Inférence de réseaux écologiques à partir d'arbres latents dans un modèle Poisson Log-Normal

Encadré par S. Robin[1] et C. Ambroise[12]

Raphaëlle Momal

[1] UMR AgroParisTech / INRA MIA-Paris
[2] LaMME, Evry

15 mai 2018

Exemple de réseau écologique

[Jakuschkin et al., 2016] :

- But : à partir de mesures d'abondance, identifier les liens de dépendance entre le champignon *E. alphitoïde* présent sur les feuilles du chêne, et les autres micro-organismes présents.

- Utile à la compréhension et au contrôle des maladies chez le chêne.

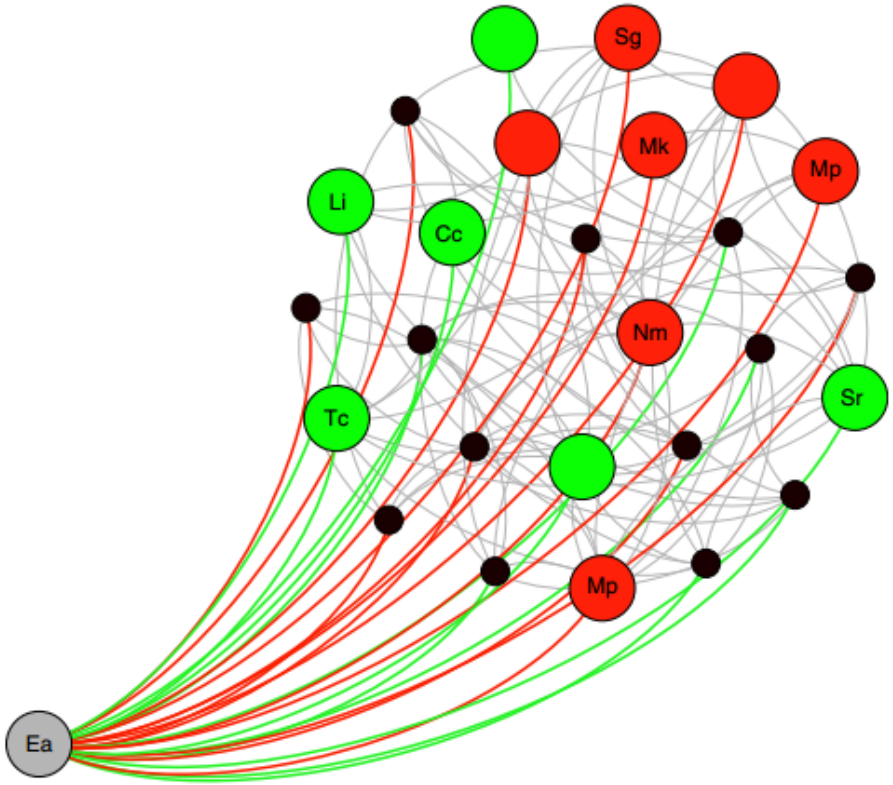
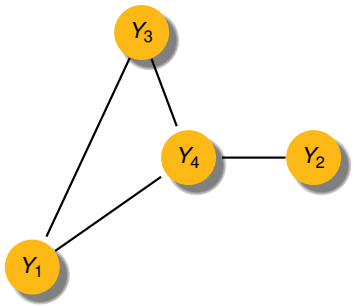


FIGURE – Model of the pathobiome *Erysiphe alphitoides* on oak leaves, source : Jakuschkin *et al.*

Notion de réseau

- Tableau de données observées Y de dimensions $n \times d$
  - abondances de d espèces, expressions RNA-seq de d gènes ...
- Réseau : représentation graphique de la structure de dépendance conditionnelle du jeu de données.
- Inférer un réseau : inférer les arêtes du graphe, *i.e.* la structure de dépendance des variables (espèces, covariables, ...) de Y

Exemple : $Y = (Y_1, ..., Y_4)$:



$\Rightarrow$

Les variables Y_2 et Y_1 sont indépendantes entre elles conditionnellement à la variable Y_4.

Cadre mathématique : modèles graphiques

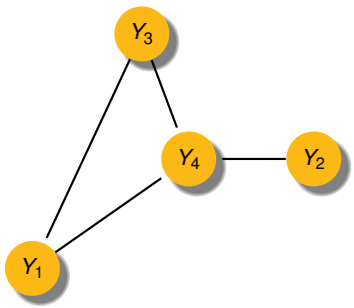
- Clique C d'un graphe G : sous-ensemble de noeuds de G qui sont tous liés entre eux.
- Clique maximale C_G : aucune autre clique de G ne la contient strictement.

Propriété de factorisation [Lauritzen, 1996]

Soit $Y = (Y_1, ..., Y_q)$ et p sa densité. p se factorise selon le graphe non orienté G si :

$$p(y) \propto \prod_{C \in C_G} \Phi_C(y^C)$$

Et alors G représente la structure d'indépendance conditionnelle entre les Y_i.



$$p(Y) = \phi_1(Y_2, Y_4) \times \phi_2(Y_1, Y_3, Y_4)$$

Gaussian Graphical Models (GGM)

Y une variable gaussienne multivariée de dimension d :

$$Y = (Y_1, ..., Y_d) \sim \mathcal{N}_d(0, \Omega^{-1}),$$
$$\Omega = (\omega_{ij})_{i,j}.$$

L'écriture de la gaussienne permet directement d'obtenir une factorisation :

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

Gaussian Graphical Models (GGM)

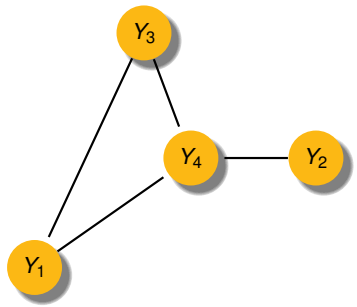
Y une variable gaussienne multivariée de dimension d :

$$Y = (Y_1, ..., Y_d) \sim \mathcal{N}_d(0, \Omega^{-1}),$$
$$\Omega = (\omega_{ij})_{i,j}.$$

L'écriture de la gaussienne permet directement d'obtenir une factorisation :

$$p(y) \propto exp(-y^T \Omega y/2)$$
$$\propto \prod_{j,k,\omega_{jk} \neq 0} exp(-y_j \omega_{jk} y_k/2)$$

$$\Omega = \begin{pmatrix} * & 0 & * & * \\ 0 & * & 0 & * \\ * & 0 & * & * \\ * & * & * & * \end{pmatrix} \Rightarrow$$



Inférence de Ω : le graphical Lasso

- La log-vraisemblance de Y s'écrit :

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

- Estimation parcimonieuse

Le graphical-Lasso (glasso) :

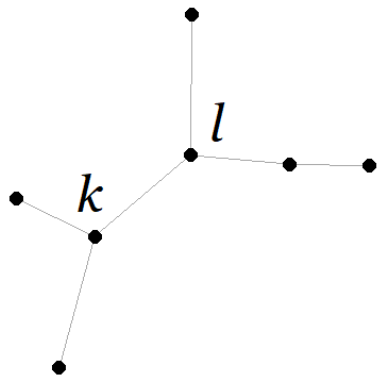
Le graphical-Lasso pénalise la norme l_1 de la matrice de précision :

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

- Choix du λ...

Données structurées par arbre

- La structure de dépendance des données s'appuie sur un arbre



- La vraisemblance des données se factorise sur les noeuds et les arêtes [Chow and Liu, 1968] :

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

Où

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

Rmq : dans le cas gaussien, $\hat{\Psi} = [\hat{\psi_{kl}}] = (1 - \hat{\rho}^2)^{-1/2}$

Loi Poisson Log-Normale (PLN)

La loi Poisson log-Normale

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

- Modélise des comptages
- S'étend facilement aux données multi-vairées (contrairement à la Binomiale Négative)
- Autorise les corrélations négatives
- Permet l'ajustement sur des covariables

Loi Poisson Log-Normale (PLN)

La loi Poisson log-Normale

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

- Modélise des comptages
- S'étend facilement aux données multi-vairées (contrairement à la Binomiale Négative)
- Autorise les corrélations négatives
- Permet l'ajustement sur des covariables

Idée : Inférer le réseau des Z, dans la couche latente gaussienne.

Modèle hiérarchique à arbre latent

1. Un arbre couvrant est tiré dans une loi décomposable sur les arêtes :

Loi décomposable pour un arbre T [Meilă and Jaakkola, 2006]

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

- Un poids β_{kl} est attribué à chaque arête (k, l)
- La probabilité de l'arbre de dépendance est proportionnelle au produit de ses poids.
- Nous considérons les **poids variants**

Modèle hiérarchique à arbre latent

1. Un arbre couvrant est tiré dans une loi décomposable sur les arêtes :

Loi décomposable pour un arbre T [Meilă and Jaakkola, 2006]

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

- Un poids β_{kl} est attribué à chaque arête (k, l)
- La probabilité de l'arbre de dépendance est proportionnelle au produit de ses poids.
- Nous considérons les **poids variants**

2. Les données sont ensuite simulées conditionnellement à l'arbre tiré :

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

Modèle hiérarchique à arbre latent

1. Un arbre couvrant est tiré dans une loi décomposable sur les arêtes :

Loi décomposable pour un arbre T [Meilă and Jaakkola, 2006]

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

- Un poids β_{kl} est attribué à chaque arête (k, l)
- La probabilité de l'arbre de dépendance est proportionnelle au produit de ses poids.
- Nous considérons les **poids variants**

2. Les données sont ensuite simulées conditionnellement à l'arbre tiré :

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

L'arbre qui structure les données est traité comme une **variable latente**.

$$\mathbb{P}(Z) = \sum_{T\in\mathcal{T}} \mathbb{P}(T)\mathbb{P}(Z|T) : \textbf{mélange d'arbres}$$

Étape E

- Vraisemblance complète :

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

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

Étape E

- Vraisemblance complète :

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

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

- Espérance conditionnelle :

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

Solution en deux étapes

Le package `PLNmodels` approche les paramètres de la loi. En utilisant `PLNmodels` :

1. Estimer $\hat{\Sigma}_Z$
2. Appliquer EM par mélange d'arbre à $Z \sim \mathcal{N}(0, \hat{\Sigma}_Z)$

Écriture simplifiée de l'espérance conditionnelle :

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

Calcul de la probabilité conditionnelle

Théorème de Kirchhoff (matrix tree, [Chaiken and Kleitman, 1978])

Pour toute matrice symétrique $W = (a_{kl})_{k,l}$, son Laplacien $Q(W)$ se définit par :

$$\mathcal{Q}_{uv}(W) = \begin{cases} -a_{uv} & 1 \leq u < v \leq n \\ \sum_{w=1}^{n} a_{wv} & 1 \leq u = v \leq n. \end{cases}$$

Alors pour tout u et v :

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

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

Algorithme EM : étape M

But : optimiser les poids β_{kl}.

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

Rappel : $B = \sum_{T \in \mathcal{T}} \prod_{k,l \in T} \beta_{kl}$, complexité combinatoire élevée :

Comment calculer $\frac{\partial B}{\partial \beta_{kl}}$?

Mise à jour des β_{kl}

Résultat de Meila [Meila and Jordan, 2000]

En inversant un mineur du Laplacien Q, on définit la matrice symétrique M :

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

On peut montrer que :

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

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

Formule de mise à jour à l'itération $h+1$

$$\hat{\beta}^{h+1}_{kl} = \frac{\tau^h_{kl}}{M^h_{kl}}$$

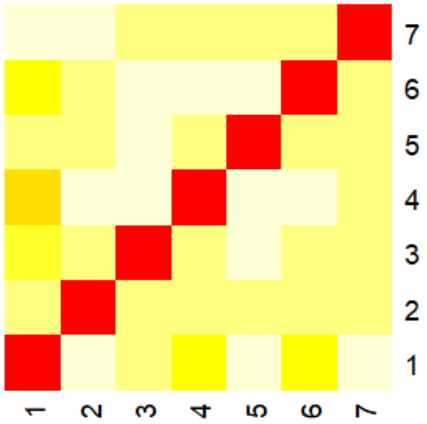
Plan de simulation

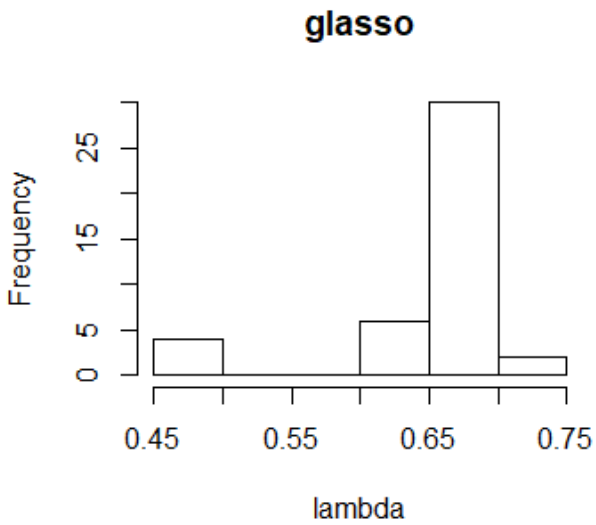
1. Tirer un graphe G
2. À partir de la matrice d'adjacence, construire Ω qui soit défini positive $\Rightarrow \Sigma_Z$
3. Par le package `PLNmodels` : ajuster le modèle de régression PLN à partir de Σ_Z, $\Rightarrow \hat{\Sigma}_Z$
4. Appliquer le glasso et notre EM à $\hat{\Sigma}_Z$
5. Comparer les graphes obtenus et G

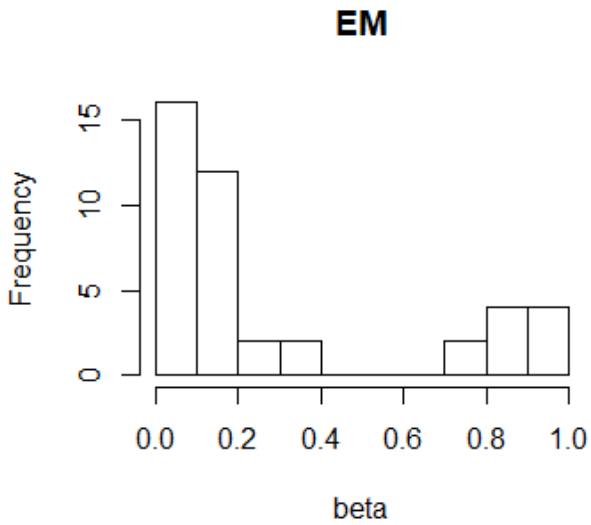
Comparaison des graphes inférés avec G

Les méthodes renvoient des matrices de scores pour chaque arête :

- Glasso : pénalités λ nécessaires pour annuler chacune des arêtes
- EM : poids β_{kl}





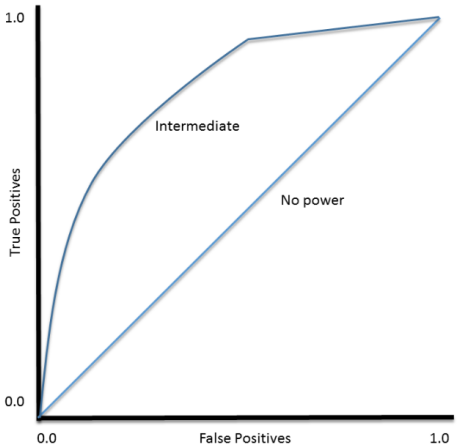


Comparaison des réseaux

- Pour un seuil fixé : arêtes identifiées (vrais positifs), manquées (faux négatifs), ajoutés (faux positif), ou absence d'arête retrouvée (vrai négatif) : construction courbe ROC

Comparaison des réseaux

- Pour un seuil fixé : arêtes identifiées (vrais positifs), manquées (faux négatifs), ajoutés (faux positif), ou absence d'arête retrouvée (vrai négatif) : construction courbe ROC



- Comparer pour tous les seuils : AUC

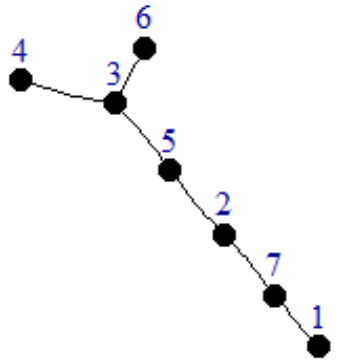


FIGURE – Vrai réseau

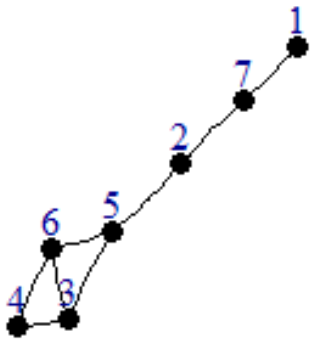
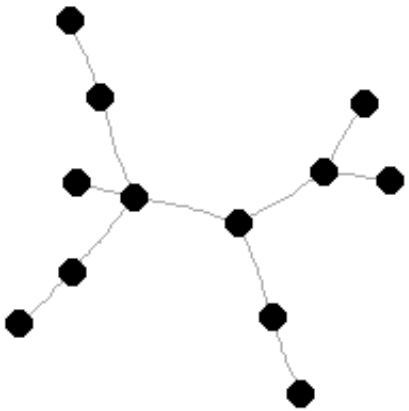


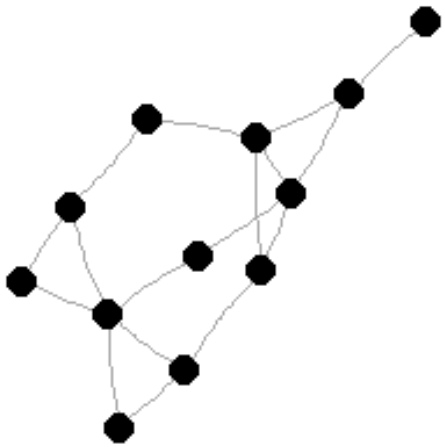
FIGURE – Exemple de réseau inféré par glasso, $\lambda_{\text{seuil}} = 0.68$

Les graphes G

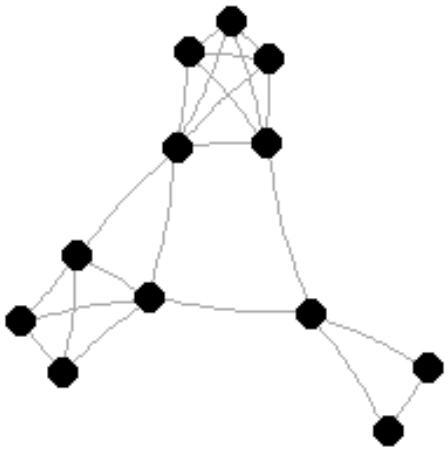
Arbre



Erdos

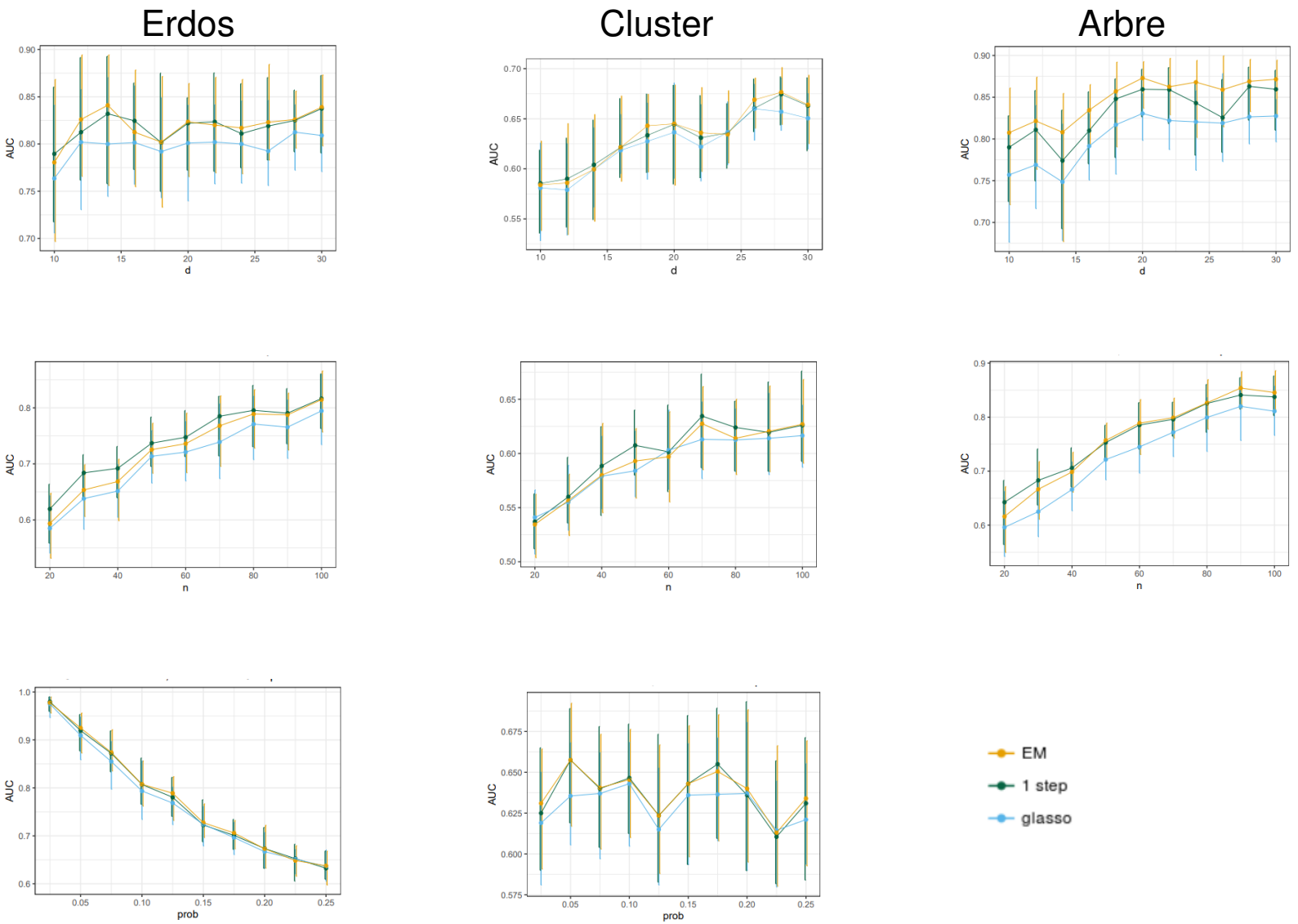


Cluster



Paramètres : nombre de noeuds, densité d'arêtes...

Résultats



Le modèle A2

- Modèle A1 : Inférence du réseau latent en deux étapes ($\hat{\Sigma}_Z$ par `PLNmodels` puis inférence du réseau à partir de $\hat{\Sigma}_Z$)
  - Résultats corrects : meilleur ou équivalent au glasso sur un panel de graphes de type et densité différents
  - facile
  - Mais l'estimation avec `PLNmodels` ajoute de la variabilité

- Modèle A2 : réécrire le Variational EM utilisé dans `PLNmodels`, en y incluant la structure de dépendance par arbre de la couche latente.
  - Permettrait de ré-estimer $\hat{\Sigma}_Z$ à chaque itération

Retour sur la loi PLN

La loi Poisson log-Normale

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

- Le package `poilog` de R calcule les densités uni et bi-variées
  - $\psi_{kl}(Y) = \frac{\mathbb{P}(Y_k, Y_l)}{\mathbb{P}(Y_k) \times \mathbb{P}(Y_l)}$ sont directement accessibles

    $\Rightarrow$ Inférence directe du réseau des espèces dans l'espace de Y ?

Modèle B :

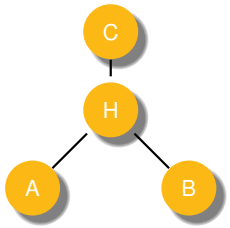
$$\mathbb{P}(Y|T) = \prod_k \mathbb{P}(Y_k) \prod_{kl \in T} \mathbb{P}(Y_k, Y_l)$$

Perspective

- Choix du seuil des matrices de scores
- Mise en oeuvre du modèle A2
- Confrontation des modèles A et B : Modèles différents avec des marginales identiques ?
- Prise en compte d'un acteur manquant

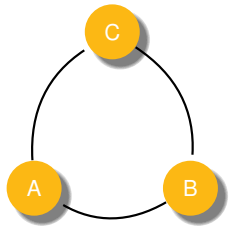
Perspective

- Choix du seuil des matrices de scores
- Mise en oeuvre du modèle A2
- Confrontation des modèles A et B : Modèles différents avec des marginales identiques ?
- Prise en compte d'un acteur manquant



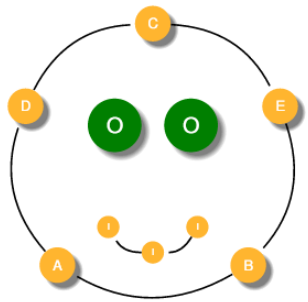
Graphe complet $\mathcal{G}$

marginalisation $\longrightarrow$



Graphe marginal $\mathcal{G}_m$

Merci pour votre attention !




Chaiken, S. and Kleitman, D. J. (1978).
Matrix tree theorems.
Journal of combinatorial theory, Series A, 24(3) :377–381.


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


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.


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.


Meila, M. and Jordan, M. I. (2000).
Learning with mixtures of trees.
Journal of Machine Learning Research, 1 :1–48.