}), (\boldsymbol{Z}_d^1, \boldsymbol{Z}_d^2)\big)$$

All these quality indicators are averaged over the 108 datasets. The results are provided in the tables 4.1 to 4.5. Each line corresponds to the 108 datasets for a given value of value of ϵ_α.

Table 4.1: Quality metrics for *sep-BiSBM*

ϵ_α	$\overline{\text{ARI}}_1$	$\overline{\text{ARI}}_2$	ARI_1	ARI_2
0.00	0	0	0	0
0.03	0	0	0	0
0.06	0.1 ± 0.01	0.08 ± 0.01	0.06 ± 0.01	0.05 ± 0.01
0.09	0.71 ± 0.02	0.7 ± 0.01	0.37 ± 0.02	0.37 ± 0.02
0.12	0.94 ± 0.01	0.93 ± 0.01	0.5 ± 0.02	0.49 ± 0.02
0.15	0.99	0.99	0.54 ± 0.02	0.49 ± 0.01
0.18	0.99	0.99	0.52 ± 0.02	0.52 ± 0.02
0.21	0.99	0.99	0.54 ± 0.02	0.52 ± 0.02
0.24	1	1	0.55 ± 0.02	0.52 ± 0.02

Table 4.2: Quality metrics for *iid-colBiSBM*

ϵ_α	$\overline{\text{ARI}}_1$	$\overline{\text{ARI}}_2$	ARI_1	ARI_2	Recovered Q_1	Recovered Q_2
0.00	0	0	0	0	1	1
0.03	0	0	0	0	1	1
0.06	0.08 ± 0.01	0.08 ± 0.01	0.08 ± 0.01	0.07 ± 0.01	1.4 ± 0.05	1.49 ± 0.05
0.09	0.72 ± 0.01	0.71 ± 0.01	0.53 ± 0.02	0.52 ± 0.02	3.4 ± 0.06	3.41 ± 0.06
0.12	0.94	0.93	0.75 ± 0.03	0.72 ± 0.03	4.06 ± 0.02	3.97 ± 0.02
0.15	0.98	0.98	0.77 ± 0.03	0.76 ± 0.03	4.11 ± 0.03	4.11 ± 0.03
0.18	0.99	0.99	0.82 ± 0.03	0.82 ± 0.03	4.15 ± 0.04	4.13 ± 0.03
0.21	0.99	0.99	0.8 ± 0.02	0.79 ± 0.03	4.35 ± 0.06	4.19 ± 0.04
0.24	0.99	0.99	0.77 ± 0.03	0.77 ± 0.03	4.3 ± 0.06	4.43 ± 0.07

Table 4.3: Quality metrics for π-*colBiSBM*

ϵ_α	$\overline{\text{ARI}}_1$	$\overline{\text{ARI}}_2$	ARI_1	ARI_2	Recovered Q_1	Recovered Q_2
0.00	0	0	0	0	1	1
0.03	0	0	0	0	1.01 ± 0.01	1
0.06	0.07 ± 0.01	0.08 ± 0.01	0.07 ± 0.01	0.06 ± 0.01	1.49 ± 0.05	1.5 ± 0.05
0.09	0.73 ± 0.02	0.72 ± 0.01	0.56 ± 0.02	0.53 ± 0.02	3.78 ± 0.07	3.37 ± 0.07
0.12	0.96	0.93	0.79 ± 0.02	0.74 ± 0.03	4.46 ± 0.07	3.95 ± 0.02
0.15	0.99	0.97	0.82 ± 0.02	0.76 ± 0.03	4.62 ± 0.08	4
0.18	1	0.98	0.83 ± 0.02	0.79 ± 0.03	4.65 ± 0.09	4
0.21	1	0.98	0.84 ± 0.02	0.79 ± 0.03	4.69 ± 0.1	4
0.24	1	0.99	0.86 ± 0.02	0.79 ± 0.03	4.74 ± 0.11	4.01 ± 0.01

Table 4.4: Quality metrics for ρ-*colBiSBM*

ϵ_α	$\overline{\text{ARI}}_1$	$\overline{\text{ARI}}_2$	ARI_1	ARI_2	Recovered Q_1	Recovered Q_2
0.00	0	0	0	0	1	1
0.03	0	0	0	0	1.01 ± 0.01	1.01 ± 0.01
0.06	0.08 ± 0.01	0.08 ± 0.01	0.06 ± 0.01	0.07 ± 0.01	1.39 ± 0.05	1.6 ± 0.06
0.09	0.72 ± 0.01	0.72 ± 0.01	0.53 ± 0.02	0.54 ± 0.02	3.39 ± 0.07	3.74 ± 0.07
0.12	0.93	0.95	0.71 ± 0.03	0.75 ± 0.02	3.95 ± 0.02	4.5 ± 0.07
0.15	0.97	0.99	0.78 ± 0.03	0.81 ± 0.02	4	4.49 ± 0.07
0.18	0.98	1	0.76 ± 0.03	0.81 ± 0.02	4.01 ± 0.01	4.71 ± 0.09
0.21	0.98	1	0.76 ± 0.03	0.81 ± 0.02	4.03 ± 0.02	4.72 ± 0.09
0.24	0.98	1	0.74 ± 0.03	0.8 ± 0.02	4.06 ± 0.02	4.8 ± 0.1

Table 4.5: Quality metrics for $\pi\rho$-*colBiSBM*

ϵ_α	$\overline{\text{ARI}}_1$	$\overline{\text{ARI}}_2$	ARI_1	ARI_2	Recovered Q_1	Recovered Q_2
0.00	0	0	0	0	1	1
0.03	0	0	0	0	1.01 ± 0.01	1.01 ± 0.01
0.06	0.07 ± 0.01	0.07 ± 0.01	0.07 ± 0.01	0.06 ± 0.01	1.48 ± 0.05	1.57 ± 0.06
0.09	0.74 ± 0.01	0.73 ± 0.01	0.56 ± 0.03	0.55 ± 0.02	3.69 ± 0.06	3.66 ± 0.06
0.12	0.96 ± 0.01	0.95 ± 0.01	0.73 ± 0.03	0.73 ± 0.03	4.31 ± 0.05	4.26 ± 0.05
0.15	0.99	0.99	0.79 ± 0.02	0.78 ± 0.03	4.31 ± 0.05	4.35 ± 0.05
0.18	1	1	0.83 ± 0.02	0.83 ± 0.02	4.31 ± 0.05	4.25 ± 0.04
0.21	1	1	0.77 ± 0.03	0.77 ± 0.03	4.42 ± 0.05	4.34 ± 0.05
0.24	1	1	0.82 ± 0.02	0.82 ± 0.02	4.25 ± 0.04	4.31 ± 0.05

Table 4.6: Proportions of models selected per ϵ_α (data for Figure 4.1)

ϵ_α	*sep-BiSBM*	*iid-colBiSBM*	π-*colBiSBM*	ρ-*colBiSBM*	$\pi\rho$-*colBiSBM*
0.00	1.00	0.00	0.00	0.00	0.00
0.03	0.95	0.04	0.01	0.00	0.00
0.06	0.39	0.33	0.06	0.15	0.06
0.09	0.07	0.01	0.12	0.25	0.55
0.12	0.00	0.08	0.06	0.13	0.72
0.15	0.00	0.12	0.08	0.08	0.71
0.18	0.00	0.11	0.04	0.06	0.79
0.21	0.00	0.19	0.04	0.07	0.69
0.24	0.00	0.09	0.06	0.08	0.77

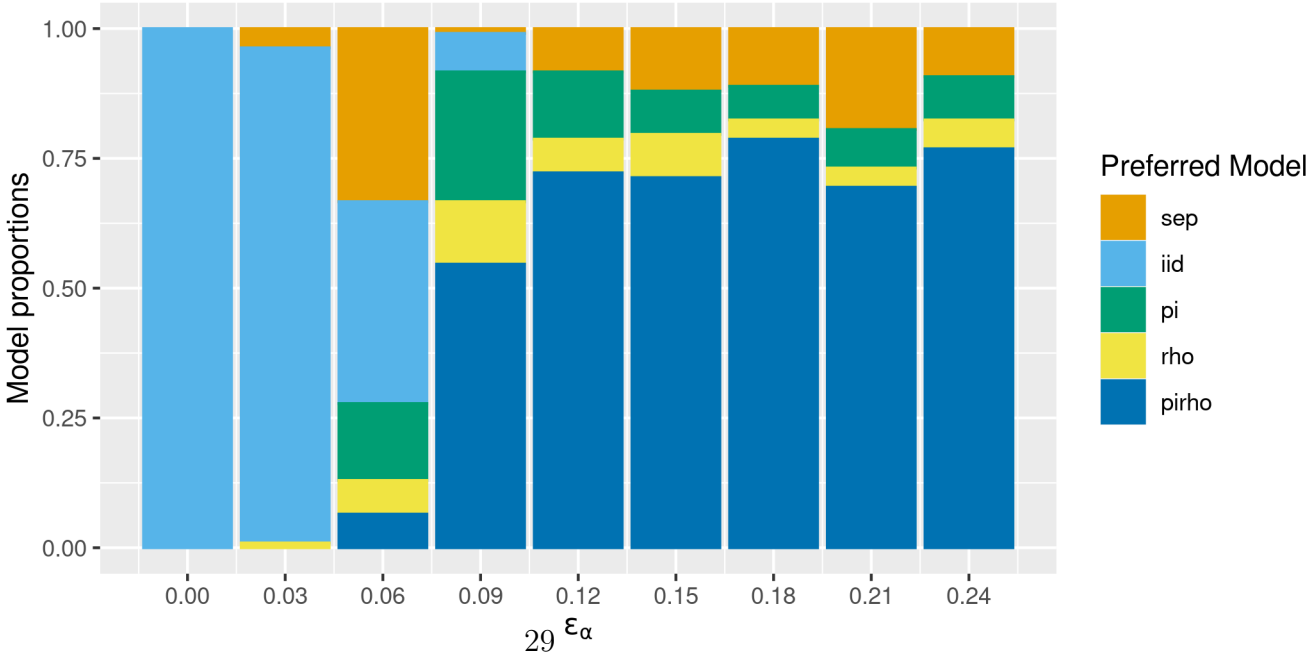


Figure 4.1: Plot of the proportions of different preferred models in function of ϵ_α

Results For the model comparison, when ϵ_α is small ($\epsilon_\alpha \in [0, .04]$), the simulation model is close to the Erdős-Reńyi network and it is very hard to find any structure beyond the one of a single block on each dimension.

On the figure 4.1 and table 4.6 we can see that from $\epsilon_\alpha = 0.12$ around 70% of the time the $\pi\rho$-*colBiSBM* model (i.e., the correct one) is selected.

An interesting result we can read in the tables is that our models outperform the *sep-BiSBM* when considering the ARI on the whole set of nodes (ARI_d). This means that our models are able to recover the block pairing *between the networks* in addition to recovering the blocks and their parameters.

4.2 Capacity to distinguish $\pi\rho$-$colBiSBM$ from iid-$colBiSBM$ and other variants

The idea of this model selection simulations is to assess how the model select the correct *colBiSBM* model among the possible ones: *iid, pi, rho, pirho*. This difference being based on the row and col block proportions.

For this task we choose the same simulation context as Chabert-Liddell et al. 2023, March 27.

Namely $n_1^m = 90, n_2^m = 90, Q_1 = Q_2 = 3$, $\boldsymbol{\alpha}, \boldsymbol{\pi}$ and $\boldsymbol{\rho}$ are set as follows:

$$\boldsymbol{\alpha} = .25 + \begin{pmatrix} 3\epsilon_\alpha & 2\epsilon_\alpha & \epsilon_\alpha \\ 2\epsilon_\alpha & 2\epsilon_\alpha & -\epsilon_\alpha \\ \epsilon_\alpha & -\epsilon_\alpha & \epsilon_\alpha \end{pmatrix}, \quad \boldsymbol{\pi}^1 = \left(\tfrac{1}{3}, \quad \tfrac{1}{3}, \quad \tfrac{1}{3}\right), \quad \boldsymbol{\pi}^2 = \sigma\left(\tfrac{1}{3} - \epsilon_\pi, \quad \tfrac{1}{3}, \quad \tfrac{1}{3} + \epsilon_\pi\right),$$

$$\boldsymbol{\rho}^1 = \left(\tfrac{1}{3}, \quad \tfrac{1}{3}, \quad \tfrac{1}{3}\right), \quad \boldsymbol{\rho}^2 = \sigma\left(\tfrac{1}{3} - \epsilon_\rho, \quad \tfrac{1}{3}, \quad \tfrac{1}{3} + \epsilon_\rho\right),$$

with $\epsilon_\alpha = 0.16$, ϵ_π and ϵ_ρ taking 9 values equally spaced in $[0, .28]$. We simulate 324 different collections for each value of ϵ_π and ϵ_ρ.

$\pi\rho$-$colBiSBM$, π-$colBiSBM$, ρ-$colBiSBM$, iid-$colBiSBM$ and *sep-BiSBM* are put in competition and the model with the greater BIC-L is selected as the *preferred model*.

When $\epsilon_\pi = 0$, $\boldsymbol{\pi}^1 = \boldsymbol{\pi}^2$, $\epsilon_\rho = 0$ and $\boldsymbol{\rho}^1 = \boldsymbol{\rho}^2$, the generated collection is an iid-$colBiSBM$. When $\epsilon_\pi > 0$ or $\boldsymbol{\pi}^1 \neq \boldsymbol{\pi}^2$, the model is a π-$colBiSBM$. When $\epsilon_\rho > 0$ or $\boldsymbol{\rho}^1 \neq \boldsymbol{\rho}^2$, the model is a ρ-$colBiSBM$. Finally, when $\epsilon_\pi > 0$ or $\boldsymbol{\pi}^1 \neq \boldsymbol{\pi}^2$ and $\epsilon_\rho > 0$ or $\boldsymbol{\rho}^1 \neq \boldsymbol{\rho}^2$, the model is a $\pi\rho$-$colBiSBM$.

Table 4.7: Model selection for varying π mixture parameters

	Models				Blocks
ϵ_π	iid-$colBiSBM$	π-$colBiSBM$	ρ-$colBiSBM$	$\pi\rho$-$colBiSBM$	Recovered Q_1
0.00	0.65	0.00	0.35	0.00	3
0.04	0.66	0.00	0.34	0.00	3
0.07	0.64	0.01	0.34	0.01	3.01 ± 0.01
0.11	0.63	0.03	0.31	0.03	3.01 ± 0.01
0.14	0.55	0.12	0.28	0.05	3
0.18	0.39	0.26	0.21	0.13	3.01
0.21	0.23	0.42	0.13	0.23	3.01
0.25	0.10	0.56	0.05	0.29	3.02 ± 0.01
0.28	0.01	0.65	0.01	0.33	3.01 ± 0.01

Table 4.8: Model selection for varying ρ mixture parameters

	Models				Blocks
ϵ_ρ	*iid-colBiSBM*	π-*colBiSBM*	ρ-*colBiSBM*	$\pi\rho$-*colBiSBM*	Recovered Q_2
0.00	0.63	0.37	0.00	0.00	3
0.04	0.65	0.34	0.00	0.01	3
0.07	0.64	0.33	0.01	0.01	3
0.11	0.64	0.31	0.03	0.02	3
0.14	0.53	0.29	0.11	0.06	3
0.18	0.42	0.20	0.24	0.14	3.01
0.21	0.25	0.12	0.40	0.22	3.01
0.25	0.08	0.06	0.58	0.29	3.01
0.28	0.01	0.01	0.65	0.32	3

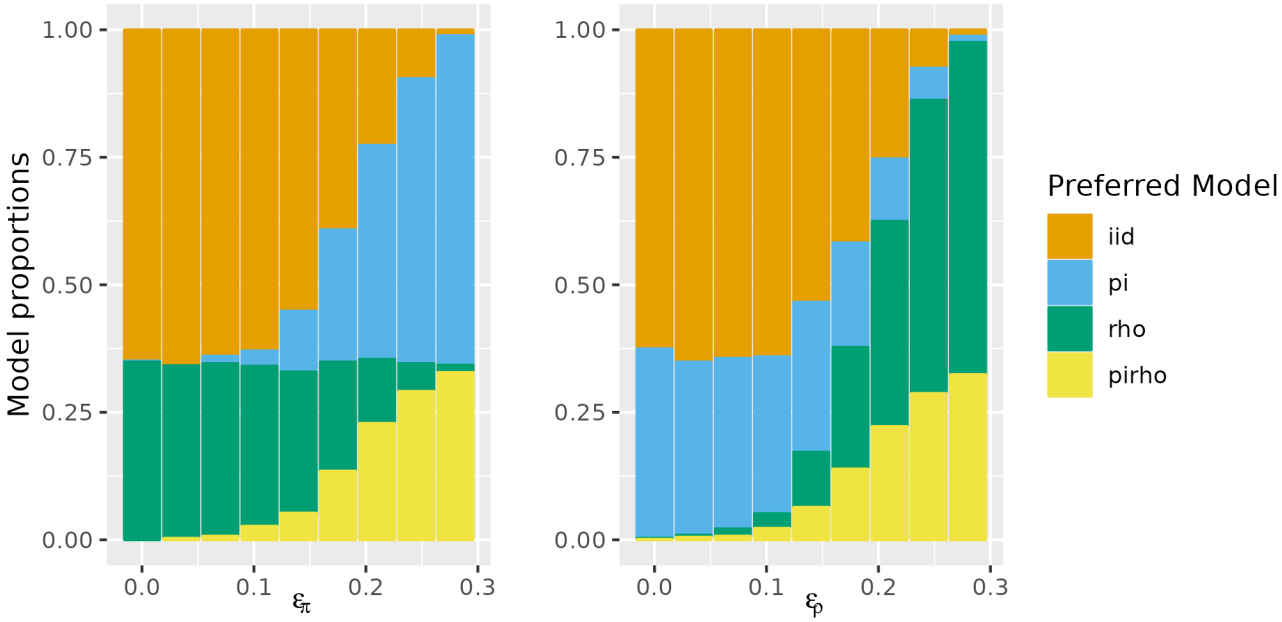


Figure 4.2: Plot of preferred model in function of ϵ_π and ϵ_ρ

Results: On the figure 4.2 and tables 4.7 and 4.8, one can see that there is a turning point around $\epsilon_\pi = 0.2$ (resp. $\epsilon_\rho = 0.2$), before which *iid-colBiSBM* and ρ-*colBiSBM* (resp. π-*colBiSBM*) are selected most of the times and after 0.2 the π-*colBiSBM* (resp. ρ-*colBiSBM*) and $\pi\rho$-*colBiSBM* gets more and more selected, highlighting our capacity to recover the simulated structure.

Remark: Please note that when "Recovered Q_1(or Q_2)" is not an integer it's because some procedures returned a value other than 3.

4.3 Network clustering of simulated networks

Simulation settings For all models we simulate $M = 9$ networks with $\forall m \in \{1 \dots M\}, n_1^m = n_2^m = 75$ with $Q_1 = Q_2 = 3$. For the simulations the proportions are the following:

$$\boldsymbol{\pi}^1 = (0.2, 0.3, 0.5) \qquad\qquad \boldsymbol{\rho}^1 = (0.2, 0.3, 0.5)$$

and for all $m = 2, \dots, 9$

$$\boldsymbol{\pi}^m = \begin{cases} \boldsymbol{\pi}^1 & \text{for } \textit{iid-colBiSBM} \\ \sigma_m^1(\boldsymbol{\pi}^1) & \text{for } \pi\textit{-colBiSBM} \text{ and } \pi\rho\textit{-colBiSBM} \end{cases}$$
$$\boldsymbol{\rho}^m = \begin{cases} \boldsymbol{\rho}^1 & \text{for } \textit{iid-colBiSBM} \\ \sigma_m^2(\boldsymbol{\rho}^1) & \text{for } \rho\textit{-colBiSBM} \text{ and } \pi\rho\textit{-colBiSBM} \end{cases}$$

where σ_m^1 and σ_m^2 are permutations of $\{1, 2, 3\}$ proper to network m and $\sigma^1(\pi) = (\pi_{\sigma^1(i)})_{i=\{1,\dots,3\}}$ and $\sigma^2(\rho) = (\rho_{\sigma^2(i)})_{i=\{1,\dots,3\}}$. The networks are divided into 3 sub-collections of 3 networks with connectivity parameters as follows:

$$\boldsymbol{\alpha}^{as} = .3 + \begin{pmatrix} \epsilon & -\frac{\epsilon}{2} & -\frac{\epsilon}{2} \\ -\frac{\epsilon}{2} & \epsilon & -\frac{\epsilon}{2} \\ -\frac{\epsilon}{2} & -\frac{\epsilon}{2} & \epsilon \end{pmatrix}, \quad \boldsymbol{\alpha}^{cp} = .3 + \begin{pmatrix} \frac{3\epsilon}{2} & \epsilon & \frac{\epsilon}{2} \\ \epsilon & \frac{\epsilon}{2} & 0 \\ \frac{\epsilon}{2} & 0 & -\frac{\epsilon}{2} \end{pmatrix}, \quad \boldsymbol{\alpha}^{dis} = .3 + \begin{pmatrix} -\frac{\epsilon}{2} & \epsilon & \epsilon \\ \epsilon & -\frac{\epsilon}{2} & \epsilon \\ \epsilon & \epsilon & -\frac{\epsilon}{2} \end{pmatrix},$$

with $\epsilon \in [.1, .4]$. $\boldsymbol{\alpha}^{as}$ represents a classical assortative community structure, while $\boldsymbol{\alpha}^{cp}$ is a layered core-periphery structure with block 2 acting as a semi-core. Finally, $\boldsymbol{\alpha}^{dis}$ is a disassortative community structure with stronger connections between blocks than within blocks. If $\epsilon = 0$, the three matrices are equal and the 9 networks have the same connection structure. Increasing ϵ differentiates the 3 sub-collections of networks.

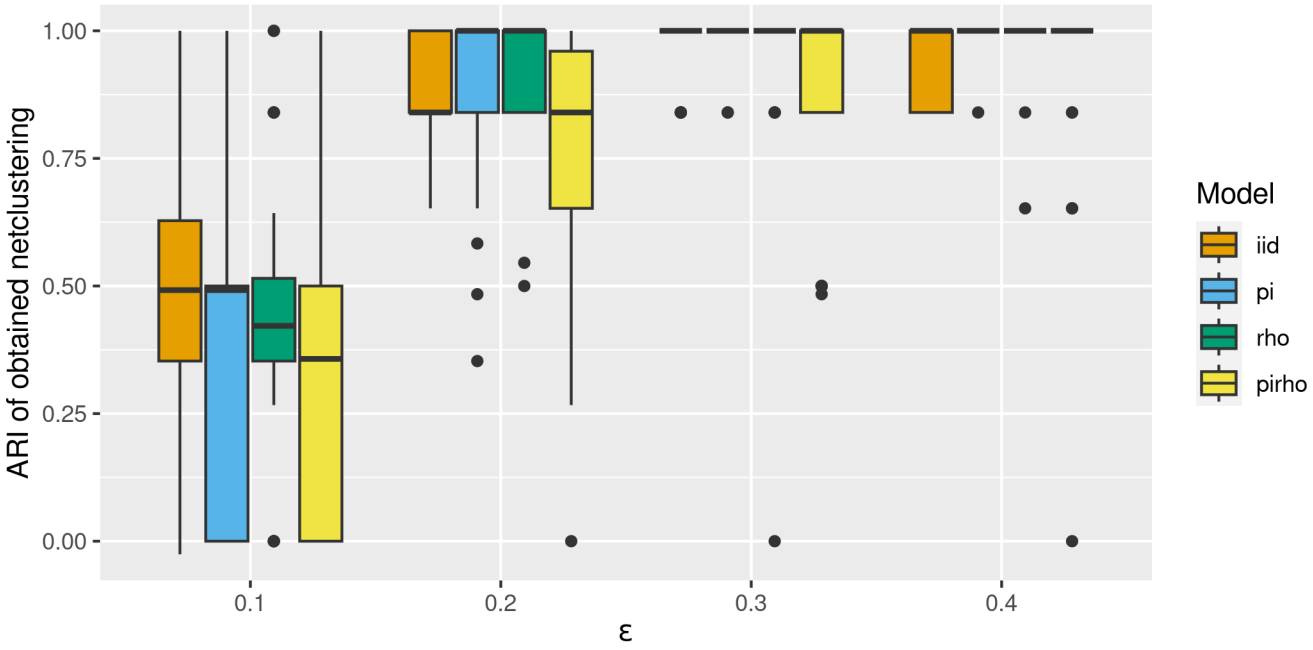


Figure 4.3: ARI of the partition obtained by clustering in function of ϵ

Results The evaluation of our method involves a comparison between the resulting partition of the network collection and the simulated partition using the ARI index. As the value of ϵ increases, our ability to distinguish between the networks improves, and this distinction becomes nearly perfect in all setups of the $colBiSBM$.

Chapter 5

Applications

5.1 Application to Doré et al. 2021 data

5.1.1 Completing raw data using CoOPLBM (Anakok et al. 2022, November 29)

Context of this analysis

After performing a netclustering on the raw data, we will see if the detect structure resulting in the clustering comes from the sampling effort. To test this we will use the CoOPLBM model by Anakok et al. 2022, November 29 to complete the data.

The CoOPLBM model assumes that the observed incidence matrix R is an element-wise product of an M matrix following an LBM and an N matrix which elements follow Poisson distributions independent on M.

The model gives us the $\widehat{M}$ matrix, the elements of which are:

$$\widehat{M_{ij}} = \mathbb{P}(M_{ij} = 1)$$

Note that if $R_{ij} = 1$ then $\widehat{M_{ij}} = 1$

- 1 if the interaction was observed
- a probability, that there should be an interaction but it wasn't observed

This *completed matrix* can be used in different manners to be fed to the colSBM model.

Threshold based completions

With the thresholds, the infered incidence matrix obtained by CoOPLBM is used to generate a completed incidence matrix by the following procedure :

$$X_{ij} = \begin{cases} 1 & \text{if the value is over the threshold} \\ 0 & \text{else} \end{cases}$$

0.5 completed threshold Here, the completion threshold is set to 0.5.

First we will compute an ARI on the collection id given by the raw data and the completed matrix.

	ARI with uncompleted data
iid	0.1142823
pi	0.0263660
rho	0.0933340
pirho	0.2158747

In the above table, one can see the network clustering obtained after applying CoOPLBM has not much in common with the clustering of the uncompleted data.

Number of sub-collections and details of each sub-collection

0.2 completed threshold

The 0.2 threshold adds a lot of interactions compared to raw matrix.

	ARI with uncompleted data
iid	0.0429465
pi	0.0330057
rho	0.0187305
pirho	0.0357728

Same as for 0.5, after applying CoOPLBM the obtained clustering doesn't match the uncompleted data.

Sample based completions

The M matrix is used to sample a new X matrix which elements are the realisation of Bernoulli distributions of probability $M_{i,j}$.

$$\mathbb{P}(X_{i,j} = 1) = M_{i,j}$$

	ARI with uncompleted data
iid	0.0148172
pi	0.0265793
rho	0.0051536
pirho	0.0152299

Bibliography

Accueil | MIA Paris-Saclay. (n.d.). Retrieved July 3, 2023, from https://mia-ps.inrae.fr/

Anakok, E., Barbillon, P., Fontaine, C., & Thebault, E. (2022, November 29). *Disentangling the structure of ecological bipartite networks from observation processes*. arXiv: 2211.16364 [stat]. Retrieved June 14, 2023, from http://arxiv.org/abs/2211.16364

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. https://doi.org/10.1109/34.865189

Chabert-Liddell, S.-C., Barbillon, P., & Donnet, S. (2023, March 27). *Learning common structures in a collection of networks. An application to food webs*. arXiv: 2206.00560 [stat]. https://doi.org/10.48550/arXiv.2206.00560

Daudin, J.-J., Picard, F., & Robin, S. (2008). A mixture model for random graphs. *Statistics and Computing*, *18*(2), 173–183. https://doi.org/10.1007/s11222-007-9046-7

Desjardins-Proulx, P., Laigle, I., Poisot, T., & Gravel, D. (2017). Ecological interactions and the Netflix problem. *PeerJ*, *5*, e3644. https://doi.org/10.7717/peerj.3644

Doré, M., Fontaine, C., & Thébault, E. (2021). Relative effects of anthropogenic pressures, climate, and sampling design on the structure of pollination networks at the global scale. *Global Change Biology*, *27*(6), 1266–1280. https://doi.org/10.1111/gcb.15474

Govaert, G. [G.], & Nadif, M. [M.]. (2005). An EM algorithm for the block mixture model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(4), 643–647. https://doi.org/10.1109/TPAMI.2005.69

Govaert, G. [Gérard], & Nadif, M. [Mohamed]. (2010). Latent Block Model for Contingency Table. *Communications in Statistics - Theory and Methods*, *39*(3), 416–425. https://doi.org/10.1080/03610920903140197

Holland, P. W., Laskey, K. B., & Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social Networks*, *5*(2), 109–137. https://doi.org/10.1016/0378-8733(83)90021-7

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification*, *2*(1), 193–218. https://doi.org/10.1007/BF01908075

Kaszewska-Gilas, K., Kosicki, J. Z., Hromada, M., & Skoracki, M. (2021). Global Studies of the Host-Parasite Relationships between Ectoparasitic Mites of the Family Syringophilidae and Birds of the Order Columbiformes. *Animals*, *11*(12), 3392. https://doi.org/10.3390/ani11123392

Pavlopoulos, G. A., Kontou, P. I., Pavlopoulou, A., Bouyioukos, C., Markou, E., & Bagos, P. G. (2018). Bipartite graphs in systems biology and medicine: A survey of methods and applications. *GigaScience*, *7*(4), giy014. https://doi.org/10.1093/gigascience/giy014

Ramos-Jiliberto, R., Domínguez, D., Espinoza, C., López, G., Valdovinos, F. S., Bustamante, R. O., & Medel, R. (2010). Topological change of Andean plant–pollinator networks along an altitudinal gradient. *Ecological Complexity*, *7*(1), 86–90. https://doi.org/10.1016/j.ecocom.2009.06.001

Snijders, T. A., & Nowicki, K. (1997). Estimation and Prediction for Stochastic Blockmodels for Graphs with Latent Block Structure. *Journal of Classification*, *14*(1), 75–100. https://doi.org/10.1007/s003579900004

List of Figures

List of Tables