

Mixture tree model for network inference

Supervision: S. Robin¹ et C. Ambroise¹²

Raphaëlle Momal

¹UMR AgroParisTech / INRA MIA-Paris

²LaMME, Evry

July 9, 2018





Final internship, multi-OMICS data analysis (Jouy)
avril 2017 - octobre 2017

Mathematical statistics Research MSc (Rennes)
2016-2017

Preparation of biostatistician engineer diploma (Rennes)
2014-2017

Prep school MPSI/MP, St Louis highschool (Paris)
2012-2014

Training courses

Cross training 53h

- Basis for teaching
- Journées Carrières en Mathématiques
- Career plan definition (FEM and course)

Pedagogical M2 "Mathématiques pour les Sciences du vivant", Orsay

- Statistiques en grande dimension - C.Giraud
- Models with Hidden Structure - S. RObin

Doctoral mission

50% teaching - 50% scientific mediation:

Teaching Statistical data interpretation with R (26h, L3 MINT Orsay)

Mediation MISS - Maison d'Initiation et de Sensibilisation aux Sciences (12 days per year, 8-15 years-old)

- Mathématiques et botanique
- Histoire des nombres

Talks

This work was presented in :

Research schools:

- Scientific seminar, Rochebrune, 25-30 March
- Chaire MMB (Modélisation Mathématique et Biodiversité), Aussois, 12-16 May

Conference: JOBIM (Journées Ouvertes de Biologie Informatique & Mathématiques), 3-6 July

The subject: preliminaries and general approach

Context

Rising interest in **jointly analysed** species abundances:

- Metagenomics
- Microbiology
- Ecology

Ecological network

Tool to better understand species interactions (direct/indirect), eco-systems organizations (clusters ?)

Allows for resilience analyses, pathogens control, ecosystem comparison, response prediction...

Data example

- **Species:** bacteria, fungi...
- **Abundances:** read counts from Next-Generation Sequencing technologies (metabarcoding)
- **Covariates:** sequencing depth, temperature, water depth...

Repeated signal : n samples, p abundances.

Data table

$$Y = [Y_{ij}]_{(i,j) \in \{1, \dots, n\} \times \{1, \dots, p\}}$$

- Y_{ij} : abundance of the j^{th} species in the i^{th} sample

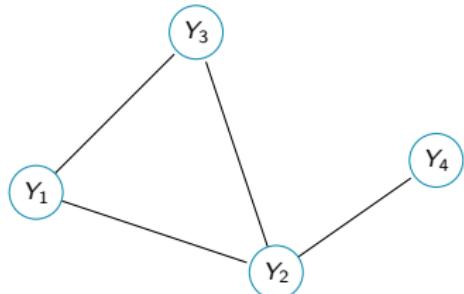
Infer the **species interaction network** from count data Y

Challenges

- Statistical network inference
- Count data
- Offsets and covariates

Graphical models: a statistical framework for network inference

Example:



- All variables are dependant
- Some are **conditionally independent** (i.e. indirectly dependant)

Y_4 is independent from (Y_1, Y_3) conditionally on Y_2

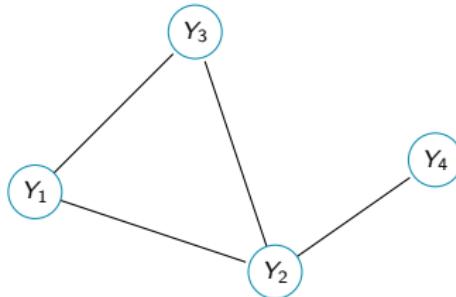
Graphical models

Definition [Lauritzen, 1996]

The joint distribution P is faithful to the graph G iff

$$P(Y_1, \dots, Y_p) \propto \prod_{C \in \mathcal{C}_G} \psi_C(Y_C)$$

where $\mathcal{C}_G = \text{set of maximal cliques of } G$.



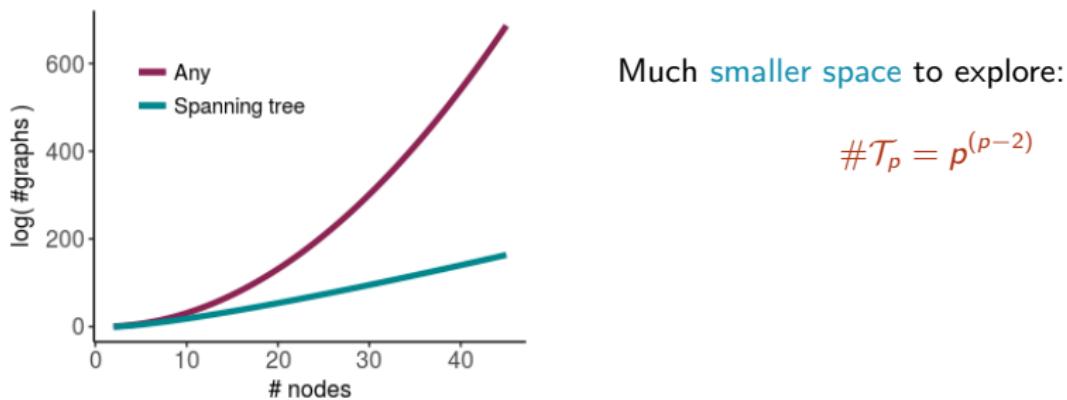
$$P(Y_1, Y_2, Y_3, Y_4) \propto \\ \psi_1(Y_1, Y_2, Y_3) \times \psi_2(Y_3, Y_4)$$

Spanning trees

Unconstrained graph \Rightarrow very large space to explore: $\#\mathcal{G}_p = 2^{\frac{p(p-1)}{2}}$

Spanning trees are a **sparse** solution :

$$\left. \begin{array}{l} G \text{ is connected} \\ G \text{ has no cycle} \end{array} \right\} G \text{ has } (p - 1) \text{ edges}$$

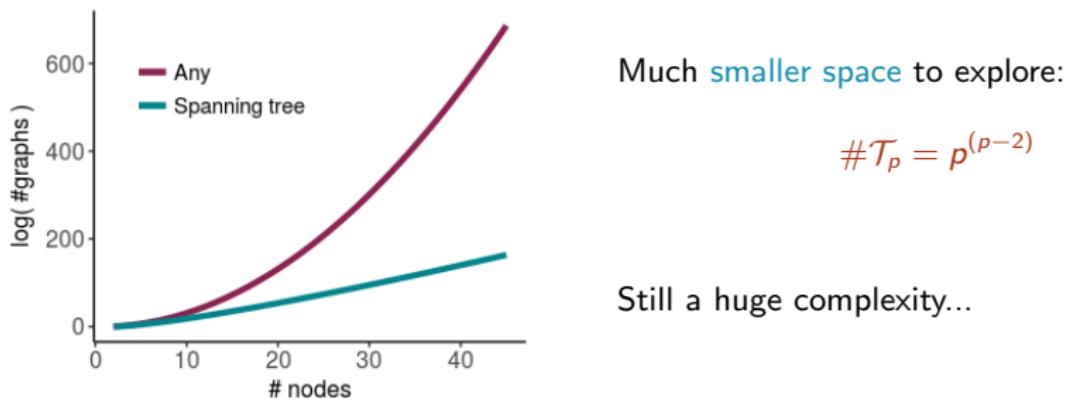


Spanning trees

Unconstrained graph \Rightarrow very large space to explore: $\#\mathcal{G}_p = 2^{\frac{p(p-1)}{2}}$

Spanning trees are a **sparse** solution :

$$\left. \begin{array}{l} G \text{ is connected} \\ G \text{ has no cycle} \end{array} \right\} G \text{ has } (p - 1) \text{ edges}$$



Maximizing and summing over spanning trees

Maximum spanning tree Kruskal's algorithm

$$\hat{T} = \underset{\mathcal{T}}{\operatorname{argmax}} \left\{ \prod_{(k,l) \in \mathcal{T}} \psi_{k,l}(Y) \right\} \rightarrow \Theta(p^2 \log(p))$$

Tree averaging Matrix tree theorem [Chaiken and Kleitman, 1978]

$$\sum_{\mathcal{T}} \prod_{(k,l) \in \mathcal{T}} \psi_{k,l}(Y) = \det(L(Y)) \rightarrow \Theta(p^3)$$

Maximizing and summing over spanning trees

Maximum spanning tree Kruskal's algorithm

$$\hat{T} = \underset{\mathcal{T}}{\operatorname{argmax}} \left\{ \prod_{(k,l) \in \mathcal{T}} \psi_{k,l}(Y) \right\} \rightarrow \Theta(p^2 \log(p))$$

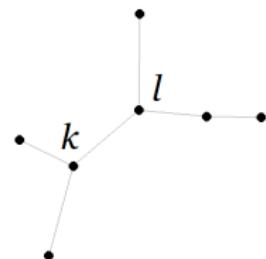
Tree averaging Matrix tree theorem [Chaiken and Kleitman, 1978]

$$\sum_{\mathcal{T}} \prod_{(k,l) \in \mathcal{T}} \psi_{k,l}(Y) = \det(L(Y)) \rightarrow \Theta(p^3)$$

Approach: infer the network by averaging spanning trees

Tree structured data

- Data dependency structure relies on a tree



- Likelihood factorizes on nodes and edges
[Chow and Liu, 1968]:

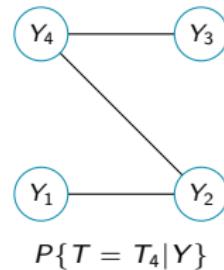
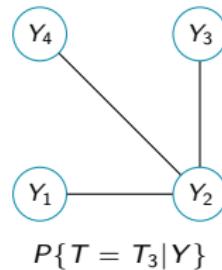
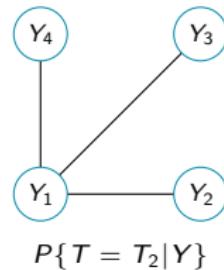
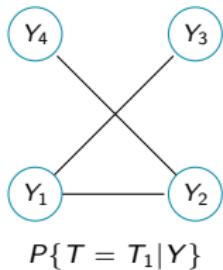
$$\mathbb{P}(Y|T) = \prod_{j=1}^d \mathbb{P}(Y_j) \prod_{k,l \in T} \psi_{kl}(Y) ,$$

Where

$$\psi_{kl}(Y) = \frac{\mathbb{P}(Y_k, Y_l)}{\mathbb{P}(Y_k) \times \mathbb{P}(Y_l)}.$$

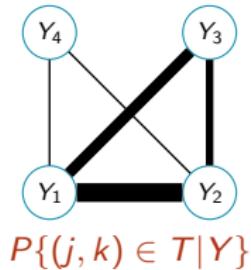
Rmq : with standardised gaussian data, $\hat{\Psi} = [\hat{\psi}_{kl}] \propto (1 - \hat{\rho}^2)^{-1/2}$

Tree averaging

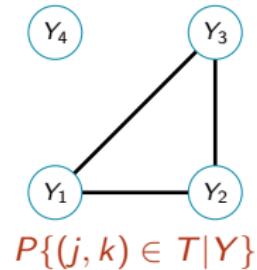


...

Compute edge probabilities:



Thresholding probabilities:



PLN model

Poisson log-Normal distribution [Aitchison and Ho, 1989]

$$\left. \begin{array}{l} Z_i \text{ iid } \sim \mathcal{N}_d(0, \Sigma) \\ (Y_{ij})_j \perp\!\!\!\perp |Z_i \\ Y_{ij}|Z_j \sim \mathcal{P}(e^{Z_{ij}}) \end{array} \right\} Y \sim \mathcal{PLN}(0, \Sigma)$$

- Dependency structure in the Gaussian latent layer
- Easy handling of multi-variate data (contrary to Negative binomial distribution)

PLN model

Poisson log-Normal distribution [Aitchison and Ho, 1989]

$$\left. \begin{array}{l} Z_i \text{ iid } \sim \mathcal{N}_d(0, \Sigma) \\ (Y_{ij})_j \perp\!\!\!\perp | Z_i \\ Y_{ij}|Z_{ij} \sim \mathcal{P}(e^{o_{ij} + x_i^T \Theta_j + Z_{ij}}) \end{array} \right\} Y \sim \mathcal{PLN}(\mathcal{O} + X^T \Theta, \Sigma)$$

- Dependency structure in the Gaussian latent layer
- Easy handling of multi-variate data (contrary to Negative binomial distribution)
- Allow adjustment for covariates and offsets
- Variational estimation algorithm [Chiquet et al., 2017]

PLN + mixture tree

G is taken as a spanning tree T , the dependency structure is encoded in Σ_T .

$$Z_T \sim \mathcal{N}(0, \Sigma_T)$$

Tree averaging (mixture model):

$$Z \sim \sum_T w_T \mathcal{N}(0, \Sigma_T)$$

$$\Rightarrow \mathbb{P}(Z) \propto \sum_T \prod_{k,l \in T} \beta_{k,l} \psi_{k,l}(Z_T)$$

Detailed method

Hierarchical model with latent tree

- 1 A spanning tree is drawn in a distribution decomposable on the edges:

Decomposable distribution for a tree T [Meilă and Jaakkola, 2006]

$$\mathbb{P}(T) = \frac{1}{B} \prod_{(k,l) \in T} \beta_{kl}, \text{ avec } B = \sum_{T \in \mathcal{T}} \prod_{(k,l) \in T} \beta_{kl}$$

- A weight β_{kl} is assigned to each edge (k, l)
- The dependence tree probability is proportional to its weights product
- We consider **varying weights**

Hierarchical model with latent tree

- 1 A spanning tree is drawn in a distribution decomposable on the edges:

Decomposable distribution for a tree T [Meilă and Jaakkola, 2006]

$$\mathbb{P}(T) = \frac{1}{B} \prod_{(k,l) \in T} \beta_{kl}, \text{ avec } B = \sum_{T \in \mathcal{T}} \prod_{(k,l) \in T} \beta_{kl}$$

- A weight β_{kl} is assigned to each edge (k, l)
- The dependence tree probability is proportional to its weights product
- We consider **varying weights**

- 2 Data is simulated conditionally on the drawn tree :

$$Z|T \sim \mathcal{N}_d(0, \Sigma_Z)$$

Hierarchical model with latent tree

- 1 A spanning tree is drawn in a distribution decomposable on the edges:

Decomposable distribution for a tree T [Meilă and Jaakkola, 2006]

$$\mathbb{P}(T) = \frac{1}{B} \prod_{(k,l) \in T} \beta_{kl}, \text{ avec } B = \sum_{T \in \mathcal{T}} \prod_{(k,l) \in T} \beta_{kl}$$

- A weight β_{kl} is assigned to each edge (k, l)
- The dependence tree probability is proportional to its weights product
- We consider **varying weights**

- 2 Data is simulated conditionally on the drawn tree :

$$Z|T \sim \mathcal{N}_d(0, \Sigma_Z)$$

The data shaping tree is treated as a **latent variable**.

$$\mathbb{P}(Z) = \sum_{T \in \mathcal{T}} \mathbb{P}(T) \mathbb{P}(Z|T) : \text{mixture tree}$$

E step

- Complete likelihood :

$$\mathbb{P}(Y, Z, T) = \mathbb{P}(T) \times \mathbb{P}(Z|T) \times \mathbb{P}(Y|Z)$$

$$\begin{aligned} \log(\mathbb{P}(Y, Z, T)) &= \sum_{k,l} \mathbb{1}_{\{(k,l) \in T\}} (\log(\beta_{kl}) + \log(\psi_{kl}(Z))) - \log(B) \\ &\quad + \sum_k (\log(\mathbb{P}(Z_k)) + \log(\mathbb{P}(Y_k|Z_k))) \end{aligned}$$

E step

- Complete likelihood :

$$\mathbb{P}(Y, Z, T) = \mathbb{P}(T) \times \mathbb{P}(Z|T) \times \mathbb{P}(Y|Z)$$

$$\begin{aligned} \log(\mathbb{P}(Y, Z, T)) &= \sum_{k,l} \mathbb{1}_{\{(k,l) \in T\}} (\log(\beta_{kl}) + \log(\psi_{kl}(Z))) - \log(B) \\ &\quad + \sum_k (\log(\mathbb{P}(Z_k)) + \log(\mathbb{P}(Y_k|Z_k))) \end{aligned}$$

- Conditional expectation :

$$\begin{aligned} \mathbb{E}_\theta[\log(\mathbb{P}(Y, Z, T))|Y] &= \sum_{k,l \in V} \mathbb{P}((k,l) \in T|Y) \log(\beta_{kl}) + \mathbb{E}[\mathbb{1}_{\{(k,l) \in T\}} \log(\psi_{kl}(Z)|Y)] \\ &\quad + \sum_k \mathbb{E}[\log(\mathbb{P}(Z_k))|Y] + \mathbb{E}[\log(\mathbb{P}(Y_k|Z_k))|Y] - \log(B) \end{aligned}$$

Two steps solution

The PLNmodels package approximates the distribution parameters. Using PLNmodels:

- 1 Estimate $\hat{\Sigma}_Z$
- 2 Apply EM mixture tree to $Z \sim \mathcal{N}(0, \hat{\Sigma}_Z)$

Simplified conditional expectation writing:

$$\mathbb{E}_{\theta}[\log(\mathbb{P}(Z, T))|Z] = \sum_{k,l} \textcolor{red}{\mathbb{P}((k, l) \in T|Z)} (\log(\beta_{kl}) + \log(\psi_{kl})) - \log(B) + \sum_k \log(\mathbb{P}(Z_k))$$

Conditional probability computation

Kirchhoff's theorem (matrix tree, [Aitchison and Ho, 1989])

For all $W = (a_{kl})_{k,l}$ a symmetric matrix, the corresponding Laplacian $Q(W)$ is defined as follows:

$$Q_{uv}(W) = \begin{cases} -a_{uv} & 1 \leq u < v \leq n \\ \sum_{i=1}^n a_{vi} & 1 \leq u = v \leq n. \end{cases}$$

Then for all u et v :

$$|Q_{uv}^*(W)| = \sum_{T \in \mathcal{T}} \prod_{\{k,l\} \in E_T} a_{kl}$$

$$\begin{aligned} \mathbb{P}((k,l) \in T | Z) &= \sum_{T \in \mathcal{T}: (k,l) \in T} \mathbb{P}(T | Z) = \frac{\sum_{(k,l) \in T} \mathbb{P}(T) \mathbb{P}(Z | T)}{\sum_T \mathbb{P}(T) \mathbb{P}(Z | T)} \\ &= 1 - \frac{|Q_{uv}^*(B\Psi^{-kl})|}{|Q_{uv}^*(B\Psi)|} \\ &= \tau_{kl} \end{aligned}$$

M step

Goal : optimization of weights β_{kl} .

$$\operatorname{argmax}_{\beta_{kl}} \left\{ \sum_{k,l \in V} \tau_{kl} (\log(\beta_{kl}) + \log(\psi_{kl})) - \log(B) + \sum_k \log(\mathbb{P}(Z_k)) \right\}$$

With high combinatorial complexity of $B = \sum_{T \in \mathcal{T}} \prod_{k,l \in T} \beta_{kl}$

How to compute $\frac{\partial B}{\partial \beta_{kl}}$?

β_{kl} update

A result from Meilă [Meilă and Jordan, 2000]

Inverting a minor of the laplacien Q , we define M :

$$\begin{cases} M_{uv} = [Q^{*-1}]_{uu} + [Q^{*-1}]_{vv} - 2[Q^{*-1}]_{uv} & u, v < n \\ M_{nv} = M_{vn} = [Q^{*-1}]_{vv} & v < n \\ M_{vv} = 0. \end{cases}$$

On peut montrer que :

$$\frac{\partial |Q_{uv}^*(W)|}{\partial \beta_{kl}} = M_{kl} \times |Q_{uv}^*(W)|$$

$$\frac{\partial \mathbb{E}_\theta[\log(\mathbb{P}(Z, T))|Z]}{\partial \beta_{kl}} = \frac{1}{\beta_{kl}} \tau_{kl} - \frac{1}{B} \frac{\partial B}{\partial \beta_{kl}}$$

Update formula at iteration $h+1$

$$\hat{\beta}_{kl}^{h+1} = \frac{\tau_{kl}^h}{M_{kl}^h}$$

Simulation design

- 1 Choose G and define Ω accordingly
- 2 Sample count data Y from $\mathcal{PLN}(0, \Omega^{-1})$ with possible covariates
- 3 Infer the network with PLN + mixture tree EM and SpiecEasi
- 4 Compare results with AUC (presence/absence of edges)

⇒ 40 replicates for each setting ($p, n, \text{edge probability}$)

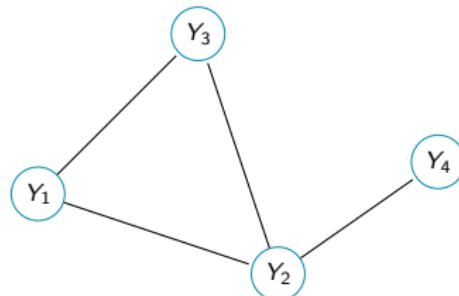
Gaussian Graphical Models (GGM)

Gaussian distribution:

$Y_i \sim \mathcal{N}_p(\mu, \Omega^{-1})$, μ = vector of means, Ω^{-1} = precision matrix.

A nice property:

$$\begin{aligned} p(Y) &\propto \exp(-Y^T \Omega Y / 2) \\ &\propto \prod_{j,k, \omega_{jk} \neq 0} \exp(-Y_j \omega_{jk} Y_k / 2) \end{aligned}$$



Inverse covariance matrix

$$\Sigma^{-1} = \Omega \propto \begin{bmatrix} 1 & .5 & .5 & 0 \\ .5 & 1 & .5 & .5 \\ .5 & .5 & 1 & 0 \\ 0 & .5 & 0 & 1 \end{bmatrix}$$

Graphical LASSO

Glasso on gaussian data:

$$L(Y, \Omega) = \frac{n}{2} \log(\det(\Omega)) - \frac{n}{2} Y^T \Omega Y + cste$$

L1 penalization of the precision matrix:

$$\widehat{\Omega}_{glasso}(\lambda) = \arg \min_{\Omega \in \mathcal{S}_d^+} \left\{ L(Y, \Omega) + \lambda \sum_{i \neq j} |\omega_{ij}| \right\}$$

SpiecEasi method [Kurtz et al., 2015]

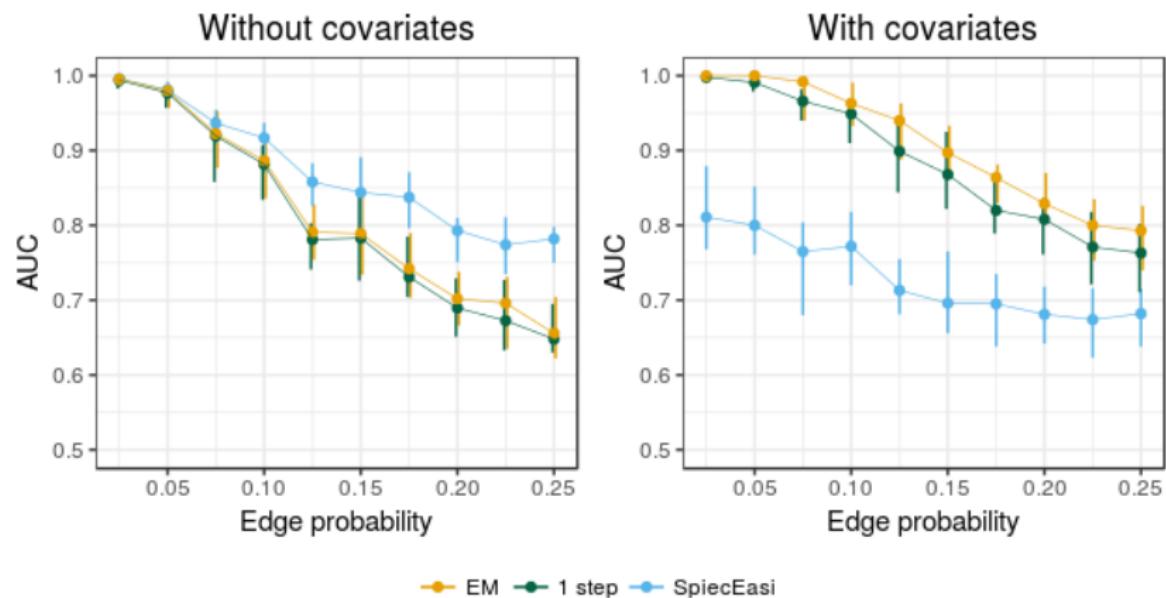
- 1 Count data centered log-ratio transformation:

$$clr(Y) = \left[\log \left(\frac{Y_{ij}}{\left(\prod_{l=1}^p Y_{lj} \right)^{1/p}} \right) \right]_{i,j}$$

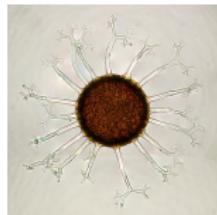
- 2 Glasso on transformed data

Output used : Matrix of first penalization constants removing each edge

AUC curves with Erdös structure, 20 nodes



Oak Mildew



Pathogen *Erysiphe alphitoides* (EA).



Oak leaf with powdery mildew.

Metabarcoding of oak tree leaves microbiome [Jakuschkin et al., 2016].

- 114 sample of 94 microbial species counts (bacteria/fungi)
- Different read depth for bacteria and fungi: unsuited for normalization with SpiecEasi
- 3 quantitative covariates

Model with covariates

F19: a major fungi of the microbiome.

Regression coefficients

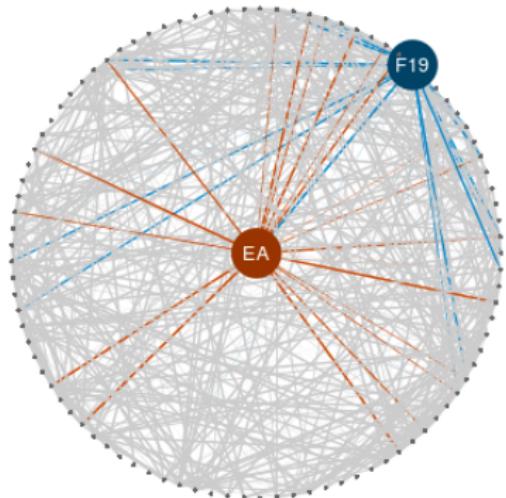
	Covariates ($\times 10^{-2}$)		
	to base	to trunk	to ground
EA	-2.00	2.15	-2.51
F19	2.19	-1.72	1.43

Degree estimation

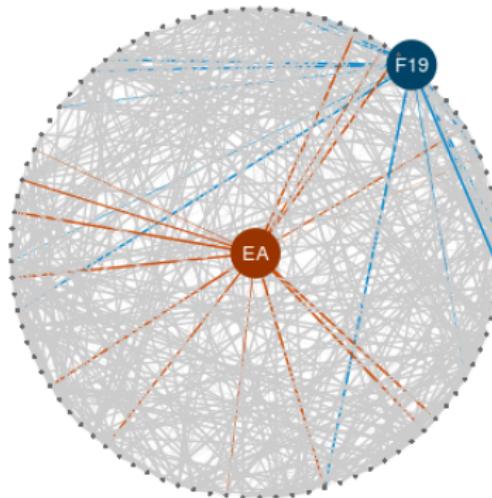
	Offset	Covariates
EA	2.20	1.86
F19	3.03	2.80

Inferred networks

Offset only



With covariates



Further developments

Latent network models

- 2 steps model: $\hat{\Sigma}_Z$ with PLNmodels, and then network inference starting from $\hat{\Sigma}_Z$
 - Formal probabilistic model for network inference with **count data**
 - Estimation algorithm (VEM and then EM)
 - Inclusion of **offsets** and **covariates**
 - Yet estimating with PLNmodels adds some variability
- VEM Model: rewrite the Variational EM used in PLNmodels, to incorporate the latent layer tree dependency structure.
 - Would allow to re-estimate $\hat{\Sigma}_Z$ at each iteration
- PLNnetworks : PLN + glasso

Direct model

The `poilog` R package computes the uni and bi-variate densities of the \mathcal{PLN} distribution

- $\psi_{kl}(Y) = \frac{\mathbb{P}(Y_k, Y_l)}{\mathbb{P}(Y_k) \times \mathbb{P}(Y_l)}$ are **directly accessible**
 \Rightarrow Direct network inference in the Y space

Direct model

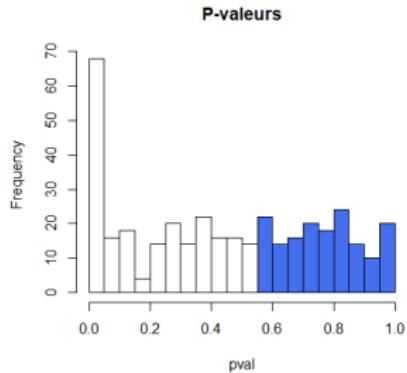
$$\mathbb{P}(Y|T) = \prod_k \mathbb{P}(Y_k) \prod_{kl \in T} \mathbb{P}(Y_k, Y_l)$$

- Metropolis-Hastings simulation strategy (it works)
- No competitor

Method for determining the threshold

$$Y_i \sim Y_{\setminus i}$$

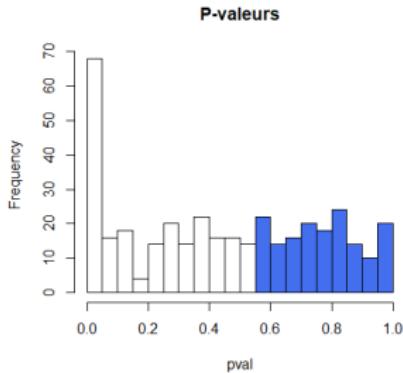
- Adjust each variable on all others
- Interpret coefficients tests in edges presence/absence:
 $(\mathcal{H}_0) : c_{ij} = 0$ vs $(\mathcal{H}_1) : c_{ij} \neq 0$
- Deduce from the p-values an estimator of the "non-edges" number



Method for determining the threshold

$$Y_i \sim Y_{\setminus i}$$

- Adjust each variable on all others
- Interpret coefficients tests in edges presence/absence:
 $(\mathcal{H}_0) : c_{ij} = 0$ vs $(\mathcal{H}_1) : c_{ij} \neq 0$
- Deduce from the p-values an estimator of the "non-edges" number

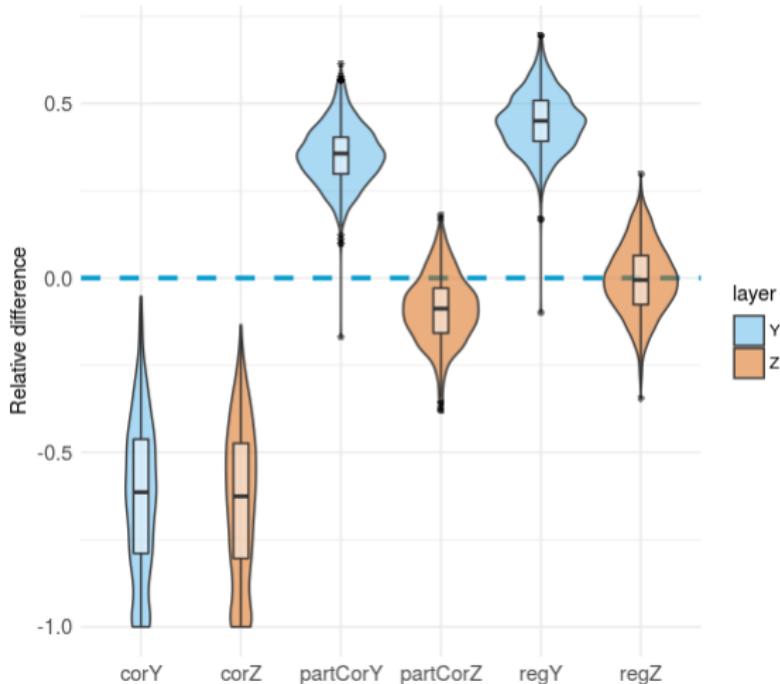


S and S^*

- $S^*(G)$: number of absent edges in graph G
- S^* estimator:

$$S(Y_G) = 2 \sum \mathbb{1}\{pval(Y_G) \geq 0.5\}$$

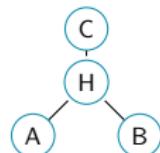
Method for determining the threshold



Switching from Gaussian data Z to count data Y: relative difference between S and S^* , related to tests of regression coefficients, correlations or partial correlations.

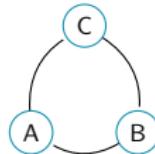
Missing major actor (species/covariable)

Missing a major actor induces spurious edges, and distort the network interpretation
 [Chandrasekaran et al., 2012] :



Complete graph \mathcal{G}

marginalization \rightarrow



Marginal graph \mathcal{G}_m

$$\mathcal{G} \Omega = \underbrace{\begin{pmatrix} \Omega_{OO} & \Omega_{OH} \\ \Omega_{HO} & \Omega_{HH} \end{pmatrix}}_{\text{edges from } E} \quad \Sigma = \begin{pmatrix} \Sigma_{OO} & \Sigma_{OH} \\ \Sigma_{HO} & \Sigma_{HH} \end{pmatrix}$$

$$\mathcal{G}_m : \Omega_m = \underbrace{\Omega_{OO} - \Omega_{OH}\Omega_{HH}^{-1}\Omega_{HO}}_{\text{edges from } E_m} \quad \Sigma_m = \Sigma_{OO}$$

Thank you for your attention!

References |



Aitchison, J. and Ho, C. (1989).

The multivariate Poisson-log normal distribution.

Biometrika, 76(4):643–653.



Chaiken, S. and Kleitman, D. J. (1978).

Matrix tree theorems.

Journal of combinatorial theory, Series A, 24(3):377–381.



Chandrasekaran, V., Parrilo, P. A., and Willsky, A. S. (2012).

Latent variable graphical model selection via convex optimization.

The Annals of Statistics, 40(4):1935–1967.



Chiquet, J., Mariadassou, M., and Robin, S. (2017).

Variational inference for probabilistic Poisson PCA.

Technical report, arXiv:1703.06633.



Chow, C. and Liu, C. (1968).

Approximating discrete probability distributions with dependence trees.

IEEE Transactions on Information Theory, 14(3):462–467.

References II

-  Jakuschkin, B., Fievet, V., Schwaller, L., Fort, T., Robin, C., and Vacher, C. (2016).
Deciphering the pathobiome: Intra- and interkingdom interactions involving the pathogen *erysiphe alphitoides*.
Microb Ecol, 72(4):870–880.
doi:10.1007/s00248-016-0777-x.
-  Kurtz, Z. D., Müller, C. L., Miraldi, E. R., Littman, D. R., Blaser, M. J., and Bonneau, R. A. (2015).
Sparse and compositionally robust inference of microbial ecological networks.
PLoS computational biology, 11(5):e1004226.
-  Lauritzen, S. L. (1996).
Graphical Models.
-  Meilă, M. and Jaakkola, T. (2006).
Tractable bayesian learning of tree belief networks.
Statistics and Computing, 16(1):77–92.
-  Meilă, M. and Jordan, M. I. (2000).
Learning with mixtures of trees.
Journal of Machine Learning Research, 1:1–48.