Received: 31 July 2020 | Accepted: 27 April 2021

DOI: 10.1111/rssc.12509

ORIGINAL ARTICLE

Accounting for missing actors in interaction network inference from abundance data

Raphaëlle Momal[1] | **Stéphane Robin**[1,2] | **Christophe Ambroise**[3]

[1]UMR MIA-Paris, AgroParisTech, INRAE, Université Paris-Saclay, Paris, France

[2]CESCO, Muséum National d'Histoire Naturelle, CNRS, Sorbonne Université, Paris, France

[3]Université Paris-Saclay, CNRS, Univ. Évry, Laboratoire de Mathématiques et Modélisation d'Évry 91037, Évry, France

Correspondence
UMR MIA-Paris, AgroParisTech, INRAE, Université Paris-Saclay, Paris, France.
Email: raphaelle.momal@inrae.fr

Abstract

Network inference aims at unravelling the dependency structure relating jointly observed variables. Graphical models provide a general framework to distinguish between marginal and conditional dependency. Unobserved variables (*missing actors*) may induce apparent conditional dependencies. In the context of count data, we introduce a mixture of Poisson log-normal distributions with tree-shaped graphical models, to recover the dependency structure, including missing actors. We design a variational EM algorithm and assess its performance on synthetic data. We demonstrate the ability of our approach to recover environmental drivers on two ecological data sets. The corresponding R package is available from github.com/Rmomal/nestor.

KEYWORDS

abundance data, graphical models, matrix tree theorem, missing actor, network inference, Poisson log-normal model, Variational EM algorithm

1 | INTRODUCTION

Network inference. Network inference (or structure inference) has become a topical problem in various fields such as biology, ecology, neurosciences, social sciences, to name a few. The aim is to unravel the dependency structure that relates a series of variables that can be jointly observed. Graphical models (see, e.g. Lauritzen, 1996) provide a natural framework to achieve this task as it allows to encode the dependency structure into a graph, the nodes of which are the variables. Two variables are connected if and only if they are dependant, conditionally on all others.

Most methodologies build on the assumption that the network is sparse, meaning that only a small fraction of variable pairs are conditionally dependent. The case of Gaussian graphical models (GGM) is especially appealing as the network corresponds to the support of the precision matrix of the joint Gaussian distribution. The use of a sparsity-inducing penalisation gives raise to the celebrated graphical lasso (Friedman et al., 2008). In a more general context, Chow and Liu (1968) consider a spanning tree structure to impose sparsity to the network, but this drastic form can be alleviated using mixtures of trees (Kirshner, 2008; Meilă & Jaakkola, 2006).

One important aspect of network inference is to distinguish between variables that are marginally dependent (possibly because of their respective dependency with some common other) from variables that are *directly related*, that is conditionally dependant. This distinction requires to account for as many confounding effects as possible, which includes not only all the other variables but also available covariates. It also requires to consider the existence of some *missing actors* (or missing nodes), that may induce an apparent direct dependency.

Abundance data. Count data is found in a multitude of fields (sociology, biology, economy, ecology, etc.). It results from the counting of events in a given setting such as crime statistics in a state or the number of produced transcripts of a gene in an experiment. The statistical processing of count data cannot always rely on classical methods developed for continuous Gaussian data and appeals for specific methods. It often exhibits specificities such as zero-inflation and a large dispersion. This work is motivated by the analysis of so-called abundance data, a count data avatar, arising from ecological studies where the number of individuals (the abundance) of a series of living species (plants or animals) is observed in a series of sites. In this context, network inference aims at understanding which pairs of species are in direct interaction. The covariates are typically environmental descriptors (altitude, temperature, distance to the see, etc.) of each collection site, while the variables are the respective abundances of each species from the community under study.

Graphical modelling for counts is not as well-developed as GGM. In his seminal paper Besag (1974) introduces generic Markov Random Field models (MRF) for lattice systems. He assumes that the graph of conditional independence is known, whereas this paper aims at inferring the graph. The likelihood-based inference is often problematic due to intractable normalising constants. Pseudo-likelihood or surrogate likelihood approaches such as neighbourhood selection (Yang et al., 2013) represent a tractable solution implemented in the `xMRF` R package (Wan et al., 2016). Inouye et al. (2016) proposes also complex models with associated parameter estimation methods using node-wise regressions with ℓ_1 regularisation and likelihood approximation methods using sampling. Bayesian computation algorithms based on Markov chain Monte Carlo (MCMC) can also be used for inferring the conditional independence graph (Roy & Dunson, 2020). Models relying on copulas (Inouye et al., 2017) have been proposed but many joint species distribution models resort to a latent Gaussian layer, which encodes the dependency structure between the species (Popovic et al., 2018, 2019; Warton et al., 2015). The Poisson log-normal model (PLN: Aitchison & Ho, 1989) enters this category: it assumes that a multivariate Gaussian random variable is associated to each species in each site and that the observed abundances are conditionally independent Poisson variables. The PLN model has already been applied to abundance data, both for dimension reduction (Chiquet et al., 2018) and network inference (Chiquet et al., 2019; Momal et al., 2020) PLN model is not directly modelling the conditional independence graph between observed variables. Nevertheless there is a strong relation between the dependence structure in the latent and observed layers. The PLN model may thus be considered as a graphical state-space model.

Missing actors. In many situations, it is likely that not all actors involved in the system have been observed. The term 'actors' refers to either species that were not observed but nonetheless influence the abundance of others, or environmental conditions that were not accounted for. Missing actors may be quantitative or qualitative. In the latter case, it defines a latent group structure (Ambroise et al., 2009).

In the perspective of unravelling the conditional independence structure, this can typically lead to the inference of spurious edges, which are links between observed actors that are not in direct interaction. In the graphical model framework, not accounting for one variable amounts to consider the marginal distribution of the rest of the system, as described in the right panel of Figure 1.

Figure 2 further illustrates the interest of accounting for covariates and missing actors when aiming at inferring a graphical model. The illustration is based on synthetic data including one covariate and one missing actor. The 'blind' inference (b), including neither the covariate nor the missing actor, yields irrelevant results. Accounting for the covariate provides an estimate of the graph marginalised with respect to the missing actor (c), that is the network resembles the original graph with additional links between the neighbours of the missing actor. Including a missing actor retrieves the original dependency structure (d).

Several approaches have been proposed for network inference accounting for quantitative missing actors in the context of GGM. Many of them (Chandrasekaran et al., 2011; Giraud & Tsybakov, 2012; Lauritzen & Meinshausen, 2012; Meng et al., 2014) adapted the principle of Robust PCA (Candès et al., 2011) to the concentration matrix, assuming it is a sum of two matrices: one low-rank and one sparse. In terms of missing actors in a network, the low-rank part corresponds to missing actors connected to all variables, whereas the sparse part refers to missing actors having a local effect. Following Robin et al. (2019) (also in the context of GGM), we focus on the later aspect, that is looking for missing actors not necessarily linked to all others. As far as we know, no model has been proposed for the inference of missing actors from abundance data.

Variational inference. The model we consider in this paper involves different types of variables, namely an unknown tree-shaped graphical model, a continuous latent layer (to induce dependence between the species) and unobserved actors. The most popular approach for the inference of such models is the EM algorithm (Dempster et al., 1977), which requires the evaluation of the conditional distribution of all unobserved variables given the data. In the problem we consider, some latent variables are (multivariate) continuous and others are discrete, and their joint conditional distribution turns out to be intractable. In this work, we resort to a variational approximation (Wainwright & Jordan, 2008) of this conditional distribution and to a variational EM algorithm for its inference (see, e.g. Blei et al., 2017).

Our contribution. In the context of the Poisson log-normal model, we propose a tree-based approach to recover the structure of latent graphical model including actors. The model we consider involves several layers of unobserved variables with intractable conditional distributions, thus we resort to a variational EM algorithm (Blei et al., 2017) for its inference. We introduce the model in Section 2 and describe its variational inference in Section 3. The performance of the algorithm is assessed via simulations in Section 4. The use of the proposed model is illustrated in Section 5, where we demonstrate its ability to recover environmental drivers on two ecological data sets. The inference procedure is implemented in the R package nestor, available at github.com/Rmomal/nestor.

FIGURE 1 Example of the marginalisation when covariate x is unobserved. *Left*: complete graphical model (including x). *Right*: marginal graphical model of the observed variables (excluding x)

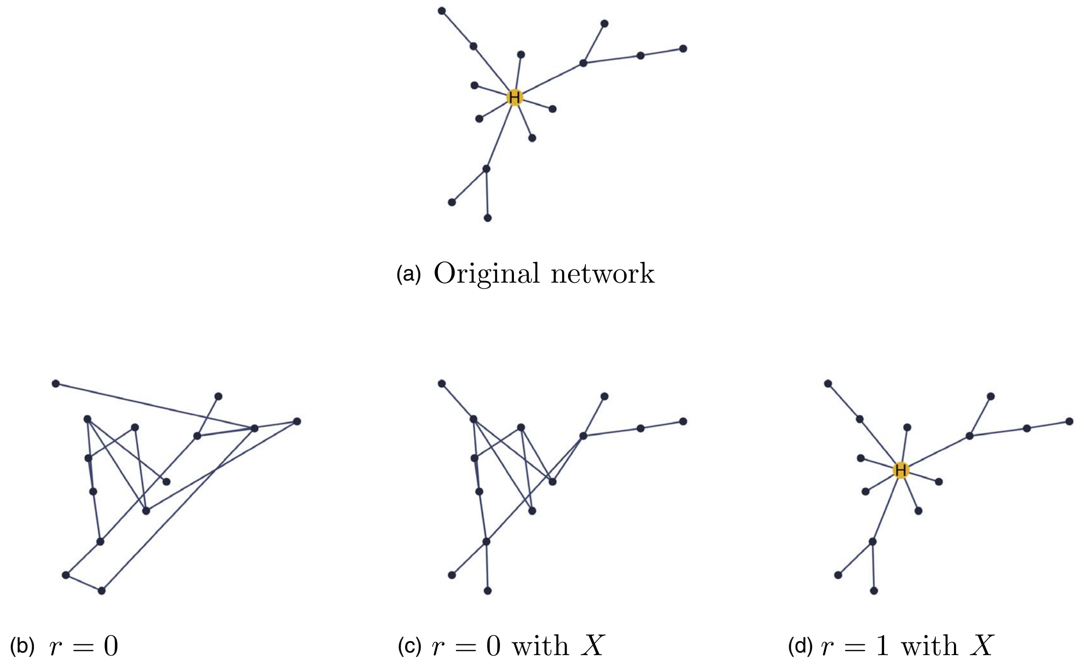

FIGURE 2 Example of network inference from counts simulated under the PLN model with $p = 14$ observed species, $r = 1$ hidden species, $n = 200$ samples, a Gaussian covariate $X \sim \mathcal{N}(1, , 1)$ and the original dependency structure (a). (b): network obtained under the null model and without any missing actors assumed. (c): network inferred while taking X into account and without missing actors assumed. (d): network inferred while accounting for both X and one missing actor, coloured in yellow [Colour figure can be viewed at wileyonlinelibrary.com]

2 | MODEL

2.1 | Poisson log-normal and tree-shaped graphical models

2.1.1 | Poisson log-normal model

We start with a reminder on the multivariate Poisson log-normal model, with the example of abundance data. The abundances of p species observed on n sites are gathered in the $n \times p$ matrix $\boldsymbol{Y}$ where Y_{ij} is the count of species j in site i, and the row i of $\boldsymbol{Y}$, denoted $\boldsymbol{Y}_i$, is the abundance vector collected on site i. A covariate vector $\boldsymbol{x}_i$ with dimension d is also measured on each site i and all covariates are gathered in the $n \times d$ matrix $\boldsymbol{X}$. The PLN model states that a (latent) Gaussian vector $\boldsymbol{U}_i$ of size p with variance matrix $\boldsymbol{R} = (\rho_{kl})_{kl}$ is associated to each site:

$$\{\mathbf{U}_i\}_{1 \le i \le n} \text{ iid}, \quad \mathbf{U}_i \sim \mathcal{N}_p(\mathbf{0}, \boldsymbol{R}), \tag{1}$$

the sites being assumed to be independent. To ensure identifiability, we let the diagonal of $\boldsymbol{R}$ be made of 1's, so $\boldsymbol{R}$ is actually a correlation matrix. All latent vectors $\boldsymbol{U}_i$ are gathered in the $n \times p$ matrix $\boldsymbol{U}$. The PLN model further assumes that species abundances in all sites are conditionally independent, and that their respective distribution only depends on the environment and the associated latent variable:

$$\{Y_{ij}\}_{1 \le i \le n,\, 1 \le j \le p} \,|\, \boldsymbol{U} \text{ independent}, \quad Y_{ij} \,|\, U_{ij} \sim \mathcal{P}\left(\exp(o_{ij} + \boldsymbol{x}_i^{\intercal} \boldsymbol{\theta}_j + \sigma_j U_{ij})\right), \tag{2}$$

where σ_j is the latent standard deviation associated with species j, and the vector $d \times 1$ of regression coefficients $\boldsymbol{\theta}_j$ describes the environmental effects on species j. The PLN model allows the specification of an offset term, denoted o_{ij} in Equation (2). In the regression literature, the offset is described as a given component in the estimation problem (Hardin & Hilbe, 2007) which can be used to account for exposure or sampling effort in count data models. The offset term can be improved with additional computation to correct for some observational bias. In genomics, effective library size are computed as offsets, in an effort to eliminate the compositional bias of RNA-seq data (Lun et al., 2016; Robinson & Oshlack, 2010). The offset can also be corrected species-wise, for example to account for species detectability in ecology (Guillera-Arroita, 2017).

An important feature of the PLN model is that the sign of the correlation between the observed counts is the same as this of correlation between the latent variables (Aitchison & Ho, 1989): $\text{sign}(\mathbb{C}\text{or}(Y_{ij}, Y_{ik})) = \text{sign}(\mathbb{C}\text{or}(U_{ij}, U_{ik}))$. To this respect the PLN model enables to capture the residual (i.e. once corrected for environmental effects) correlation structure between the species abundances through the corresponding latent variable.

2.1.2 | Tree-shaped graphical models

Network inference relies on the assumption that few species are directly dependent on one another, meaning that the underlying graphical model is sparse. In the framework of the PLN model, the graphical model of interest rules the distribution of the latent vectors $\boldsymbol{U}_i$ and is encoded in the precision matrix $\boldsymbol{\Omega} := \boldsymbol{R}^{-1}$. A way to foster sparsity is to impose $\boldsymbol{\Omega}$ to be faithful to a spanning tree T, that is: $\boldsymbol{U}_i \sim \mathcal{N}_p(\mathbf{0}, \boldsymbol{\Omega}_T^{-1})$ where the non-zero terms of $\boldsymbol{\Omega}_T$ correspond to the edges of the tree T. However, this hypothesis is very restrictive as it allows only $p-1$ links among p species (Chow & Liu, 1968). A more flexible approach consists in assuming that the latent vectors are drawn from a mixture of Gaussian distributions, each faithful to a tree T (Kirshner, 2008; Meilă & Jaakkola, 2006; Meilă & Jordan, 2000; Schwaller et al., 2019):

$$\boldsymbol{U}_i \sim \sum_{T \in \mathcal{T}_p} p(T) \mathcal{N}_p(\mathbf{0}, \boldsymbol{\Omega}_T^{-1}), \tag{3}$$

where $\mathcal{T}_p$ is the set of all spanning trees with p nodes. As a consequence, the mixture model involves $|\mathcal{T}_p| = p^{p-2}$ components. We further assume that the tree distribution $\{p(T)\}_{T \in \mathcal{T}_p}$ can be written as a product over the edges:

$$p(T) = B^{-1} \prod_{(j,k) \in T} \beta_{jk}, \quad \text{with} \quad B = \sum_{T \in \mathcal{T}_p} \prod_{(j,k) \in T} \beta_{jk}. \tag{4}$$

The weights β_{jk} are gathered in the $p \times p$ symmetric matrix $\boldsymbol{\beta}$ with diagonal zero. These weights are defined up to a multiplicative constant, so only $p(p-1)/2-1$ of them may vary independently. This PLN model with latent tree-shaped dependency structure is similar to that considered by Momal et al. (2020). Note that the weight β_{jk} associated with the edge (j,k) is not the same as the probability for this edge to be part of the random tree T, which is $\Pr\{(j,k) \in T\} = \sum_{T \ni (j,k)} p(T)$. (see Figure 3 of Momal et al., 2020, for an illustration). The marginal edge probability $\Pr\{(j,k) \in T\}$ obviously increases with β_{jk}, but the relation between the two is not straightforward because of the spanning constraint.

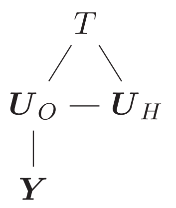

FIGURE 3 Graphical model linking the count data $\boldsymbol{Y}$, the latent layer of Gaussian parameters $\boldsymbol{U} = (\boldsymbol{U}_O, \boldsymbol{U}_H)$, and the latent tree T

The proposed model actually deals with the graphical model in the latent layer, which does not necessarily coincide with the graphical model of the observed layer (see, e.g. Figure 1 in the ArXiv report by Chiquet et al., 2018). This limitation is common to all models for multivariate count data that rely on a latent layer.

2.2 | Introducing the missing actor

2.2.1 | PLN model with missing actors

We now introduce the concept of missing actors, which corresponds to variables that are involved in the graphical model but are not associated to observed variables. To involve such actors in the model, we assume that a complete latent vector $\boldsymbol{U}_i$ with dimension $p + r$ is associated to site i, where r is the number of missing actors. This complete vector can be decomposed as $\boldsymbol{U}_i^\intercal = [\boldsymbol{U}_{Oi}^\intercal \ \boldsymbol{U}_{Hi}^\intercal]$ where $\boldsymbol{U}_{Oi}$ (with dimension p) corresponds to observed species and $\boldsymbol{U}_{Hi}$ (with dimension r) corresponds to the missing actors. The complete $n \times (p + r)$ latent matrix $\boldsymbol{U}$ can be decomposed in the same way as $\boldsymbol{U} = [\boldsymbol{U}_O \ \boldsymbol{U}_H]$, $\boldsymbol{U}_O$ and $\boldsymbol{U}_H$ having dimension $n \times p$ and $n \times r$, respectively. The model we consider states that

1. The complete latent vectors $\boldsymbol{U}_i$ are all iid and distributed according to a mixture similar to (3) and (4) but with Gaussian distributions (and matrices $\boldsymbol{\Omega}_T$ and $\boldsymbol{\beta}$) of dimension $(p + r)$, and trees drawn from $\mathcal{T}_{p+r}$;
2. the abundances Y_{ij} of the p observed species are distributed according to (2), replacing $\boldsymbol{U}$ with $\boldsymbol{U}_O$,

In the sequel, we shall refer to the elements of $\boldsymbol{U}_O$ and $\boldsymbol{U}_H$, respectively, as ‘observed’ and ‘hidden’ (or ‘missing’) latent variables, whereas obviously none of them are actually observed. Figure 3 displays the graphical model of the quadruplet $(T, \boldsymbol{U}_O, \boldsymbol{U}_H, \boldsymbol{Y})$. The observed data $\boldsymbol{Y}$ still arise from an PLN model, but the graphical model of the observed latent $\boldsymbol{U}_O$ may not be sparse due to the marginalisation over the hidden latent $\boldsymbol{U}_H$. Our main goal is to infer the dependency structure of the complete latent vectors, that is to estimate the elements of the matrices $\boldsymbol{\Omega}_T$ and the edges weights $\boldsymbol{\beta}$. The latent dependency structure is similar to this considered by Robin et al. (2019), but the inference strategy much differs, because of the additional hidden layer.

2.2.2 | Identifiability restriction

The proposed model only makes sense because the graphical model of the complete latent vectors $\boldsymbol{U}_i^\intercal = [\boldsymbol{U}_{Oi}^\intercal \ \boldsymbol{U}_{Hi}^\intercal]$ is supposed to be sparse. Missing actors could obviously not be identified from a regular PLN model, without restriction on the precision matrix $\boldsymbol{\Omega}$, as only the marginal precision matrix of the $\boldsymbol{U}_{Oi}$ could be recovered. Still, to ensure identifiability, we impose the same restriction as Robin et al. (2019) that missing latent variables are not connected with each other (the block corresponding to $\boldsymbol{U}_H \times \boldsymbol{U}_H$ is diagonal in each $\boldsymbol{\Omega}_T$).

3 | INFERENCE

As said in the introduction, we resort to a variational EM algorithm to perform the inference due to the complex latent structure.

3.1 | Variational inference

The log-likelihood of the so-called *complete* data, that is ($\boldsymbol{Y}$, $\boldsymbol{U}$, T), writes

$$\begin{aligned}\log p_{\theta,\beta,\boldsymbol{\Omega}}(\boldsymbol{Y},\boldsymbol{U},T) &= \log p_{\beta}(T)+\log p_{\boldsymbol{\Omega}}(\boldsymbol{U}|T)+\log p_{\theta}(\boldsymbol{Y}|\boldsymbol{U}) \\ &= \log p_{\beta}(T)+\log p_{\boldsymbol{\Omega}}(\boldsymbol{U}|T)+\log p_{\theta}(\boldsymbol{Y}|\boldsymbol{U}_O)\end{aligned}$$

where $\boldsymbol{\Omega}$ stands for the set of all tree-specific precision matrices: $\boldsymbol{\Omega} = \{\boldsymbol{\Omega}_T, T \in \mathcal{T}_{p+r}\}$ and where the second equality is a consequence of the graphical model given in Figure 3. The conditional distributions of the latent variables $\boldsymbol{U}$ and of the tree T given the data $\boldsymbol{Y}$ are both intractable. Variational inference then aims at maximising a lower bound of the log-likelihood of the observed data, which writes in our context as

$$\begin{aligned}\mathcal{J}(\boldsymbol{\theta},\boldsymbol{\beta},\Omega;q) &= \log p_{\theta,\beta,\boldsymbol{\Omega}}(\boldsymbol{Y}) - KL\left(q(\boldsymbol{U},T)\|p_{\theta,\beta,\boldsymbol{\Omega}}(\boldsymbol{U},T|\boldsymbol{Y})\right) \\ &= \mathbb{E}_q \log p_{\theta,\beta,\boldsymbol{\Omega}}(\boldsymbol{Y},\boldsymbol{U},T)+\mathcal{H}(q(\boldsymbol{U},T)),\end{aligned} \tag{5}$$

where $q(\boldsymbol{U}, T)$ stands for the approximate joint conditional distribution of the latent layer and of the tree: $q(\boldsymbol{U}, T) \simeq p(\boldsymbol{U}, T|\boldsymbol{Y})$.

3.1.1 | Approximate distribution

The efficiency of variational inference mostly depends on the choice of $q(\boldsymbol{U}, T)$, which is a balance between computational ease and adequation to the target distribution $p(\boldsymbol{U}, T|\boldsymbol{Y})$. We adopt here a classical product form for the approximate distribution: we impose to the latent variables $\boldsymbol{U}$ and to the tree T to be independent according to q (whereas actually they are not conditional on the data), with respective marginals h and g:

$$q(\boldsymbol{U}, T) = h(\boldsymbol{U})g(T).$$

Because the sites are independent, and without further assumption, the distribution h is a product over all sites. Following Chiquet et al. (2018), we approximate the conditional distribution of each latent vector $\boldsymbol{U}_i$ with a Gaussian distribution, that is:

$$h(\boldsymbol{U}) = \prod_i \mathcal{N}_{p+r}(\boldsymbol{U}_i;\boldsymbol{m}_i,\boldsymbol{S}_i)$$

with all $\boldsymbol{S}_i$ diagonal. We gather all the mean vectors $\boldsymbol{m}_i$ in the $n \times (p + r)$ matrix $\boldsymbol{M}$ and pile up the diagonals of all the variance matrices $\boldsymbol{S}_i$ in the $n \times (p + r)$ matrix denoted $\boldsymbol{S}$.

3.1.2 | Variational EM

The variational EM algorithm then consists in maximising the lower bound $\mathcal{J}$ defined in Equation (5) with respect to the parameters (M step), and to the approximate distributions (VE step), alternatively.

M step: At iteration $t + 1$, given the current approximate distribution $q^t(\boldsymbol{U}, T) = g^t(T)h^t(\boldsymbol{U})$, the M step consists in the update of the model parameters, solving

$$\begin{aligned} &\boldsymbol{\theta}^{t+1} = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{h^t}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{Y}|\boldsymbol{U}_O)\right], \qquad \boldsymbol{\Omega}^{t+1} = \arg\max_{\boldsymbol{\Omega}} \mathbb{E}_{q^t}\left[\log p_{\boldsymbol{\Omega}}(\boldsymbol{U}|T)\right], \\ &\boldsymbol{\beta}^{t+1} = \arg\max_{\boldsymbol{\beta}} \mathbb{E}_{g^t}\left[\log p_{\boldsymbol{\beta}}(T)\right]. \end{aligned} \tag{6}$$

Observe that the matrix of edge weights $\boldsymbol{\beta}$ is considered here as a parameter to be estimated, as opposed to Robin et al. (2019), where is was kept fixed and supposed to be given.

VE step: Maximising $\mathcal{J}$ with respect to (wrt) q is equivalent to minimising the Küllback–Leibler divergence between $q(\boldsymbol{U}, T)$ and $p_{\boldsymbol{\theta},\boldsymbol{\beta},\boldsymbol{\Omega}}(\boldsymbol{U}, T \mid \boldsymbol{Y})$ that appears in (5). Because we adopted a product form for q, the solution of the VE step for both g and h is known to be a mean-field approximation (Wainwright & Jordan, 2008). More specifically, maximising $\mathcal{J}$ gives

$$\begin{aligned} g^{t+1}(T) &\propto \exp\left\{\mathbb{E}_{h^t}\left[\log p_{\boldsymbol{\theta}^{t+1},\, \boldsymbol{\beta}^{t+1},\, \Omega^{t+1}}(\boldsymbol{Y}, \boldsymbol{U}, T)\right]\right\} \\ &\propto \exp\left\{\log p_{\boldsymbol{\beta}^{t+1}}(T) + \mathbb{E}_{h^t}\left[\log p_{\boldsymbol{\Omega}^{t+1}}(\boldsymbol{U}|T)\right]\right\}, \end{aligned} \tag{7}$$

and

$$\begin{aligned} h^{t+1}(\boldsymbol{U}) &\propto \exp\left\{\mathbb{E}_{g^{t+1}}\left[\log p_{\boldsymbol{\theta}^{t+1},\, \boldsymbol{\beta}^{t+1},\, \Omega^{t+1}}(\boldsymbol{Y}, \boldsymbol{U}, T)\right]\right\} \\ &\propto \exp\left\{\mathbb{E}_{g^{t+1}}\left[\log p_{\boldsymbol{\Omega}^{t+1}}(\boldsymbol{U}|T)\right] + \log p_{\boldsymbol{\theta}^{t+1}}(\boldsymbol{Y}|\boldsymbol{U}_O)\right\}. \end{aligned} \tag{8}$$

Observing that $\log p_{\boldsymbol{\beta}}(T) + \log p_{\boldsymbol{\Omega}}(\boldsymbol{U} \mid T)$ can be written as a sum over all the edges present in T, we see that $g^{t+1}(T)$ has a product form. So, without any further assumption, we may parameterise $g(T)$ in the same way as $p_{\boldsymbol{\beta}}(T)$:

$$g(T) = \prod_{jk \in T} \tilde{\beta}_{jk} / \tilde{B} \quad \text{where} \quad \tilde{B} = \sum_{T \in \mathcal{T}_{p+r}} \prod_{jk \in T} \tilde{\beta}_{jk}. \tag{9}$$

We gather the $\tilde{\beta}_{jk}$'s in the $(p + r) \times (p + r)$ matrix $\tilde{\boldsymbol{\beta}}$. The parameters $\tilde{\boldsymbol{\beta}}$, $\boldsymbol{M}$ and $\boldsymbol{S}$ are called the variational parameters, in the sense that it is equivalent to optimise $\mathcal{J}$ wrt (g, h) or wrt $(\tilde{\boldsymbol{\beta}}, \boldsymbol{M}, \boldsymbol{S})$.

3.2 | Proposed algorithm

The model we consider is an extension of the PLN model, for which an efficient inference algorithm have been implemented in the `PLNmodels`, an R package available on CRAN (Chiquet et al., 2018, 2019).

3.2.1 | Prior estimates of $\boldsymbol{\theta}$, $\boldsymbol{M}_O$ and $\boldsymbol{S}_O$

To alleviate the computational burden of the inference, we take advantage of this available tool to get an estimate of the regression coefficient matrix $\widehat{\boldsymbol{\theta}}$ and an approximation of the parameters of the observed latent variable conditional distribution $h_O(\boldsymbol{U}_O) \simeq p(\boldsymbol{U}_O \mid \boldsymbol{Y})$. These latter parameters are $\boldsymbol{M}_O$ and $\boldsymbol{S}_O$ (first p columns of $\boldsymbol{M}$ and $\boldsymbol{S}$, respectively) and we denote $\tilde{\boldsymbol{M}}_O$ and $\boldsymbol{S}t_O$ their approximation. The quantities $\widehat{\boldsymbol{\theta}}$, $\tilde{\boldsymbol{M}}_O$ and $\tilde{\boldsymbol{S}}_O$ are kept fixed in the rest of the algorithm, so the VEM algorithm only deals with the remaining unknown quantities: the model parameters $\boldsymbol{\beta}$, $\boldsymbol{\Omega}$, and the variational parameters $\tilde{\boldsymbol{\beta}}$, $\boldsymbol{M}_H$, $\boldsymbol{S}_H$. As a consequence, the final estimates we get yield a lower value of the objective function $\mathcal{J}$ as compared to an optimisation wrt to all model and variational parameters.

3.2.2 | M step

This steps deals with the update of the model parameters $\boldsymbol{\beta}$ and $\boldsymbol{\Omega}_T$. Some of the calculations are tedious and postponed to Appendix B.

Edges weights $\boldsymbol{\beta}$: As shown in Equation (6), the maximisation of $\mathcal{J}$ requires the computation of the derivative of $\mathbb{E}_{g^t}[\log p_{\boldsymbol{\beta}}(T)]$ wrt $\boldsymbol{\beta}$, which includes the derivative of the normalising constant B. The latter can be computed via an extension of the matrix tree theorem (see Meilă & Jaakkola, 2006, Lemma 1 reminded in Appendix A). Setting the derivative of the expectation to 0 yields the following update (same as in Momal et al. (2020) and detailed in appendix B.1):

$$\beta_{kl}^{t+1} = \frac{P_{kl}^t}{M(\boldsymbol{\beta}^t)_{kl}},$$

where $M(\boldsymbol{\beta})$ is defined in Lemma 1 and P_{kl}^t is the probability that the edge (k, l) belongs to the tree T according to g^t:

$$P_{kl}^t = \mathbb{P}_{g^t}\{kl \in T\} = \sum_{\substack{T \in \mathcal{T}: \\ T \ni kl}} g^t(T) = \frac{1}{\tilde{B}^t} \sum_{\substack{T \in \mathcal{T}: \\ T \ni kl}} \prod_{uv \in T} \tilde{\beta}_{uv}^t.$$

P_{kl}^t is computed using a result from Kirshner (2008) (reminded as Lemma 2 in appendix A). We now define the binary variable I_{Tkl} which indicates the presence of the edge kl in tree T, so $P_{kl}^t = \mathbb{E}_{g^t}[I_{Tkl}]$ and $I_T = [I_{Tkl}]_{1 \le k,l \le (p+r)}$ is the adjacency matrix of tree T.

Precision matrices $\boldsymbol{\Omega}_T$: For a given dependency structure in the Gaussian graphical model framework, Lauritzen (1996) gives maximum likelihood estimates for the precision matrix. These estimators are given as functions of sufficient statistics of the multivariate Gaussian distribution. Indeed in the exponential family framework, the M step of any EM algorithm requires the computation of the expectation of a sufficient statistic, under the current fit of the variational laws (see McLachlan and Krishnan (2007)). Here as $\boldsymbol{U}|T$ is centred, a sufficient statistic is $\boldsymbol{U}^\intercal \boldsymbol{U}$. We now let SSD denote the matrix defined as

$$SSD^t = \mathbb{E}_{h^t}(\boldsymbol{U}^\intercal \boldsymbol{U}) = (\boldsymbol{M}^t)^\intercal \boldsymbol{M}^t + \boldsymbol{S}_+^t$$

where $\boldsymbol{S}_+^t = \sum_i \boldsymbol{S}_i^t$. Applying Lauritzen's formulas, we get:

$$\begin{aligned}
\omega_{Tkl}^{t+1} &= \begin{cases} \dfrac{-ssd_{kl}^t/n}{1-(ssd_{kl}^t/n)^2} & \text{if } kl \in T \\ 0 & \text{otherwise} \end{cases}, \\
\omega_{Tkk}^{t+1} &= 1 + \sum_l I_{Tkl} \frac{(ssd_{kl}^t/n)^2}{1-(ssd_{kl}^t/n)^2},
\end{aligned} \tag{10}$$

where ssd_{kl}^t stands for the entry kl of the matrix SSD^t. The calculations are postponed to Appendix B.2. Observe that estimates of the off-diagonal entries ω_{Tkl}^{t+1} do not depend on T provided that the edge (k, l) belongs to T. Thus the estimates of the off-diagonal terms of the precision matrices $\mathbf{\Omega}_T$ are common to all trees sharing a given edge. This does not result from any assumption on the shape of $\mathbf{\Omega}_T$, but from the properties of the maximum likelihood estimate of Gaussian variance matrix. In the sequel we will simply denote off-diagonal terms by ω_{kl} (as opposed to ω_{Tkk} which still depends on T).

Other quantities are needed for later computations. Lauritzen gives the maximum likelihood estimator of every entry of the correlation matrix $\boldsymbol{R}_T$ corresponding to an edge kl being part of T, which is $\boldsymbol{R}_{Tkl}^{t+1} = ssd_{kl}^t/n$. Hereafter for any matrix A, $A_{[kl]}$ refers to the bloc kl of A: $A_{[kl]} = (a_{ij})_{\{i,j\} \in \{k,l\}}$. The determinant of $\mathbf{\Omega}_T^{t+1}$ factorises on the edges of T and writes as a function of blocs of the correlation matrix as follows:

$$|\mathbf{\Omega}_T^{t+1}| = \left(\prod_{kl \in T} |\boldsymbol{R}_{T[kl]}^{t+1}| \right)^{-1} \quad \text{and for any } kl \in T,\ |\boldsymbol{R}_{T[kl]}^{t+1}| = 1 - (ssd_{kl}^t/n)^2. \tag{11}$$

Finally, we define the matrix $\overline{\mathbf{\Omega}}^{t+1} = \mathbb{E}_{g^t}[\mathbf{\Omega}_T^{t+1}]$. Noticing that, for $k \neq l$, $\mathbb{E}_{g^t}[\mathbf{\Omega}_T^{t+1}]_{kl} = \mathbb{E}_{g^t}[\mathbf{\Omega}^{t+1} \odot I_T]_{kl}$, edges probabilities appear as follows:

$$\overline{\omega}_{kl}^{t+1} = -P_{kl}^t \frac{ssd_{kl}^t/n}{1-(ssd_{kl}^t/n)^2}, \quad \overline{\omega}_{kk}^{t+1} = 1 + \sum_l P_{kl}^t \frac{(ssd_{kl}^t/n)^2}{1-(ssd_{kl}^t/n)^2}.$$

3.2.3 | VE step

This step deals with the update of the approximate conditional distributions g and h_H, namely the update of the corresponding variational parameters $\tilde{\boldsymbol{\beta}}$, $\boldsymbol{M}_H$ and $\boldsymbol{S}_H$.

Approximate conditional tree distribution $g(T)$: Computing the expression (7) yields the following, where the constant term 'cst' does not depend on a specific edge:

$$\begin{aligned}
\log g^{t+1}(T) &= \log p_{\beta^{t+1}}(T) + \mathbb{E}_{h^t}\left[\log p_{\mathbf{\Omega}^{t+1}}(\boldsymbol{U}|T)\right] + \text{cst} \\
&= \sum_{kl \in T} \log \beta_{kl}^{t+1} - \frac{n}{2} \log |\boldsymbol{R}_{[kl]}^{t+1}| - \omega_{kl}^{t+1} \left[(\boldsymbol{M}^t)^\intercal \boldsymbol{M}^t\right]_{kl} + \text{cst}
\end{aligned}$$

Then remembering the product form of g^{t+1} given in (9), we obtain the expression for each edge variational weight:

$$\tilde{\beta}_{kl}^{t+1} = \beta_{kl}^{t+1} \left| \boldsymbol{R}_{[kl]}^{t+1} \right|^{-n/2} \exp\left(-\omega_{kl}^{t+1} \left[(\boldsymbol{M}^t)^\intercal \boldsymbol{M}^t\right]_{kl} \right). \tag{12}$$

Approximate Gaussian distribution h: According to (8), we have that

$$\log h^{t+1}(\boldsymbol{U}) = \mathbb{E}_{g^{t+1}} \log p(\boldsymbol{Y} \mid \boldsymbol{U}_O) - \frac{1}{2} \mathrm{tr}\left(\overline{\boldsymbol{\Omega}}_T^{t+1} (\boldsymbol{U}^\intercal \boldsymbol{U})\right) + \mathrm{cst}.$$

Using the properties of the conditional Gaussian distribution we have that

$$h^{t+1}(\boldsymbol{U}_H \mid \boldsymbol{U}_O) = \mathcal{N}\left(\boldsymbol{U}_H; -\boldsymbol{U}_O \overline{\boldsymbol{\Omega}}_{OH}^{t+1} \left(\overline{\boldsymbol{\Omega}}_H^{t+1}\right)^{-1}, \left(\overline{\boldsymbol{\Omega}}_H^{t+1}\right)^{-1}\right).$$

Now, to get $h_H^{t+1}(\boldsymbol{U}_H)$, it suffices to integrate $h^{t+1}(\boldsymbol{U}_H \mid \boldsymbol{U}_O)$ wrt h_O (the parameter of which are kept fixed along iterations) to get

$$\boldsymbol{M}_H^{t+1} = -\tilde{\boldsymbol{M}}_O \overline{\boldsymbol{\Omega}}_{OH}^{t+1} \left(\overline{\boldsymbol{\Omega}}_H^{t+1}\right)^{-1}, \quad \boldsymbol{S}_H^{t+1} = \left(\overline{\boldsymbol{\Omega}}_H^{t+1}\right)^{-1}.$$

3.3 | Algorithm peculiarities

3.3.1 | Initialisation

As for any EM algorithm, the choice of the starting point is paramount. The initialisation we use here takes the primary estimate $\tilde{\boldsymbol{M}}_O$ as an input.

Initial clique: As a starting point, we look for a clique of species as potential neighbours of the missing actor h. There are many different ways to do so, and if any prior knowledge exists on that matter it should be used. Otherwise, such a clique can be found using sparse principal component analysis (sPCA; Erichson et al., 2020), where principal components are formed using only a few of the original variables, which is consistent with the assumption that each missing actor is connected only to some actors in the network. When applying sPCA to $\tilde{\boldsymbol{M}}_O$, the set of non-zero loadings of each principal components provides us with an initial clique of neighbours of each missing actor.

Parameters initialisation: The eigenvectors resulting from the sPCA also provide us with a starting value $\boldsymbol{M}_H^0$, as well as a first estimate of the latent correlation matrix $\boldsymbol{R}^0$. The parameter $\boldsymbol{\beta}$ is uniformly initialised.

3.3.2 | Numerical issues

Because the Matrix tree theorem and Kirshner's formula, respectively, resort to the calculation of a determinant and a matrix inversion, the proposed algorithm is exposed to numerical instabilities. To circumvent these issues, we rely on both multiple-precision arithmetic and likelihood tempering (via a parameter α, similarly to Schwaller & Robin, 2017). More details are given in Appendix B.4.

3.3.3 | Model selection

In practical analyses, the number of missing actors r needs to be chosen in some way. A natural way to select r is to resort to a penalised likelihood criterion, such as BIC (Schwarz, 1978) or ICL (Biernacki et al., 2000). The variational counterpart of these criteria remains an open question for many models. In our case, the simple plug-in of the lower-bound $\mathcal{J}$ in place of the log-likelihood yielded poor results. For the time being, we propose to choose r using a cross-validation heuristic described in Section 5.1 and Appendix D.

4 | SIMULATIONS

4.1 | Count data sets

For the simulation study, 300 count data sets of 15 species in total including one missing actor are generated, thus $p = 14$ and $r = 1$. We simulate scale-free graphs for the dependency structures, using the R package `huge` (Zhao et al., 2012) available on CRAN. The missing species h is chosen as the one with highest degree. Then, the variance–covariance matrix for the latent layer $\boldsymbol{U}$ is built as the observed part of the inverse of the adjacency graph matrix made positive definite. Finally, $\boldsymbol{U}$ and the observed abundances $\boldsymbol{Y}$ are simulated according to the PLN model defined in Section 2. This protocol yields over-dispersed counts which vary from 0 to about 500 on average (see Appendix C). To focus on the missing actor reconstruction, this simulation study does not involve covariates.

We further measure the *influence* of the missing actor with its degree, distinguishing three influence classes: *Minor* (degree ≤ 5), *Medium* ($5 <$ degree ≤ 7) and *Major* (degree ≥ 8).

4.2 | Experiment and measures

For each simulated data set, the VEM algorithm is initialised as described in Section 3.3. More specifically and because we only look for one missing actor, we consider the cliques corresponding to each of the first two principal components of sPCA, and their respective complements, which provides us with four cliques. Then four VEM algorithms, as described in Section 3.2, are run starting from each of the four candidate cliques, and the one yielding the highest lower bound $\mathcal{J}$ is kept. For all simulations, we set the precision of the convergence criterion to $\varepsilon = 10^{-3}$, the tempering parameter to $\alpha = 0.1$ and the maximal number of iterations to 100. The inference quality is assessed regarding the global network inference, the missing actor's position in the network, and its values along the n sites. We refer to this first procedure as the *blind* procedure. Additionally, we define the *oracle* procedure as running the VEM with the set of true neighbours of the missing actor as initial clique.

For each procedure, a general measure of the whole network inference quality is first given by comparing the inferred edge probabilities to the original dependency structure. This is done using the area under the ROC curve (AUC) criteria. Then, to be more specific and target the neighbours of node h specifically, the probabilities of edges involving h are transformed into binary values using the 0.5 threshold. The values are then compared to the original links of h and yield quantities of true/false (*T*/*P*) positives/negatives (*P*/*N*), from which are built the *precision* (also known as the positive predictive value, *TP*/(*TP*+*FP*)) and the *recall* (also known as the true positive rate, *TP*/(*TP*+*FN*)) criteria. Finally, we assess the ability to reconstruct the missing actor across the sites by computing the absolute correlation between its inferred vector of means ($\boldsymbol{M}_h$) and its original latent Gaussian vector $\boldsymbol{U}_h$.

4.3 | Results

Simulations performance measures are gathered in Table 1 and Table 2 for blind and oracle procedures, respectively. The distributions of the quality measures are displayed in Figure 4.

Table 1 shows the network is well inferred, as all AUC means are above 0.85, with almost perfect inference when the influence of the missing actor is major. Its neighbours and values per site are very well retrieved in these cases with mean recall values above 0.9 and mean correlation above 0.8, with a great confidence in the algorithm outputs as mean precision is above 0.95. However, there exists a

TABLE 1 Blind procedure using cliques from initialisation. The influence of the missing actor is measured with its degree, distinguishing three influence classes: *Minor* (degree ≤ 5), *Medium* (5 < degree ≤ 7) and *Major* (degree ≥ 8). For each class of influence, the following quantities are reported: number of simulated graphs (N), means and standard deviations of AUC, Precision, Recall, Correlation between missing actor inferred vector of means and original latent vector, and running times in seconds. AUC measures the retrieval of the dependence structure between all variables (observed and missing), whereas precision and recall are specific to the missing actor links

	N	AUC	Precision	Recall	Correlation	Time (s)
Major	100	0.98 (0.06)	0.96 (0.14)	0.94 (0.17)	0.83 (0.10)	2.36 (0.91)
Medium	132	0.93 (0.12)	0.83 (0.26)	0.81 (0.30)	0.73 (0.17)	2.69 (1.15)
Minor	68	0.89 (0.10)	0.61 (0.34)	0.66 (0.36)	0.59 (0.21)	3.08 (1.14)

TABLE 2 Oracle procedure using true clique as starting point. The influence of the missing actor is measured with its degree, distinguishing three influence classes: *Minor* (degree ≤ 5), *Medium* (5 < degree ≤7) and *Major* (degree ≥ 8). For each class of influence, the following quantities are reported: number of simulated graphs (N), means and standard deviations of AUC, Precision, Recall, Correlation between missing actor inferred vector of means and original latent vector, and running times in seconds. AUC measures the retrieval of the dependence structure between all variables (observed and missing), whereas precision and recall are specific to the missing actor links

	N	AUC	Precision	Recall	Cor.	t(s)
Major	100	1 (0.00)	1 (0.00)	1 (0.01)	0.86 (0.02)	1.28 (0.21)
Medium	132	1 (0.02)	1 (0.00)	0.99 (0.04)	0.83 (0.02)	1.38 (0.46)
Minor	68	0.98 (0.04)	0.99 (0.03)	0.96 (0.12)	0.8 (0.04)	1.56 (0.69)

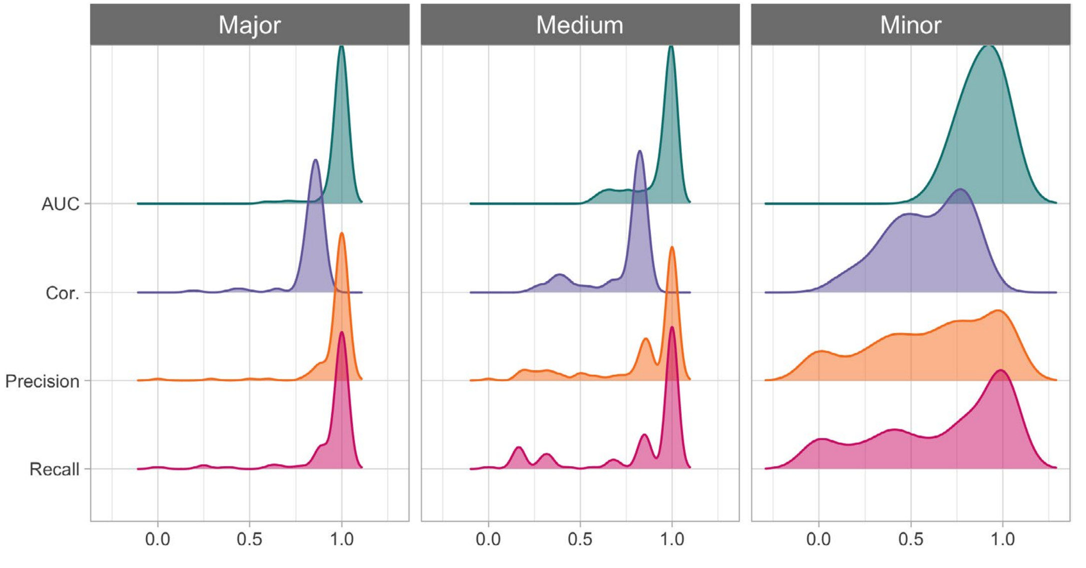

FIGURE 4 The influence of the missing actor is measured with its degree, distinguishing three influence classes: *Minor* (degree ≤ 5), *Medium* (5 < degree ≤ 7) and *Major* (degree ≥ 8). The distributions of performance measures are displayed for each class of influence: AUC measures the retrieval of the dependence structure between all variables, observed and missing. Precision and recall are specific to the missing actor links [Colour figure can be viewed at wileyonlinelibrary.com]

clear deterioration of all performance as the influence decreases with lower means are greater deviations, down to about 0.6 mean values for all measures when the influence is minor. Moreover, the algorithm takes more and more time to converge as the influence decreases, although it stays at about 3*s* for minor cases which is reasonable. Figure 4 shows that as the influence decreases, the densities present with several modes and dilute towards 0, illustrating that even if some networks are still well-inferred, there also are more and more cases where the algorithm fails. In particular, the performance decrease of medium cases seems to be only due to a greater number of failed inferences.

All these elements point to minor cases being harder problems to solve, unsurprisingly. Yet as oracle results show in Table 2, it is possible to carry out almost-perfect inference in all cases, if the algorithm is initialised with the true clique; the deterioration is still present in all measures, but stays marginal. Thus the harsh decrease in the blind procedures seems to be mainly due to the proposed initialisation method failing at correctly finding some of the small cliques of neighbours.

About intialisation. Figure 5 compares the initialisation quality and the corresponding final inferred neighbours, in terms of initial (-i) and final (-f) false negative (FNR, also 1-TPR) and positive rates (FPR). It clearly appears that final measures mostly increase with false negatives of the initial clique. This means that not including a neighbour in the initialisation is much worse for the inference than falsely including a node. The increase of FNR-f is bigger than that of FPR-f, meaning that a wrong initialisation leads to a set of inferred neighbours which most part can be trusted, but which will be largely incomplete. This advocates for bigger initialisation cliques when no prior information is available.

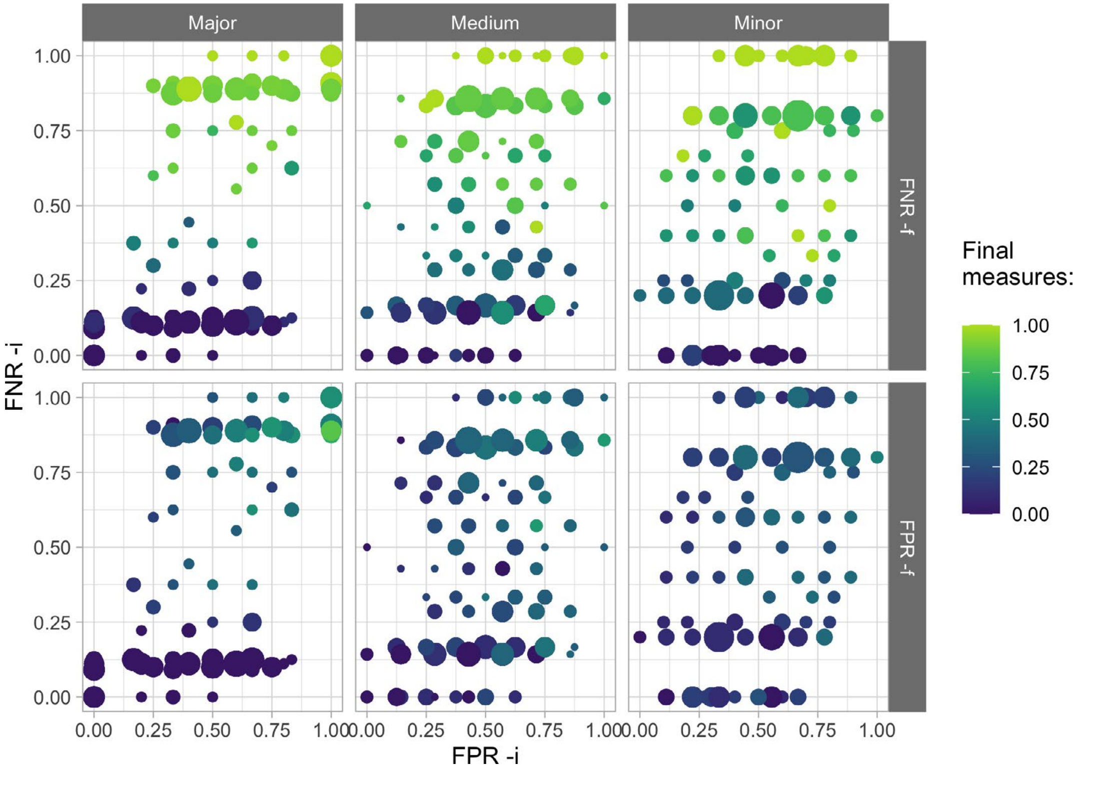

FIGURE 5 Comparison of initial and final FPR and FNR, for cliques of neighbours of one missing actor obtained with the sparse PCA method. Position of dots are defined according to initial values, their color according to the final FPR and FNR. Sizes are proportional to the density of dots on a given position [Colour figure can be viewed at wileyonlinelibrary.com]

5 | APPLICATIONS

We now illustrate the use of the proposed model on two abundance data sets published by Fossheim et al. (2006) and Baran (1995), respectively. In both cases, the species abundance data are accompanied with environmental covariates describing each site. The importance of accounting for environmental effects to better understand the dependency structure between species has been widely discussed in the literature about general joint species distribution models (see, e.g. Popovic et al., 2018, 2019; Warton et al., 2015) and, more specifically for the PLN model (see Chiquet et al., 2018, 2019). One aim of the present section is to assess the ability of the missing actor to reveal some unknown underlying effect. To this aim, we chose to keep the available covariate aside (that is: we did not include them in the model) to, then, compare the inferred missing actor with them.

5.1 | Cross-validation criterion for model selection

The proposed model obviously raises the problem of choosing the number of missing actors r (which may be zero). Variational-based inference often relies on approximate versions of the BIC or ICL criteria for model selection. Few theoretical guaranties exist about these approximate criteria and, in the present case, we observed that BIC and ICL penalisations did not yield consistent results. Therefore, we resort to V-fold cross-validation to determine the number of missing actors.

More specifically, we split the original data set $\boldsymbol{Y}$ ($\boldsymbol{X}$ is dropped here for the sake of clarity) into V subsets with almost equal sizes $m_1, \dots m_V$ ($\sum_{v=1}^{V} m_v = n$), which we denote $\{\boldsymbol{Y}^v\}_{v=1,\dots V}$. For each subset v, we define its complement $\boldsymbol{Y}^{-v}$ on which we fit a model with r missing actors and get a parameter estimate $\Gamma_r^{-v} = (\boldsymbol{\theta}_r^{-v}, \boldsymbol{\sigma}_r^{-v}, \boldsymbol{\beta}_r^{-v}, \boldsymbol{\Omega}_r^{-v})$ and measure the fit of Γ_r^{-v} to the test data set $\boldsymbol{Y}^v$.

To avoid the integration over the $(p + r)$-dimensional Gaussian latent layer, we measure the fit with the pairwise composite likelihood (PCL: Lindsay, 1988). For any given tree T and parameter $\boldsymbol{\Gamma}$, the bivariate Poisson log-normal pdf $p_{PLN}\left((Y_{ij}, Y_{ik}); \Gamma, T\right)$ can be easily computed for any sample i and pair of species (j, k) with available tools such as the `poilog` R package (Vidar & Steinar, 2008) available on CRAN, which resorts to a two-dimensional numerical integration. The cross-validation criterion is defined as

$$PCL_r(\boldsymbol{Y}) = \frac{1}{V}\sum_{v}\frac{1}{B}\sum_{b=1}^{B}\frac{1}{m_v}\sum_{i=1}^{m_v}\sum_{j<k}\log p_{PLN}\left((Y_{ij}^v, Y_{ik}^v); \Gamma_r^{-v}, T_{r,b}^{-v}\right)$$

where the tree samples $\{T_{r,b}^{-v}\}_{b=1\dots B}$ are iid according to $p_{\beta_r^{-v}}(T)$.

The sampling procedure for spanning trees is given in Appendix D.1; the complete procedure for the calculation of $PCL_r(\boldsymbol{Y})$ is described by Algorithm 1, given in Appendix D. Note that this criterion measures the fit of the model in terms of abundance prediction, whereas our interest is mostly focused on the inference of the dependency structure. In other words, our goal is identification that is selecting the smallest model and not the best model in terms of prediction (Arlot & Celisse, 2010).

We did not include this computationally greedy procedure in the complete simulation study, but tested it on a reduced number of data sets as described in Appendix D.2. The results suggest that this procedure is conservative, meaning that it has a higher probability of not detecting a missing actor when $r = 0$ (true negative) than of detecting one when $r = 1$ (true positive).

We then applied it to the two ecological data sets that will be described in the next two sections. Figure 6 shows the computed PCL_r criterion on a grid of r values from 0 to 2. Selecting the maximal

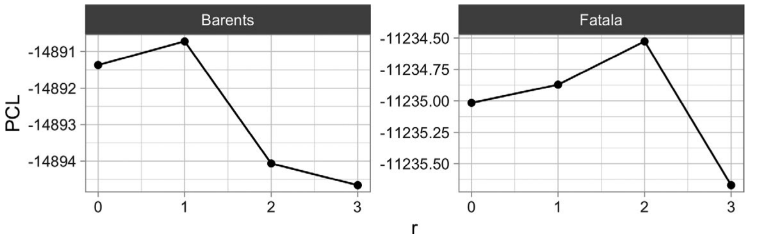

FIGURE 6 Pairwise composite likelihoods estimates of Barents and Fatala data sets for models including 0 to 3 missing actors

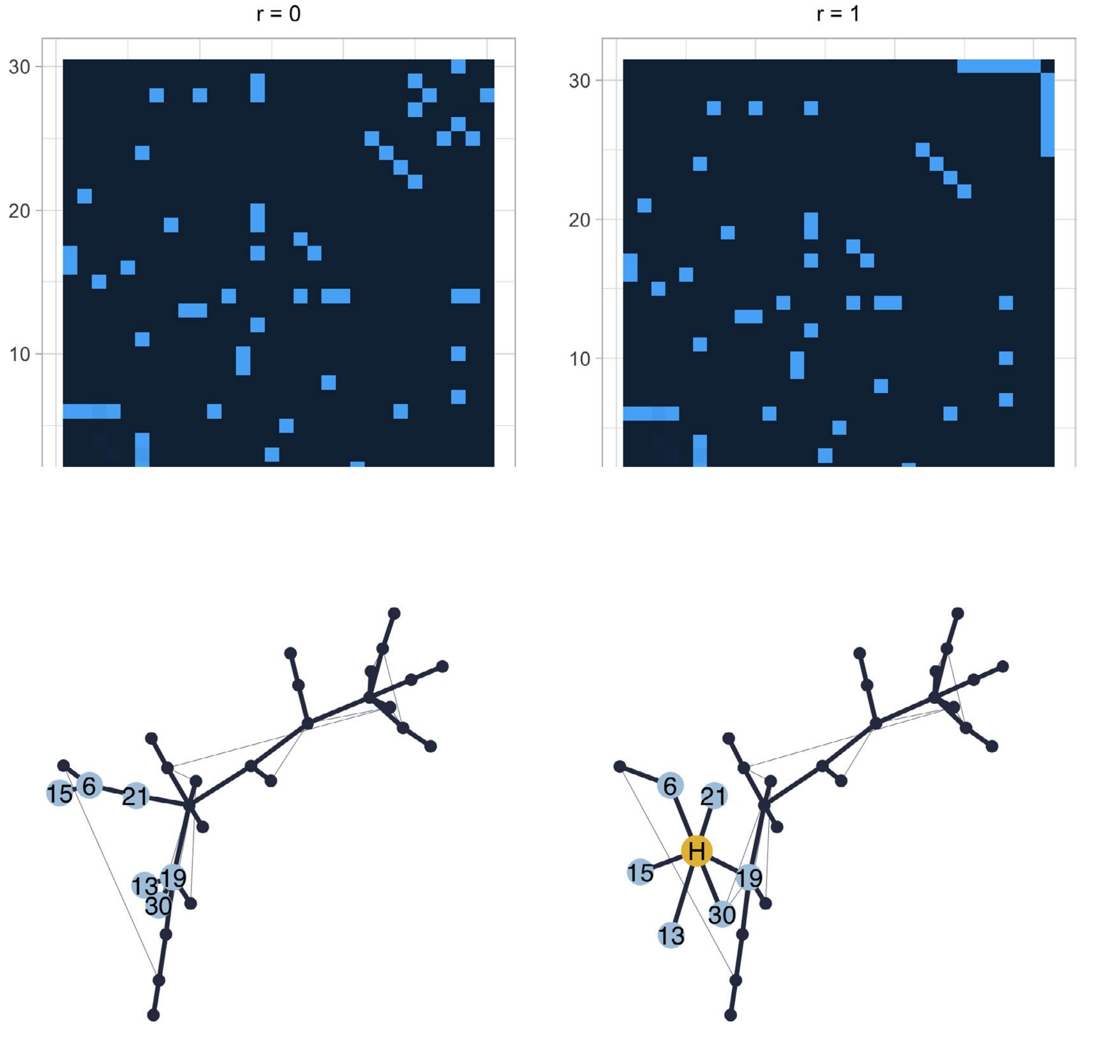

FIGURE 7 *Top left:* adjacency matrix of the Barents Sea fishes interaction network for $r = 0$ missing actor. The inferred neighbours are gathered in the last 6 columns, so that their interactions are observable in the upper-right corner. *Top right:* adjacency matrix for $r = 1$ missing actor. The last column gathers the interactions of the inferred missing actor. *Bottom*: Inferred interaction network with $r = 0$ (left) and $r = 1$ (right). Coloured nodes refer to the inferred neighbours (blue) of the missing actor (yellow). The width of the edges is proportional to their probability P_{kl} [Colour figure can be viewed at wileyonlinelibrary.com]

PCL_r value yields $r = 1$ missing actor for the Barents Sea data set, and $r = 2$ missing actors for the Fatala River one.

Regarding the initialisation, we performed a wider exploration as compared to the simulation study. To enlarge the list of possible cliques, we applied a resampling version of the procedure described in Section 3.3, and applied it to 200 subsamples, each consisting in 80% of the whole data set. This yielded 200 lists of r initial cliques, from which duplicates were removed.

5.2 | Barents Sea

The data set was first published by Fossheim et al. (2006) and consists of the abundance of 30 fish species measured in 89 sites in the Barents See in April-May 1997. In addition to abundances, the water temperature was measured in each site. The complete data set is available at www.fbbva.es/microsite/multivariate-statistics/data.html. Fishes distributions are known to be greatly linked with the temperature. As explained above, we present the results of the model fitted without any covariate (that is not accounting for the temperature), but including one missing actor (as suggested by Figure 6). To assess the ability of the proposed methodology to retrieve the influence of temperature as a missing actor, we report the empirical correlation between the temperature and the conditional expectation of the missing actor M_h, which we denote $\rho(H, \text{temp})$.

The resampling initialisation procedure yielded in 14 different cliques, for each of which a VEM algorithm was run: the mean running time was 6.63mins with deviation 0.70 mins.

The edge probabilities involving node h as an endpoint were either very close to 0 or very close to 1, yielding a total of 6 highly probable neighbours of h. Figure 7 shows that many direct interactions are inferred between the corresponding six species in absence of a missing actor, which vanish when it is introduced. It also shows that accounting for this actor has only a local effect and that the direct interactions among the other species are preserved, which is consistent with our notion of a missing actor.

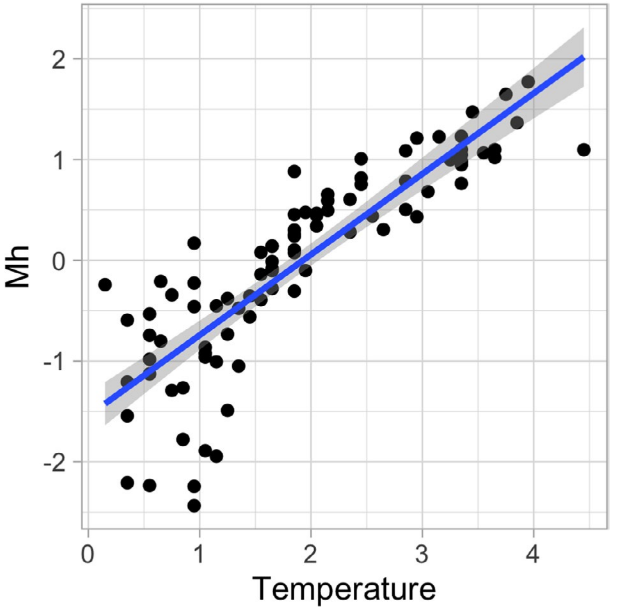

FIGURE 8 Missing actor estimated vector of means M_h as a function of the temperature. $\rho(H,\text{temp}) = 0.85$ [Colour figure can be viewed at wileyonlinelibrary.com]

In terms of interpretation, Figure 8 shows that the missing actor is highly correlated with the temperature. It also appears that the abundances of the species neighbour to the missing actor are much more correlated with the temperature (mean correlation = 0.78, sd = .06) than the abundances of the non-neighbour species (mean correlation = 0.46, sd = .27). This example shows the ability of the method to recover an underlying effect that would not be recorded in the data.

5.3 | Fatala River

Baran (1995) collected the abundances of 33 fish species in 90 sites along the Fatala River in Guinea between June 1993 and February 1994. The data are available from the R package `ade4` on CRAN (Dray & Dufour, 2007), along with the date and site of collection, from which we deduce the season (dry or rainy). Again the model was fitted without any covariates, but with two missing actors, as suggested by Figure 6.

The resampling initialisation procedure yielded in 60 different cliques, for each of which a VEM algorithm was run: the mean running time was 11.33 min (sd = 1.47 mn). 14 VEM did not reach convergence (with tolerance $\varepsilon = 1e - 3$) after 100 iterations. We filtered out the results obtained from the different initialisations, when the algorithm obviously ended in a degenerate solution ($\mathbb{V}(M_h) < \exp(-20)$).

Figure 9 shows the scatterplot of the estimated conditional mean of the two missing actors (M_{h_1}, M_{h_2}) in each site, coloured with either one of the available covariates (site and season). The

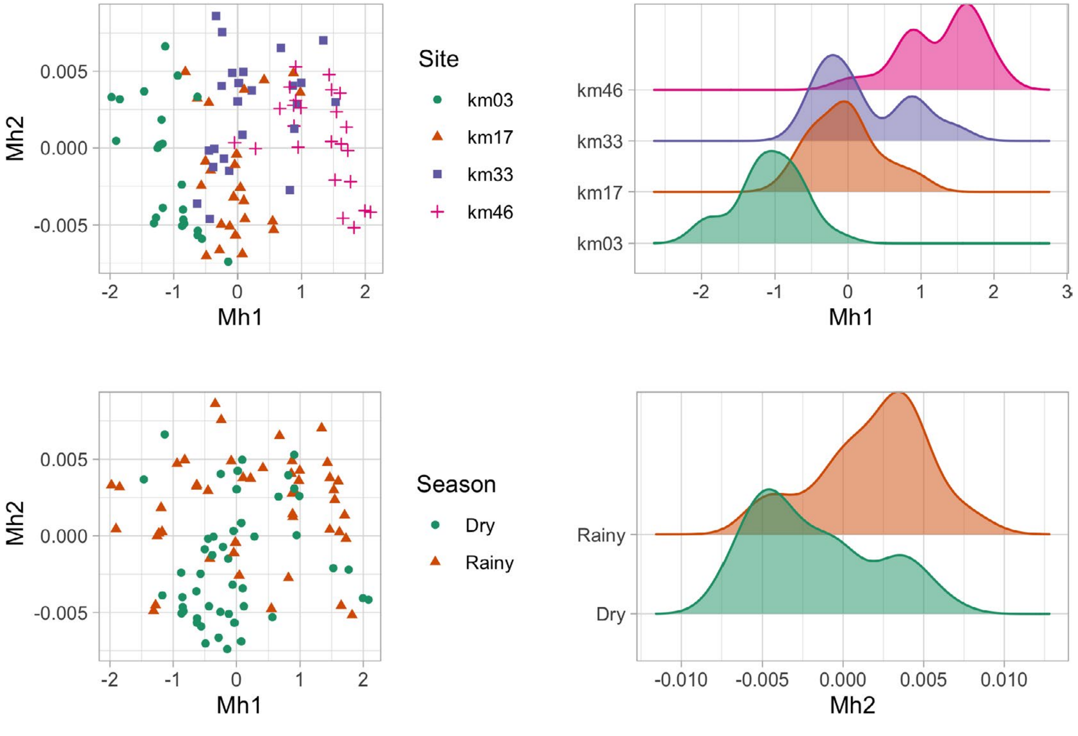

FIGURE 9 Estimated means M_{h_1} and M_{h_2} of the two inferred missing actors. Left column: scatterplots M_{h_1} vs M_{h_2} with site (top) and season (bottom) colour code. Right: distribution of the estimated means across sites. Top right: distribution of M_{h_1} in each location, bottom right: distribution of M_{h_2} in each season [Colour figure can be viewed at wileyonlinelibrary.com]

missing actor h_1 is obviously linked to the site and separates most upstream locations (kilometre 3) from most downstream locations (kilometre 46). This actor has 11 highly probable neighbour species. Again, this retrieved missing actor corresponds to an underlying effect (in this case: geography) that rules fish species abundances.

The second missing actor seems to be linked with the season but with a less clear separation. Also the variability of M_{h_2} is much smaller than this of M_{h_1}. This effect is therefore questionable, which brings us back to model selection. As mentioned above, we used a procedure based on cross-validation, which may be prone to select too complex model (Arlot & Celisse, 2010; Friedman et al., 2001; Shao, 1993). The definition of a grounded model selection criterion for structure inference in presence of missing actors remains open.

6 | DISCUSSION

We introduced a novel approach for network inference for count data, including missing actors. Although several methods have been previously proposed for network inference, this is, to our knowledge, the first both to deal with count data and to account for a missing actor. The proposed model is similar to Poisson log-normal, where the latent layer is enriched with few missing variables each corresponding to an unrecorded actor of the network. To manage complex hidden structure of the model, the inference strategy resorts to a variational approximation of the likelihood. We demonstrated the interest of this approach to detect underlying drivers of species abundances in community ecology.

Although variational approximations have been proven to be accurate in terms of parameter estimation for many models, they may raise problems in terms of model selection. In the present context, model selection is needed to chose the number of missing actors (which can be zero). A common—and heuristic—approach consists in applying some standard penalisation (such as BIC or ICL), originally derived for the log-likelihood, to the variational lower-bound itself. This turned out to fail for the present model, so we proposed a heuristic criterion based on cross-validation, which provides satisfying results, but is time consuming. The problem of defining general model selection criteria consistent with variational approximations remains open at this time.

The proposed approach aims at inferring the latent network that is the structure of the graphical model of the latent variable (including missing actors). This graphical model does not necessarily coincide with this of the observed variables, which is a limitation of all network inference methods relying on a latent layer. Still, the Gaussian framework remains predominant (because reasonably manageable) when accounting for the existence of missing actors. This explains why we opted for the Poisson log-normal model. A generic framework for network inference in the observed layer with missing actors for counts or, more generally, for non-Gaussian data remains to be defined.

ACKNOWLEDGEMENTS

This work was partly supported by the French ANR-18-CE02-0010 Ecological Networks (EcoNet) project and by the French ANR-11-LABX-0056-LMH LabEx Laboratoire de Mathématique Hadamard.

ORCID

Raphaëlle Momal https://orcid.org/0000-0002-1550-4530

REFERENCES

Aitchison, J. & Ho, C.H. (1989) The multivariate Poisson-log normal distribution. *Biometrika*, 76(4), 643–653.

Ambroise, C., Chiquet, J. & Matias, C. (2009) Inferring sparse Gaussian graphical models with latent structure. *Electronic Journal of Statistics*, 3, 205–238.

Arlot, S. & Celisse, A. (2010) A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4, 40–79.

Baran, E. (1995) Dynamique spatio-temporelle des peuplements de Poissons estuariens en Guinée (Afrique de l'Ouest). PhD Thesis, Thèse de Doctorat, Université de Bretagne Occidentale.

Besag, J. (1974) Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2), 192–225.

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), 719–25.

Blei, D.M., Kucukelbir, A. & McAuliffe, J.D. (2017) Variational inference: a review for statisticians. *Journal of the American Statistical Association*, 112(518), 859–877.

Candès, E.J., Li, X., Ma, Y. & Wright, J. (2011) Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3), 1–37.

Chaiken, S. & Kleitman, D.J. (1978) Matrix tree theorems. *Journal of Combinatorial Theory, Series A*, 24(3), 377–381.

Chandrasekaran, V., Sanghavi, S., Parrilo, P.A. & Willsky, A.S. (2011) Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21, 572–596.

Chiquet, J., Mariadassou, M. & Robin, S. (2018Variational inference for sparse network reconstruction from count data. Technical Report 1806.03120, arXiv.

Chiquet, J., Mariadassou, M. & Robin, S. (2018) Variational inference for probabilistic Poisson PCA. *The Annals of Applied Statistics*, 12(4), 2674–2698.

Chiquet, J., Mariadassou, M., Robin, S. (2019) Variational inference for sparse network reconstruction from count data. In: *International Conference on Machine Learning*.

Chow, C. & Liu, C. (1968) Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3), 462–467.

Dempster, A.P., Laird, N.M. & Rubin, D.B. (1977) Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 39, 1–38.

Devroye, L. (1986) *Non-uniform random variate generation*. Berlin: Springer.

Dray, S., Dufour, A.-B. (2007) The ade4 package: implementing the duality diagram for ecologists. *Journal of Statistical Software*, 22(4), 1–20.

Durfee, D., Kyng, R., Peebles, J., Rao, A.B. & Sachdeva, S. (2017) Sampling random spanning trees faster than matrix multiplication. In: *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 730–742.

Erichson, N.B., Zheng, P., Manohar, K., Brunton, S.L., Kutz, J.N. & Aravkin, A.Y. (2020) Sparse principal component analysis via variable projection. *SIAM Journal on Applied Mathematics*, 80(2), 977–1002.

Fossheim, M., Nilssen, E.M. & Aschan, M. (2006) Fish assemblages in the Barents Sea. *Marine Biology Research*, 2(4), 260–269.

Friedman, J., Hastie, T. & Tibshirani, R. (2001) *The elements of statistical learning*, Vol. 1. New York: Springer Series in Statistics.

Friedman, J., Hastie, T. & Tibshirani, R. (2008) Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3), 432–441.

Giraud, C. & Tsybakov, A. (2012) Discussion of "latent variable graphical model selection via convex optimization". *Annals of Statistics*, 40(4), 1984–1988.

Guillera-Arroita, G. (2017) Modelling of species distributions, range dynamics and communities under imperfect detection: advances, challenges and opportunities. *Ecography*, 40(2), 281–295.

Hardin, J.W. & Hilbe, J.M. (2007) *Generalized linear models and extensions*. College Station: Stata Press.

Inouye, D., Ravikumar, P. & Dhillon, I. (2016) Square root graphical models: multivariate generalizations of univariate exponential families that permit positive dependencies. In: *International Conference on Machine Learning*, pp. 2445–2453. PMLR.

Inouye, D.I., Yang, E., Allen, G.I. & Ravikumar, P. (2017) A review of multivariate distributions for count data derived from the poisson distribution. *Wiley Interdisciplinary Reviews: Computational Statistics*, 9(3), e1398.

Kirshner, S. (2008) Learning with tree-averaged densities and distributions. In: *Advances in Neural Information Processing Systems*, pp. 761–768.

Lauritzen, S.L. (1996) *Graphical models*. Oxford Statistical Science Series. Oxford: Clarendon Press.

Lauritzen, S. & Meinshausen, N. (2012) Discussion: latent variable graphical model selection via convex optimization. *The Annals of Statistics*, 40(4), 1973–1977.

Lindsay, B.G. (1988) Composite likelihood methods. *Contemporary Mathematics*, 80(1), 221–239.

Lucas, A., Scholz, I., Boehme, R., Jasson, S. & Maechler, M. (2020) GMP: Multiple Precision Arithmetic. Available from: https://CRAN.R-project.org/package=gmp. R package version 0.5-13.6.

Lun, A.T.L., Chen, Y. & Smyth, G.K. (2016) It's de-licious: a recipe for differential expression analyses of rna-seq experiments using quasi-likelihood methods in edger. *Methods in Molecular Biology*, 1418, 391–416.

McLachlan, G.J. & Krishnan, T. (2007) *The EM algorithm and extensions*, Vol. 382. Hoboken: John Wiley & Sons.

Meilă, M. & Jaakkola, T. (2006) Tractable Bayesian learning of tree belief networks. *Statistics and Computing*, 16(1), 77–92.

Meilă, M. & Jordan, M.I. (2000) Learning with mixtures of trees. *Journal of Machine Learning Research*, 1, 1–48.

Meng, Z., Eriksson, B. & Hero III, A.O. (2014) Learning latent variable Gaussian graphical models. *Proceedings of the 31 International Conference on Machine Learning*, 32, 1269–1277.

Momal, R., Robin, S. & Ambroise, C. (2020) Tree-based inference of species interaction networks from abundance data. *Methods in Ecology and Evolution*, 11, 621–632. doi: 10.1111/2041-210X.13380. https://besjournals.onlinelibrary.wiley.com/doi/abs/10.1111/2041-210X.13380.

Popovic, G.C., Hui, F.K.C. & Warton, D.I. (2018) A general algorithm for covariance modeling of discrete data. *Journal of Multivariate Analysis*, 165, 86–100.

Popovic, G.C., Warton, D.I., Thomson, F.J., Hui, F.K.C. & Moles, A.T. (2019) Untangling direct species associations from indirect mediator species effects with graphical models. *Methods in Ecology and Evolution*, 10(9), 1571–1583.

Robin, G., Ambroise, C. & Robin, S. (2019) Incomplete graphical model inference via latent tree aggregation. *Statistical Modelling*, 19(5), 545–568.

Robinson, M.D. & Oshlack, A. (2010) A scaling normalization method for differential expression analysis of rna-seq data. *Genome Biology*, 11(3), 1–9.

Roy, A. & Dunson, D.B. (2020) Nonparametric graphical model for counts. *Journal of Machine Learning Research*, 21(229), 1–21.

Schwaller, L. & Robin, S. (2017) Exact Bayesian inference for off-line change-point detection in tree-structured graphical models. *Statistics and Computing*, 27 (5), 1331–1345. http://dx.doi.org/10.1007/s11222-016-9689-3.

Schwaller, L., Robin, S. & Stumpf, M. (2019) Bayesian inference of graphical model structures using trees. *Journal of Sociology France Statistics*, 160(2), 1–23.

Schwarz, G. (1978) Estimating the dimension of a model. *The Annals of Statistics*, 6(2), 461–464.

Shao, J. (1993) Linear model selection by cross-validation. *Journal of the American statistical Association*, 88(422), 486–494.

Vidar, G. & Steinar, E. (2008) Poilog: Poisson lognormal and bivariate Poisson lognormal distribution. R package version 0.4.

Wainwright, M.J. & Jordan, M.I. (2008) Graphical models, exponential families, and variational inference. *Foundations and Trends® Machine Learning*, 1(1–2), 1–305.

Wan, Y.-W., Allen, G.I., Baker, Y., Yang, E., Ravikumar, P., Anderson, M. et al. (2016) Xmrf: an r package to fit Markov networks to high-throughput genetics data. *BMC Systems Biology*, 10(3), 69.

Warton, D.I., Blanchet, F.G., O'Hara, R.B., Ovaskainen, O., Taskinen, S., Walker S.C. et al. (2015) So many variables: joint modeling in community ecology. *Trends in Ecology & Evolution*, 30(12), 766–779.

Yang, E., Ravikumar, P., Allen, G.I. & Liu, Z. (2013) On Poisson graphical models. *Advances in Neural Information Processing Systems*, 26, 1718–1726.

Zhao, T., Liu, H., Roeder, K., Lafferty, J. & Wasserman, L. (2012) The huge package for high-dimensional undirected graph estimation in R. *The Journal of Machine Learning Research*, 13(1), 1059–1062.

How to cite this article: Momal R, Robin S, Ambroise C. Accounting for missing actors in interaction network inference from abundance data. *J R Stat Soc Series C*. 2021;70:1230–1258. https://doi.org/10.1111/rssc.12509

APPENDIX A

ALGEBRAIC TOOLS

We here present some algebraic results about spanning tree structures which are used during the computations. Theorem 1, Lemma 1 as well as Lemma 2 use the notion of Laplacian matrix $\boldsymbol{Q}$ of a symmetric matrix $\boldsymbol{W} = [w_{jk}]_{1\le j,k\le p}$, which is defined as follows:

$$[\boldsymbol{Q}]_{jk} = \begin{cases} -w_{jk} & 1 \le j < k \le p \\ \sum_{u=1}^{p} w_{ju} & 1 \le j = k \le p. \end{cases}$$

We further denote $\boldsymbol{W}^{uv}$ the matrix $\boldsymbol{W}$ deprived from its uth row and vth column and we remind that the (u, v)-minor of $\boldsymbol{W}$ is the determinant of this deprived matrix, that is $|\boldsymbol{W}^{uv}|$. The following Theorem 1 is the extension of Kirchhoff's Theorem to the case of weighted graphs (Chaiken & Kleitman, 1978; Meilă & Jaakkola, 2006).

Theorem 1 *(Matrix Tree Theorem) For any symmetric weight matrix W with all positive entries, the sum over all spanning trees of the product of the weights of their edges is equal to any minor of its Laplacian. That is, for any $1 \le u,v \le p$,*

$$W := \sum_{T\in\mathcal{T}} \prod_{(j,k)\in T} w_{jk} = |\boldsymbol{Q}^{uv}|.$$

In the following, without loss of generality, we will choose $\boldsymbol{Q}^{11}$. As an extension of this result, Meilă and Jaakkola (2006) provide a close form expression for the derivative of W with respect to each entry of $\boldsymbol{W}$.

Lemma 1 *(Meilă and Jaakkola (2006)) Define the entries of the symmetric matrix $\boldsymbol{M}$ as*

$$[\boldsymbol{M}]_{jk} = \begin{cases} \left[(\boldsymbol{Q}^{11})^{-1}\right]_{jj} + \left[(\boldsymbol{Q}^{11})^{-1}\right]_{kk} - 2\left[(\boldsymbol{Q}^{11})^{-1}\right]_{jk} & 1 < j < k \le p \\ \left[(\boldsymbol{Q}^{11})^{-1}\right]_{jj} & k = 1, 1 < j \le p \\ 0 & j = k. \end{cases}$$

it then holds that

$$\partial_{w_{jk}} W = [\boldsymbol{M}]_{jk} \times W.$$

Kirshner (2008) build on Lemma 1 to provide an efficient computation of all edges probabilities.

Lemma 2 *(Kirshner (2008)) Let p_W be a distribution on the space of spanning trees, such that $p_W(T) = \prod_{kl\in T} w_{kl}/W$, where W is defined as in Theorem 1. Taking the symmetric matrix $\boldsymbol{M}$ as defined in Lemma 1, the probability for an edge kl to be in the tree $\boldsymbol{T}^*$ writes:*

$$\mathbb{P}\{kl \in T^*\} = \sum_{T \in \mathcal{T}} p_W(T) = w_{kl}\boldsymbol{M}_{kl}$$

APPENDIX B

COMPUTATIONS

B.1 Update of β

As in Momal et al. (2020), the update of $\boldsymbol{\beta}$ is such that:

$$\boldsymbol{\beta}^{t+1} = \arg\max_{\boldsymbol{\beta}} \mathbb{E}_{g^t}\left[\log p_{\boldsymbol{\beta}}(T)\right].$$

By definition of $p_{\boldsymbol{\beta}}(T)$:

$$\mathbb{E}_{g^t}\left[\log p_{\boldsymbol{\beta}}(T)\right] = \sum_{kl} P^t_{kl}\log\beta_{kl} - \log B\,, \quad B = \sum_{T \in \mathcal{T}}\prod_{kl \in T}\beta_{kl}.$$

Computing the derivative with respect to the edge weight β_{kl} gives:

$$\partial_{\beta_{kl}}\mathbb{E}_{g^t}\left[\log p_{\boldsymbol{\beta}}(T)\right] = \frac{P^t_{kl}}{\beta_{kl}} - \frac{\partial_{\beta_{kl}}B^t}{B^t}$$

According to Lemma 1: $\partial_{\beta_{kl}}B^t = [\boldsymbol{M}]_{kl} \times B$. Finally setting the derivative to 0 yields the update formula $\beta^{t+1}_{kl} = \frac{P^t_{kl}}{M(\boldsymbol{\beta}^t)_{kl}}$.

B.2 Update of Ω_T

The update of $\boldsymbol{\Omega}_T$ respects

$$\boldsymbol{\Omega}^{t+1} = \arg\max_{\boldsymbol{\Omega}} \mathbb{E}_{q^t}\left[\log p_{\boldsymbol{\Omega}}(\boldsymbol{U}|T)\right].$$

This is a problem of parameter optimisation in the context of Gaussian graphical models (GGM). In what follows, for any $q \times q$ matrix A, $A_{[kl]}$ will refer to the bloc kl of A: $A_{[kl]} = (a_{ij})_{\{i,j\}\in\{k,l\}}$. $[A_{[kl]}]^q$ will then denote the matrix obtained by filling up with zero entries to obtain full dimension $q \times q$, so that:

$$([A_{[kl]}]^q)_{ij} = \begin{cases} a_{ij} \text{ if } \{i,j\} \in \{k,l\} \\ 0 \text{ if } \{i,j\} \in \{1,\ldots,q\}_{\backslash kl} \end{cases}$$

In its proposition 5.9, Lauritzen (1996) states that in a GGM with p variables and associated with the decomposable graph $\mathcal{G}$, the maximum likelihood of the precision matrix exists if and only if $n > \max_{C\in\mathcal{C}}|C|$. It is then given as

$$\widehat{\boldsymbol{\Omega}} = n\left(\sum_{C \in \mathcal{C}}[SSD^{-1}_{[C]}]^p - \sum_{S \in \mathcal{S}}\nu(S)[SSD^{-1}_{[S]}]^p\right)$$

where $\mathscr{C}$ is the set of cliques and $\mathscr{S}$ the set of separators of $\mathscr{G}$, with associated multiplicities $\nu(S)$.

In our context, $\mathscr{G}$ is a spanning tree and so all cliques are edges and separators are nodes. The multiplicity of a given node k as a separator in the graph is $\nu(k) = d(k) - 1$, where $d(k)$ is its degree. Therefore the estimator $\widehat{\mathbf{\Omega}}_T$ writes as the following

$$\begin{aligned}\widehat{\mathbf{\Omega}}_T &= n\sum_{kl\in T}[(SSD_{[kl]})^{-1}]^{p+r} - n\sum_{k}(d(k)-1)[(SSD_{kk})^{-1}]^{p+r}\\ &= n\sum_{kl\in T}[(SSD_{[kl]})^{-1} - (SSD_{kk})^{-1} - (SSD_{ll})^{-1}]^{p+r} + n\sum_{k}[(SSD_{kk})^{-1}]^{p+r}\end{aligned}$$

As SSD has diagonal n, the expression simplifies. Denoting I_d the identity matrix of dimension d we obtain:

$$\widehat{\mathbf{\Omega}}_T = n\sum_{kl\in T}[(SSD_{[kl]})^{-1} - \frac{1}{n}I_2]^{p+r} + I_{p+r}.$$

Detailing each bloc matrices as follows gives the update formulas in (10):

$$n \times [(SSD_{[kl]})^{-1} - \frac{1}{n}I_2] = \frac{1}{1-(ssd_{kl}/n)^2}\begin{pmatrix}(ssd_{kl}/n)^2 & -ssd_{kl}/n\\ -ssd_{kl}/n & (ssd_{kl}/n)^2\end{pmatrix}$$

B.3. Determinant of $\mathbf{\Omega}_T$

The determinant of a precision matrix of a GGM with a decomposable graph is expressed as follows (Lauritzen, 1996):

$$|\Omega| = \frac{\prod_{C\in\mathscr{C}}|\Sigma_C|^{-1}}{\prod_{S\in\mathscr{S}}|\Sigma_S|^{-\nu(S)}},$$

where $\Sigma = \Omega^{-1}$. As $\mathbf{\Omega}_T$ is tree-structured, its determinant factorises on the edges of T. It is expressed with the correlation matrix $\boldsymbol{R}_T$ as follows:

$$|\mathbf{\Omega}_T| = \frac{\prod_{kl\in T}|\boldsymbol{R}_{Tkl}|^{-1}}{\prod_k|\boldsymbol{R}_{Tkk}|^{1-d(k)}}$$

Using that $\boldsymbol{R}_T$ has diagonal 1, we obtain for step $t+1$ of the algorithm:

$$|\mathbf{\Omega}_T^{t+1}| = \left(\prod_{kl\in T}|\boldsymbol{R}_{T[kl]}^{t+1}|\right)^{-1}.$$

B.4. Numerical issues

B.4.1 Exact computations

Our algorithm requires the computation of determinants (from the Matrix Tree Theorem) and inverses (in Kirshner's formula) of Laplacian of weight matrices. As we deal with highly variable weights, numerical issues arise: infinite determinants or matrix numerically non-invertible due to either the maximal machine precision (about $1.7 \cdot 10^{308}$), or with machine zero (about $2.2 \cdot 10^{-16}$). To enhance the precision of such computations, we rely on multiple-precision arithmetic which allows the digit of precision of numbers to be limited only by the available memory instead of 64 bits. We implemented matrix inversion and log-determinant computation using both, symbolic computation and multiple precision arithmetic, relying on the `gmp` R package, which uses (Lucas et al., 2020), the C library GMP (GNU Multiple Precision Arithmetic).

B.4.2 Tempering parameter α

Moreover, weights $\tilde{\beta}$ are mechanically linked to the quantity of data available n. To avoid reaching maximal precision when computing the determinant, a tempering parameter α is applied to every quantity proportional to n, so that the actual update performed is

$$\log\tilde{\beta}_{kl} = \log\beta_{kl} - \alpha(\frac{n}{2}\log|\hat{\boldsymbol{R}}_{Tkl}| + \hat{\omega}_{Tkl}[M^{\intercal}M]_{kl}).$$

We provide hereafter a heuristic to set the parameter α. The proposed algorithm requires the computation of the normalising constant $\tilde{B}$, which is the determinant of any minor of the Laplacian of the $q \times q$ variational weights matrix $\tilde{\boldsymbol{\beta}}$. As these weights mechanically increase with the quantity of available data n, this step is numerically very sensitive. Hereafter we denote $|\boldsymbol{Q}^{uv}|$ this determinant and Δ the maximal machine precision. In order to ease the computations, we define the tempering parameter α as

$$\log\tilde{\beta}_{kl} = \log\beta_{kl} - \alpha(\frac{n}{2}\log|\hat{\boldsymbol{R}}_{Tkl}| + \hat{\omega}_{Tkl}[M^{\intercal}M]_{kl}), \quad \text{under constraint} \quad |\boldsymbol{Q}^{uv}| \leq \Delta.$$

Let us first detail the expression for $\tilde{\beta}_{kl}$. Following the definition of the *SSD* matrix, and update formulas (10) and (11), we obtain:

$$\log\tilde{\beta}_{kl} = \log\beta_{kl} + \alpha n \left\{ \frac{(ssd_{kl}/n)^2}{1-(ssd_{kl}/n)^2} - \frac{1}{2}\log[1-(ssd_{kl}/n)^2] \right\}$$

For large n, we thus have

$$\tilde{\beta}_{kl} \approx \exp[\alpha n \cdot f(ssd_{kl}/n)], \quad \text{with } f(x) = x/(1-x) - \log(\sqrt{1-x}),\ x \in [0,\ 1].$$

We then denote $ssd_{max} = \max\{ssd_{kl}/n, k \neq l\}$. By definition, $\boldsymbol{Q}^{uv}$ is positive-definite, so its determinant is upper bounded by the product of its diagonal terms (Hadamard's inequality). Namely:

$$\begin{aligned} |\boldsymbol{Q}^{uv}| &\leq \prod_{i=1}^{q-1} \boldsymbol{Q}^{uv}_{ii} \leq \prod_{i=1}^{q-1}\sum_{i=1}^{q-1} \exp(\alpha n f(ssd_{max})) \\ &\leq \left[(q-1)\exp(\alpha n f(ssd_{max}))\right]^{q-1} \end{aligned}$$

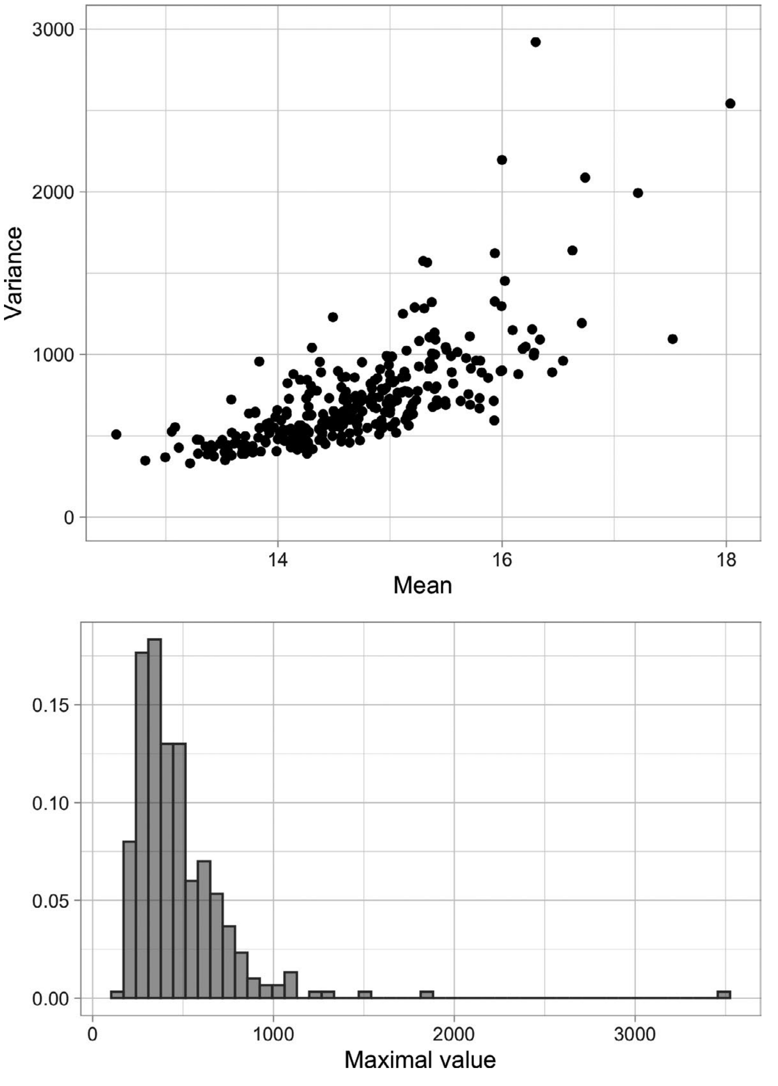

FIGURE 10 Descriptive elements on 400 count Data sets simulated following the protocol of section 1. *Top:* mean-variance relationship. *Bottom:* distribution of the data sets respective maximal value; the mean value is about 470

Then applying the constraint yields:

$$|\boldsymbol{Q}^{uv}| \leq \Delta \Longleftrightarrow \alpha \leq \frac{1}{nf(ssd_{max})}\left[\frac{1}{q-1}\log\Delta - \log(q-1)\right]$$

With $ssd_{max} = 0.8$, $n = 200$ and $q = 15$, we get $\alpha \leq 1.05 \cdot 10^{-1}$.

APPENDIX C

SIMULATED COUNTS DESCRIPTION

Figure 10 gives descriptive details on typical count data sets simulated according to the Poisson log-normal model, as detailed in section 4.1. It shows the varying ranges of the resulting counts and illustrates their clear over-dispersion compared to the Poisson law.

APPENDIX D

MODEL SELECTION AND CROSS-VALIDATION

D.1 Sampling spanning trees

Sampling non-uniform spanning trees (i.e. sampling T from p_β) is a research topic by itself, especially for large networks (see Durfee et al., 2017, for a review). For moderate size networks, a rejection algorithm (Devroye, 1986) can be defined in the following way:

1. Sample T from a distribution q, such that there exist a constant M, that ensures that, for all $T,\ Mq(T) > p_\beta(T)$;
2. Keep T with probability $M^{-1}p_\beta(T)/q(T)$ or try step 1 again.

The efficiency of such an algorithm strongly relies on the choice of the proposal distribution. Here we adopt the following proposal:

1. Sample a connected graph G with independent edges, each drawn with probability $Q_{jk} \propto P_{jk} = \Pr_\beta\{jk \in T\}$;
2. Sample T uniformly among the spanning trees of G.

Evaluation of the proposal. To evaluate the proposal distribution for each sampled tree, we may observe that, the probability for a graph drawn from the proposal to contain a given tree T is approximately

$$\Pr_q\{G \ni T\} \approx \prod_{jk \in T} Q_{jk},$$

the approximation being due to the connectivity constraint. This constraint can be almost surely satisfied by taking Q_{jk}'s large enough. So, denoting $|\mathcal{T}(G)|$ the number of spanning trees in G, we have that

$$q(T) = \sum_{G \ni T} q(T \mid G) q(G) = \sum_{G \ni T} \frac{q(G)}{|\mathcal{T}(G)|} = \Pr_q\{G \ni T\}\ \mathbb{E}\left(|\mathcal{T}(G)|^{-1} | G \ni T\right).$$

The last expectation can be evaluated via Monte Carlo, by sampling a series of graphs G according to the proposal q but forcing all edges from T to be part of G.

Upper bounding constant M. To evaluate the upper bounding constant M, we may observe that finding the tree T^* such that

$$m_\beta := \frac{\Pr_q\{G \ni T^*\}}{p_\beta(T^*)} = \min_{T \in \mathcal{T}} \frac{\Pr_q\{G \ni T\}}{p_\beta(T)} = \min_{T \in \mathcal{T}} \prod_{jk \in T} \frac{Q_{jk}}{\beta_{jk}}$$

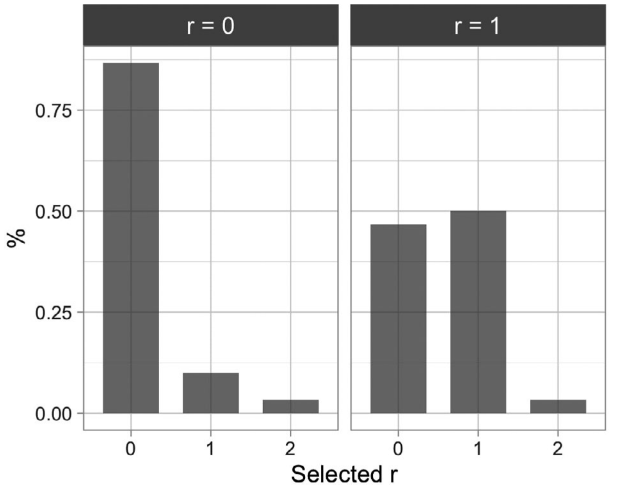

FIGURE 11 Model selection simulation results using the cross-validation procedure described in Section 1. Simulated data sets involve $n = 200$ samples of $p = 14$ species, and $r = 0$ or $r = 1$ missing actor (30 data sets in each category)

is a minimum spanning tree problem. Then, obviously, for any tree T: $\Pr_q\{G \ni T\} \geq m_\beta p_\beta(T)$. Now, because the maximum number of spanning trees within a graph is p^{p-2}, we have

$$Mq(T) = M \sum_{G \ni T} \frac{q(G)}{|\mathcal{T}(G)|} \geq \frac{M}{p^{p-2}} \sum_{G \ni T} q(G) = \frac{M}{p^{p-2}} \Pr_q\{G \ni T\} \geq M \frac{m_\beta}{p^{p-2}} p_\beta(T)$$

So we may set $M = p^{p-2}/m_\beta$. Still, in practice, this bounds turns out to be far too large and needs to be tuned down to preserve computational efficiency.

D.2 Cross-validation for model selection

We conducted a simulation study to assess the performances of the cross-validation procedure described in Section 5.1. We simulated data sets as described in Section 4, with p=14 species and $n = 200$ observations. 30 data sets were simulated with $r = 0$ and another 30 with $r = 1$. Figure 11 shows that when no actor was missing ($r = 0$), the proposed criterion gave the right result for 87% of data sets. When one actor was missing ($r = 1$), the criterion gave the right result for 50% of the data sets. This suggests that using the proposed cross-validated PCL criterion is conservative.

Algorithm 1: Cross-validation for model selection with r missing actors

```
// 0. INITIALIZATION;
```
Divide the dataset $\boldsymbol{Y}$ into V subset $\boldsymbol{Y}^1, \dots \boldsymbol{Y}^V$;

for $v \in \{1, \cdots, V\}$ **do**

  // 1. Apply the VEM algorithm to the train dataset $\boldsymbol{Y}^{-v}$;

  $\boldsymbol{\Gamma}_r^{-v} \leftarrow (\boldsymbol{\theta}_r^{-v}, \boldsymbol{\sigma}_r^{-v}, \boldsymbol{\beta}_r^{-v}, \boldsymbol{\Omega}_r^{-v})$ // 2. MONTE CARLO APPROXIMATION OF COMPLETE LOG-LIKELIHOOD EXPECTATION;

  for $b \in \{1, \cdots, B\}$ **do**

    // 2.1 Draw tree (see Section D.1);

    $T_{r,b}^{-v} \sim p_{\boldsymbol{\beta}_r^{-v}}$

    // 2.2. Build the precision matrix having non-nul entries determined by $T_{r,b}^{-v}$ and values stored in $\boldsymbol{\Omega}_r^{-v}$, and its diagonal terms according to (10);

    $\boldsymbol{\Omega}_{T^b} \leftarrow f(T_{r,b}^{-v}, \boldsymbol{\Omega}_r^{-v})$

    // 2.3. Compute the marginal variance matrix;

    $\boldsymbol{\Sigma}_{T^bO} \leftarrow \boldsymbol{\Omega}_{T^bOO} - \boldsymbol{\Omega}_{T^bOH}\boldsymbol{\Omega}_{T^bHH}^{-1}\boldsymbol{\Omega}_{T^bHO}$;

    // 2.4. Compute the bivariate Poisson log-normal density in test sites;

    for site $i \in v$ **do**

      for pairs of species (j, k) **do**

        $p_{PLN}\left((Y_{ij}^v, Y_{ik}^v); \boldsymbol{\Gamma}_r^{-v}, T_{r,b}^{-v}\right)$ with means $\boldsymbol{x}_i^\intercal \boldsymbol{\theta}_{r,j}^{-v}$ and $\boldsymbol{x}_i^\intercal \boldsymbol{\theta}_{r,k}^{-v}$ and variance matrix $[\boldsymbol{\Sigma}_{T^bO}]_{[jk,jk]}$

    // 2.5. Compute the average;

$$PCL_{rvb}(\boldsymbol{Y}^v, \boldsymbol{\Gamma}_r^{-v}, T^b) = \frac{1}{m_v}\sum_{i=1}^{m_v}\sum_{j<k} \log p_{PLN}\left((Y_{ij}^v, Y_{ik}^v); \boldsymbol{\Gamma}_r^{-v}, T_{r,b}^{-v}\right)$$

// 3. AVERAGE OVER SUBSETS;

$$PCL_r(\boldsymbol{Y}) = \frac{1}{V}\sum_v PCL_{rv}(\boldsymbol{Y}^v, \boldsymbol{\Gamma}_r^{-v}).$$