Modélisation de la variabilité inter-individuelle dans les modèles de croissance de plantes

Charlotte Baey

Colloque Jeunes Probabilistes et Statisticiens - 8 avril 2014

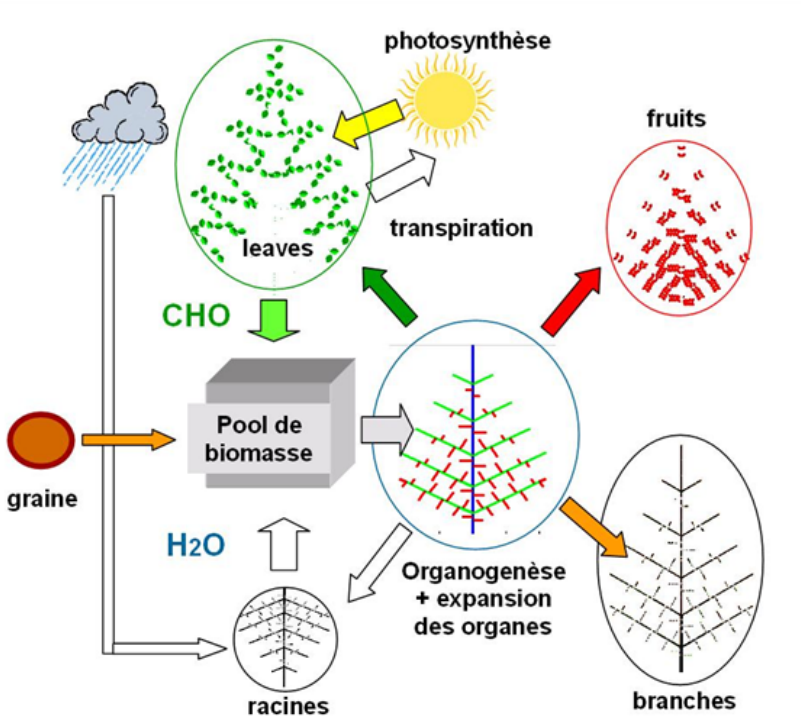
Introduction

Différents types de modèles de croissance de plantes existent, avec des objectifs variés :

- modèles architecturaux
  - ↪ simulation de paysages, plantes virtuelles, ...
- modèles de culture
  - ↪ aide à la décision, prévision de rendement, ...
- modèles structure-fonction
  - ↪ description plus fine des processus biologiques, lien entre paramètres du modèle et génotype, ...

Introduction : modèle Greenlab

Modèle Greenlab (de Reffye and Hu, 2003) :



- modèle individu-centré
- de type structure-fonction
- système dynamique :

$$X_{n+1} = F_n(X_n, U_n, P)$$

  - X_n : variables d'état (masses des organes)
  - $F_n \Rightarrow$ lois biophysiques
  - P : paramètres du modèle
  - U_n : variables de contrôle

Motivation

La plupart des approches courantes en modélisation de la croissance des plantes :

- sont basées sur le comportement moyen des plantes dans le champ
- ne prennent pas en compte la variabilité inter-individuelle
- proposent des prévisions moyennes

Et pourtant :

- il existe une variabilité génétique entre plantes, même de la même variété
- les conditions environnementales peuvent varier localement dans le champ
- définition difficile de la plante moyenne

$\Rightarrow$ il y a une forte variabilité entre les différents individus d'une population de plantes, qui peut avoir un impact majeur à l'échelle de l'agrosystème

Motivation

La plupart des approches courantes en modélisation de la croissance des plantes :

- sont basées sur le comportement moyen des plantes dans le champ
- ne prennent pas en compte la variabilité inter-individuelle
- proposent des prévisions moyennes

Et pourtant :

- il existe une variabilité génétique entre plantes, même de la même variété
- les conditions environnementales peuvent varier localement dans le champ
- définition difficile de la plante moyenne

$\Rightarrow$ il y a une forte variabilité entre les différents individus d'une population de plantes, qui peut avoir un impact majeur à l'échelle de l'agrosystème

Motivation

- L'extrapolation des modèles individus-centrés de croissance n'est pas immédiate
- Premières tentatives :
  - compétition pour la lumière
  - propagation d'incertitudes basée sur le développement en séries de Taylor
- L'utilisation de modèles de population stochastiques semble plus appropriée

$\rightarrow$ approche possible par l'utilisation de modèles mixtes

Modèles mixtes - formulation

On note $\{y_{ij}\}_{i=1,\dots,s,j=1,\dots,n_i}$: observation de la plante i sous la condition t_{ij}.

- **Étape 1** (variabilité intra-individuelle) : comment évoluent les mesures d'un même individu ?
  - un même jeu d'équations permet de modéliser cette évolution pour chaque individu de la population
  - mais, certains paramètres sont spécifiques à cet individu

$$y_{ij} = g(t_{ij}, \phi_i) + \varepsilon_{ij}, \quad \varepsilon_{ij} \sim \mathcal{N}(0, \sigma^2),$$

- **Étape 2** (variabilité inter-individuelle) :
  - les paramètres individuels obtenus à l'étape précédente sont considérés comme des variables aléatoires
  - on s'intéresse alors à la caractérisation de leur variation dans la population (moyenne et variance)

$$\phi_i = A_i\beta + \xi_i, \quad \xi_i \sim \mathcal{N}_P(0, \Gamma),$$

Modèles mixtes - formulation

On note $\{y_{ij}\}_{i=1,\dots,s,j=1,\dots,n_i}$: observation de la plante i sous la condition t_{ij}.

- **Étape 1** (variabilité intra-individuelle) : comment évoluent les mesures d'un même individu ?
  - un même jeu d'équations permet de modéliser cette évolution pour chaque individu de la population
  - mais, certains paramètres sont spécifiques à cet individu

$$y_{ij} = g(t_{ij}, \phi_i) + \varepsilon_{ij}, \quad \varepsilon_{ij} \sim \mathcal{N}(0, \sigma^2),$$

- **Étape 2** (variabilité inter-individuelle) :
  - les paramètres individuels obtenus à l'étape précédente sont considérés comme des variables aléatoires
  - on s'intéresse alors à la caractérisation de leur variation dans la population (moyenne et variance)

$$\phi_i = A_i\beta + \xi_i, \quad \xi_i \sim \mathcal{N}_P(0, \Gamma),$$

Modèles mixtes - formulation

On note $\{y_{ij}\}_{i=1,\dots,s,j=1,\dots,n_i}$: observation de la plante i sous la condition t_{ij}.

- **Étape 1** (variabilité intra-individuelle) : comment évoluent les mesures d'un même individu ?
  - un même jeu d'équations permet de modéliser cette évolution pour chaque individu de la population
  - mais, certains paramètres sont spécifiques à cet individu

$$y_{ij} = g(t_{ij}, \phi_i) + \varepsilon_{ij}, \quad \varepsilon_{ij} \sim \mathcal{N}(0, \sigma^2),$$

- **Étape 2** (variabilité inter-individuelle) :
  - les paramètres individuels obtenus à l'étape précédente sont considérés comme des variables aléatoires
  - on s'intéresse alors à la caractérisation de leur variation dans la population (moyenne et variance)

$$\phi_i = A_i\beta + \xi_i, \quad \xi_i \sim \mathcal{N}_P(0, \Gamma),$$

Modèles mixtes - formulation

On note $\{y_{ij}\}_{i=1,\dots,s,j=1,\dots,n_i}$: observation de la plante i sous la condition t_{ij}.

- **Étape 1** (variabilité intra-individuelle) : comment évoluent les mesures d'un même individu ?
  - un même jeu d'équations permet de modéliser cette évolution pour chaque individu de la population
  - mais, certains paramètres sont spécifiques à cet individu

$$y_{ij} = g(t_{ij}, \phi_i) + \varepsilon_{ij}, \quad \varepsilon_{ij} \sim \mathcal{N}(0, \sigma^2),$$

- **Étape 2** (variabilité inter-individuelle) :
  - les paramètres individuels obtenus à l'étape précédente sont considérés comme des variables aléatoires
  - on s'intéresse alors à la caractérisation de leur variation dans la population (moyenne et variance)

$$\phi_i = A_i\beta + \xi_i, \quad \xi_i \sim \mathcal{N}_P(0, \Gamma),$$

Modèles mixtes - formulation

On note $\{y_{ij}\}_{i=1,\dots,s,j=1,\dots,n_i}$: observation de la plante i sous la condition t_{ij}.

- **Étape 1** (variabilité intra-individuelle) : comment évoluent les mesures d'un même individu ?
  - un même jeu d'équations permet de modéliser cette évolution pour chaque individu de la population
  - mais, certains paramètres sont spécifiques à cet individu

$$y_{ij} = g(t_{ij}, \phi_i) + \varepsilon_{ij}, \quad \varepsilon_{ij} \sim \mathcal{N}(0, \sigma^2),$$

- **Étape 2** (variabilité inter-individuelle) :
  - les paramètres individuels obtenus à l'étape précédente sont considérés comme des variables aléatoires
  - on s'intéresse alors à la caractérisation de leur variation dans la population (moyenne et variance)

$$\phi_i = A_i\beta + \xi_i, \quad \xi_i \sim \mathcal{N}_P(0, \Gamma),$$

Modèles mixtes - estimation par MV

- Vecteur de paramètres : $\theta = (\beta, \Gamma, \sigma^2)$
- Vraisemblance :

$$L(\theta) := f(y;\theta) = \int_{\mathbb{R}^{P\times s}} f(y,\phi;\theta)d\phi = \int_{\mathbb{R}^{P\times s}} f(y \mid \phi;\theta) f(\phi;\theta)d\phi$$

- La non linéarité de la fonction $g(t_{ij}, \phi_i) = \mathbb{E}(y_{ij}|\phi_i)$ rend en général le calcul de cette intégrale impossible analytiquement
- Mais, les modèles mixtes peuvent être vus comme un problème de données incomplètes, en considérant les effets aléatoires comme des données manquantes

$\Rightarrow$ on peut alors utiliser une variante appropriée de l'algorithme d'Espérance-Maximisation (EM) (Dempster et al., 1977).

Algorithme EM

Idée principale de l'algorithme : travailler avec la densité complète $f(y, \phi; \theta)$

À l'itération k de l'algorithme, on a les deux étapes suivantes :

- **Étape E** (Espérance) : on calcule

$$Q(\theta; \theta^k) = \mathbb{E}\left(\log f(y, \phi; \theta) \mid y; \theta^k\right).$$

- **Étape M** (Maximisation) :

$$\theta^{(k+1)} = \arg\max_{\theta \in \Theta} Q(\theta; \theta^k).$$

- Lorsque la densité complète appartient à la famille exponentielle, les deux étapes s'écrivent simplement en fonction des statistiques exhaustives

Algorithme EM

Idée principale de l'algorithme : travailler avec la densité complète $f(y, \phi; \theta)$

À l'itération k de l'algorithme, on a les deux étapes suivantes :

- **Étape E** (Espérance) : on calcule

$$Q(\theta; \theta^k) = \mathbb{E}\left(\log f(y, \phi; \theta) \mid y; \theta^k\right).$$

- **Étape M** (Maximisation) :

$$\theta^{(k+1)} = \arg\max_{\theta \in \Theta} Q(\theta; \theta^k).$$

- Lorsque la densité complète appartient à la famille exponentielle, les deux étapes s'écrivent simplement en fonction des statistiques exhaustives

Algorithme EM - approximation de l'étape E

- À chaque itération de l'algorithme EM, on est ramené à des calculs d'espérance conditionnelle de différentes statistiques sous la loi des effets aléatoires ϕ sachant les observations y, $f(\phi \mid y; \theta)$
- Ce calcul est en général non explicite

→ variantes stochastiques de l'algorithme EM

- Méthodes basées sur des simulations de type Monte Carlo par chaîne de Markov (MCMC) :
  - l'algorithme MCMC-EM (Wei and Tanner, 1990) : on génère une nouvelle chaîne à chaque itération de l'algorithme
  - l'algorithme (MCMC-)SAEM (Delyon et al., 1999 ; Kuhn and Lavielle, 2005) : on réutilise les simulations des itérations précédentes
- La convergence des algorithmes a été étudiée dans le cas de la famille exponentielle

Algorithme EM - approximation de l'étape E

- À chaque itération de l'algorithme EM, on est ramené à des calculs d'espérance conditionnelle de différentes statistiques sous la loi des effets aléatoires ϕ sachant les observations y, $f(\phi \mid y; \theta)$
- Ce calcul est en général non explicite

→ variantes stochastiques de l'algorithme EM

- Méthodes basées sur des simulations de type Monte Carlo par chaîne de Markov (MCMC) :
  - l'algorithme MCMC-EM (Wei and Tanner, 1990) : on génère une nouvelle chaîne à chaque itération de l'algorithme
  - l'algorithme (MCMC-)SAEM (Delyon et al., 1999 ; Kuhn and Lavielle, 2005) : on réutilise les simulations des itérations précédentes
- La convergence des algorithmes a été étudiée dans le cas de la famille exponentielle

Algorithme EM - approximation de l'étape E

- À chaque itération de l'algorithme EM, on est ramené à des calculs d'espérance conditionnelle de différentes statistiques sous la loi des effets aléatoires ϕ sachant les observations y, $f(\phi \mid y; \theta)$
- Ce calcul est en général non explicite

$\rightarrow$ variantes stochastiques de l'algorithme EM

- Méthodes basées sur des simulations de type Monte Carlo par chaîne de Markov (MCMC) :
  - l'algorithme MCMC-EM (Wei and Tanner, 1990) : on génère une nouvelle chaîne à chaque itération de l'algorithme
  - l'algorithme (MCMC-)SAEM (Delyon et al., 1999 ; Kuhn and Lavielle, 2005) : on réutilise les simulations des itérations précédentes
- La convergence des algorithmes a été étudiée dans le cas de la famille exponentielle

Algorithme MCMC-EM (Wei and Tanner, 1990)

- À chaque itération k de l'algo MCMC-EM, on génère une nouvelle chaîne de Markov $(\phi^{k,(1)}, \ldots, \phi^{k,(m_k)})$ et on approche la fonction Q par

$$\hat{Q}(\theta; \theta^k) = \frac{1}{m_k} \sum_{m=1}^{m_k} \log f(y, \phi^{k,(m)}; \theta)$$

- La taille de la chaîne m_k doit augmenter pour compenser l'erreur de Monte Carlo générée par les simulations

$\rightarrow$ utilisation d'un algorithme automatique (Caffo et al., 2005), basé sur la propriété de monotonie de l'algorithme EM

  - basé sur le calcul d'un intervalle de confiance pour ΔQ entre deux itérations successives
  - borne inférieure $< 0 \rightarrow$ rejet du candidat θ^k et la chaîne continue
  - borne supérieure $<$ à un seuil $\rightarrow$ définition d'une règle d'arrêt

Algorithme (MCMC-)SAEM (Delyon et al., 1999)

- À chaque itération k de l'algo (MCMC)-SAEM, on génère une nouvelle chaîne de Markov $(\phi^{k,(1)}, \ldots, \phi^{k,(m_k)})$ et on réutilise les simulations précédentes grâce à une approximation stochastique (Robbins and Monro, 1951) :

$$\hat{Q}(\theta;\theta^k) = \hat{Q}(\theta;\theta^{k-1}) + \gamma_k \left[\frac{1}{m_k} \sum_{m=1}^{m_k} \log f(y, \phi^{k,(m)};\theta) - \hat{Q}(\theta;\theta^{k-1}) \right]$$

- La convergence s'obtient avec une taille de chaîne m_k constante et faible
- On choisit (Kuhn and Lavielle, 2005)

$$\gamma_k = \begin{cases} 1 & \text{pour } 1 \leq k \leq K_1 \\ \dfrac{1}{k - K_1 + 1} & \text{pour } K_1 < k \leq K_1 + K_2 \end{cases}$$

où K_1 et K_2 sont fixés

Modèle Greenlab de population - données simulées

Comparaison des algorithmes MCMC-EM et SAEM sur données simulées

- 3 paramètres aléatoires : μ (efficience), s^{pr}, a_r (allocation aux racines)
- 50 plantes virtuelles
- MCMC-EM : version automatique
- SAEM : K_1 et K_2 fixés
- Comparaison de différents algorithmes MCMC pour identifier le plus approprié
- Dix réalisations indépendantes de chaque algorithme
- Comparaison des intervalles de confiance calculés par méthode de Louis (Louis, 1982) (matrice d'information de Fisher) et Bootstrap

Modèle Greenlab de population - simulations MCMC

Simulations MCMC pour générer la chaîne à chaque itération de l'algorithme :

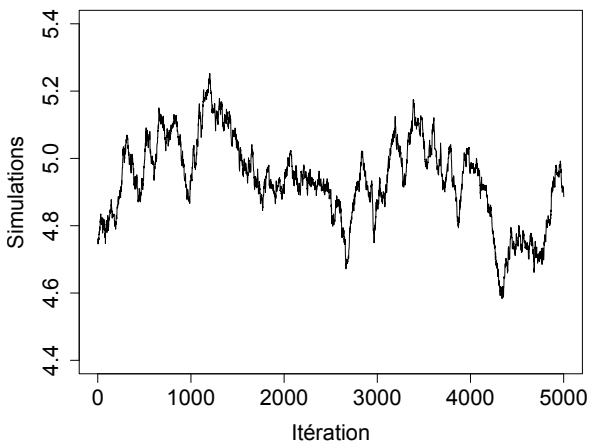
- par Metropolis-Hastings ou échantillonneur de Gibbs hybride $\rightarrow$ choix d'une loi instrumentale
- ce choix influence la vitesse de convergence vers la loi cible
  - loi marginale (multi ou unidimensionnelle)
  - marche aléatoire $\rightarrow$ attention au choix de la variance

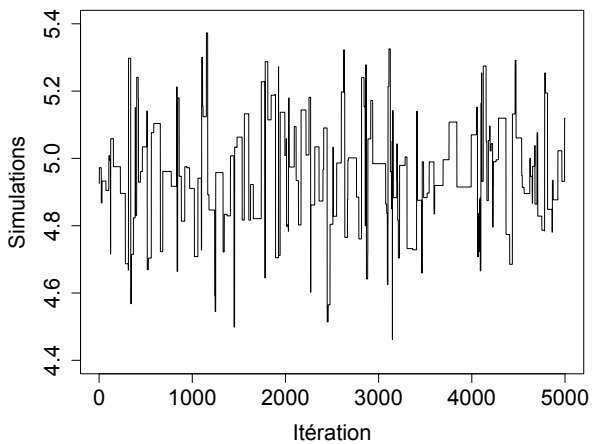
$\Rightarrow$ utilisation de marches aléatoires adaptatives (Andrieu and Thoms, 2008)

Modèle Greenlab de population - simulations MCMC

Simulations MCMC pour générer la chaîne à chaque itération de l'algorithme :

- par Metropolis-Hastings ou échantillonneur de Gibbs hybride → choix d'une loi instrumentale
- ce choix influence la vitesse de convergence vers la loi cible
  - loi marginale (multi ou unidimensionnelle)
  - marche aléatoire → attention au choix de la variance





⇒ utilisation de marches aléatoires adaptatives (Andrieu and Thoms, 2008)

Modèle Greenlab de population - simulations MCMC

Performances des différentes lois instrumentales (algo MCMC-EM automatique) : Moyenne (Min - Max) sur 10 réalisations indépendantes

	Itérations	**Taille finale de la chaîne**	**Temps d'exécution**
		Metropolis-Hastings	
Marginale	500 (500 - 500)	1119 (337 - 2966)	12h08 (8h21 - 21h24)
AMH	48 (21 -72)	21811 (3614 - 46461)	5h12 (1h37 - 14h29)
AMH Global	42 (16 - 57)	23028 (13354 - 36157)	5h04 (2h46 - 6h52)
		Échantillonneur de Gibbs hybride	
Marginale	91 (55 - 128)	22980 (7917 - 40151)	11h39 (5h48 - 20h11)
AhGs CW	72 (30 - 145)	23465 (5015 - 41604)	3h40 (1h11 - 5h34)
AhGs Global	45 (26 - 72)	28656 (15877 - 45080)	10h55 (6h32 - 18h31)

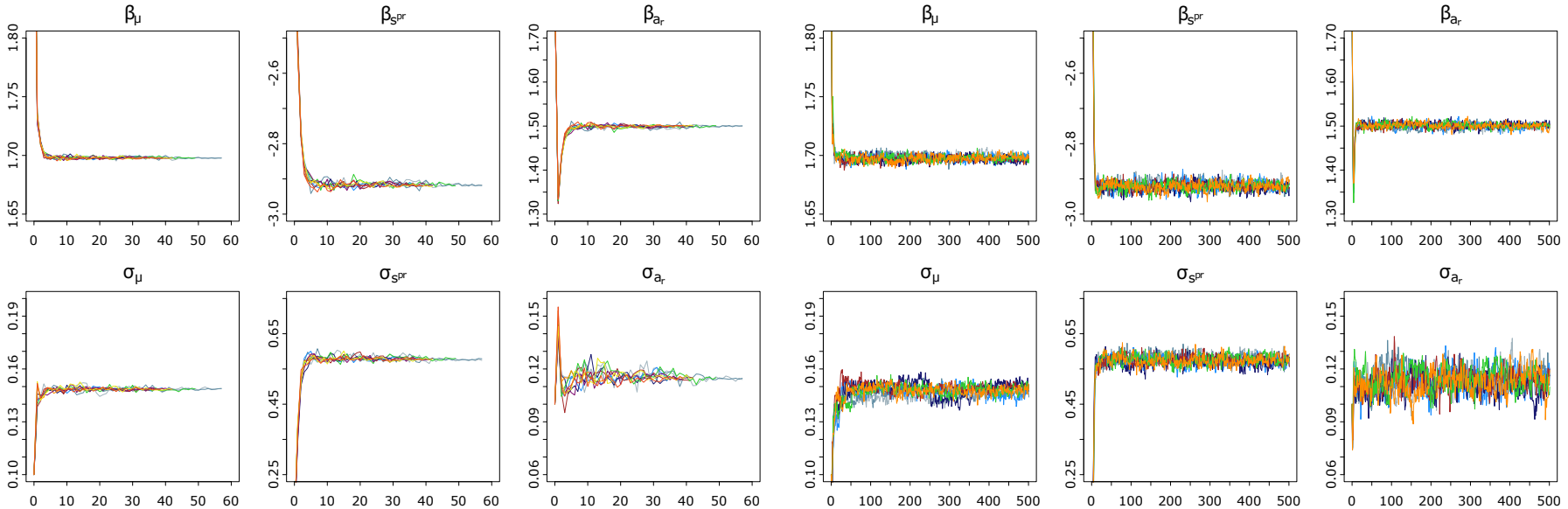
Modèle Greenlab de population - simulations MCMC

Performances des différentes lois instrumentales (algo MCMC-EM automatique) : Moyenne (Min - Max) sur 10 réalisations indépendantes

	Itérations	**Taille finale de la chaîne**	**Temps d'exécution**
		Metropolis-Hastings	
Marginale	500 (500 - 500)	1119 (337 - 2966)	12h08 (8h21 - 21h24)
AMH	48 (21 -72)	21811 (3614 - 46461)	5h12 (1h37 - 14h29)
AMH Global	42 (16 - 57)	23028 (13354 - 36157)	5h04 (2h46 - 6h52)
		Échantillonneur de Gibbs hybride	
Marginale	91 (55 - 128)	22980 (7917 - 40151)	11h39 (5h48 - 20h11)
AhGs CW	72 (30 - 145)	23465 (5015 - 41604)	3h40 (1h11 - 5h34)
AhGs Global	45 (26 - 72)	28656 (15877 - 45080)	10h55 (6h32 - 18h31)

Modèle Greenlab de population - données simulées

Algorithme MCMC-EM



(a) MH - MA adaptative globale

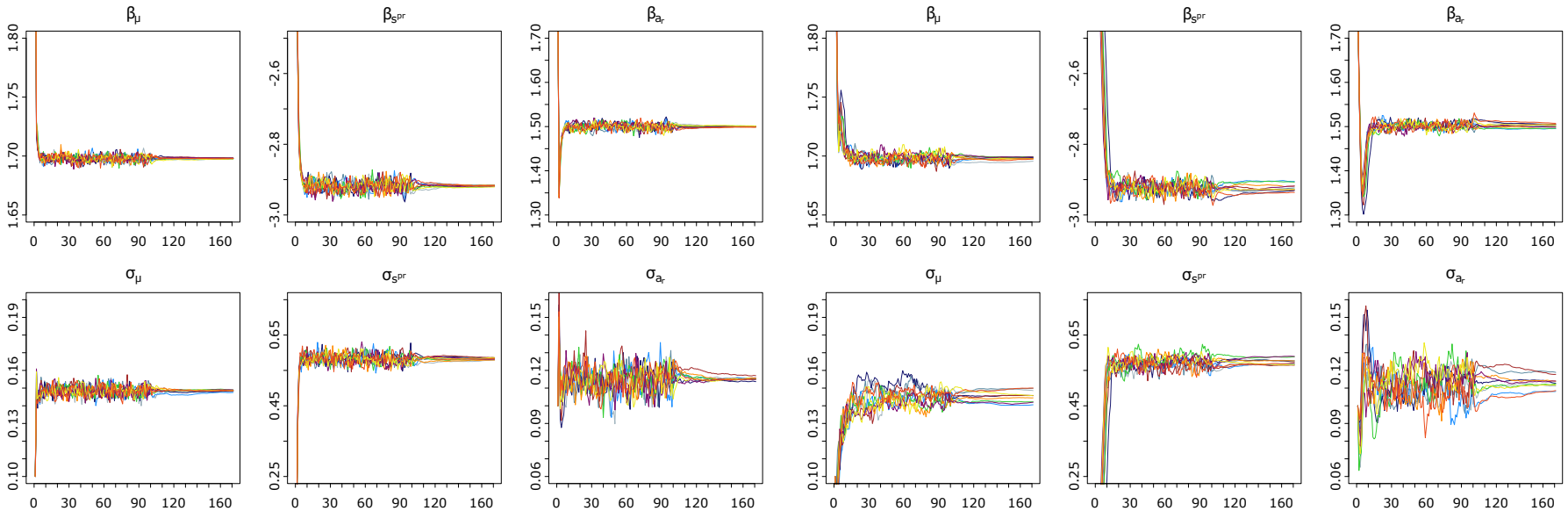
(b) MH - loi marginale

- Loi marginale non adaptée : plus grande variabilité

→ Marche aléatoire à schéma adaptatif plus efficace

Modèle Greenlab de population - données simulées

Algorithme SAEM



(c) MH - MA adaptative globale

(d) MH - loi marginale

- Loi marginale non adaptée : plus grande variabilité

→ Marche aléatoire à schéma adaptatif plus efficace

Modèle Greenlab de population - données simulées

- Résultats satisfaisants pour les deux algorithmes
- Algorithme automatique pour MCMC-EM permet de réduire le temps de calcul par rapport à une augmentation déterministe
- Algorithme SAEM plus rapide dans sa version non automatique sur données simulées
- Meilleure approximation de la loi cible par marche aléatoire adaptative globale et composante par composante :
  - augmentation suffisante de la taille de la chaîne
  - réduction de la variabilité entre réalisations indépendantes
  - diminution du temps de calcul
- Résultats similaires pour intervalles de confiance par méthode de Louis (matrice de Fisher) ou Bootstrap

Modèle Greenlab de population - données réelles

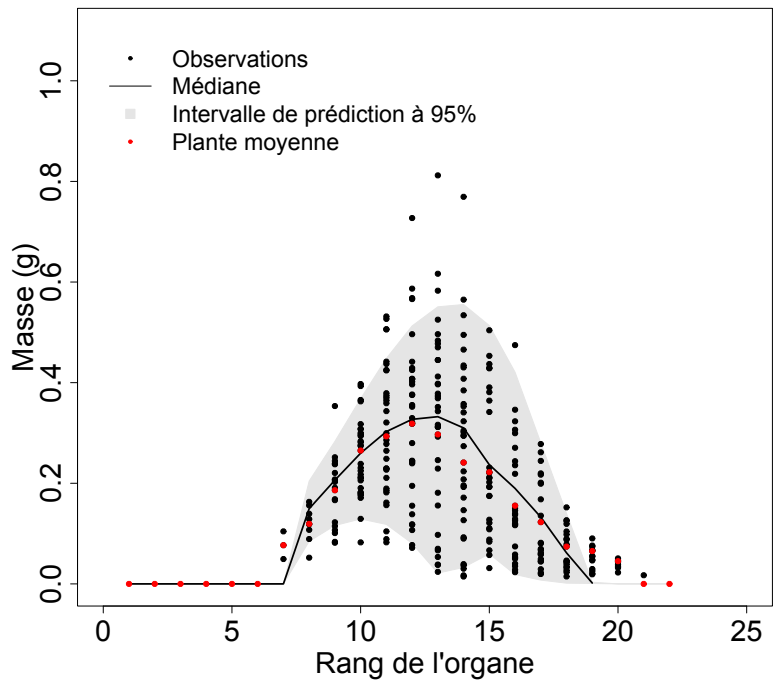
→ Application aux données colza (INRA Grignon, UMR EGC (A. Mathieu, A. Jullien)) :

- profils foliaires de 34 plantes
- stade rosette : un seul type d'organe, 4 paramètres : μ, s^{pr}, a_l, b_l
  - comparaison des modèles contenant 2, 3 ou 4 paramètres aléatoires (AICc et BIC)
  - comparaison de deux modèles d'erreur (additive ou log-additive)
- simulations MCMC par algorithme Metropolis-Hastings et marche aléatoire adaptative globale

Modèle Greenlab de population - données réelles

- Modèles à erreur additive meilleurs qu'avec erreur log-additive
- Deux paramètres aléatoires sélectionnés : μ (efficience de conversion de la lumière) et a_l (paramètre pour l'allocation de biomasse) :

  - paramètre a_l lié à la première phase de la courbe d'allocation : certains organes sont encore en expansion
  - ajout du paramètre de compétition s^{pr} ne permet pas d'améliorer les résultats : faible effet de la compétition au stade rosette (Jullien et al., 2011)



Modèle Greenlab de population

Conclusion

- Extension du modèle Greenlab à l'échelle de la population pour prendre en compte la variabilité
- Algorithme MCMC-EM automatique moins variable que SAEM (non auto) mais des problèmes numériques peuvent survenir
- Résultats satisfaisants sur le colza

Perspectives

- Développer une version automatique de l'algorithme SAEM
- Prendre en compte les effets fixes
- Matrice de covariance non diagonale pour les effets aléatoires
- Prise en compte des bruits de modélisation (Trevezas and Cournède, 2013)
- Validation plus complète du modèle proposé

Modèle Greenlab de population

Conclusion

- Extension du modèle Greenlab à l'échelle de la population pour prendre en compte la variabilité
- Algorithme MCMC-EM automatique moins variable que SAEM (non auto) mais des problèmes numériques peuvent survenir
- Résultats satisfaisants sur le colza

Perspectives

- Développer une version automatique de l'algorithme SAEM
- Prendre en compte les effets fixes
- Matrice de covariance non diagonale pour les effets aléatoires
- Prise en compte des bruits de modélisation (Trevezas and Cournède, 2013)
- Validation plus complète du modèle proposé