

Développement d'un modèle joint de distribution des espèces pour la réalisation d'une carte de biodiversité à Madagascar

Jeanne Clément

Rapport de stage, Février à Août 2019

Enseignant référent : Benoite De Saporta

Encadrant : Ghislain Vieilledent



Master Maths-Biostatistique

Université Montpellier 2

UMR AMAP - Montpellier



botAnique et Modélisation
de l'Architecture des Plantes et des végétations



Remerciements

J'aimerai adresser mes plus sincères remerciements à G. Vieilledent qui m'a encadrée et conseillée durant ce stage riche en découvertes puisque le langage C++, la construction de packages R ainsi que les modèles joints de distribution des espèces m'étaient inconnus. Il m'a beaucoup appris et encouragée à trouver des solutions par moi même. Je remercie également les chercheurs et autres stagiaires de l'UMR AMAP pour leur accueil chaleureux et leur bonne humeur communicative qui font du laboratoire un cadre de travail idéal et tout particulièrement G. Le Moguedec qui fut une référence précieuse en statistiques ainsi que l'instigateur de pic-niques au lac du Crès qui nous ont bien aidé à supporter la canicule.

Sommaire

Introduction	1
1 Définition des modèles joints de distribution des espèces envisagés	2
1.1 Modèle linéaire mixte généralisé (GLMM)	2
1.2 Modèle à variable latente (LVM)	2
2 Méthodes d'inférence bayesienne selon la fonction de lien choisie	3
2.1 Principe d'un échantillonneur de Gibbs	3
2.2 Modèle probit : échantillonneur de Gibbs et priors conjugués	3
2.2.1 Définition du modèle probit	3
2.2.2 Priors utilisés	4
2.2.3 Propositions sur les priors conjugués	4
2.2.4 Echantillonneur de Gibbs et priors conjugués	7
2.3 Modèle logit : échantillonneur de Gibbs et algorithme de Metropolis adaptatif	8
2.3.1 Définition du modèle logit	8
2.3.2 Priors utilisés	8
2.3.3 Principe d'un algorithme de Metropolis adaptatif	8
2.3.4 Echantillonneur de Gibbs et algorithme de Metropolis adaptatif	9
2.4 Evaluation de la fiabilité de ces méthodes sur des données simulées	11
2.4.1 Simulation des données	11
2.4.2 Représentation des paramètres estimés	11
2.4.3 Evaluation du temps de calcul et de la pertinence des résultats	14
3 Comparaison des packages boral et jSDM	15
3.1 Performances sur des données simulées	15
3.2 Comparaison des résultats sur des jeux de données réels	15
3.2.1 Jeux de données utilisés	15
3.2.2 Comparaison des résultats	15
4 Application aux données collectées à Madagascar	16
4.1 Description des données	16
4.2 Estimation des paramètres et représentation des résultats	17
4.2.1 Représentation des résultats pour évaluer la convergence de l'algorithme	17
4.2.2 Représentations spatiales des paramètres associés aux sites et des probabilités de présence estimées	21
4.2.3 Estimation de la richesse spécifique propre à chaque site et comparaison à celle observée	22
4.3 Comparaison des effets sites obtenus avec différentes méthodes d'interpolation spatiale	23
4.3.1 Thin plate spline	23
4.3.2 Ordinary kriging	23
4.3.3 Inverse distance weighted	23
4.3.4 Comparaison des méthodes par validation croisée sur les effets sites estimés	23
4.4 Résultats de l'interpolation spatiale par krigeage ordinaire	25
4.5 Analyse des résultats et mise en évidence de lieux refuges de la biodiversité	26
4.6 Prédictions avec auto-corrélation spatiale	26
Conclusion	27
Bibliographie	27
Annexe	28

Liste des figures

1	Représentation des $(\alpha_i)_{i=1,\dots,500}$ estimés en fonction de ceux simulés pour le modèle probit	11
2	Représentation des $(W_{il})_{i=1,\dots,500}^{l=1,2}$ estimés en fonction de ceux simulés pour le modèle probit	12
3	Représentation des $(\beta_{jk})_{j=1,\dots,100}^{k=1,2,3}$ et $(\lambda_{jl})_{j=1,\dots,100}^{l=1,2}$ estimés en fonction de ceux simulés pour le modèle probit	12
4	Représentation des $(\text{probit}(\theta_{ij}), Z_{ij}, \theta_{ij})_{i=1,\dots,500}^{j=1,\dots,100}$ estimés en fonction de ceux simulés pour le modèle probit	12
5	Représentation des $(\alpha_i)_{i=1,\dots,500}$ estimés en fonction de ceux simulés pour le modèle logit	13
6	Représentation des $(W_{il})_{i=1,\dots,500}^{l=1,2}$ estimés en fonction de ceux simulés pour le modèle logit	13
7	Représentation des $(\beta_{jk})_{j=1,\dots,100}^{k=1,2,3}$ et $(\lambda_{jl})_{j=1,\dots,100}^{l=1,2}$ estimés en fonction de ceux simulés pour le modèle logit	13
8	Représentation des $(\text{logit}(\theta_{ij}), \theta_{ij})_{i=1,\dots,500}^{j=1,\dots,100}$ estimés en fonction de ceux simulés pour le modèle probit	14
9	Traces et densités de la déviance du modèle	17
10	Traces et densités des effets espèces fixes $(\beta_{jk})_{j=1}^{k=0,\dots,10}$ estimés pour la deuxième espèce	18
11	Traces et densités des effets espèces fixes $(\lambda_{jq})_{j=1,2}^{q=1,2}$ estimés pour les deux premières espèces	19
12	Traces et densités des variables latentes W_1 et W_2 estimées pour un site	20
13	Trace et densité d'un effet site et de la variance associée aux effets sites estimés	20
14	Représentation spatiale des variables latentes W_1 et W_2 estimées pour chaque site	21
15	Représentation spatiale des effets sites $(\alpha_i)_{i=1,\dots,753}$ estimés	21
16	Représentation spatiale des probabilités de présence estimées pour l'espèce Ocotea laevis	22
17	Représentation spatiale des occurrences observées de l'espèce Ocotea laevis	22
18	Représentation spatiale de la richesse spécifique estimée pour chaque site	22
19	Représentation spatiale de la richesse spécifique observée pour chaque site	22
20	Représentation richesse spécifique estimée en fonction de celle observée	23
21	Comparaison des cartes obtenues pour l'effet site par différentes méthodes d'interpolation	24
22	Interpolation spatiale des effets sites par krigeage ordinaire	25
23	Interpolation spatiale des variables latentes par krigeage ordinaire	25

Liste des tableaux

1	Temps de calcul nécessaires à l'ajustement des modèles logit et probit ainsi que les NRMSE (Normalized Root Mean Square Error) et les pourcentages de déviance expliquée obtenus	14
2	Variables bioclimatiques considérées	16
3	Temps de calcul nécessaire à l'ajustement du modèle sur les données de Madagascar et nombre de paramètres à estimer	17

Introduction

J'ai effectué mon stage au sein de l'UMR AMAP (botAnique et Modélisation de l'Architecture des Plantes et des végétations), qui se trouve à Montpellier. Il s'agit d'une unité interdisciplinaire hébergée par le Cirad ou « Centre de Coopération Internationale en Recherche Agronomique pour le Développement», qui mène des recherches sur les plantes et les végétations, dans le but de prévoir la réponse des écosystèmes aux forçages environnementaux.

Ce stage s'inscrit dans le cadre du projet BioSceneMada qui vise à fournir des scénarios d'évolution de la biodiversité sous l'effet conjoint du changement climatique et de la déforestation à Madagascar. Pour ce faire, plusieurs jeux de données sur la biodiversité ont été collectés et regroupés pour différents groupes taxonomiques (mammifères, oiseaux, reptiles, amphibiens, arbres, plantes herbacées, invertébrés), parmi lesquels j'ai utilisé des inventaires forestiers répertoriant l'absence ou la présence d'espèces d'arbres sur différents sites de l'île ainsi que des variables bioclimatiques afin d'ajuster un modèle joint de distribution des espèces permettant d'estimer la niche des espèces, de prédire leur distribution, tout en prenant en compte les interactions entre espèces (Warton et al. (2015)).

Dans un premier temps j'ai implémenté différents échantillonneurs de Gibbs en C++ permettant d'estimer les paramètres de modèles joints de distribution des espèces (JSDM) comportant des variables latentes, à l'aide du package Rcpp. La construction du package R <https://ecology.ghislainv.fr/jSDM/> autour de l'une de ces fonctions ainsi que sa présentation à la conférence useR 2019 ont constitué une partie importante de mon stage. L'ajustement d'un JSDM sur des données d'inventaires forestiers collectées à Madagascar ainsi que des variables climatiques et environnementales, m'a permis d'obtenir des cartes reflétant la biodiversité sur l'île afin de par la suite identifier des zones refuges de la biodiversité sous l'effet du changement climatique en utilisant les variables bioclimatiques fournies par les scénarios du GIECC. Ces résultats seront utilisés pour des préconisations de gestion de la biodiversité dans le cadre du projet BioSceneMada.

1 Définition des modèles joints de distribution des espèces envisagés

Les données dont on dispose pour ajuster ce type de modèle sont les réalisations d'une variable réponse, $Y = (y_{ij})_{j=1,\dots,J}^{i=1,\dots,I}$ telle que :

$$y_{ij} = \begin{cases} 0 & \text{si l'espèce } j \text{ est absente du site } i \\ 1 & \text{si l'espèce } j \text{ est présente sur le site } i, \end{cases}$$

ainsi que de variables explicatives $X = (X_i)_{i=1,\dots,I}$ avec $X_i = (X_{i1}, \dots, X_{ip}) \in \mathbb{R}^p$ où p est le nombre de variables bioclimatiques considérées pour chaque site.

On note θ_{ij} , la probabilité de présence de l'espèce j sur le site i .

L'article Warton et al. (2015) développe deux approches hiérarchiques pouvant être utilisées à la spécification d'un modèle joint de distribution des espèces.

1.1 Modèle linéaire mixte généralisé (GLMM)

D'une part on pourrait utiliser un modèle linéaire mixte généralisé (**GLMM**) de la forme :

$$\begin{aligned} g(\theta_{ij}) &= \alpha_i + \beta_{j0} + X_i \beta_j + u_{ij}, \\ y_{ij} \mid u_{ij}, \alpha_i &\sim \text{Bernoulli}(\theta_{ij}), \\ u_i &\sim \mathcal{N}_J(0_{\mathbb{R}^J}, \Sigma) \text{ iid}, \\ \alpha_i &\sim \mathcal{N}(0, V_\alpha) \text{ iid et indépendant de } u_i. \end{aligned}$$

où $g :]0, 1[\rightarrow]-\infty, +\infty[$ est une fonction de lien, $\beta_j = (\beta_{j1}, \dots, \beta_{jp})'$ et β_{j0} sont les coefficients de régression correspondants aux variables bioclimatiques et l'intercept pour l'espèce j qui est supposé être un effet fixe, α_i représente l'effet aléatoire du site i , et $u_i = (u_{i1}, \dots, u_{iJ})$ est un effet aléatoire multivarié corrélé dont la matrice de variance covariance Σ contrôle la corrélation entre les espèces et est supposée être complètement non structurée.

Cette dernière partie du modèle est problématique lorsque le nombre d'espèces J est important car le nombre de paramètres dans Σ augmente quadratiquement avec J .

1.2 Modèle à variable latente (LVM)

D'autre part en posant $u_{ij} = W_i \lambda_j$, avec $W_i = (W_{i1}, \dots, W_{iq})$ les q variables latentes (ou "variables latentes") considérées et $\lambda_j = (\lambda_{j1}, \dots, \lambda_{jq})'$ les coefficients associés, on obtient le modèle à variables latentes (**LVM**) suivant :

$$\begin{aligned} g(\theta_{ij}) &= \alpha_i + \beta_{j0} + X_i \beta_j + W_i \lambda_j \\ y_{ij} \mid W_i, \alpha_i &\sim \text{Bernoulli}(\theta_{ij}), \\ W_i &\sim \mathcal{N}(0, I_q) \text{ iid} \\ \alpha_i &\sim \mathcal{N}(0, V_\alpha) \text{ iid et indépendant de } W_i \end{aligned}$$

Ce qui revient à un cas particulier de GLMM multivarié auquel on impose la contrainte $\Sigma = \Lambda \Lambda'$ avec

$$\Lambda := \begin{pmatrix} \lambda_{11} & \dots & \lambda_{1q} \\ \vdots & \ddots & \vdots \\ \lambda_{J1} & \dots & \lambda_{Jq} \end{pmatrix}$$

On préférera ce dernier modèle, en effet il comporte potentiellement beaucoup moins de paramètres que le GLMM précédent car Λ a autant de colonne qu'il y a de variables latentes (q) tandis que Σ présente autant de colonnes de paramètres qu'il y a d'espèces (J).

On peut choisir de modéliser l'abondance absolue plutôt que l'abondance relative en supprimant les effets sites aléatoires α_i du modèle.

2 Méthodes d'inférence bayesienne selon la fonction de lien choisie

2.1 Principe d'un échantillonneur de Gibbs

Dans le cadre bayésien, l'algorithme de Gibbs permet d'obtenir une réalisation du paramètre $\theta = (\theta_1, \dots, \theta_m)$ suivant la loi *a posteriori* $\pi(\theta | x)$ dès que l'on est capable d'exprimer les lois conditionnelles : $\pi(\theta_i | \theta_1, \dots, \theta_{i-1}, \theta_{i+1}, \dots, \theta_m, x)$ pour $i = 1, \dots, m$.

L'échantillonnage de Gibbs consiste à :

- **Initialisation** : choix arbitraire de $\theta^{(0)} = (\theta_1^{(0)}, \dots, \theta_m^{(0)})$.
- **Itération t** : Générer $\theta^{(t)}$ de la manière suivante :
 - $\theta_1^{(t)} \sim \pi(\theta_1 | \theta_2^{(t-1)}, \dots, \theta_m^{(t-1)}, x)$
 - $\theta_2^{(t)} \sim \pi(\theta_2 | (\theta_1^{(t)}, \theta_3^{(t-1)}, \dots, \theta_m^{(t-1)}, x))$
 - $\theta_m^{(t)} \sim \pi(\theta_m | \theta_1^{(t)}, \dots, \theta_{m-1}^{(t)}, x)$

Les itérations successives de cet algorithme génèrent les états d'une chaîne de Markov $\{\theta^{(t)}, t > 0\}$ à valeurs dans \mathbb{R}^m , on montre que cette chaîne admet une mesure invariante qui est la *loi a posteriori*.

Pour un nombre d'itérations suffisamment grand, le vecteur θ obtenu peut donc être considéré comme étant une réalisation de la loi *a posteriori* jointe $\pi(\theta | x)$.

Par conséquent l'implémentation d'un échantillonneur de Gibbs nécessite la connaissance des distributions *a posteriori* de chacun des paramètres conditionnellement aux autres paramètres du modèle, qui se déduisent des formules de priors conjugués dans le cas du modèle probit mais ne sont pas explicitement exprimables dans le cas où on utilise une fonction de lien logit.

2.2 Modèle probit : échantillonneur de Gibbs et priors conjugués

D'une part, on utilise un fonction de lien probit : $p \rightarrow \Phi^{-1}(p)$ où Φ correspond à la fonction de répartition d'une loi normale centrée réduite.

2.2.1 Définition du modèle probit

D'après l'article Albert and Siddhartha (1993), une modélisation possible est de supposer l'existence d'une variable latente sous-jacente liée à notre variable binaire observée en utilisant la proposition suivante :

Proposition 2.2.1.1 (Modèle probit par l'intermédiaire d'une variable latente).

Si $Z_{ij} = \alpha_i + \beta_{j0} + X_i \beta_j + W_i \lambda_j + \epsilon_{ij}$, $\forall i, j$ avec $\epsilon_{ij} \sim \mathcal{N}(0, 1)$ iid et tel que :

$$y_{ij} = \begin{cases} 1 & \text{si } Z_{ij} > 0 \\ 0 & \text{sinon.} \end{cases}$$

Alors on a $y_{ij} | Z_{ij} \sim \text{Bernoulli}(\theta_{ij})$ avec $\text{probit}(\theta_{ij}) = \alpha_i + \beta_{j0} + X_i \beta_j + W_i \lambda_j$.

Preuve 2.2.1.1.

$$\begin{aligned} \mathbb{P}(y_{ij} = 1) &= \mathbb{P}(Z_{ij} > 0) \\ &= \mathbb{P}(\alpha_i + \beta_{j0} + X_i \beta_j + W_i \lambda_j + \epsilon_{ij} > 0) \\ &= \mathbb{P}(\epsilon_{ij} > -(\alpha_i + \beta_{j0} + X_i \beta_j + W_i \lambda_j)) \\ &= \mathbb{P}(\epsilon_{ij} \leq \alpha_i + \beta_{j0} + X_i \beta_j + W_i \lambda_j) \\ &= \Phi(\alpha_i + \beta_{j0} + X_i \beta_j + W_i \lambda_j) \end{aligned}$$

De la même façon on a :

$$\begin{aligned}\mathbb{P}(y_{ij} = 0) &= \mathbb{P}(Z_{ij} \leq 0) \\ &= 1 - \Phi(\alpha_i + \beta_{j0} + X_i \beta_j + W_i \lambda_j)\end{aligned}$$

On définit le modèle probit à l'aide d'une variable latente afin d'être en mesure d'utiliser les propriétés des priors conjugués pour échantillonner les paramètres du modèle selon leurs distributions conditionnelles *a posteriori*.

2.2.2 Priors utilisés

Afin d'utiliser une méthode d'inférence bayesienne on détermine une distribution *a priori* pour chacun des paramètres du modèle :

$$\begin{aligned}V_\alpha &\sim \mathcal{IG}(\text{shape} = 0.5, \text{rate} = 0.005) \text{ avec rate} = \frac{1}{\text{scale}}, \\ \beta_{jk} &\sim \mathcal{N}(0, 10^6) \text{ pour } j = 1, \dots, J \text{ et } k = 1, \dots, p, \\ \lambda_{jl} &\sim \begin{cases} \mathcal{N}(0, 10) & \text{si } l < j \\ \mathcal{N}(0, 10) \text{ tronquée à gauche par 0} & \text{si } l = j \\ P \text{ tel que } \mathbb{P}(\lambda_{jl} = 0) = 1 & \text{si } l > j \end{cases} \\ &\text{pour } j = 1, \dots, J \text{ et } l = 1, \dots, q.\end{aligned}$$

En effet pour assurer l'identifiabilité du modèle les valeurs de Λ sont contraintes à des valeurs strictement positives sur la diagonale et nulles au dessus de celle-ci, Λ est ainsi supposée être triangulaire inférieure d'après l'article Warton et al. (2015).

La fonction *boral()* du package du même nom, permettant d'ajuster différents modèles utilise ces distributions *a priori* pour le modèle qui nous intéresse. Dans l'article Warton et al. (2015) l'ajustement de modèles joints de distributions des espèces est réalisé à l'aide de boral qui fonctionne avec JAGS (Just Another Gibbs Sampler) un programme de simulation à partir de modèles hiérarchiques bayésiens utilisant des méthodes MCMC, implémenté en C++. Cependant la fonction *jSDM_probit_block()* du package jSDM que j'ai implémentée utilise une distribution *a priori* jointe pour les effets espèces fixes de la manière qui suit.

2.2.3 Propositions sur les priors conjugués

Effets espèces fixes :

On se ramène à un modèle de la forme $Z^* = X\beta + \epsilon$, en posant $Z_{ij}^* = Z_{ij} - \alpha_i = \beta_{j0} + X_i \beta_j + W_i \lambda_j + \epsilon_{ij}$, afin d'estimer simultanément les β_j et λ_j pour chacune des espèces j , ce qui revient en écriture matricielle à :

$$\begin{aligned}Z_j^* &= \begin{pmatrix} Z_{1j}^* \\ \vdots \\ Z_{Ij}^* \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & X_{11} & \dots & X_{1p} & W_{11} & \dots & W_{1q} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & X_{I1} & \dots & X_{Ip} & W_{I1} & \dots & W_{Iq} \end{pmatrix}}_D \underbrace{\begin{pmatrix} \beta_{j0} \\ \beta_{j1} \\ \vdots \\ \beta_{jp} \\ \lambda_{j1} \\ \vdots \\ \lambda_{jq} \end{pmatrix}}_{P_j} + \begin{pmatrix} \epsilon_{1j} \\ \vdots \\ \epsilon_{Ij} \end{pmatrix} \\ &= DP_j + \epsilon_j \quad \text{avec } \epsilon_j \sim \mathcal{N}_I(0_{\mathbb{R}^I}, I_I).\end{aligned}$$

On suppose que $P_j \sim \mathcal{N}_{p+q+1}(m, V)$ avec $m = 0_{\mathbb{R}^{p+q+1}}$ et $V = \text{diag}(\underbrace{10^6, \dots, 10^6}_{\times p+1}, \underbrace{10, \dots, 10}_{\times q})$, par exemple.

Bien que cette distribution *a priori* ne prenne pas en compte les contraintes sur Λ , elle permet l'échantillonnage selon une loi normale multivariée des effets espèce fixes. On imposera les contraintes aux λ_{jl} concernés après les avoir simulés.

On applique la proposition suivante :

Proposition 2.2.3.1.

$$\begin{cases} Y \mid \beta \sim \mathcal{N}_n(X\beta, I_n) \\ \beta \sim \mathcal{N}_p(m, V) \end{cases} \Rightarrow \begin{cases} \beta \mid Y \sim \mathcal{N}_p(m^*, V^*) \text{ avec} \\ m^* = (V^{-1} + X'X)^{-1}(V^{-1}m + X'Y) \\ V^* = (V^{-1} + X'X)^{-1} \end{cases}$$

Preuve 2.2.3.1.

$$\begin{aligned}
p(\beta \mid Y) &\propto p(Y \mid \beta) p(\beta) \\
&\propto \frac{1}{(2\pi)^{\frac{n}{2}}} \exp\left(-\frac{1}{2}(Y - X\beta)'(Y - X\beta)\right) \frac{1}{(2\pi)^{\frac{p}{2}}|V|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\beta - m)'V^{-1}(\beta - m)\right) \\
&\propto \exp\left(-\frac{1}{2}((\beta - m)'V^{-1}(\beta - m) + (Y - X\beta)'(Y - X\beta))\right) \\
&\propto \exp\left(-\frac{1}{2}(\beta'V^{-1}\beta + m'V^{-1}m - m'V^{-1}\beta - \beta'V^{-1}m + Y'Y + \beta'X'X\beta - Y'X\beta - \beta'X'Y)\right) \\
&\propto \exp\left(-\frac{1}{2}(\beta'(V^{-1} + X'X)\beta - \beta'(V^{-1}m + X'Y) - (Y'X + m'V^{-1})\beta + m'V^{-1}m + Y'Y)\right) \\
&\propto \exp\left(-\frac{1}{2}(\beta'(V^{-1} + X'X)\beta - \beta'(V^{-1}m + X'Y) - (X'Y + V^{-1}m)\beta + m'V^{-1}m + Y'Y)\right) \\
&\propto \exp\left(-\frac{1}{2}(\beta - (V^{-1} + X'X)^{-1}(V^{-1}m + X'Y))'(V^{-1} + X'X)(\beta - (V^{-1} + X'X)^{-1}(V^{-1}m + X'Y))\right. \\
&\quad \left.- (V^{-1}m + X'Y)'(V^{-1} + X'X)^{-1}(V^{-1}m + X'Y) + m'V^{-1}m + Y'Y\right) \\
&\propto \exp\left(-\frac{1}{2}\left(\beta - \underbrace{(V^{-1} + X'X)^{-1}(V^{-1}m + X'Y)}_{m^*}\right)' \underbrace{(V^{-1} + X'X)}_{V^{*-1}} (\beta - (V^{-1} + X'X)^{-1}(V^{-1}m + X'Y))\right)
\end{aligned}$$

On obtient :

$$\begin{cases} Z_j^* \mid P_j \sim \mathcal{N}_I(DP_j, I_I) \\ P_j \sim \mathcal{N}_{p+q+1}(m, V) \end{cases} \Rightarrow \begin{cases} P_j \mid Z_j^* \sim \mathcal{N}_{p+q+1}(m^*, V^*) \text{ avec} \\ m^* = (V^{-1} + D'D)^{-1}(V^{-1}m + D'Z_j^*) \\ V^* = (V^{-1} + D'D)^{-1} \end{cases}$$

Variables latentes (prédicteurs non mesurés) :

De la même façon, on pose : $Z_{ij}^* = Z_{ij} - \alpha_i - \beta_{j0} - X_i\beta_j = W_i\lambda_j + \epsilon_{ij}$, afin d'estimer W_i pour chaque site i . En appliquant la proposition précédente, on obtient :

$$\begin{cases} Z_i^* := (Z_{i1}^*, \dots, Z_{iJ}^*)' \mid W_i \sim \mathcal{N}_J(\Lambda W'_i, I_J) \\ W'_i \sim \mathcal{N}_q(0_{\mathbb{R}^q}, I_q) \end{cases} \Rightarrow \begin{cases} W'_i \mid Z_i^* \sim \mathcal{N}_q(m^*, V^*) \text{ avec} \\ m^* = (I_q + \Lambda'\Lambda)^{-1}(\Lambda'Z_i^*) \\ V^* = (I_q + \Lambda'\Lambda)^{-1} \end{cases}$$

Effets sites aléatoires et variance associée :

En ce qui concerne les effets sites aléatoires $(\alpha_i)_{i=1,\dots,I}$, on pose $Z_{ij}^* = Z_{ij} - D_i P_j = \alpha_i + \epsilon_{ij}$, avec $D_i = (1, X_{i1}, \dots, X_{ip}, W_{i1}, \dots, W_{iq})$. On a ainsi $Z_{ij}^* \mid \alpha_i \sim \mathcal{N}(\alpha_i, 1)$ iid pour $j = 1, \dots, J$, puis on applique la proposition suivante :

Proposition 2.2.3.2.

$$\begin{cases} x_i \mid \theta \sim \mathcal{N}(\theta, \sigma^2) \text{ iid pour } i = 1, \dots, n \\ \theta \sim \mathcal{N}(\mu_0, \tau_0^2) \\ \sigma^2 \text{ connu} \end{cases} \Rightarrow \begin{cases} \theta \mid x_1, \dots, x_n \sim \mathcal{N}(\mu_1, \tau_1^2) \text{ avec} \\ \mu_1 = \frac{\tau_0^{-2}\mu_0 + \sigma^{-2}\sum_{i=1}^n x_i}{\tau_0^{-2} + n\sigma^{-2}} \\ \tau_1^{-2} = \tau_0^{-2} + n\sigma^{-2} \end{cases}$$

Preuve 2.2.3.2.

$$\begin{aligned}
p(\theta \mid x_1, \dots, x_n) &\propto p(\theta)p(x_1, \dots, x_n \mid \theta) \\
&\propto \frac{1}{(2\pi\tau_0^2)^{\frac{1}{2}}} \exp\left(-\frac{1}{2\tau_0^2}(\theta - \mu_0)^2\right) \prod_{i=1}^n \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp\left(-\frac{1}{2\sigma^2}(x_i - \theta)^2\right) \\
&\propto \exp\left(-\frac{1}{2\tau_0^2}(\theta - \mu_0)^2 - \frac{1}{2\sigma^2} \sum_{i=1}^n (x_i - \theta)^2\right) \\
&\propto \exp\left(-\frac{1}{2\tau_0^2}(\theta^2 - 2\mu_0\theta) - \frac{1}{2\sigma^2} \sum_{i=1}^n (\theta^2 - 2\theta x_i)\right) \\
&\propto \exp\left(-\frac{1}{2}\left(\theta^2(\tau_0^{-2} + n\sigma^{-2}) - 2\mu_0\theta\tau_0^{-2} - 2\theta\sigma^{-2} \sum_{i=1}^n x_i\right)\right) \\
&\propto \exp\left(-\frac{1}{2(\tau_0^{-2} + n\sigma^{-2})^{-1}} \left(\theta - \frac{\mu_0\tau_0^{-2} + \sigma^{-2} \sum_{i=1}^n x_i}{\tau_0^{-2} + n\sigma^{-2}}\right)^2\right)
\end{aligned}$$

On obtient ainsi :

$$\begin{cases} Z_{ij}^* \mid \alpha_i \sim \mathcal{N}(\alpha_i, 1), \text{iid } \forall j = 1, \dots, J \\ \alpha_i \sim \mathcal{N}(0, V_\alpha) \end{cases} \Rightarrow \begin{cases} \alpha_i \mid Z_{i1}^*, \dots, Z_{iJ}^* \sim \mathcal{N}(\mu_1, \tau_1^{-2}) \text{ avec} \\ \mu_1 = \frac{\sum_{j=1}^J Z_{ij}^*}{V_\alpha^{-1} + J} \text{ et } \tau_1^{-2} = V_\alpha^{-1} + J. \end{cases}$$

Finalement pour estimer V_α , la variance des effets sites aléatoires $(\alpha_i)_{i=1, \dots, I}$, on utilise la proposition suivante :

Proposition 2.2.3.3. Si

$$\begin{cases} x \mid \sigma^2 \sim \mathcal{N}_n(\theta, \sigma^2 I_n) \\ \sigma^2 \sim \mathcal{IG}(a, b) \\ \theta \text{ connu} \end{cases} \Rightarrow \begin{cases} \sigma^2 \mid x \sim \mathcal{IG}(a', b') \text{ avec} \\ a' = a + \frac{n}{2} \text{ et } b' = \frac{1}{2} \sum_{i=1}^n (x_i - \theta)^2 + b. \end{cases}$$

Preuve 2.2.3.3.

$$\begin{aligned}
p(\sigma^2 \mid x) &\propto p(x \mid \sigma^2) p(\sigma^2) \\
&\propto \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp\left(-\frac{1}{2\sigma^2}(x - \theta)'(x - \theta)\right) \frac{b^a}{\Gamma(a)} (\sigma^2)^{-(a+1)} \exp\left(-\frac{b}{\sigma^2}\right) \\
&\propto (\sigma^2)^{-\left(\frac{n}{2} + a + 1\right)} \exp\left(-\frac{1}{\sigma^2} \left(b + \frac{1}{2} \sum_{i=1}^n (x_i - \theta)^2\right)\right)
\end{aligned}$$

On a donc :

$$\begin{cases} (\alpha_1, \dots, \alpha_I)' \mid V_\alpha \sim \mathcal{N}_n(0_{\mathbb{R}^I}, V_\alpha I_n) \\ V_\alpha \sim \mathcal{IG}(a, b) \end{cases} \Rightarrow \begin{cases} V_\alpha \mid \alpha_1, \dots, \alpha_I \sim \mathcal{IG}(a', b') \text{ avec} \\ a' = a + \frac{I}{2} \text{ et } b' = b + \frac{1}{2} \sum_{i=1}^I \alpha_i^2. \end{cases}$$

2.2.4 Echantillonneur de Gibbs et priors conjugués

L'algorithme utilisé pour estimer les paramètres du modèle logit est donc le suivant :

- Définir les constantes N_{Gibbs} , N_{burn} , N_{thin} telles que N_{Gibbs} correspond au nombre d'itérations effectuées par l'échantillonneur de Gibbs, N_{burn} au nombre d'itérations nécessaires pour le burn-in ou temps de chauffe et $N_{samp} = \frac{N_{Gibbs} - N_{burn}}{N_{thin}}$ au nombre de valeurs estimées retenues pour chaque paramètre. En effet on enregistre les paramètres estimés à certaines itérations, afin d'obtenir un échantillon de N_{samp} valeurs distribuées selon la distribution *a posteriori* pour chacun des paramètres.
- Initialiser tous les paramètres à 0 par exemple, excepté les valeurs diagonales de Λ initialisées à 1 et $V_\alpha^{(0)} = 1$.

- *Gibbs sampler* : à chaque itération t pour $t = 1, \dots, N_{Gibbs}$ on répète chacune de ces étapes :

- Générer la **variable latente** $Z^{(t)} = (Z_{ij}^{(t)})_{i=1, \dots, I}^{j=1, \dots, J}$ telle que

$$Z_{ij}^{(t)} \sim \begin{cases} \mathcal{N}(\alpha_i^{(t1)} + \beta_{j0}^{(t1)} + X_i \beta_j(t1) + W_i^{(t1)} \lambda_j^{(t1)}, 1) & \text{tronquée à droite par 0} \quad \text{si } y_{ij} = 0 \\ \mathcal{N}(\alpha_i^{(t1)} + \beta_{j0}^{(t1)} + X_i \beta_j(t1) + W_i^{(t1)} \lambda_j^{(t1)}, 1) & \text{tronquée à gauche par 0} \quad \text{si } y_{ij} = 1 \end{cases}$$

, la variable latente est ainsi initialisée à la première itération en la générant selon ces lois normales centrées.

- Générer les **effets espèces fixes** $P_j^{(t)} = (\beta_{j0}^{(t)}, \beta_{j1}^{(t)}, \dots, \beta_{jp}^{(t)}, \lambda_{j1}^{(t)}, \dots, \lambda_{jq}^{(t)})'$ pour $j = 1, \dots, J$ selon :

$$P_j^{(t)} \mid Z^{(t)}, W_1^{(t-1)}, \alpha_1^{(t-1)}, \dots, W_I^{(t1)}, \alpha_I^{(t-1)} \sim \mathcal{N}_{p+q+1}(m^*, V^*), \text{ avec}$$

$$m^* = (V^{-1} + D^{(t)'} D^{(t)})^{-1} (V^{-1} m + D^{(t)'} Z_j^*), \text{ et } V^* = (V^{-1} + D^{(t)'} D^{(t)})^{-1},$$

$$\text{où } Z_j^* = (Z_{1j}^*, \dots, Z_{Ij}^*)' \text{ tel que } Z_{ij}^* = Z_{ij}^{(t)} - \alpha_i^{(t-1)}.$$

Afin de contraindre les valeurs diagonales de $\Lambda = (\lambda_{jl})_{j=1, \dots, J}^{l=1, \dots, q}$ à des valeurs positives et de rendre la matrice triangulaire inférieure, on modifie les valeurs des $P^{(t)}$ simulées aléatoirement selon les conditions suivantes :

$$P_{jp+1+l}^{(t)} = \lambda_{jl}^{(t)} \leftarrow \begin{cases} 0 & \text{si } l > j \\ \lambda_{jl}^{(t-1)} & \text{si } l = j \text{ et } \lambda_{jl}^{(t)} < 0. \end{cases}$$

On pose $P^{(t)} = (P_1^{(t)} | \dots | P_J^{(t)})$.

- Générer les **variables latentes** (ou prédicteurs non mesurés) $W_i^{(t)}$ pour $i = 1, \dots, I$ selon :

$$W_i^{(t)} \mid Z^{(t)}, P^{(t)}, \alpha_i^{(t-1)} \sim \mathcal{N}_q \left((I_q + \Lambda^{(t)'} \Lambda^{(t)})^{-1} (\Lambda^{(t)'} Z_i^{**}), (I_q + \Lambda^{(t)'} \Lambda^{(t)})^{-1} \right),$$

$$\text{où } Z_i^{**} = (Z_{i1}^{**}, \dots, Z_{iJ}^{**})' \text{ tel que } Z_{ij}^{**} = Z_{ij}^{(t)} - \alpha_i^{(t-1)} \beta_{j0}^{(t)} - X_i \beta_j^{(t)}.$$

On pose $D_i^{(t)} = (1, X_{i1}, \dots, X_{ip}, W_{i1}^{(t)}, \dots, W_{iq}^{(t)})$.

- Générer les **effets sites aléatoires** $\alpha_i^{(t)}$ pour $i = 1, \dots, I$ selon :

$$\alpha_i \mid Z^{(t)}, P^{(t)}, W_i^{(t)} \sim \mathcal{N} \left(\frac{\sum_{j=1}^J Z_{ij}^{(t)} - D_i^{(t)} P_j^{(t)}}{V_\alpha^{(t-1)} + J}, \left(\frac{1}{V_\alpha^{(t-1)}} + J \right)^{-1} \right)$$

- Générer la **variance des effets sites aléatoires** $V_\alpha^{(t)}$ selon :

$$V_\alpha^{(t)} \mid \alpha_1^{(t)}, \dots, \alpha_I^{(t)} \sim \mathcal{IG} \left(\text{shape} = 0.5 + \frac{I}{2}, \text{rate} = 0.005 + \frac{1}{2} \sum_{i=1}^I (\alpha_i^{(t)})^2 \right)$$

2.3 Modèle logit : échantillonneur de Gibbs et algorithme de Metropolis adaptatif

D'autre part on considère une fonction de lien logit : $p \rightarrow \ln\left(\frac{p}{1-p}\right) = F^{-1}(p)$, avec $F : x \rightarrow \frac{1}{1+e^{-x}}$ la fonction de répartition appelée sigmoïde d'une loi logistique standard.

2.3.1 Définition du modèle logit

De la même façon que pour le modèle probit, d'après Marine Guillerm (2016) on peut définir le modèle logit par l'intermédiaire d'une variable latente : $Z_{ij} = \alpha_i + \beta_{j0} + X_i \beta_j + W_i \lambda_j + \epsilon_{ij}$ pour $i = 1, \dots, I$ et $j = 1, \dots, J$, avec $\epsilon_{ij} \sim \text{logistique}(0, 1)$ iid et telle que :

$$y_{ij} = \begin{cases} 1 & \text{si } Z_{ij} > 0 \\ 0 & \text{sinon.} \end{cases}$$

Cependant dans ce cas les distributions *a priori* de la variable latente et des paramètres n'étant pas conjuguées, on n'est pas en mesure d'utiliser les propriétés des priors conjugués donc la modélisation à l'aide d'une variable latente ne présente pas d'intérêt.

Dans ce cas on suppose que

$$y_{ij} | \theta_{ij} \sim \mathcal{B}(n_i, \theta_{ij})$$

, avec $\text{probit}(\theta_{ij}) = \alpha_i + \beta_{j0} + X_i \beta_j + W_i \lambda_j$ et n_i le nombre de visites du site i .

Par conséquent on échantillonnera les paramètres de ce modèle selon une estimation de leurs distributions conditionnelles *a posteriori* à l'aide d'un algorithme de Metropolis adaptatif.

2.3.2 Priors utilisés

On détermine une distribution *a priori* pour chacun des paramètres du modèle :

$$\begin{aligned} V_\alpha &\sim \mathcal{IG}(\text{shape} = 0.5, \text{rate} = 0.005) \text{ avec rate} = \frac{1}{\text{scale}}, \\ \beta_{jk} &\sim \mathcal{N}(0, 10^6) \text{ pour } j = 1, \dots, J \text{ et } k = 1, \dots, p, \\ \lambda_{jl} &\sim \begin{cases} \mathcal{N}(0, 10) & \text{si } l < j \\ \mathcal{U}(0, 10) & \text{si } l = j \\ P \text{ tel que } \mathbb{P}(\lambda_{jl} = 0) = 1 & \text{si } l > j \end{cases} \\ &\text{pour } j = 1, \dots, J \text{ et } l = 1, \dots, q. \end{aligned}$$

2.3.3 Principe d'un algorithme de Metropolis adaptatif

Cet algorithme appartient aux méthodes MCMC et permet d'obtenir une réalisation du paramètre $\theta = (\theta_1, \dots, \theta_m)$ selon leurs distributions conditionnelles *a posteriori* $\pi(\theta_i | \theta_1, \dots, \theta_{i-1}, \theta_{i+1}, \dots, \theta_m, x)$, pour $i = 1, \dots, m$ connues à une constante multiplicative près.

On le qualifie d'adaptatif car la variance de la densité instrumentale conditionnelle utilisée est adaptée en fonction du nombre d'acceptation lors des dernières itérations.

- **Initialisation** : $\theta^{(0)} = (\theta_1^{(0)}, \dots, \theta_m^{(0)})$ fixés arbitrairement, les nombres d'acceptation $(n_i^A)_{i=1, \dots, m}$ sont initialisés à 0 et les variances $(\sigma_i^2)_{i=1, \dots, m}$ sont initialisées à 1.
- **Itération t** : pour $i = 1, \dots, m$
 - Générer $\theta_i^* \sim q(\theta_i^{(t-1)}, .)$, avec une densité instrumentale conditionnelle $q(\theta_i^{(t-1)}, \theta_i^*)$ symétrique, on choisira une loi $\mathcal{N}(\theta_i^{(t-1)}, \sigma_i^2)$ par exemple.
 - Calculer la probabilité d'acceptation :

$$\gamma = \min \left(1, \frac{\pi(\theta_i^* | \theta_1^{(t-1)}, \dots, \theta_{i-1}^{(t-1)}, \theta_{i+1}^{(t-1)}, \dots, \theta_m^{(t-1)}, x)}{\pi(\theta_i^{(t-1)} | \theta_1^{(t-1)}, \dots, \theta_{i-1}^{(t-1)}, \theta_{i+1}^{(t-1)}, \dots, \theta_m^{(t-1)}, x)} \right)$$

- Retenir

$$\theta_i^{(t)} = \begin{cases} \theta_i^* & \text{avec probabilité } \gamma \\ \theta_i^{(t-1)} & \text{si on est dans ce cas le nombre d'acceptation devient : } n_i^A \leftarrow n_i^A + 1 \\ & \text{avec probabilité } 1 - \gamma. \end{cases}$$

— Durant le **burn-in**, toutes les DIV itérations, avec

$$\text{DIV} = \begin{cases} 100 & \text{si } N_{Gibbs} \geq 1000 \\ \frac{N_{Gibbs}}{10} & \text{sinon} \end{cases}$$

, où N_{Gibbs} est le nombre total d'itérations effectuées.

On modifie les variances en fonction des nombres d'acceptation de la manière suivante pour $i = 1, \dots, m$:

- On calcule le taux d'acceptation : $r_i^A = \frac{n_i^A}{\text{DIV}}$.
- On adapte les variances selon le taux d'acceptation et une constante fixée R_{opt} :

$$\sigma_i \leftarrow \begin{cases} \sigma_i \left(2 - \frac{1 - r_i^A}{1 - R_{opt}} \right) & \text{si } r_i^A \geq R_{opt} \\ \frac{\sigma_i}{2 - \frac{1 - r_i^A}{1 - R_{opt}}} & \text{sinon} \end{cases}$$

- On réinitialise les nombres d'acceptation : $n_i^A \leftarrow 0$.
- Toutes les $\frac{N_{Gibbs}}{10}$ itérations, on calcule et affiche les taux d'acceptation moyen $m^A = \frac{1}{m} \sum_{i=1, \dots, m} r_i^A$.

2.3.4 Echantillonneur de Gibbs et algorithme de Metropolis adaptatif

On utilise un algorithme de Metropolis adaptatif pour échantillonner les paramètres du modèle selon leurs distributions conditionnelles *a posteriori* estimées à une constante multiplicative près.

Dans un premier temps on définit la fonction f permettant de calculer la vraisemblance du modèle en fonction des paramètres estimés :

$$f : \lambda_j, \beta_{j0}, \beta_j, \alpha_i, W_i, X_i, y_{ij}, n_i \rightarrow f(\lambda_j, \beta_{j0}, \beta_j, \alpha_i, W_i, X_i, y_{ij}, n_i) = L(\theta_{ij})$$

- Calcul $\text{logit}(\theta_{ij}) = \alpha_i + \beta_{j0} + X_i \beta_j + W_i \lambda_j$.
- Calcul $\theta_{ij} = \frac{1}{1 + \exp(-\text{logit}(\theta_{ij}))}$.
- Renvoi $L(\theta_{ij}) = p(y_{ij} | \theta_{ij}, n_i) = \binom{n_i}{y_{ij}} (\theta_{ij})^{y_{ij}} (1 - \theta_{ij})^{n_i - y_{ij}}$.

On répète ces étapes pour $i = 1, \dots, I$ et $j = 1, \dots, J$, et on pose $\theta = (\theta_{ij})_{i=1, \dots, I}^{j=1, \dots, J}$.

On peut ainsi calculer la vraisemblance du modèle : $L(\theta) = \prod_{\substack{1 \leq i \leq I \\ 1 \leq j \leq J}} L(\theta_{ij})$.

D'après la formules de Bayes on a

$$p(\theta | Y) \propto \pi(\theta) L(\theta).$$

On utilise donc les relations suivantes pour approcher les densités conditionnelles *a posteriori* de chacun des paramètres avec $\pi(\cdot)$ les densités correspondants à leurs lois *a priori*.

$$\begin{aligned} p(\beta_{jk} | \beta_{j0}, \beta_{j1}, \dots, \beta_{jk-1}, \beta_{jk+1}, \dots, \beta_{jp}, \lambda_j, \alpha_1, \dots, \alpha_I, W_1, \dots, W_I, Y) &\propto \pi(\beta_{jk}) \prod_{1 \leq i \leq I} L(\theta_{ij}) \\ p(\lambda_{jl} | \lambda_{j1}, \dots, \lambda_{jl-1}, \lambda_{jl+1}, \dots, \lambda_{jq}, \beta_j, \beta_{j0}, \alpha_1, \dots, \alpha_I, W_1, \dots, W_I, Y) &\propto \pi(\lambda_{jl}) \prod_{1 \leq i \leq I} L(\theta_{ij}) \\ p(W_{il} | W_{i1}, \dots, W_{il-1}, W_{il+1}, \dots, W_{iq}, \alpha_i, \beta_{10}, \dots, \beta_{J0}, \beta_1, \dots, \beta_J, \lambda_1, \dots, \lambda_J, Y) &\propto \pi(W_{il}) \prod_{1 \leq j \leq J} L(\theta_{ij}) \\ p(\alpha_i | W_i, \beta_{10}, \dots, \beta_{J0}, \beta_1, \dots, \beta_J, \lambda_1, \dots, \lambda_j, V_\alpha, Y) &\propto \pi(\alpha_i | V_\alpha) \prod_{1 \leq j \leq J} L(\theta_{ij}) \end{aligned}$$

, pour $i = 1, \dots, I$, $j = 1, \dots, J$, $k = 1, \dots, p$ et $l = 1, \dots, q$.

L'algorithme utilisé pour estimer les paramètres du modèle logit est donc le suivant :

- Définition des constantes N_{Gibbs} , N_{burn} , N_{thin} et R_{opt} tels que N_{Gibbs} correspond au nombre d'itérations effectuées par l'algorithme, N_{burn} au nombre d'itérations nécessaires pour le burn-in ou temps de chauffe, $N_{samp} = \frac{N_{Gibbs} - N_{burn}}{N_{thin}}$ correspondant au nombre de valeurs estimées retenues pour chaque paramètre. En effet on enregistre les paramètres estimés à certaines itérations afin d'obtenir N_{samp} valeurs, nous permettant de représenter une distribution *a posteriori* pour chacun des paramètres.

On fixe R_{opt} le ratio d'acceptation optimal utilisé dans les algorithmes de Metropolis adaptatifs implémentés pour chacun des paramètres du modèle.

- Initialiser tous les paramètres à 0 par exemple, excepté les valeurs diagonales de Λ initialisées à 1 et $V_\alpha^{(0)} = 1$. Le nombre d'acceptation de chaque paramètre est initialisé à 0 et les variances de leur densités instrumententales conditionnelles prennent la valeur 1.
- Gibbs sampler : à chaque itération t pour $t = 1, \dots, N_{Gibbs}$ on répète chacune de ces étapes :

- Générer les **effets sites aléatoires** $\alpha_i^{(t)}$ pour $i = 1, \dots, I$ selon un algorithme de Metropolis adaptatif simulant $\alpha_i^* \sim \mathcal{N}(\alpha_i^{(t-1)}, \sigma_{\alpha_i}^2)$ puis calculant le taux d'acceptation de la manière suivante :

$$\gamma = \min \left(1, \frac{\pi(\alpha_i^* | V_\alpha^{(t-1)}) \prod_{1 \leq j \leq J} f(\alpha_i^*, W_i^{(t-1)}, \beta_{j0}^{(t-1)}, \beta_j^{(t-1)}, \lambda_j^{(t-1)}, X_i, y_{ij}, n_i)}{\pi(\alpha_i^{(t-1)} | V_\alpha^{(t-1)}) \prod_{1 \leq j \leq J} f(\alpha_i^{(t-1)}, W_i^{(t-1)}, \beta_{j0}^{(t-1)}, \beta_j^{(t-1)}, \lambda_j^{(t-1)}, X_i, y_{ij}, n_i)} \right).$$

- Générer la **variance des effets sites aléatoires** $V_\alpha^{(t)}$ selon :

$$V_\alpha^{(t)} | \alpha_1^{(t)}, \dots, \alpha_I^{(t)} \sim \mathcal{IG} \left(\text{shape} = 0.5 + \frac{I}{2}, \text{rate} = 0.005 + \frac{1}{2} \sum_{i=1}^I (\alpha_i^{(t)})^2 \right)$$

- Générer les **variables latentes** (ou prédicteurs non mesurés) $W_{il}^{(t)}$ pour $i = 1, \dots, I$ et $l = 1, \dots, q$ selon un algorithme de Metropolis adaptatif simulant $W_{il}^* \sim \mathcal{N}(W_{il}^{(t-1)}, \sigma_{W_{il}}^2)$ puis calculant le taux d'acceptation de la manière suivante :

$$\gamma = \min \left(1, \frac{\pi(W_{il}^*) \prod_{1 \leq j \leq J} f(W_{il}^*, \alpha_i^{(t)}, \beta_{j0}^{(t-1)}, \beta_j^{(t-1)}, \lambda_j^{(t-1)}, X_i, y_{ij}, n_i)}{\pi(W_{il}^{(t-1)}) \prod_{1 \leq j \leq J} f(W_{il}^{(t-1)}, \alpha_i^{(t)}, \beta_{j0}^{(t-1)}, \beta_j^{(t-1)}, \lambda_j^{(t-1)}, X_i, y_{ij}, n_i)} \right).$$

- Générer les **effets espèces fixes** $\beta_{jk}^{(t)}$ pour $j = 1, \dots, J$ et $k = 0, \dots, p$ selon un algorithme de Metropolis adaptatif simulant $\beta_{jk}^* \sim \mathcal{N}(\beta_{jk}^{(t-1)}, \sigma_{\beta_{jk}}^2)$ puis calculant le taux d'acceptation de la manière suivante :

$$\gamma = \min \left(1, \frac{\pi(\beta_{jk}^*) \prod_{1 \leq i \leq I} f(\beta_{j0}^{(t)}, \dots, \beta_{jk-1}^{(t)}, \beta_{jk}^*, \beta_{jk+1}^{(t-1)}, \dots, \beta_{jp}^{(t-1)}, \lambda_j^{(t-1)}, \alpha_1^{(t)}, W_1^{(t)}, \dots, \alpha_I^{(t)}, W_I^{(t)}, X_i, y_{ij}, n_i)}{\pi(\beta_{jk}^{(t-1)}) \prod_{1 \leq i \leq I} f(\beta_{j0}^{(t)}, \dots, \beta_{jk-1}^{(t)}, \beta_{jk}^{(t-1)}, \beta_{jk+1}^{(t-1)}, \dots, \beta_{jp}^{(t-1)}, \lambda_j^{(t-1)}, \alpha_1^{(t)}, W_1^{(t)}, \dots, \alpha_I^{(t)}, W_I^{(t)}, X_i, y_{ij}, n_i)} \right).$$

- Générer les **effets espèces fixes liés aux variables latentes** $\lambda_{jl}^{(t)}$ pour $j = 1, \dots, J$ et $l = 1, \dots, q$ selon un algorithme de Metropolis adaptatif pour $l \geq j$, simulant $\lambda_{jl}^* \sim \mathcal{N}(\lambda_{jl}^{(t-1)}, \sigma_{\lambda_{jl}}^2)$ puis calculant le taux d'acceptation de la manière suivante :

$$\gamma = \min \left(1, \frac{\pi(\lambda_{jl}^*) \prod_{1 \leq i \leq I} f(\lambda_{j1}^{(t)}, \dots, \lambda_{jl-1}^{(t)}, \lambda_{jl}^*, \lambda_{jl+1}^{(t-1)}, \dots, \lambda_{jq}^{(t-1)}, \beta_{j0}^{(t)}, \beta_j^{(t)}, \alpha_1^{(t)}, W_1^{(t)}, \dots, \alpha_I^{(t)}, W_I^{(t)}, X_i, y_{ij}, n_i)}{\pi(\lambda_{jl}^{(t-1)}) \prod_{1 \leq i \leq I} f(\lambda_{j1}^{(t)}, \dots, \lambda_{jl-1}^{(t)}, \lambda_{jl}^{(t-1)}, \lambda_{jl+1}^{(t-1)}, \dots, \lambda_{jq}^{(t-1)}, \beta_{j0}^{(t)}, \beta_j^{(t)}, \alpha_1^{(t)}, W_1^{(t)}, \dots, \alpha_I^{(t)}, W_I^{(t)}, X_i, y_{ij}, n_i)} \right).$$

Dans le cas $l > j$, on pose $\lambda_{jl}^{(t)} = 0$.

2.4 Evaluation de la fiabilité de ces méthodes sur des données simulées

2.4.1 Simulation des données

On simule pour chacun des sites i deux variables bioclimatiques selon une loi normale centrée réduite et on note $X_i = (1, X_{i1}, X_{i2})$ pour $i = 1, \dots, 500$. On considérera deux variables latentes.

On fixe les paramètres suivants afin de simuler des données de présence absence pour les 100 espèces et 500 sites considérés selon les deux modèles définis précédemment.

Les effets espèces fixes générés selon des lois uniformes :

$$\beta_{jk} \sim \mathcal{U}(-2, 2) \text{ pour } j = 1, \dots, 100 \text{ et } k = 0, 1, 2.$$

$$\lambda_{jl} \begin{cases} \sim \mathcal{U}(-2, 2) & \text{si } l < j, \\ \sim \mathcal{U}(0, 2) & \text{si } l = j, \\ = 0 & \text{si } l > j, \end{cases}$$

pour $j = 1, \dots, 100$ et $l = 1, 2$.

Les variables latentes $W_{il} \sim \mathcal{N}(0, 1)$ pour $i = 1, \dots, 500$ et $l = 1, 2$.

Les effets sites aléatoires générés selon $\alpha_i \sim \mathcal{N}(0, V_{alpha})$ avec V_{alpha} fixé à 0.5.

Puis on simule les données de présence absence $Y = (y_{ij})_{i=1, \dots, 500}^{j=1, \dots, 100}$ de la manière appropriée pour chacun des modèles envisagés en fonction des paramètres établis :

Modèle probit :

Pour $i = 1, \dots, 500$ et $j = 1, \dots, 100$:

Générer $\epsilon_{ij} \sim \mathcal{N}(0, 1)$ iid.

Calculer $Z_{ij} = \alpha_i + \beta_{j0} + X_{i1}\beta_j + W_{i1}\lambda_j + \epsilon_{ij}$

Simuler $y_{ij} = \begin{cases} 1 & \text{si } Z_{ij} > 0 \\ 0 & \text{sinon.} \end{cases}$

Modèle logit :

Pour $i = 1, \dots, 500$ et $j = 1, \dots, 100$:

Calculer $\text{logit}(\theta_{ij}) = \alpha_i + \beta_{j0} + X_{i1}\beta_j + W_{i1}\lambda_j$.

Calculer $\theta_{ij} = \frac{1}{1 + \exp(-\text{logit}(\theta_{ij}))}$.

Simuler $y_{ij} \sim \text{Bernoulli}(\theta_{ij})$.

2.4.2 Représentation des paramètres estimés

Les paramètres à estimer sont au nombre de $(p+1)J+qJ-\sum_{j=1}^{q-1}(q-j)+qI+I+1 = 3 \times 100 + 2 \times 100 - 1 + 2 \times 500 + 500 + 1 = 2000$,

pour chacun d'entre eux les fonctions implémentées retournent un échantillon de N_{samp} valeurs dont on fait la moyenne pour obtenir un estimateur du paramètre.

On note $\widehat{\beta}_{jk} = \sum_{n=1}^{N_{samp}} \frac{\beta_{jk}^{(n)}}{N_{samp}}$ l'estimateur de β_{jk} et on utilise les mêmes notations pour les autres estimateurs.

On effectue 40000 itérations au total dont 35000 de burn-in puis 5000 durant lesquelles les fonctions implémentées enregistrent les paramètres estimés à partir des données simulées précédemment, toutes les 5 itérations et renvoient un échantillon de $N_{samp} = 1000$ valeurs pour chacun des paramètres du modèle considéré.

Modèle probit :

Figure 1 – Représentation des $(\alpha_i)_{i=1, \dots, I}$ estimés en fonction de ceux simulés pour le modèle probit

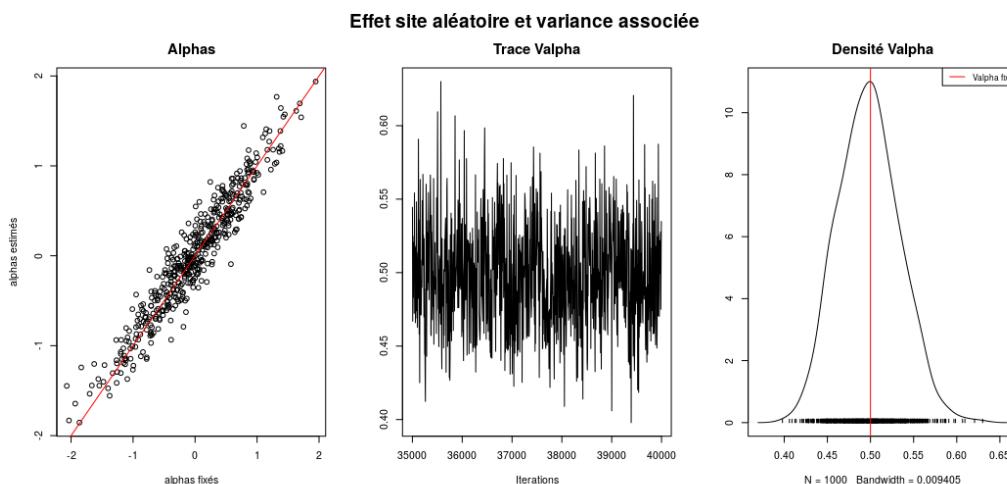


Figure 2 – Représentation des $(W_{il})_{i=1,\dots,500}^{l=1,2}$ estimés en fonction de ceux simulés pour le modèle probit

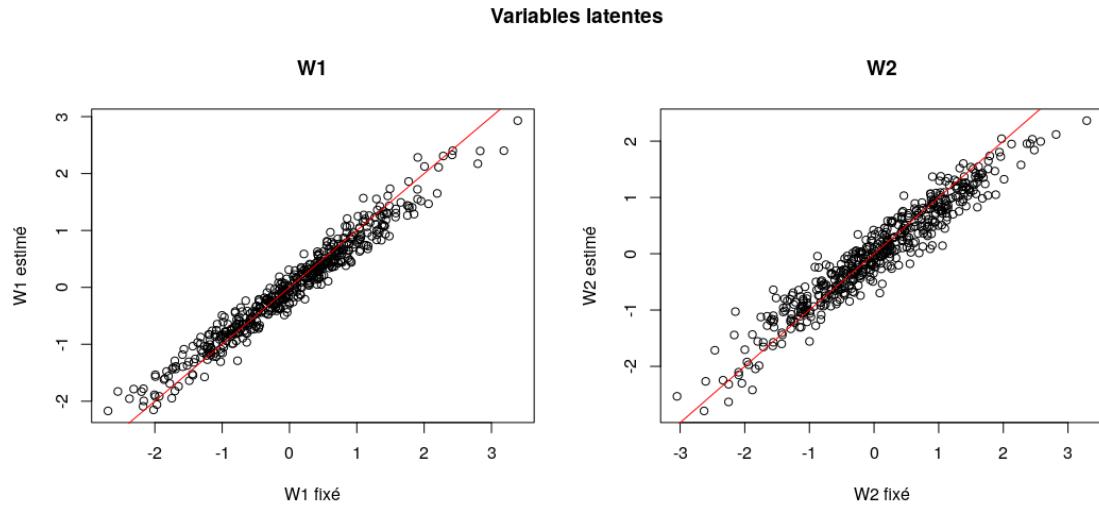


Figure 3 – Représentation des $(\beta_{jk})_{j=1,\dots,100}^{k=1,2,3}$ et $(\lambda_{jl})_{j=1,\dots,100}^{l=1,2}$ estimés en fonction de ceux simulés pour le modèle probit

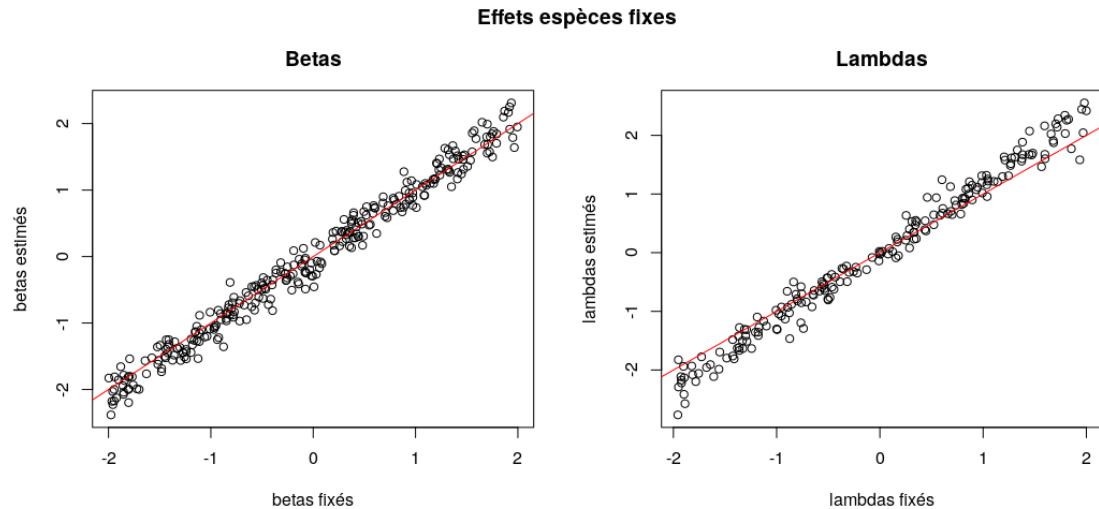
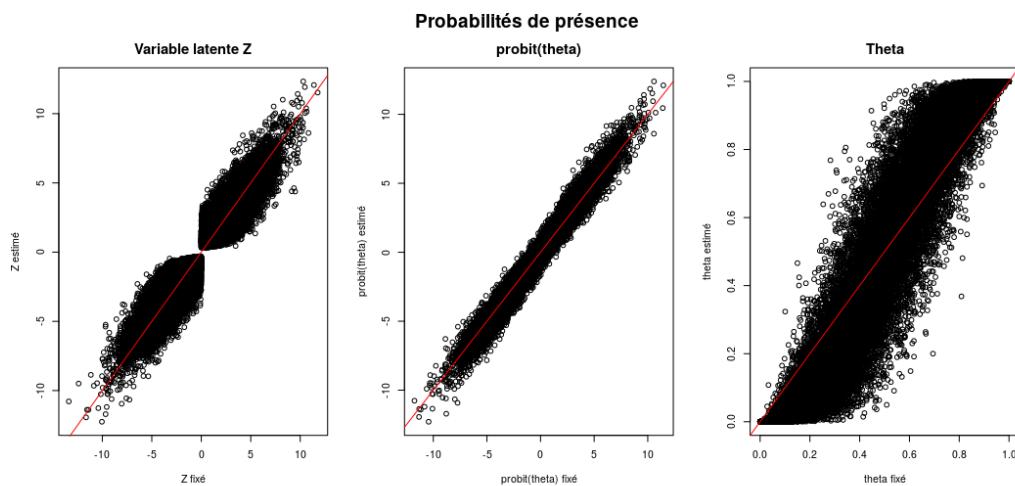


Figure 4 – Représentation des $(\text{probit}(\theta_{ij}))_{i=1,\dots,500}^{j=1,\dots,100}$, $(Z_{ij})_{i=1,\dots,500}^{j=1,\dots,100}$ et $(\theta_{ij})_{i=1,\dots,500}^{j=1,\dots,100}$ estimés en fonction de ceux simulés pour le modèle probit



Modèle logit :

Figure 5 – Représentation des $(\alpha_i)_{i=1,\dots,I}$ estimés en fonction de ceux simulés pour le modèle logit

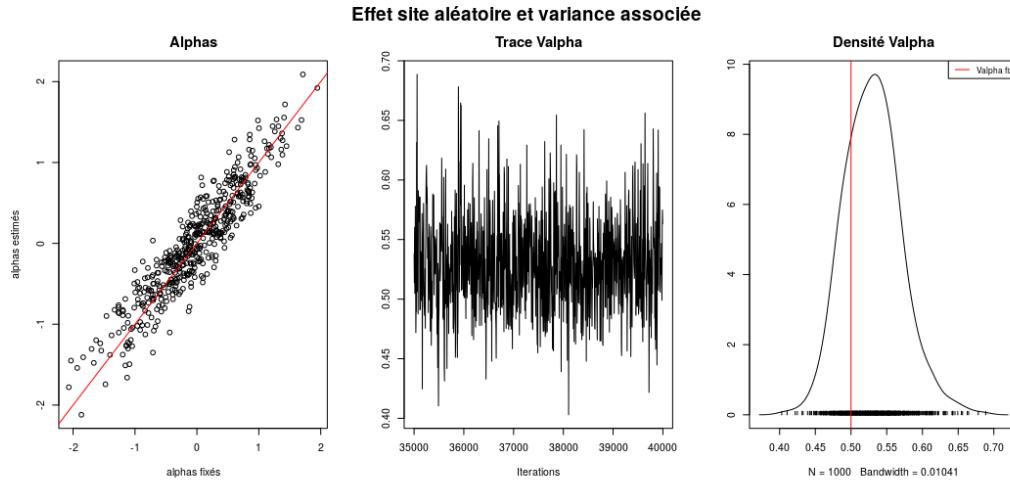


Figure 6 – Représentation des $(W_{il})_{i=1,\dots,500}^{l=1,2}$ estimés en fonction de ceux simulés pour le modèle logit

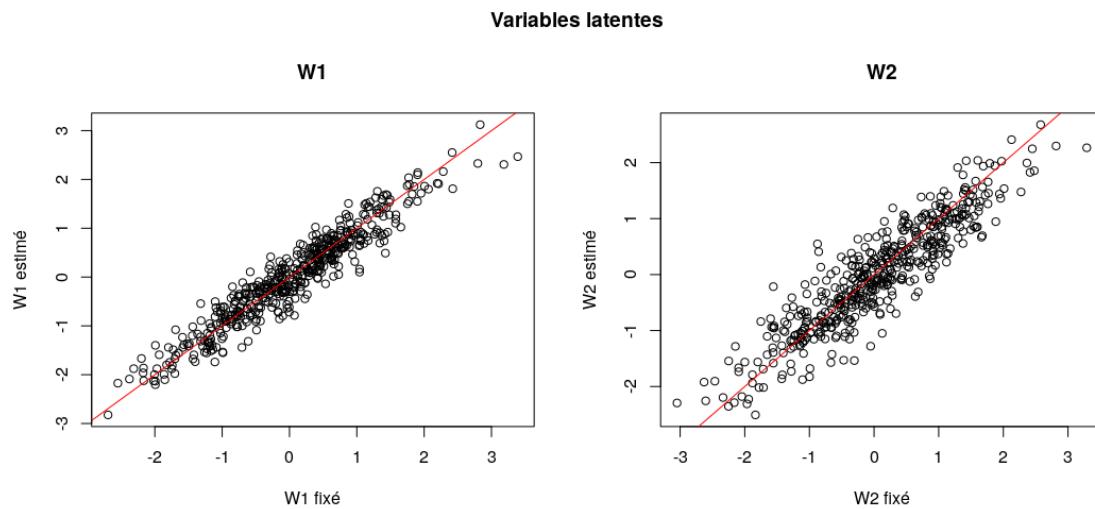


Figure 7 – Représentation des $(\beta_{jk})_{j=1,\dots,100}^{k=1,2,3}$ et $(\lambda_{jl})_{j=1,\dots,100}^{l=1,2}$ estimés en fonction de ceux simulés pour le modèle logit

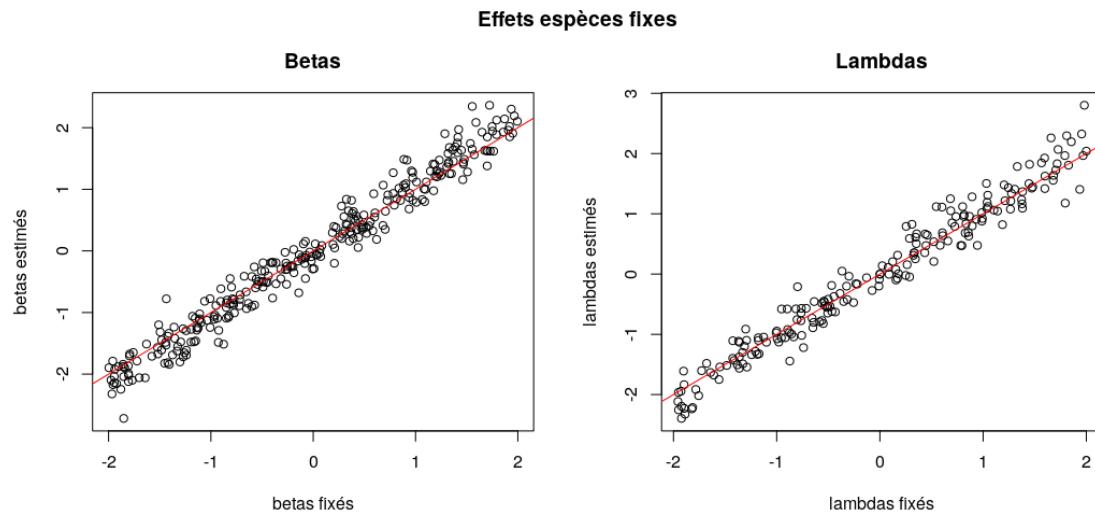
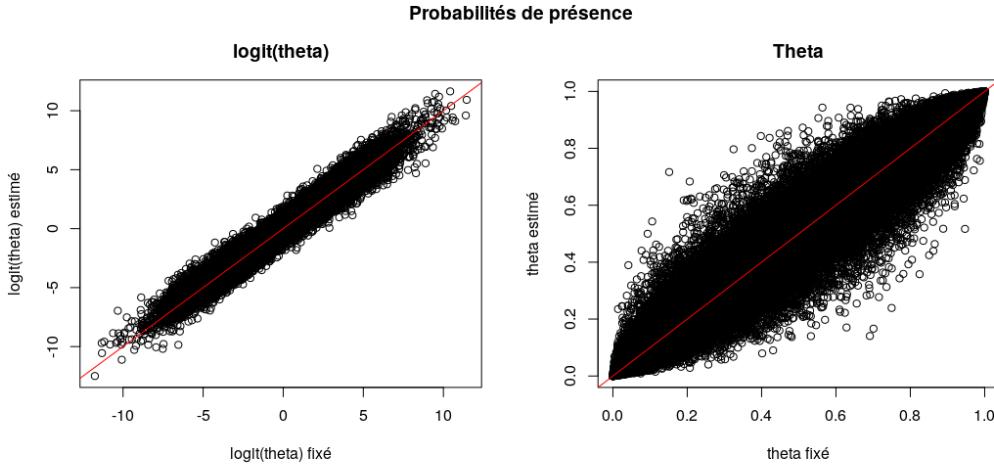


Figure 8 – Représentation des $(\text{logit}(\theta_{ij}))_{i=1, \dots, 500}^{j=1, \dots, 100}$ et $(\theta_{ij})_{i=1, \dots, 500}^{j=1, \dots, 100}$ estimés en fonction de ceux simulés pour le modèle probit



2.4.3 Evaluation du temps de calcul et de la pertinence des résultats

Table 1 – Temps de calcul nécessaires à l'ajustement des modèles logit et probit ainsi que le pourcentage de déviance expliquée et la racine de la moyenne des carrés des erreurs normalisée (NRMSE) obtenus pour chaque modèle de la manière suivante avec une fonction de lien g probit ou logit :

$$\text{NRMSE}_g = \frac{\text{RMSE}_g}{g(\bar{\theta})} \text{ avec } \bar{g}(\bar{\theta}) = \sum_{\substack{1 \leq i \leq I \\ 1 \leq j \leq J}} \frac{g(\theta_{ij})}{IJ} \text{ et } \text{RMSE} = \sqrt{\frac{1}{IJ} \sum_{\substack{1 \leq i \leq I \\ 1 \leq j \leq J}} \left(g(\theta_{ij}) - \widehat{g}(\theta_{ij}) \right)^2}.$$

Le pourcentage de déviance expliquée est défini par $1 - \frac{D}{D_0}$ avec $D = -2(l(\hat{\theta}) - l_s)$ la déviance du modèle considéré, où l_s est la log-vraisemblance du modèle saturé comportant autant de paramètres qu'il y a d'observations et dont l'ajustement est supposé parfait par conséquent dans le cas de données binaires, les probabilités ajustées prenant les valeurs 0 ou 1 en fonction de celles de Y , on a $l_s = 0$ car la vraisemblance des données observées sous le modèle saturé est de 1.

De plus la déviance nulle est définie par $D_0 = -2(l(\hat{\theta}_0) - l_s)$ avec $l(\hat{\theta}_0)$ la log-vraisemblance du modèle nul défini par $g(\theta_{ij}) = \mu$, pour lequel le seul paramètre est l'intercept μ .

On ajuste le modèle nul avec la fonction `jSDM_binomial()` du package `jSDM` pour le modèle logit et avec `Rcpp_hSDM_binomial_probit()` pour le modèle probit.

Les log-vraisemblances des différents modèles sont calculées avec les paramètres estimés à chaque itération $l(\hat{\theta}) = \sum_{\substack{1 \leq i \leq I \\ 1 \leq j \leq J}} \log(L(\hat{\theta}_{ij}))$ et les valeurs utilisées pour le calcul du pourcentage de déviance expliquée correspondent à la moyenne des N_{samp} valeurs retournées par chacune des fonctions implémentées.

	Temps de calcul (mins)	NRMSE	Déviance expliquée (%)
Modèle probit	12.5	3.2	37.8
Modèle logit	64.0	3.7	58.6

Afin d'ajuster un modèle joint de distribution des espèces à partir des données collectées à Madagascar, on utilisera donc la fonction implémentée pour le modèle probit qui présente un pourcentage de déviance expliquée supérieur, un RMSE inférieur ainsi qu'un temps de calcul bien inférieur à ceux associés au modèle logit, indiquant que les paramètres estimés sont plus en accord avec les données et proches des valeurs attendues pour le modèle logit.

3 Comparaison des packages boral et jSDM

3.1 Performances sur des données simulées

On ajuste le modèle joint de distribution des espèces défini précédemment à partir des données simulées selon le modèle probit à l'aide du package jSDM d'une part et boral d'autre part, puis on compare les résultats obtenus et les temps de calculs nécessaires.

3.2 Comparaison des résultats sur des jeux de données réels

3.2.1 Jeux de données utilisés

3.2.2 Comparaison des résultats

4 Application aux données collectées à Madagascar

4.1 Description des données

On dispose d'inventaires forestiers réalisés sur 753 sites de l'île de Madagascar et répertoriant la présence ou l'absence de 555 espèces végétales sur chacun de ces sites.

Parmi les données climatiques et environnementales concernant Madagascar à l'heure actuelle (interpolations de données observées représentatives des années 1950-2000), disponibles sur le site <https://madaclim.cirad.fr>, on choisit d'utiliser les variables suivantes :

Table 2 – Variables bioclimatiques considérées affichées pour quelques sites :

Les températures (temp) et précipitations (prec) moyennes annuelles qui sont exprimées respectivement en $^{\circ}\text{C} \times 10$ et millimètres.

La saisonnalité des températures (sais_temp) correspond à l'écart type des températures mensuelles multiplié par 100 ainsi que la saisonnalité des précipitations (sais_prec) sous la forme d'un coefficient de variation.

Le déficit hydrique climatique (cwd) annuel est calculé en fonction des précipitations et des évapotranspirations potentielles mensuelles (pet) qui sont définies comme la quantité d'évaporation qui se produirait en un mois si une source d'eau suffisante était disponible : $\text{cwd} = \sum_{m=1}^{12} \min(0, \text{prec}_m - \text{pet}_m)$, il est exprimé en millimètres.

Les trois dernières colonnes correspondent aux coordonnées du site considéré en latitude (lat) et longitude (long) ainsi qu'à son identifiant (site).

temp	prec	sais_temp	sais_prec	cwd	long	lat	site
241	1302	1316	110	498	49.145	-12.440	1
243	1288	1320	111	521	49.136	-12.449	2
238	1321	1321	110	475	49.155	-12.449	3
239	1321	1323	110	482	49.145	-12.449	4
238	1325	1329	109	466	49.145	-12.458	5
196	1498	1366	97	216	49.181	-12.586	6

On considère également les carrés de ces variables climatiques afin d'effectuer un régression quadratique, plus adaptée pour ajuster un modèle de niche qu'une régression linéaire. On centre et on réduit ces variables afin de former une matrice de design X telle que $X_i = (1, \text{temp}_i, \text{prec}_i, \text{sais_temp}_i, \text{sais_prec}_i, \text{cwd}_i, \text{temp}_i^2, \text{prec}_i^2, \text{sais_temp}_i^2, \text{sais_prec}_i^2, \text{cwd}_i^2)$ pour $i = 1, \dots, 753$.

Les coordonnées des sites seront utilisées par la suite dans le cadre de l'interpolation spatiale des résultats et représenter spatialement les résultats.

4.2 Estimation des paramètres et représentation des résultats

4.2.1 Représentation des résultats pour évaluer la convergence de l'algorithme

On ajuste un modèle joint de distribution des espèces de fonction de lien probit en considérant deux variables latentes à l'aide de la fonction `jSDM_probit_block()` du package jSDM à partir des données décrites précédemment, en effectuant 100000 itérations dont 90000 de burn-in et on retient $N_{samp} = 1000$ valeurs pour chaque paramètre du modèle que l'on va représenter en fonction du nombre d'itérations effectuées afin d'évaluer la convergence de l'algorithme de Gibbs.

De plus on affiche une estimation de la densité des échantillons obtenus qui devrait correspondre aux distributions *a posteriori* définies et donc être de forme gaussienne. On met en évidence les moyennes des N_{samp} -échantillons en bleu, que l'on utilisera comme estimateur pour les paramètres.

Table 3 – Temps de calcul nécessaire à l'ajustement du modèle sur les données de Madagascar et nombre de paramètres à estimer

Nombre d'observations	Paramètres à estimer	Itérations effectuées	Temps de calcul (heures)
417915	9474	1e+05	12

Figure 9 – Traces et densités de la déviance du modèle

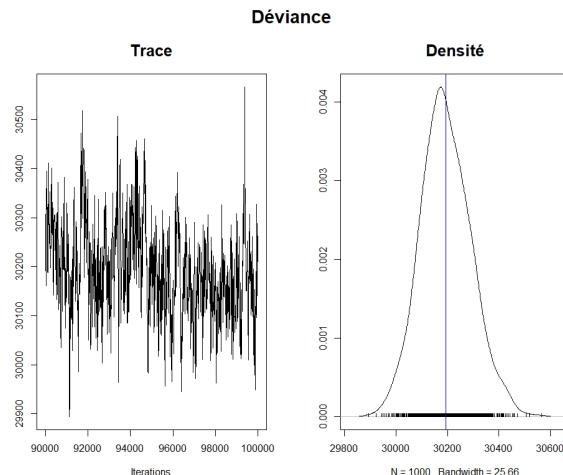


Figure 10 – Traces et densités des effets espèces fixes $(\beta_{jk})_{j=2}^{k=0, \dots, 10}$ estimés pour la deuxième espèce

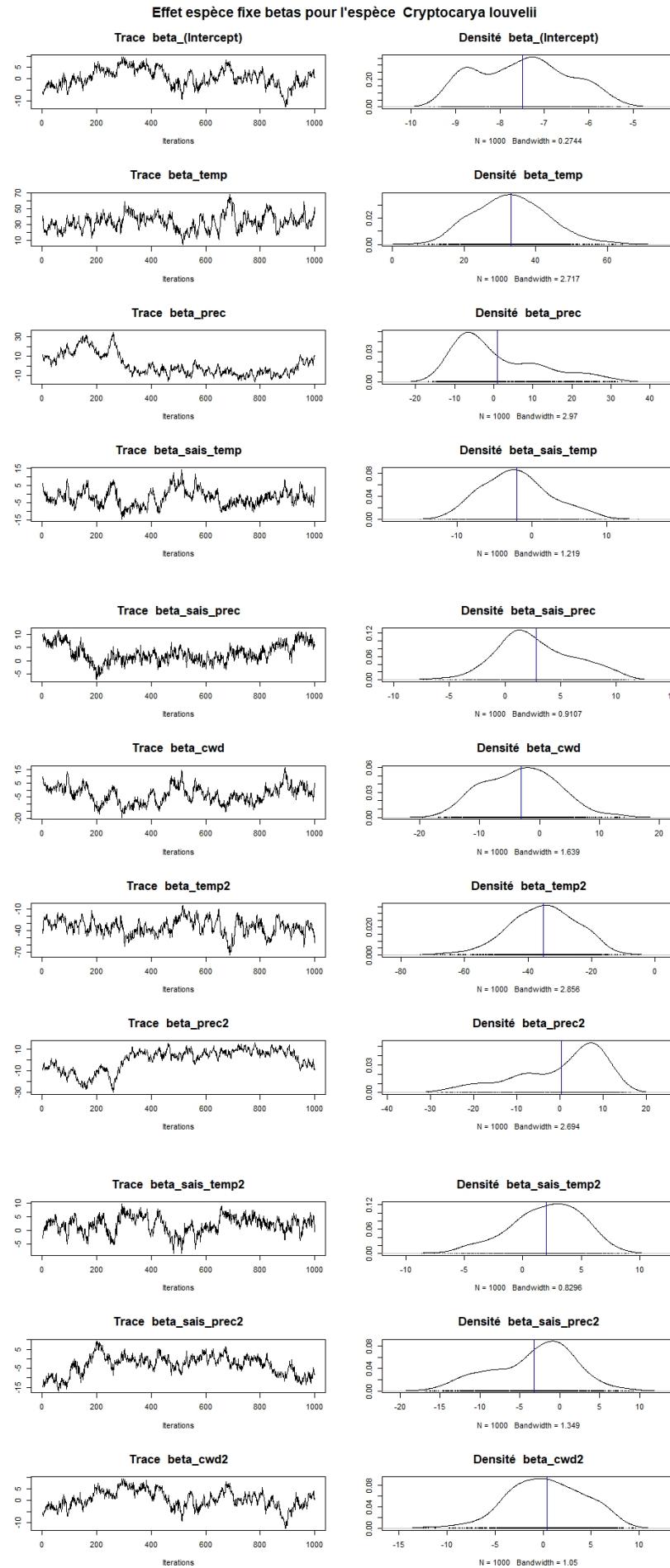


Figure 11 – Traces et densités des effets espèces fixes $(\lambda_{jq})_{j=1,2}^{q=1,2}$ estimés pour les deux premières espèces

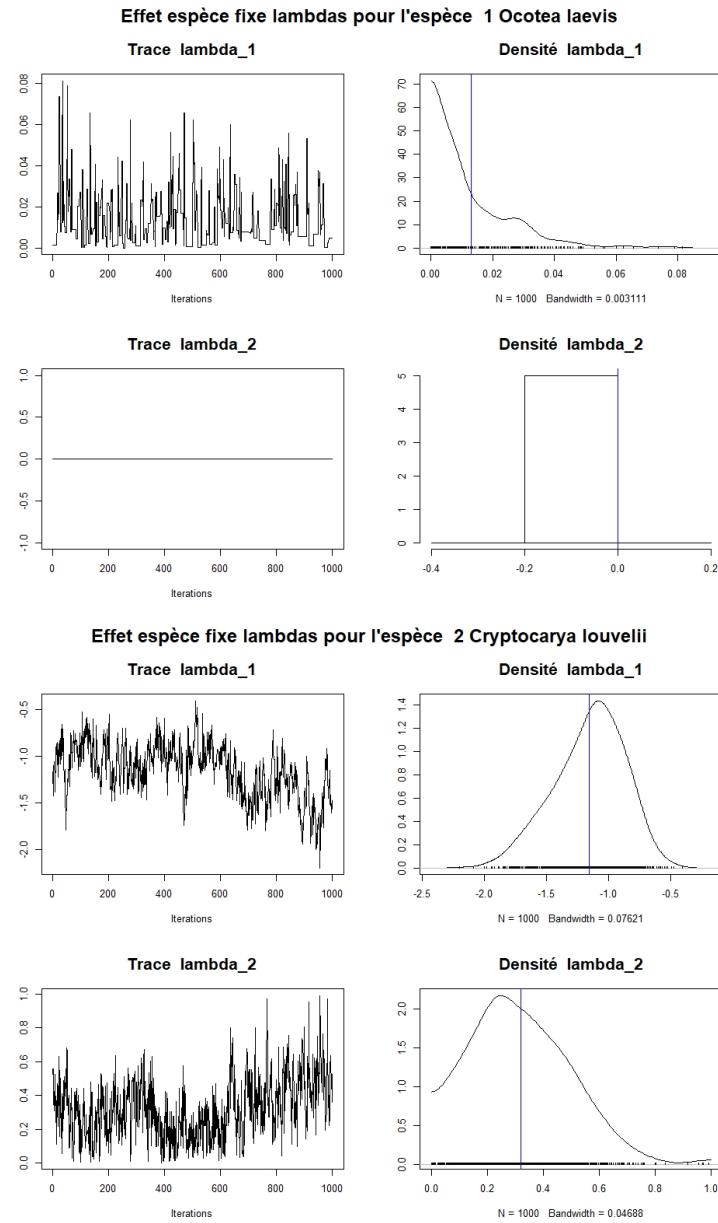


Figure 12 – Trace et densité estimées des variables latentes W_1 et W_2 estimées pour un site

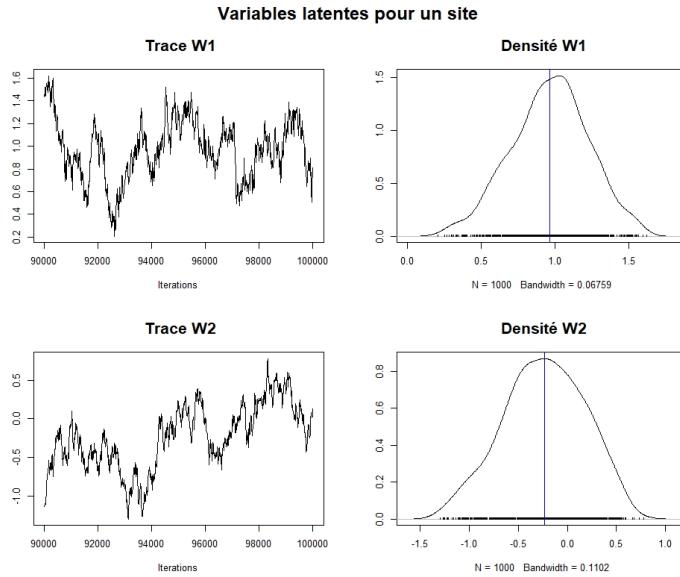
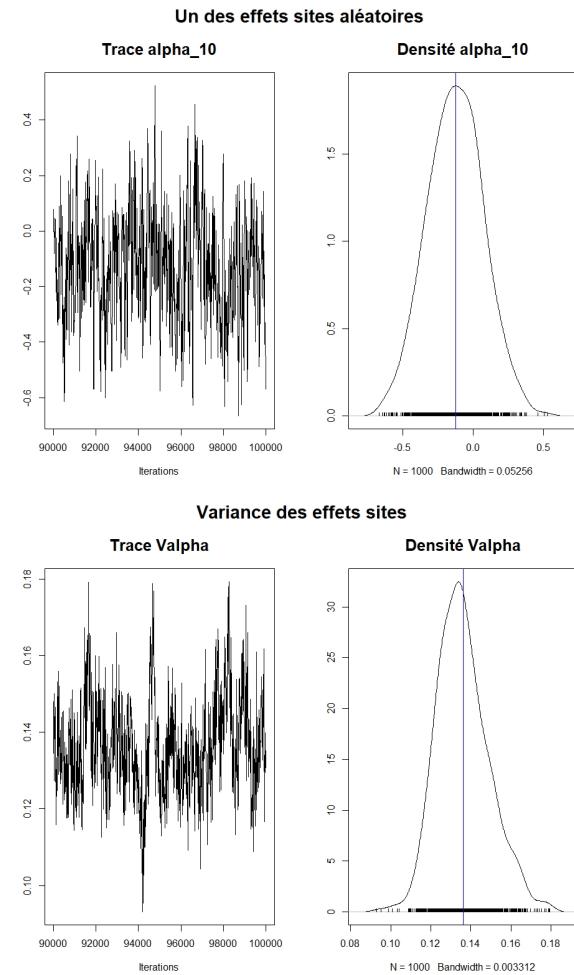


Figure 13 – Trace et densité d'un effet site et de la variance associée aux effets sites estimés



4.2.2 Représentations spatiales des paramètres associés aux sites et des probabilités de présence estimées

Figure 14 – Représentation spatiale des variables latentes W_1 et W_2 estimées pour chaque site

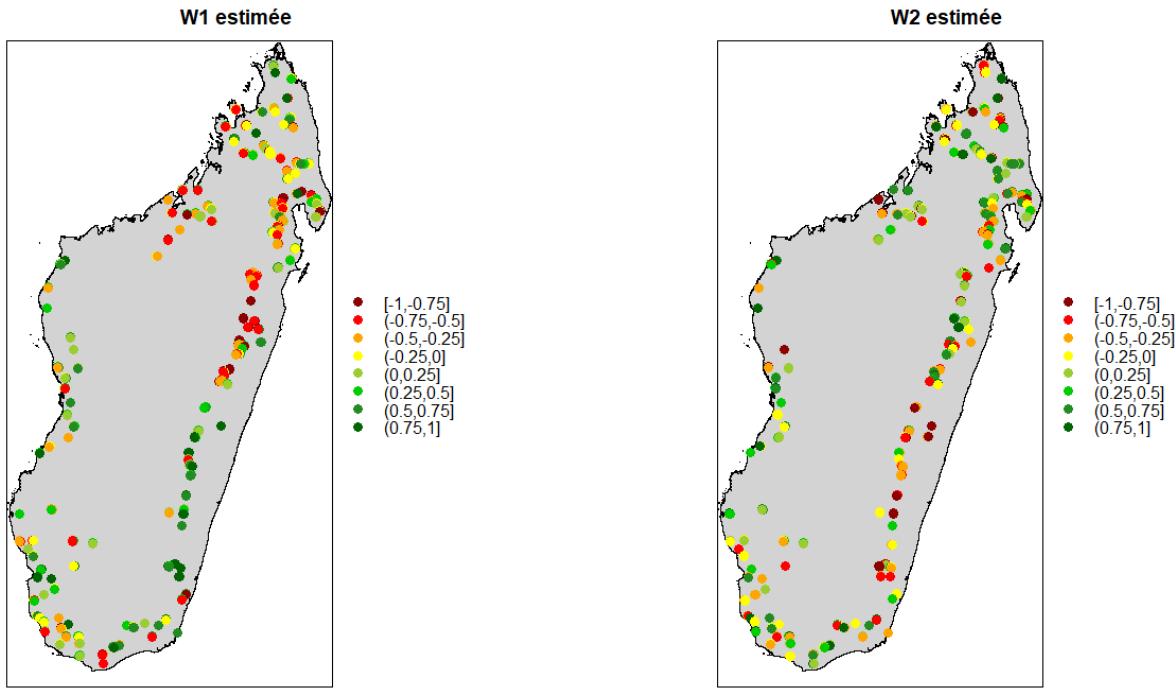


Figure 15 – Représentation spatiale des effets sites estimés

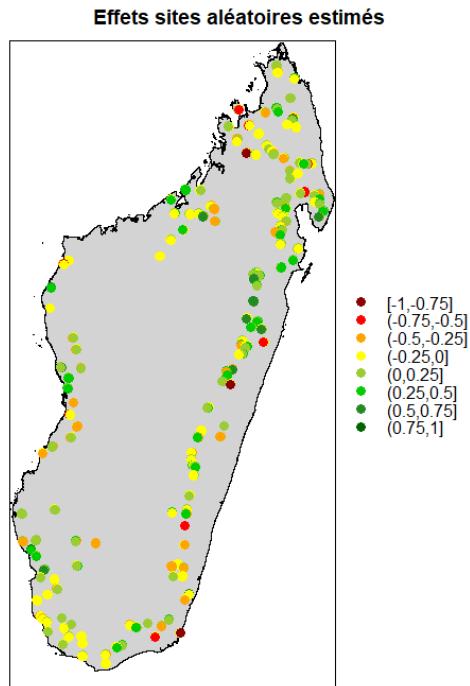


Figure 16 – Représentation spatiale des probabilités de présence estimées pour l'espèce *Ocotea laevis*

Probabilités de présence d'*Ocotea laevis* estimées

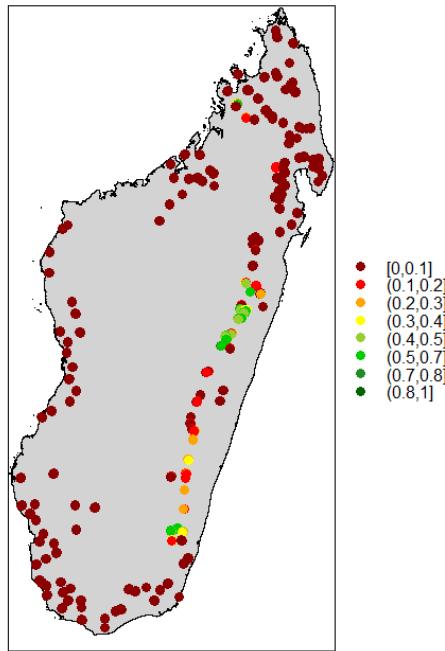


Figure 17 – Représentation spatiale des occurrences observées de l'espèce *Ocotea laevis*

Présences d'*Ocotea laevis* observées



4.2.3 Estimation de la richesse spécifique propre à chaque site et comparaison à celle observée

Figure 18 – Représentation spatiale de la richesse spécifique estimée pour chaque site par $\widehat{R}_i = \sum_{j=1}^{555} \widehat{\theta}_{ij}$

Richesse spécifique estimée

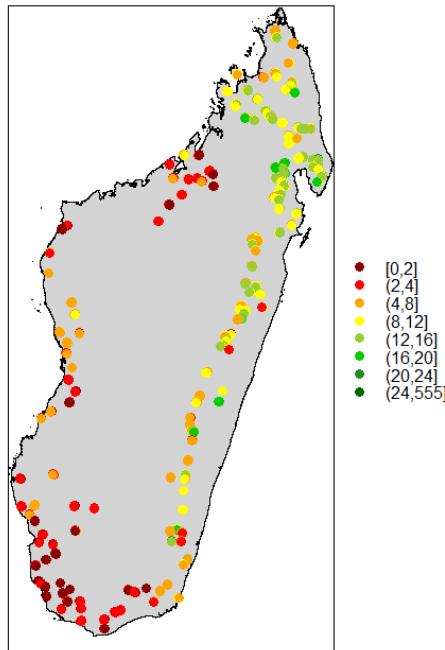


Figure 19 – Représentation spatiale de la richesse spécifique observée pour chaque site calculée par $R_i = \sum_{j=1}^{555} y_{ij}$

Richesse spécifique observée

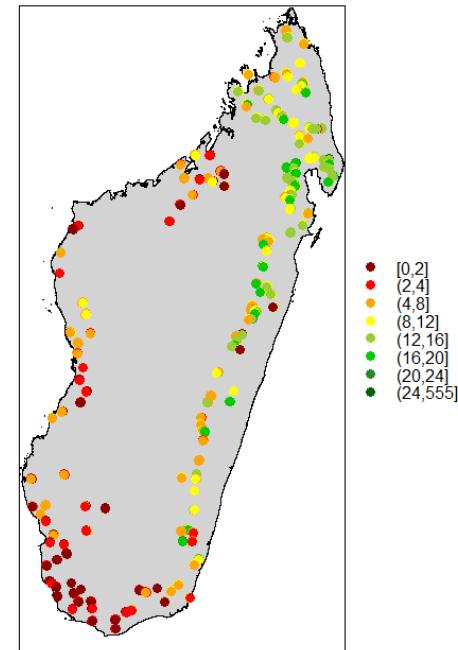
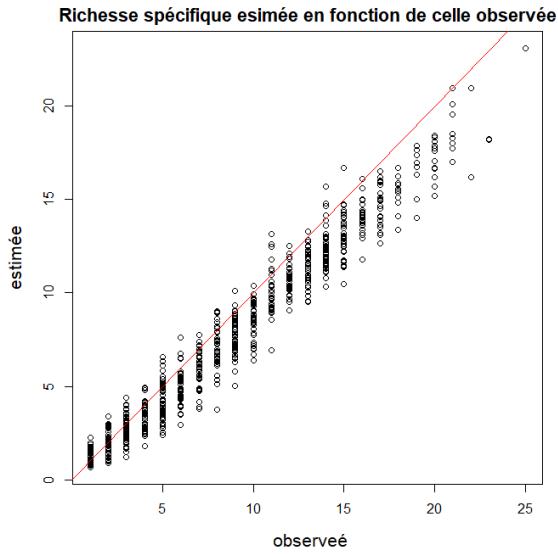


Figure 20 – Représentation richesse spécifique estimée en fonction de celle observée



4.3 Comparaison des effets sites obtenus avec différentes méthodes d'interpolation spatiale

définition interpolation et expliquer méthodes

4.3.1 Thin plate spline

4.3.2 Ordinary kriging

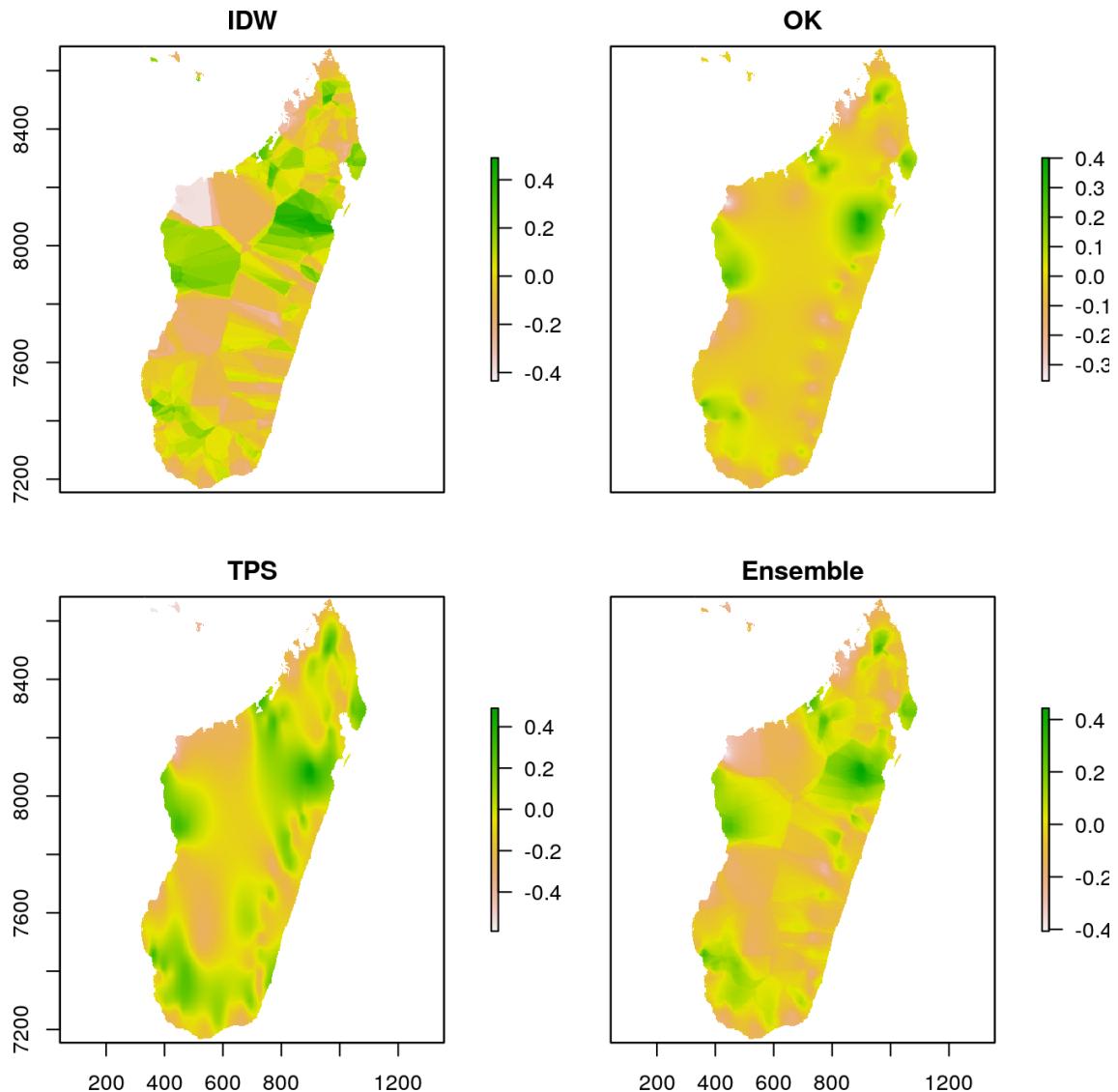
4.3.3 Inverse distance weighted

4.3.4 Comparaison des méthodes par validation croisée sur les effets sites estimés

expliquer validation croisée et calcul RMSE

Figure 21 – Comparaison des cartes obtenues pour l’effet site par une interpolation avec la méthode du krigage ordinaire (OK), de l’inverse de la distance pondéré (IDW) et de thin plate spline (TPS).

La carte intitulée ensemble représente la moyenne des résultats obtenus avec les différentes méthodes d’interpolation pondérés par le RMSE de la méthode considérée divisée par la somme des RMSE, qui sont calculés durant la validation croisée et correspondent à l’écart entre les effets sites interpolés et ceux estimés pour les sites sur lesquels on a collecté des données.



4.4 Résultats de l'interpolation spatiale par krigeage ordinaire

Figure 22 – Interpolation spatiale des effets sites par krigeage ordinaire

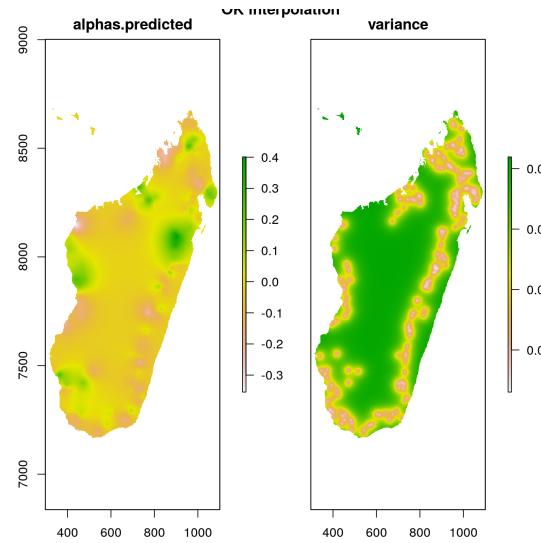
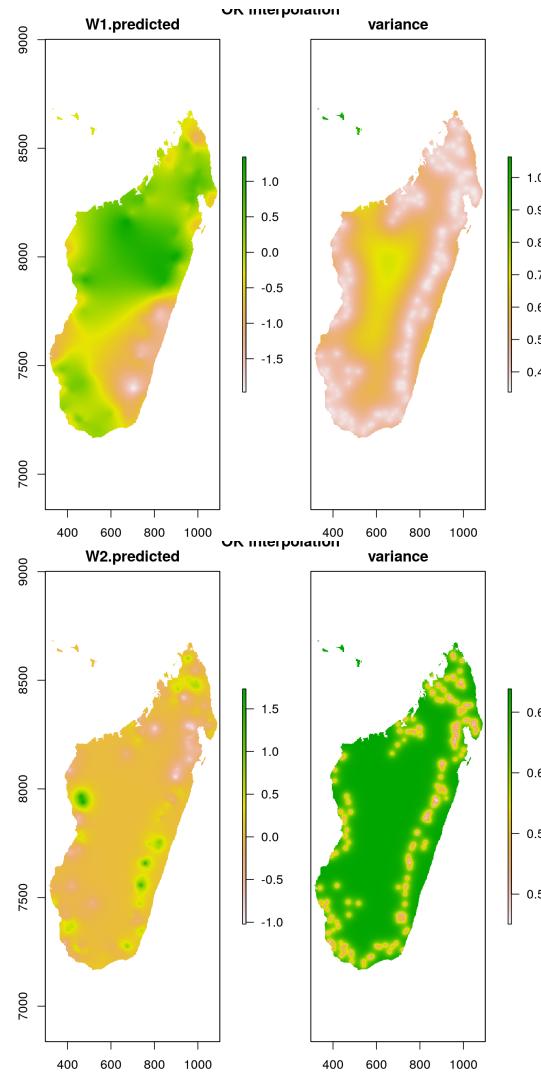


Figure 23 – Interpolation spatiale des variables latentes par krigeage ordinaire



4.5 Analyse des résultats et mise en évidence de lieux refuges de la biodiversité

résultats species richness theta et probit theta

4.6 Prédictions avec auto-corrélation spatiale

Expliquer méthode CAR + 2D spline qui sont implémentées mais pas abouties les résultats ne sont pas satisfaisants pour l'instant.

Conclusion

données GIECC pour évolution biodiversité avec changement climatique, intégrer traits spécifiques dans le modèle, enrichir le package,

Bibliographie

Albert, James H., and Chib Siddhartha. 1993. "Bayesian Analysis of Binary and Polychotomous Response Data." *Journal of the American Statistical Association* 88 (422) : 669–79. <https://doi.org/10.1080/01621459.1993.10476321>.

Marine Guillerm, Pauline Givord et. 2016. "Les Modèles Multiniveaux." *Méthodologie Statistique INSEE*.

Warton, David I., F. Guillaume Blanchet, Robert B. O'Hara, Otso Ovaskainen, Sara Taskinen, Steven C. Walker, and Francis K.C. Hui. 2015. "So Many Variables : Joint Modeling in Community Ecology." *Trends in Ecology & Evolution* 30 (12) : 766–79. <https://doi.org/10.1016/j.tree.2015.09.007>.

Annexes

Fonction utilisée pour le modèle probit

```
#include <RcppArmadillo.h>
#include <gsl/gsl_rng.h>
#include <gsl/gsl_randist.h>
#include <gsl/gsl_cdf.h>
#include <cmath>
#include "Rcpp_jSDM_useful.h"

// [[Rcpp::depends(RcppArmadillo)]]
// [[Rcpp::depends(RcppGSL)]]

using namespace arma;
using namespace std;

/* **** */
/* Gibbs sampler function */

// [[Rcpp::export]]
Rcpp::List Rcpp_jSDM_probit_block(const int ngibbs, int nthin, int nburn,
                                  arma::umat Y,
                                  arma::umat T,
                                  arma::mat X,
                                  arma::mat param_start,
                                  arma::mat Vparam,
                                  arma::vec muparam,
                                  arma::mat VW,
                                  arma::mat W_start,
                                  arma::vec alpha_start,
                                  double Valpha_start,
                                  double shape,
                                  double rate,
                                  const int seed,
                                  const int verbose) {

///////////////////////////////
// Defining and initializing objects//

// Initialize random number generator
gsl_rng *s = gsl_rng_alloc(gsl_rng_mt19937);
```

```

gsl_rng_set(s, seed);

// Redefining constants

const int NGIBBS = ngibbs;
const int NTHIN = nthin;
const int NBURN = nburn;
const int NSAMP = (NGIBBS-NBURN)/NTHIN;
const int NSITE = Y.n_rows;
const int NP = X.n_cols;
const int NSP = Y.n_cols;
const int NL = W_start.n_cols;

///////////////////////////////
// Declaring new objects to store results //
/* Parameters */

arma::Cube<double> param; param.zeros(NSAMP, NSP, NP+NL);
arma::Cube<double> W; W.zeros(NSAMP, NSITE, NL);
arma::mat alpha; alpha.zeros(NSAMP, NSITE);
arma::vec Valpha; Valpha.zeros(NSAMP);

/* Latent variable */

arma::mat probit_theta_pred; probit_theta_pred.zeros(NSITE, NSP);
arma::mat Z_latent; Z_latent.zeros(NSITE, NSP);

/* Deviance */

arma::vec Deviance; Deviance.zeros(NSAMP);

///////////////////////////////
// Initializing running parameters //

// mat of species effects parameters and coefficients for latent variables (nl+np,nspl)
arma::mat param_run = param_start;
// alpha vec of sites effects (nsite)
arma::vec alpha_run = alpha_start;
double Valpha_run = Valpha_start;
// w latent variables (nsite*nl)
arma::mat W_run = W_start;
// Z latent (nsite*nspl)
arma::mat Z_run; Z_run.zeros(NSITE,NSP);
// probit_theta_ij = X_i*beta_j + W_i*lambda_j + alpha_i
arma::mat probit_theta_run; probit_theta_run.zeros(NSITE,NSP);
// data
arma::mat data = arma::join_rows(X,W_run);

```

```

///////////
// Message//
Rprintf("\nRunning the Gibbs sampler. It may be long, please keep cool :)\n\n");
R_FlushConsole();

///////////////////////////////
//%%%%%%%%%%%%%
// Gibbs sampler

for (int g=0; g < NGIBBS; g++) {

/////////////////////////////
// latent variable Z //

for (int j=0; j<NSP; j++) {
    for (int i=0; i<NSITE; i++) {
        // Actualization
        if (Y(i,j) == 0) {
            Z_run(i,j) = rtnorm(s, R_NegInf, 0, probit_theta_run(i,j), 1);
        } else {
            Z_run(i,j) = rtnorm(s, 0, R_PosInf, probit_theta_run(i,j), 1);
        }
    } // loop on sites
} // loop on species

/////////////////////////////
// mat param: Gibbs algorithm //

// Loop on species
for (int j=0; j<NSP; j++) {
    // small_v
    arma::vec small_v = inv(Vparam)*muparam + data.t()*(Z_run.col(j) - alpha_run);
    // big_V
    arma::mat big_V = inv(inv(Vparam)+data.t()*data);
    // Draw in the posterior distribution
    arma::vec param_prop = arma_mvgauss(s, big_V*small_v, chol_decomp(big_V));

    // constraints on lambda
    for (int l=0; l<NL; l++) {
        if (l > j) {

```

```

    param_prop(NP+1) = 0;
}

if ((l==j) & (param_prop(NP+1) < 0)) {
    param_prop(NP+1) = param_run(NP+1,j);
}
}

param_run.col(j) = param_prop;
}// loop on species

///////////////////////////////
// mat latent variable W: Gibbs algorithm //

// Loop on sites
for (int i=0; i<NSITE; i++) {
    arma::mat beta_run = param_run.submat(0,0,NP-1,NSP-1);
    arma::mat lambda_run = param_run.submat(NP,0,NP+NL-1,NSP-1);
    // big_V
    arma::mat big_V = inv(inv(VW)+lambda_run*lambda_run.t());
}

// small_v
arma::vec small_v = lambda_run*(Z_run.row(i)-X.row(i)*beta_run-alpha_run(i)).t();

// Draw in the posterior distribution
arma::vec W_i = arma_mvgauss(s, big_V*small_v, chol_decomp(big_V));
W_run.row(i) = W_i.t();
}

data = arma::join_rows(X, W_run);

/////////////////////////////
// vec alpha : Gibbs algorithm //

// Loop on sites
double sum = 0.0;
for (int i=0; i<NSITE; i++) {
    // small_v
    double small_v = arma::sum(Z_run.row(i)-data.row(i)*param_run);
    // big_V
    double big_V = 1/(1/Valpha_run + NSP);
}

```

```

// Draw in the posterior distribution

alpha_run(i) = big_V*small_v + gsl_ran_gaussian_ziggurat(s, std::sqrt(big_V));
sum += alpha_run(i)*alpha_run(i);

}

////////////////////

// Valpha

double shape_posterior = shape + 0.5*NSITE;
double rate_posterior = rate + 0.5*sum;

Valpha_run = rate_posterior/gsl_ran_gamma_mt(s, shape_posterior, 1.0);

///////////////////

// Deviance //

// logLikelihood

double logL = 0.0;
for ( int i = 0; i < NSITE; i++ ) {
    for ( int j = 0; j < NSP; j++ ) {
        // probit(theta_ij) = X_i*beta_j + W_i*lambda_j + alpha_i
        probit_theta_run(i,j) = arma::as_scalar(data.row(i)*param_run.col(j) + alpha_run(i));
        // link function probit is the inverse of N(0,1) repartition function
        double theta = gsl_cdf_ugaussian_P(probit_theta_run(i,j));
        /* log Likelihood */
        logL += R::dbinom(Y(i,j), T(i,j), theta, 1);
    } // loop on species
} // loop on sites

// Deviance

double Deviance_run = -2 * logL;

////////////////////

// Output

if (((g+1)>NBURN) && (((g+1)%(NTHIN))==0)) {
    int isamp=((g+1)-NBURN)/(NTHIN);
    for ( int j=0; j<NSP; j++ ) {
        param.tube(isamp-1,j) = param_run.col(j);
        for ( int i=0; i<NSITE; i++ ) {
            W.tube(isamp-1,i) = W_run.row(i);
            Z_latent(i,j) += Z_run(i,j) / NSAMP; // We compute the mean of NSAMP values
            probit_theta_pred(i,j) += probit_theta_run(i,j)/NSAMP;
        }
    }
}

```

```

    }

    alpha.row(isamp-1) = alpha_run.t();
    Valpha(isamp-1) = Valpha_run;
    Deviance(isamp-1) = Deviance_run;
}

///////////////////////////////
// Progress bar

double Perc=100*(g+1)/(NGIBBS);
if (((g+1)%(NGIBBS/100))==0 && (verbose==1)) {
    Rprintf(" *");
    R_FlushConsole();
    //R_ProcessEvents(); for windows
    if (((g+1)%(NGIBBS/10))==0) {
        Rprintf(":%.1f%% \n",Perc);
        R_FlushConsole();
        //R_ProcessEvents(); for windows
    }
}
}

///////////////////
// User interrupt //

R_CheckUserInterrupt(); // allow user interrupts

} // Gibbs sampler

// Free memory
gsl_rng_free(s);

// Return results as a Rcpp::List
Rcpp::List results = Rcpp::List::create(Rcpp::Named("param") = param,
                                         Rcpp::Named("W") = W,
                                         Rcpp::Named("alpha") = alpha,
                                         Rcpp::Named("Valpha") = Valpha,
                                         Rcpp::Named("Deviance") = Deviance,
                                         Rcpp::Named("Z_latent") = Z_latent,
                                         Rcpp::Named("probit_theta_pred") = probit_theta_pred);

return results;

} // end Rcpp_jSDM_probit_block

```