Université Paris-Sud

Mémoire de Magistère de Mathématiques

2015 - 2019

Inférence bayésienne sur des modèles de croissance de plantes hétérogènes en interaction.

Julie HEMONT

Table des matières

Première partie

Cursus au sein du magistère

À l'issue de mes trois années en classes préparatoires en filière Physique et Sciences de l'ingénieur, effectuées au Lycée Pothier à Orléans, indécise, j'étais attirée par les mathématiques théoriques, mais pas seulement. J'avais hâte de pouvoir les appliquer dans des domaines tels que la physique ou la biologie. Attirée également par les métiers de l'enseignement et soutenue par mes professeurs de l'époque, je postule au Magistère de Mathématiques du Département de Mathématiques d'Orsay. J'ai eu la chance de pouvoir intégrer cette formation en septembre 2015.

1 Licence 3 : Mathématiques Fondamentales et Appliquées

Durant l'année universitaire 2015-2016, j'ai suivi le parcours Mathématiques Fondamentales et Appliquées en licence 3. L'année de L3 a été l'occasion pour moi de consolider mes bases en mathématiques, notamment dans les domaines peu étudiés dans la filière que j'avais choisie en classe prépa.

Au premier semestre, j'ai suivi l'enseignement d'Algèbre qui m'a introduit les corps finis. J'ai compris l'importance de faire des dessins avant de se lancer dans une démonstration avec les cours d'Intégration et Analyse de Fourier et de Topologie et Calcul Différentiel. J'ai également travaillé mon esprit de groupe durant le cours d'Anglais. Pour compléter cette formation, j'ai choisi de suivre les options d'Algorithmique et de Mathématiques et Biologie. Le magistère m'a offert la possibilité de suivre deux options au lieu d'une, ce qui m'a été profitable puisque ces deux enseignements sont très certainement ceux qui m'ont le plus servi lors de mon stage de fin d'études.

Au second semestre, j'ai suivi le cours de Théorie de la Mesure et Probabilités, où l'humour était au rendez-vous. J'ai aussi suivi le cours d'Equations Différentielles. Le cours d'Algèbre m'a montré que le travail et la persévérance font des miracles. J'ai découvert l'analyse complexe avec l'enseignement de Fonctions Holomorphes. De plus, des cours d'Algèbre effective et de Modélisation en analyse et probabilités m'ont permis d'appliquer, sur ordinateur, les connaissances acquises durant l'année. Le cursus du magistère prévoit un cours spécifique de Topologie Générale. Je ai compris par la suite, en préparant l'agrégation, l'importance des objets utilisés dans ce cours très théorique. Ce second semestre de licence 3 a été l'occasion pour moi de travailler sur mon premier projet de recherche. J'ai étudié, encadrée par Clémence LABROUSSE, le théorème de Jordan.

En fin d'année, j'ai effectué un stage de trois semaines à l'Institut de Biologie Intégrative de la Cellule de l'Université Paris-Sud. J'ai travaillé sous la direction de Jean LEHMANN au sein de l'équipe de Daniel GAUTHERET. J'y ai découvert le fonctionnement d'un laboratoire de recherche. J'ai eu l'opportunité de présenter mes travaux, à la frontière entre Mathématiques et Biologie, à l'équipe

de bio-informaticiens. Ce stage a conforté mon envie de poursuivre en Master de Mathématiques pour les Sciences du Vivants.

2 Master 1 : Mathématiques Fondamentales

Bien entourée, cette année de Master 1 (2016-2017) a été l'année la plus riche en rencontres de tout mon cursus universitaire. J'ai eu la chance d'y recevoir des enseignements de grande qualité et de travailler avec un groupe d'amis soudés. Je me dois d'évoquer le BDE Math$\sim$ que j'ai eu l'honneur de présider.

Les cours que j'ai suivi étaient extrêmement variés. C'était un choix personnel lié à mes ambitions futures : obtenir l'agrégation de Mathématiques. Ce choix m'a permis d'avoir une culture plus large et de découvrir qu'avec le travail, tout domaine devient accessible.

J'ai choisi de suivre les enseignements de Probabilités et de Statistiques, complétés par un module de Mathématiques Assistées par Ordinateur. Les cours d'Analyse et Mathématiques Générales m'ont donné des bases plus solides pour l'agrégation. Plus particulièrement, les enseignements de Pierre-Guy PLAMONDON m'ont réconcilié avec l'algèbre. J'ai également suivi un cours aussi difficile qu'intéressant de Théorie Spectrale et Analyse Harmonique, dispensé par Frédéric PAULIN, qui s'est révélé très utile pour mes deux Masters 2, et un cours de Traitement et Analyse d'Images qui avait lieu à l'ENS Cachan.

Encadrée par Laurent MOONENS et en binôme avec Christian MARGUERITE, mon sujet de TER portait sur l'étude des théorèmes fondamentaux de l'intégration en dimension n. J'avais choisi ce sujet dans le but d'approfondir le cours d'Analyse qui nous avait été donné au premier semestre. Ce TER a été un vrai plaisir : j'ai appris à travailler en groupe, à communiquer des résultats, à partager mes doutes, à se questionner ensemble et à rédiger rigoureusement un document de Mathématiques.

3 Année de césure : Concours de l'agrégation

Après mon année de Master 1, j'ai décidé de préparer le concours de l'agrégation dans le Master 2 de l'Université Paris-Sud, Formation à l'Enseignement Supérieur. Grâce à mes capacités acquises les années précédentes, j'étais particulièrement à l'aise dans l'épreuve d'option Probabilités et Statistiques. La préparation de cette option a renforcé mes automatismes d'algorithmique.

J'ai choisi de réaliser un stage en Lycée pour voir le métier d'enseignant. J'ai été frustrée de ne pas avoir complètement la main et de ne pas pouvoir enseigner librement. J'ai confirmé pendant ce stage ma volonté de devenir enseignante.

C'est avec une grande fierté que je suis devenue professeur agrégée en Juin 2018, à l'issu de trois années au Département de Mathématiques d'Orsay, riches autant en mathématiques qu'en humanité.

4 Master 2 : Mathématiques pour les Sciences du Vivant

Durant l'année universitaire 2018-2019, j'ai réalisé une année de Master 2 de recherche en Mathématiques pour les Sciences du Vivant, ce que je voulais depuis mon entrée au Magistère. Ce Master a la particularité d'être à l'interface entre les Mathématiques et la Biologie. La diversité des profils des étudiants en fait sa richesse. De plus, une ambiance intimiste s'est vite installée.

Un enseignement de Biologie est dispensé de manière intensive en début d'année pour se familiariser avec les problématiques particulières de ce Master. Le premier semestre est composé d'un tronc commun, idéal pour établir des bases solides et replacer tous les étudiants sur un pied d'égalité. J'ai donc étudié les Statistiques en Grande Dimension avec Christophe GIRAUD. Ce cours avait l'avantage d'être commun avec le Master de Mathématiques de l'Aléatoire, diversifiant encore davantage les profils des étudiants qui n'hésitaient pas, grâce à la bienveillance qui régnait, à poser toutes leurs questions. Les enseignements d'Optimisation et de Systèmes Dynamiques étaient pour moi l'occasion de réviser les pans de l'analyse les plus utiles pour les problèmes de Biologie. Le TP dispensé par Sylvain FAURE a été particulièrement intéressant. Un médecin est venu nous présenter les données et la modélisation. Nous avons travaillé sur des images 3D d'IRM de bustes dans le but d'estimer les paramètres dans un système dynamique.

Le second semestre m'a fourni les dernières clés pour débuter dans le domaine de la recherche. J'ai suivi les enseignements de Detection de ruptures d'Emilie LEBARBIER et de Modèles à structures latentes de Stéphane ROBIN, très en lien avec mon projet. Afin d'approfondir le domaine des systèmes dynamique, j'ai suivi les cours de Modèles d'équations aux dérivées partielles pour l'écologie, dispensé par Gael RAOUL et de Modèles aléatoires de population, donnés par Vincent Bansaye et Sylvie Méléard. Des modèles d'évolution de population nous ont été enseignés. Enfin, j'ai découvert les Modèles à effets mixtes par les enseignements de Marc LAVIELLE. J'y ai appris à me servir du logiciel Monolix, que j'ai utilisé ensuite au cours de mon stage.

Dans le cadre du magistère, j'ai eu le plaisir de suivre, comme cours supplémentaire, l'enseignement d'Outils probabilistes et statistiques pour l'étude de la diversité génétique d'une population. Amandine VEBER a su me captiver lors de ces pauses culturelles bien agréables.

Mon projet de M2 a été réalisé en binôme avec Perrine LACROIX. Encadrées par Marie-Laure MARTIN-MAGNIETTE et Etienne DELANNOY, nous avons travaillé sur le sujet "Réseaux de gènes et Stochastic Block Model". Ce projet nous a demandé beaucoup d'investissement. Nous avons dû anticiper la plupart des enseignements. Nous avons appris à manipuler le logiciel R, très utilisé en recherche en statistiques.

J'ai effectué mon stage de fin d'études au sein de l'équipe Biomathematics du Laboratoire de Mathématiques et Informatique pour la Complexité et les Systèmes (MICS) de CentraleSupelec, sous la tutelle de Paul-Henry COURNEDE, directeur de ce laboratoire, du 1er avril au 30 août 2019 (5 mois). Ce stage,

intitulé "Inférence bayésienne sur des modèles de croissance de plantes hétérogènes en interaction", m'a donné l'opportunité d'appliquer les notions que j'ai assimilées tout au long de mon cursus universitaire. J'ai utilisé mes supports de cours allant de la L3 au M2 recherche, comme par exemple pour l'optimisation, les simulations de lois ou encore les modèles mixtes. J'ai à nouveau appris un nouveau langage de programmation : Julia. Ce stage conclut parfaitement mes 4 années de travail au sein de l'Université Paris-Saclay.

5 Et ensuite ?

À la rentrée de septembre 2019, j'ai décidé de prendre un poste d'enseignante stagiaire. J'ai été affectée au lycée Jean Jaurès, à Chatenay-Malabry. Malgré une légère appréhension, je suis impatiente de faire face à ce nouveau défi. Le challenge est grand mais je sais pouvoir y arriver, mes anciens enseignants étant des exemples à suivre.

Deuxième partie

Présentation d'un sujet de recherche : Inférence bayésienne sur des modèles de croissance de plantes hétérogènes en interaction.

1 Contextes biologique et mathématique.

Même au sein d'une même espèce, les individus présentent des variations génétiques. Pour les plantes, elles s'observent par exemple par une capacité de résistance plus ou moins grande aux maladies, ou aux insectes ravageurs. D'un autre point de vue, au sein d'une parcelle agricole, les plantes sont soumises à des conditions environnementales différentes et la qualité du sol peut varier par endroits. Cette variabilité inter-individuelle peut également être attribuées à des micro-variations de l'environnement.

Un intérêt particulier est porté à cette variabilité et plus largement, aux cultures mixtes, où les plantes sont toutes différentes. En effet, les populations hétérogènes, par exemple lorsque l'on mélange différentes variétés de blé [Borg et al., 2018] ou différentes espèces [Tang et al., 2018], semblent présenter des avantages tels que la résistance à certaines maladies ou le transfert d'azote entre les plantes.

Ces exemples montrent la nécessité de tenir compte de la variabilité entre les plantes dans les modèles mathématiques que l'on utilise. Une possibilité est d'introduire des modèles de population qui permettent de décrire un comportement général de la population tout en préservant l'idée de variabilité entre les individus. Les modèle mixtes, inférés par exemple par des méthodes de Monte-Carlo par chaîne de Markov (MCMC) et algorithmes Espérance-Maximisation (EM) [Kuhn, 2004], trouvent leurs applications dans plusieurs domaines du vivant comme la pharmacodynamique [Comets et al., 2008] ou l'écologie [Bolker et al., 2009] mais également pour les modèles de croissance de plantes [Baey, 2014].

D'autre part, les plantes produisent de la biomasse par photosynthèse. La lumière, l'eau et les nutriments du sol sont des ressources vitales. Lorsque les plantes sont en forte densité, comme dans un champs, elles entrent en compétition pour les ressources.

Les plantes poussent donc en interaction : la croissance d'une plante dépend de la croissance des plantes qui lui sont voisines. Si les effets de la compétition pour de la lumière ont été étudiés pour des populations homogènes [Cournede et al., 2008], l'influence de la compétition sur la croissance de plantes formant une population hétérogène est, à notre connaissance, peu étudiée.

Le modèle de croissance considéré par la suite est un modèle dynamique, qui s'écrit sous la forme d'un modèle à incrément donnant l'état du système au temps $t + \Delta t$ en fonction de l'état au temps t et du taux de croissance associé au pas de temps Δt.

Les plantes sont représentées par des grandeurs caractéristiques telles que la taille de la tige ou la surface foliaire, etc. L'interaction est représentée par des facteurs de compétition, calculés entre paires d'individus.

Lors de la prise en compte de l'interaction entre les individus et lorsque la population est importante, l'estimation des paramètres d'un modèle de population hétérogène à partir de données expérimentales est difficile. Une approche bayésienne est privilégiée dans le cadre de ce travail pour permettre d'intégrer des connaissances *a priori* et pallier au problème de la faible quantité de données usuellement rencontré en estimation fréquentiste. On souhaite donc inférer des modèles hiérarchiques bayésiens, c'est-à-dire retrouver la loi *a posteriori* π des paramètres du modèle sachant un ensemble d'observations. L'inférence par méthode de type MCMC nécessite de simuler toute la population sans possibilité de paralléliser les calculs.

2 Problème d'inférence bayésienne sur un modèle de population pour un système dynamique.

2.1 Problématique

Modèle d'évolution - La première caractéristique du modèle étudié est qu'il s'agit d'un modèle dynamique. Chaque plante est décrite par son état au cours du temps. Alors, l'état du système est décrit par un système dynamique de la forme suivante :

$$\forall t > 0,\ \forall 1 \leq i \leq N,\ E_i(t+1) = F_{ti}\left(E_i(t), \theta_i, (E_{i'}(t), \theta_{i'})_{1 \leq i' \neq i \leq N}\right),\ \text{où}$$

- N est le nombre de plantes ;
- chaque plante est indexée par $i \in \{1, \cdots, N\}$;
- E_i est l'**état** de la plante i : ce sont les caractéristiques de la plante qui varient au cours du temps comme par exemple la taille de la tige ou la surface foliaire ;
- θ_i est le vecteur des **paramètres individuels** de la plante i : ce sont les caractéristiques intrinsèques de la plante, supposées invariantes au cours du temps ;
- F_t est la fonction de transition de la plante i au temps t : elle modélise l'influence de la population, représentée par la variable $(E_{i'}(t), \theta_{i'})_{1 \leq i' \neq i \leq N}$, sur le développement de la plante i.

Modèle de population - L'hétérogénéité de la population est représentée par une mesure de probabilité p_0^θ sur l'espace des paramètres individuels, $\theta = (\theta_i)_{1 \leq i \leq N}$, qui n'est pas réduite à une mesure de Dirac. Cette distribution donne la répartition des valeurs prises par les paramètres individuels. Les paramètres de cette loi, η_θ, sont qualifiés de *paramètres de population* et font partie des

hyperparamètres du modèle. Les paramètres individuels sont alors supposés distribués de manière indépendante selon $p_0^\theta(\,\cdot\,|\eta_\theta)$.

La fonction F_t est également paramétrée par des hyperparamètres : les *paramètres d'interaction*, η_F. Ces hyperparamètres sont supposés indépendants des paramètres de population.

Modèle d'observation - Les observations de ce système sont supposées bruitées. Les observations, effectuées aux temps t_j pour $j \in \{1, \cdots, M\}$, sont donc la somme de la solution du système dynamique au temps t_j et d'un bruit gaussien.

$$E^o_{i,j} = E_i(t_j) + \varepsilon_{i,j} \text{ où } \varepsilon_{i,j} \underset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2).$$

Les observations $E^o = \left(E^o_{i,j}\right)_{1\leq i\leq N,\ 1\leq j\leq M}$ constituent une base de données.

Modèle bayésien - Le problème statistique consiste à identifier les hyperparamètres du modèle, qui donnent la distribution des paramètres individuels et la fonction de transition F_t en se reposant sur les observations collectées. L'approche bayésienne est privilégiée. Des lois *a priori* sont donc ajoutées sur les hyperparamètres et sur la variance d'observation σ^2.

Modèle graphique - Le modèle construit est résumé par le modèle graphique de la figure 2.1.

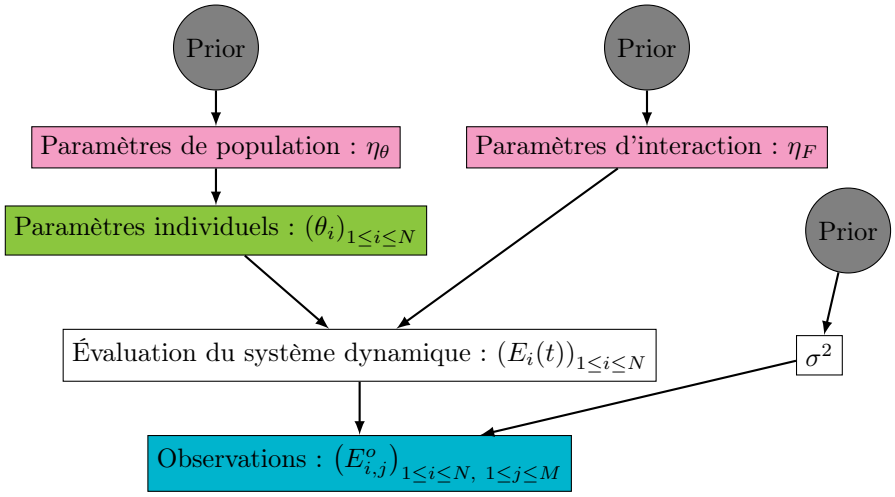


Figure 2.1 – Modèle graphique d'un modèle de population bayésien sur un système dynamique. En bleu, les données, en vert, les paramètres individuels, en rose les hyperparamètres et en gris, les loi *a priori*.

Comme il s'agit d'un modèle bayésien, l'objectif est de déterminer la loi *a posterori*, π, des paramètres du modèle, $x = \left(\eta_\theta, \eta_F, \theta, \sigma^2\right)$, sachant les données, E^o : $\pi(x) = p\left(\eta_\theta, \eta_F, \theta, \sigma^2 | E^o\right)$. Par le théorème de Bayes, la densité *a posteriori* est proportionnelle à la densité jointe :

$$p\left(\eta_\theta, \eta_F, \theta, \sigma^2 | E^o\right) = \frac{p\left(E^o, \eta_\theta, \eta_F, \theta, \sigma^2\right)}{p\left(E^o\right)} = \frac{p\left(E^o, \eta_\theta, \eta_F, \theta, \sigma^2\right)}{\int p\left(E^o, x\right) dx} \propto p\left(E^o, \eta_\theta, \eta_F, \theta, \sigma^2\right).$$

où $\propto$ désigne la proportionnalité sous x : ne sont conservés que les facteurs dépendants de x.

Le calcul de la densité *a posteriori* requiert le calcul d'une intégrale sur l'espace des paramètres. Ce calcul peut potentiellement être complexe et sur un espace de grande dimension, ce qui le rend infaisable. De plus, par la règle de Bayes et par les indépendances liées au modèle établi précédemment,

$$p\left(E^o, \eta_\theta, \eta_F, \theta, \sigma^2\right) = \underbrace{\prod_{i=1}^{N}\prod_{j=1}^{M} p\left(E^o_{i,j}|\eta_F, \theta, \sigma^2\right)}_{\text{Vraisemblance}} \times \underbrace{\prod_{i=1}^{N} p\left(\theta_i|\eta_\theta\right)}_{\substack{\text{Distribution} \\ \text{de population}}} \times \underbrace{p(\eta_\theta)p(\eta_F)p(\sigma^2)}_{prior}$$

Lorsque le modèle est complexe, par exemple pour les modèles non-linéaires, cette densité n'est pas calculable. Cependant, il existe des méthodes permettant de simuler une loi π dont on connaît l'expression analytique.

La méthode par acceptation/rejet n'est pas envisageable. En effet, la constante permettant une majoration de la densité par la densité d'une loi auxiliaire doit être connue, ce qui n'est pas possible en raison des calculs d'intégrales liées à cette densité particulièrement complexe.

On souhaite simuler des vecteurs x suivant une distribution de probabilité π. On cherche donc à avoir une suite de K vecteurs $(x_0, x_1, \ldots, x_{K-1})$ telle que la distribution des x_i approche π. L'idée des méthodes suivantes va être d'utiliser les propriétés d'ergodicité des chaînes de Markov.

Méthodes MCMC - Étant donnée une loi π, toute méthode consistant à produire une chaîne de Markov de loi invariante π est appelée méthode de *Monte-Carlo par chaîne de Markov* (MCMC). Les théorèmes ergodiques justifient l'utilisation de ces méthodes. [Tatarinova and Schumitzky, 2015]

L'échantillonnage de Gibbs, [Geman and Geman, 1984], introduit en 1984 par les frères Geman dans le cadre de la restauration d'images, est particulièrement adapté à l'inférence des modèles hiérarchiques. Avec cet algorithme, on simplifie les problèmes, notamment lorsque les lois conditionnelles sont connues et faciles à simuler. Si ce n'est pas le cas, on peut faire appel à l'algorithme de Metropolis-Hastings [Metropolis et al., 1953],[Hastings, 1970] pour chaque loi conditionnelle dans un algorithme dit *hybride*.

2.2 Metropolis-Hastings dans Gibbs

Les algorithmes de Metropolis-Hastings et d'échantillonnage de Gibbs sont combinés dans des algorithmes dits *hybrides*. Ces algorithmes sont utilisés par exemple lorsque les lois conditionnelles de la loi cible sont inconnues. L'idée est de combiner des algorithmes qui laissent invariants la distribution π. On obtient à nouveau un algorithme qui laisse lui aussi invariant la distribution π. L'algorithme Metropolis-Hastings Within Gibbs (MHWG) [Gilks et al., 1995] est un exemple d'algorithme hybride.

L'algorithme hybride MHWG est un algorithme itératif, reposant sur l'échantillonnage de Gibbs (on met à jour la variable d'intérêt x coordonnée par coordonnée) auquel on ajoute une étape de type acceptation-rejet de Metropolis-Hastings. Supposons que les distributions conditionnelles de π ne soient pas

simulables et ne soient connues qu'à constante multiplicative près. Ajouter une étape de Metropolis-Hastings permet de palier à ces difficultés. L'algorithme 1 présente une version de l'algorithme de Metropolis-Hastings Within Gibbs (MHWG) avec une loi de proposition q quelconque.

Algorithme 1 : Metropolis-Hastings Within Gibbs

Choisir une valeur initiale $x^{(0)}$
pour $k = 0 : K$ **faire**
  pour $i = 1 : d$ **faire**
    Simuler $x_i^* \sim q\left(x_i^{(k)}, \cdot\right)$
    Calculer

$$r_i(x_i^{(k)}, x_i^*) = \frac{\pi\left(x_i^* | x_1^{(k+1)}, \cdots, x_{i-1}^{(k+1)}, x_{i+1}^{(k)}, \cdots, x_d^{(k)}\right) q\left(x_i^{(k)}, x_i^*\right)}{\pi\left(x_i^{(k)} | x_1^{(k+1)}, \cdots, x_{i-1}^{(k+1)}, x_{i+1}^{(k)}, \cdots, x_d^{(k)}\right) q\left(x_i^*, x_i^{(k)}\right)}$$

    Simuler $u \sim \mathcal{U}([0,1])$
    Mettre à jour :

$$x_i^{(k+1)} = \begin{cases} x_i^* & \text{si } u \leq r_i(x_i^{(k)}, x_i^*) \\ x_i^{(k)} & \text{sinon} \end{cases}$$

  fin
fin
retourner $x^{(0:K)}$.

Lorsque la densité complète du modèle $\pi(x, E^o)$ est connue, pour obtenir la densité conditionnelle de x_i sachant toutes les autres variables, notée $\pi(x_i|\cdots)$, à constante près, il suffit d'utiliser la formule de Bayes. En effet, $\pi(x_i|\cdots) = \pi(\cdots)^{-1}\pi(x_i, \cdots)$ donc il suffit de ne conserver que les termes qui impliquent x_i dans la densité complète $\pi(x_i, \cdots) = \pi(x, E^o)$. Un exemple classique, utilisé dans la Section 4, d'algorithme de MHWG est d'utiliser une marche aléatoire comme loi de proposition : $x_i^* \sim \mathcal{N}(x_i^{(k)}, \sigma_i^2)$. Se pose alors la question du choix des variances d'exploration σ_i^2.

3 Adaptation d'un modèle GreenLab (Colza).

Dans cette section, on s'appuie sur le travail de Charlotte Baey. Dans cet article [Baey et al., 2018], un modèle GreenLab sur le colza a été appliqué à des données expérimentales dans le cadre d'un modèle mixte ne prenant pas en compte l'interaction entre les individus. On présente ici le modèle GreenLab modifié qui permet d'inclure de la compétition entre les individus. L'inférence sur ce modèle à partir de données expérimentales est réalisée dans la section 4.

Ce modèle est adapté au Colza en stade rosette. Il tient compte de la structure et du fonctionnement de la plante. Chaque feuille est caractérisée par son rang sur la tige principale. Les feuilles sont numérotées du pied au sommet. On considère une population de plantes de N individus. On notera $X_i = (x_i, y_i)$ la position de l'individu i pour $1 \leq i \leq N$.

Temps thermique et phyllochrone - Les plantes sont sensibles à leur environnement. Les plantes ne sont pas sensibles au temps calendaire mais au

temps thermique. Le temps thermique de l'expérience se calcule de la manière suivante. On relève $T(s)$ la température moyenne au jour s pour tout $0 \leq s < t$ puis on calcule le temps thermique au jour t :

$$\tau(t) = \int_0^t \max\left(T(s) - T_b, 0\right)\ ds \simeq \sum_{s=0}^{t-1} \max\left(T(s) - T_b, 0\right)$$

où T_b est la température de base, un seuil de température considéré comme constant. Dans le cas du colza, $T_b = 4,5°C$.

Le temps thermique doit dépasser une certaine valeur τ_i^{init} pour que la plante i émerge. Passé ce stade, la plante commence à intercepter de la lumière et débute donc sa production de biomasse par photosynthèse.

Le temps thermique qui sépare l'apparition de deux feuilles successives est appelé le *phyllochrone*. Dans ce modèle, il y a deux phases, l'une avant et l'une après le temps thermique de rupture τ^R. Si on note $\tau_{i,j}^{init}$ le temps thermique d'apparition de la feuille de rang j sur la plante i et $\omega_i^{(1)}$ (réciproquement $\omega_i^{(2)}$) le phyllochrone de la plante i pendant la période avant (récip. après) le temps thermique de rupture, alors le nombre de feuilles de la plante i au jour t est

$$N_i^{leaves}(t) = 1 + \left\lfloor \frac{\tau(t) - \tau_i^{init}}{\omega_i^{(1)}} \mathbb{1}_{\tau(t) \geq \tau_i^{init}} + \left(\frac{1}{\omega_i^{(1)}} - \frac{1}{\omega_i^{(2)}}\right)\left(\tau(t) - \tau^R\right)\mathbb{1}_{\tau(t) \geq \tau^R} \right\rfloor.$$

En particulier, si n_i feuilles sont apparues sur la plante i avant la rupture,

$$\begin{cases} \tau_{i,j}^{init} = \tau_i^{init} + (j-1)\ \omega_i^{(1)} & \text{pour tout } 1 \leq j \leq n_i; \\ \tau_{i,j}^{init} = \tau_i^{init} + (n_i - 1)\ \omega_i^{(1)} + (j - n_i)\ \omega_i^{(2)} & \text{pour tout } j \geq n_i. \end{cases}$$

Répartition de la biomasse - Comme les plantes de colza sont encore au stade rosette, on suppose que la biomasse n'est située que dans les feuilles, la graine étant attribuée à la feuille 1. Les feuilles reçoivent de la biomasse lorsqu'elles sont en phase d'expansion. Cette phase a une durée thermique fixe, notée τ_e. La fonction de distribution, appelée *puits de biomasse*, dans une feuille au rang j de la plante i est supposée proportionnelle à une loi bêta, paramétrée par a, un paramètre de population et b, une constante du modèle :

$$s_{i,j}(t) = \left(\frac{\tau(t) - \tau_{i,j}^{init}}{\tau_e}\right)^{a-1}\left(1 - \frac{\tau(t) - \tau_{i,j}^{init}}{\tau_e}\right)^{b-1} \mathbb{1}_{\tau_{i,j}^{init} \leq \tau(t) \leq \tau_{i,j}^{init} + \tau_e}.$$

Le quantité de biomasse totale demandée par la plante i au jour t est donc

$$d_i(t) = \sum_{j=1}^{N_i^{leaves}(t)} s_{i,j}(t).$$

Alors, la quantité $s_{i,j}(t)/d_i(t)$ représente la proportion de la biomasse qui sera allouée à la feuille j de la plante i au jour t. Si on note $F_i(t)$ la biomasse accumulée durant le jour t par la plante i, alors la masse totale d'une feuille au rang $j > 1$ de la plante i au début du jour t est

$$q_{i,j}(t) = \sum_{s=1}^{t-1} F_i(s)\frac{s_{i,j}(s)}{d_i(s)}.$$

Dans le cas particulier de la première feuille, il nous faut considérer la biomasse issue de la graine, notée q_0. On a alors

$$q_{i,1}(t) = q_0 + \sum_{s=1}^{t-1} F_i(s) \frac{s_{i,1}(s)}{d_i(s)}.$$

D'autre part, la quantité de biomasse totale produite par la plante i au début du jour t vaut

$$Q_i(t) = \sum_{s=1}^{t-1} F_i(s).$$

Surface active - La biomasse générée par une plante provient principalement de la photosynthèse. Il est donc nécessaire de calculer la *surface active*, c'est-à-dire la surface de feuille capable d'effectuer de la photosynthèse. Chaque feuille a un temps de vie τ_l pendant lequel elle peut faire de la photosynthèse. Passé ce délai, elle n'est plus considérée comme active. La surface active au début du jour t se calcule par l'intermédiaire de la masse active

$$q_i^{act}(t) = \sum_{j=1}^{N_i^{leaves}(t)} q_{i,j}(t) \mathbb{1}_{\tau(t) - \tau_{i,j}^{init} < \tau_{i,j}^{l}}.$$

En divisant par la masse de feuille par unité de surface e, on obtient la surface active au début du jour t : $s_i^{act}(t) = \dfrac{q_i^{act}(t)}{e}$. Le calcul de la surface active au jour t permet d'obtenir la quantité de biomasse produite durant le jour t.

Production de biomasse - La quantité initiale de biomasse est donnée par la masse de la graine, q_0. Après l'apparition de la première feuille, c'est-à-dire dès que le temps thermique vérifie $\tau(t) > \tau_{i,1}^{init}$, de la biomasse est produite grâce à la photosynthèse. La quantité de biomasse produite au jour t par la plante i, $F_i(t)$, est classiquement donnée par une loi de Beer-Lambert : pour tout $t \in \mathbb{N}^*$ tel que $\tau(t) > \tau_{i,1}^{init}$,

$$F_i(t) = u_t \ \mu \ s^{pr} \left(1 - \exp\left(-k_B \frac{s_i^{act}(t)}{s^{pr}}\right)\right),$$

où u_t est le rayonnement photo-synthétiquement actif (PAR) et traduit l'effet des conditions environnementales du jour t, μ traduit l'efficacité de la plante pour convertir la lumière en biomasse, s^{pr} traduit l'effet de la densité en plantes dans laquelle pousse la plante i, k_B est le coefficient d'absorption de la loi de Beer-Lambert et $s_i^{act}(t)$ est la surface de feuille active pour la photosynthèse au début du jour t.

Dans cette étape du modèle GreenLab, il est alors possible d'introduire de la compétition avec une fonction de compétition. Cette fonction de compétition, pour une paire d'individus i et i', dépend de la biomasse totale de chacun des individus, Q_i et $Q_{i'}$, et de la distance entre les plantes, $\|X_i - X_{i'}\|$. La compétition exercée par la plante i' sur la plante i est :

$$C\left(Q_i, Q_{i'}, \|X_i - X_{i'}\|\right) = \frac{\tanh\left(\dfrac{Q_{i'}}{s_m}\right)}{2\left(1 + \dfrac{\|X_i - X_{i'}\|^2}{\sigma_x^2}\right)} \left(1 + \tanh\left(\frac{Q_{i'} - Q_i}{\sigma_S}\right)\right),$$

où s_m est une constante et σ_x^2 et σ_S sont des paramètres de compétition. La fonction de production au jour t devient :

$$F_i(t) = u_t\, \mu\, s^{pr} \left(1 - \exp\left(-k_B \frac{s_i^{act}(t)}{s^{pr}}\right)\right) \left[1 - \frac{1}{N-1} \sum_{i' \neq i} C\left(Q_i(t), Q_{i'}(t), \|X_i - X_{i'}\|\right)\right].$$

Fonction de transition - La population de plantes que l'on considère peut être caractérisée, au jour t, par un état E_t qui contient l'état, $e_{i,t}$, de chaque individu i tel que $1 \leq i \leq N$, au jour t. D'après ce qui précède, pour tout individu $i \in \{1, \cdots, N\}$, $e_{i,t}$ contient

- $N_i^{leaves}(t)$ le nombre de feuilles au début du jour t ;
- $Q_i(t)$ la quantité de biomasse totale au début du jour t ;
- $q_i^{act}(t)$ la masse active au début du jour t ;
- $s_i^{act}(t)$ la surface active au début du jour t ;
- $F_i(t)$ la quantité de biomasse produite durant le jour t ;
- pour chaque feuille $j \in \{1, \cdots, N_i^{leaves}(t)\}$, $s_{i,j}(t)$ la fonction de puits de biomasse et $q_{i,j}(t)$ la quantité de biomasse au début du jour t.

Pour tout jour $t \in \mathbb{N}^*$, l'équation de transition de l'état E_t à l'état E_{t+1} s'écrit

$$E_{t+1} = F_t(E_t, \mathrm{Env}_t, x, \eta, \nu),$$

où

- Env_t regroupe les paramètres environnementaux du jour t, c'est-à-dire la température, $T(t)$, et le PAR, u_t ;
- $x = (\mu, a, \sigma_x, \sigma_S)$, les paramètres du modèle ;
- $\eta = (\tau_e, \tau_l, T_b, k_B, s^{pr}, e, q_0, s_m, \tau^{init}, \tau^R, b)$, les constantes du modèle ;
- $\nu = (\nu_i)_{1 \leq i \leq N}$ contient les phyllochrones individuels : $\forall i$, $\nu_i = (\omega_i^{(1)}, \omega_i^{(2)})$;
- F_t est la fonction de transition.

Modélisation. - Comme la compétition n'intervient pas dans le nombre de feuilles, on va distinguer deux modèles : un modèle lié au nombre de feuilles sur les plantes et un modèle lié à la production de biomasse.

Dans un premier temps, intéressons-nous au modèle lié au nombre de feuilles. On fait l'hypothèse que l'on est dans un modèle de population, c'est-à-dire un modèle mixte. Les effets fixes sont τ^R et τ^{init} et les effets mixtes portent sur les phyllochrones. Comme les phyllochrones sont positifs, il est préférable de travailler avec leurs logarithmes. Alors, pour tout individu i,

$$\begin{cases} \log\left(\omega_i^{(1)}\right) = \log\left(\omega_{pop}^{(1)}\right) + \eta_i \text{ où } \eta_i \underset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_1^2); \\ \log\left(\omega_i^{(2)}\right) = \log\left(\omega_{pop}^{(2)}\right) + \xi_i \text{ où } \xi_i \underset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_2^2). \end{cases}$$

On suppose de plus que l'on dispose d'une base de données d'observations bruitées du nombre de feuilles à différents jours : pour tout individu i tel que $1 \leq i \leq N$ et tout temps t_j tel que $1 \leq j \leq M$,

$$N_{i,j}^o = N_i^{leaves}(t_j) + \varepsilon_{i,j} \text{ où } \varepsilon_{i,j} \underset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2).$$

Dans le second modèle, les paramètres liés au nombre de feuilles sont supposés connus et ne sont donc pas des paramètres du modèle. Dans ce modèle lié

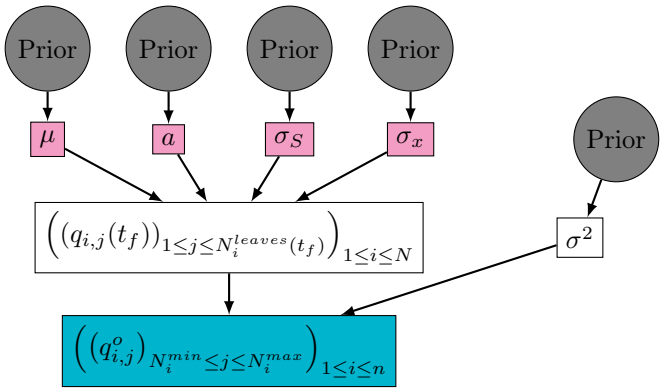


Figure 3.3 – Modèle graphique du modèle GreenLab avec compétition. En gris, les lois *a priori*, en rose, les hyperparamètres, en bleu, les données.

à la production de biomasse, les paramètres sont les paramètres de population μ et a et les paramètres de compétition σ_x et σ_S.

Les paramètres sont positifs, donc les lois *a priori* choisies sont des lois Log-Normales ou des Inverse-Gamma, paramétrées selon les connaissances biologiques comme par exemple les ordres de grandeurs des observations typiques.

$$\begin{cases} \mu \sim \mathcal{LN}(m_\mu, \sigma_\mu) \\ a \sim 1 + \mathcal{LN}(m_a, \sigma_a) \end{cases} \quad \text{et} \quad \begin{cases} \sigma_S \sim \mathcal{IG}(\alpha_S, \beta_S) \\ \sigma_x^2 \sim \mathcal{IG}(\alpha_x, \beta_x) \end{cases} \tag{1}$$

On suppose que l'on dispose d'une base de données d'observations portant sur n individus. Les données pour un individu i sont des relevés de biomasse de certaines feuilles, $N_i^{min} \leq j \leq N_i^{max}$, au temps final t_f :
$q^o_{i,j} = q_{i,j}(t_f) + \varepsilon_{i,j}$ où $\varepsilon_{i,j} \underset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$.

On obtient ainsi un profil pour chaque individu. Un profil typique est tracé sur la figure 3.2.

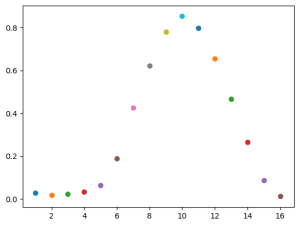


Figure 3.2 – Profil typique de la quantité de biomasse (en g) en fonction du rang sur la tige.

On ajoute une loi *a priori* sur la variance d'observation : $\sigma^2 \sim \mathcal{IG}(e, f)$.

Sans variabilité individuelle sur les paramètres, l'étage des paramètres individuels n'apparaît pas dans le modèle graphique résumé sur la figure 3.3.

Ce modèle graphique nous permet d'écrire la densité complète.

$$p(q^o, x, \sigma^2) = \underbrace{\prod_{i=1}^{n} \prod_{j=N_i^{min}}^{N_i^{max}} p(q^o_{i,j}|x, \sigma^2)}_{\text{Vraisemblance}} \underbrace{p(\mu|m_\mu, \sigma_\mu)\ p(a|m_a, \sigma_a)\ p(\sigma_x^2|\alpha_x, \beta_x)\ p(\sigma_S|\alpha_S, \beta_S)\ p(\sigma^2|e, f)}_{Prior}$$

4 Application sur des données expérimentales.

Les données à notre disposition proviennent d'expérimentations sur du colza, réalisées en 2012-2013 à la station expérimentale de l'INRA à Grignon (78), sur la variété Pollen [Baey et al., 2018].

Deux types de mesures ont été effectués : des relevés du nombre de feuilles ont été faits de manière hebdomadaire et des mesures de masses des feuilles ont été réalisées à la sortie de l'hiver, après 200 jours d'expérience.

Les données sur le nombre de feuilles permettent l'estimation des paramètres d'organogenèse reposant sur le modèle présenté dans la section 3. Les paramètres individuels ont été estimés sur 83 plantes avec un modèle mixte et à l'aide du logiciel Monolix [Lixoft SAS, 2018].

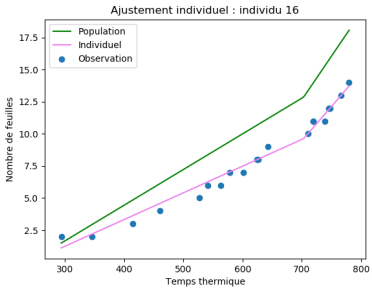


Figure 4.4 – Ajustement individuel avec Monolix pour l'individu 16 sur le modèle d'organogenèse.

Les calculs de critères BIC ont permis de déterminer les effets que l'on peut considérer comme fixes : τ^R et τ^{init}. Les paramètres individuels sont donc les deux phyllochrones $\omega^{(1)}$ et $\omega^{(2)}$. Un exemple d'ajustement individuel réalisé avec Monolix est tracé sur la figure 4.4. Les résultats obtenus pour les paramètres de population sont résumés dans le tableau 4.1.

τ^{init}	τ^R	$\omega^{(1)}_{pop}$	σ_1	$\omega^{(2)}_{pop}$	σ_2	σ	$Correlation$
240.26	703.10	35.97	0.10	14.72	0.21	0.63	0.62

Table 4.1 – Paramètres de population estimés avec le logiciel Monolix dans le modèle d'organogenèse GreenLab.

Des calculs préalables sont nécessaires pour la mise en oeuvre de l'algorithme. Pour appliquer l'algorithme de MHWG (algorithme 1), on a vu dans la section 2.2 qu'il nous suffisait d'écrire les densité des lois conditionnelles à constante multiplicative près en ne conservant que les termes faisant intervenir le paramètre. La notation $p(x|\cdots)$ désigne la densité conditionnelle de x sachant tous les autres paramètres du modèle.

Pour un paramètre x, la notation $q^{mod}_{i,j}(x)$ désigne la quantité de biomasse au temps final sur la feuille de rang j de l'individu i calculée avec le modèle où tous les paramètres sont fixés sauf x. Le candidat pour l'étape d'acceptation/rejet, x^*, est à distinguer du paramètre courant, x. D'autre part, N_{obs} désigne le nombre d'observations.

Par exemple, pour le paramètre μ, la densité conditionnelle est

$$p(\mu|\cdots) \propto p(q^o|\mu, a, \sigma_x^2, \sigma_S)\, p(\mu|m_\mu, \sigma_\mu) \propto \frac{1}{\mu} \exp\left[-\frac{\log(\mu) - m_\mu}{2\sigma_\mu^2} - \frac{\sum_{i,j} \left(q^o_{i,j} - q^{mod}_{i,j}(\mu)\right)^2}{2\sigma^2} \right] \mathbb{1}_{\mu>0}$$

Afin d'optimiser au mieux les calculs et d'éviter les zéros machines, on travaille avec le logarithme du rapport. La loi *a priori* sur σ^2 est conjuguée. On obtient

$$p(\sigma^2|\cdots) = \mathcal{IG}\left(\frac{N_{obs}}{2} + e, \frac{1}{2}\sum_{i,j}\left(q_{i,j}^{o} - q_{i,j}^{mod}\right)^2 + f\right).$$

Le tableau 4.2 regroupe les valeurs des paramètres des lois *a priori* et des paramètres d'inférence utilisés pour appliquer un algorithme de MHWG. Les variances d'exploration sont différentes selon les paramètres. Comme le *prior* sur σ^2 est conjugué, l'étape d'acceptation/rejet de Metropolis-Hastings n'est pas nécessaire.

Lois *a priori*									
m_μ	σ_μ	m_a	σ_a	e	f	α_S	β_S	α_x	β_x
1.0	1.0	0	1.0	2.0	1.0	2.0	1.0	2.0	1.0

Paramètres d'inférence		
Variances d'exploration		Nombre d'itérations
$\sigma_{\mu,a}$	$\sigma_{\sigma_S,\sigma_x}$	K
0.01	0.1	100 000

Table 4.2 – Valeurs des paramètres utilisés pour inférer le modèle adapté de GreenLab avec compétition à l'aide d'un Metropolis-Hastings Within Gibbs.

L'algorithme a été initialisé sur une approximation numérique de l'estimateur des moindres carrés, θ_{LS}. La figure 4.5 présente l'évolution des états de la chaîne de Markov au cours des itérations de l'algorithme. On observe que l'algorithme a bien convergé. On peut donc établir la distribution empirique de la distribution stationnaire, représentée sur la figure 4.6.

Le résultat obtenu sur la figure 4.6 est satisfaisant. Cependant, une variabilité individuelle sur le paramètre a de la fonction de puits pourrait améliorer considérablement le modèle. Malheureusement, les temps de calculs importants nécessaires pour inférer ce modèle incluant de la variabilité individuelle sur l'un des paramètres empêchent d'observer la convergence de l'algorithme. On touche ici les limites des modèles de croissance de plantes en population hétérogène qui prennent en compte les interactions entre les plantes.

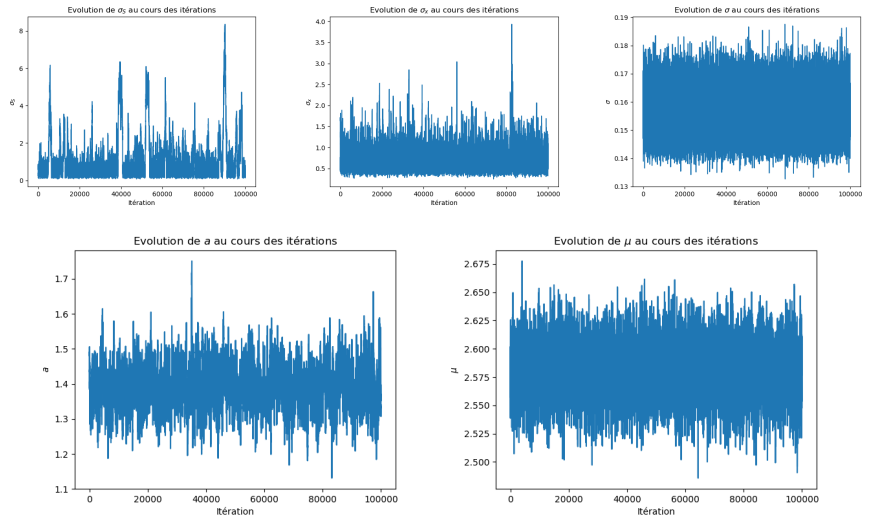


Figure 4.5 – Évolution de l'état des paramètres de la chaîne de Markov, pour le modèle adapté de GreenLab avec compétition, au cours des itérations de l'algorithme de Metropolis-Hastings Within Gibbs initialisé sur l'estimateur des moindres carrés θ_{LS}.

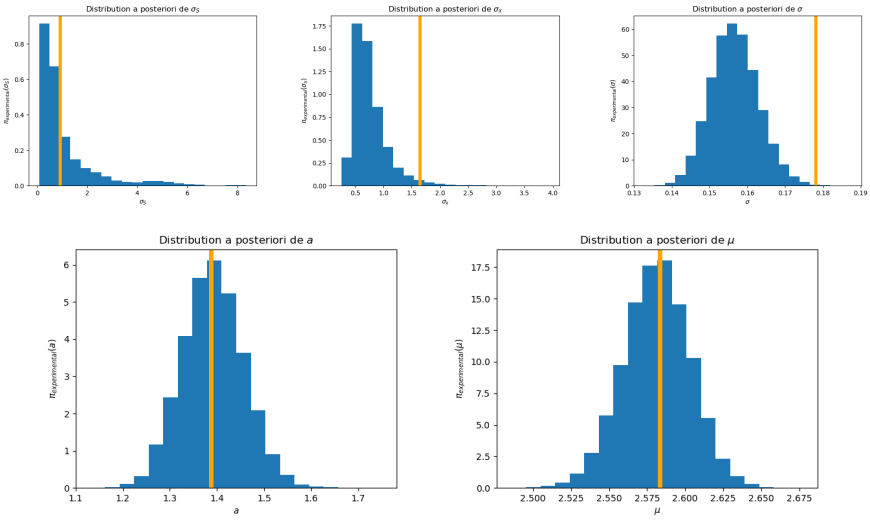


Figure 4.6 – En bleu, distributions empiriques, pour le modèle adapté de GreenLab avec compétition, des paramètres de la chaîne de Markov obtenue avec l'algorithme Metropolis-Hastings Within Gibbs initialisé sur l'estimateur des moindres carrés θ_{LS}. En orange, θ_{LS}.

Références

[Baey, 2014] Baey, C. (2014). Modelling inter-individual variability in plant growth models and model selection for prediction. Theses, Ecole Centrale Paris.

[Baey et al., 2018] Baey, C., Mathieu, A., Jullien, A., Trevezas, S., and Cournède, P.-H. (2018). Mixed-Effects Estimation in Dynamic Models of Plant Growth for the Assessment of Inter-individual Variability. Journal of Agricultural, Biological, and Environmental Statistics, 23(2) :208–232.

[Bolker et al., 2009] Bolker, B. M., Brooks, M. E., Clark, C. J., Geange, S. W., Poulsen, J. R., Stevens, M. H. H., and White, J.-S. S. (2009). Generalized linear mixed models : a practical guide for ecology and evolution. Trends in Ecology & Evolution, 24(3) :127 – 135.

[Borg et al., 2018] Borg, J., Kiær, L., Lecarpentier, C., Goldringer, I., Gauffreteau, A., Saint-Jean, S., Barot, S., and Enjalbert, J. (2018). Unfolding the potential of wheat cultivar mixtures : A meta-analysis perspective and identification of knowledge gaps. Field Crops Research, 221 :298 – 313.

[Comets et al., 2008] Comets, E., Brendel, K., and Mentré, F. (2008). Computing normalised prediction distribution errors to evaluate nonlinear mixed-effect models : The npde add-on package for R. Computer Methods and Programs in Biomedicine, 90(2) :154–66.

[Cournede et al., 2008] Cournede, P., Mathieu, A., Houllier, F., Barthélémy, D., and De Reffye, P. (2008). Computing competition for light in the GREENLAB model of plant growth : a contribution to the study of the effects of density on resource acquisition and architectural development. Annals of Botany, 101 :1207–1219.

[Geman and Geman, 1984] Geman and Geman (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. IEEE Transactions On Pattern Analysis And Machine Intelligence.

[Gilks et al., 1995] Gilks, W. R., Best, N. G., and Tan, K. K. C. (1995). Adaptive rejection metropolis sampling within gibbs sampling. Applied Statistics, 44(4) :455.

[Hastings, 1970] Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications. Biometrika, 57(1) :97–109.

[Kuhn, 2004] Kuhn, Estelle, L. M. (2004). Coupling a stochastic approximation version of em with an mcmc procedure. ESAIM : Probability and Statistics, 8 :115–131.

[Lixoft SAS, 2018] Lixoft SAS (2018). Monolix version 2018R1. Antony, France : Lixoft SAS. http ://lixoft.com/products/monolix/.

[Metropolis et al., 1953] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. The Journal of Chemical Physics, 21(6) :1087–1092.

[Tang et al., 2018] Tang, Q., Tewolde, H., LIU, H., Ren, T., Jiang, P., Zhai, L., Lei, B., Lin, T., and Liu, E. (2018). Nitrogen uptake and transfer in broad bean and garlic strip intercropping systems. Journal of Integrative Agriculture, 17(1) :220 – 230.

[Tatarinova and Schumitzky, 2015] Tatarinova, T. and Schumitzky, A. (2015). Nonlinear Mixture Models. IMPERIAL COLLEGE PRESS.

Troisième partie

Mémoires et rapports réalisés au cours de ma scolarité au magistère

1 Projet de recherche de L3 : Étude du théorème de Jordan.

THÉORÈME DE JORDAN

JULIE HÉMONT & STEPHAN KUNNE

Table des matières

1. Introduction

"Toute courbe continue fermée qui ne se croise pas elle-même sépare le plan en deux, l'intérieur et l'extérieur de la courbe."

L'apparence extrêmement simple de cette assertion est trompeuse. Au début du dix-neuvième, Bernhard Bolzano est le premier mathématicien à considérer ce problème comme un théorème nécessitant une preuve rigoureuse. Une première démonstration en est donnée par Camille Jordan à la fin du dix-neuvième siècle.

On donne ici deux démonstrations.

La première, purement géométrique, n'est valable que dans l'hypothèse d'une courbe régulière, c'est-à-dire de classe $\mathcal{C}^1$ par morceaux et dont la dérivée ne s'annule pas. En l'absence de cette hypothèse de régularité, il est difficile de montrer qu'au voisinage des points de la courbe le plan est effectivement séparé en deux.

La seconde démonstration fait intervenir des résultats d'analyse complexe, et relie la notion de point intérieur ou extérieur à une courbe à l'existence de logarithmes continus de fonctions définies sur la courbe. On montre en fait un résultat légérement plus général que le théorème de Jordan : si deux compacts du plan sont homéomorphes, alors leurs complémentaires admettent le même nombre de composantes connexes.

2. Définitions

Définition 1. *On dit qu'un chemin $\gamma : [a;b] \to X$ est un chemin de Jordan si γ est injectif. On dit qu'un lacet $\gamma : [a;b] \to X$ est un lacet de Jordan si la restriction de γ à $[a;b[$ est injective.*

Une courbe de Jordan simple est l'image d'un chemin de Jordan. Une courbe de Jordan fermée est l'image d'un lacet de Jordan. Un tel chemin (ou lacet) est appelé un paramétrage de la courbe.

Définition 2. *Soit $p \in \mathbb{C}$. Si γ est un lacet dans $\mathbb{C} \setminus \{p\}$, l'indice de γ par rapport au point p est le nombre* $\mathrm{Ind}(\gamma, p) = \frac{1}{2\pi i} \int_\gamma \frac{\mathrm{d}z}{z-p}$.

Définition 3. *On dit que deux lacets γ_0,$\gamma_1 : J \to \Omega$ sont homotopes dans Ω s'il existe une application continue $H : [0,1] \times J \to \Omega$ vérifiant les propriétés suivantes :*
(a) Pour tout $s \in [0,1]$, le chemin Γ_s défini par $\Gamma_s(t) = H(s,t)$ est un lacet;
(b) $\Gamma_0 = \gamma_0$ et $\Gamma_1 = \gamma_1$
On dit que l'appication H est une homotopie antre γ_0 et γ_1.

On fera référence à plusieurs reprises au théorème suivant :

Théorème 2.1. *Théorème de Borsuk : Soit K un compact de $\mathbb{R}^n$. Pour une application continue $f : K \to \mathbb{C}^*$, les propriétés suivantes sont équivalentes :*

1. *f admet un logarithme continu ;*
2. *f est homotope dans $\mathbb{C}^*$ à la fonction 1 ;*
3. *f possède une extension continue $F : \mathbb{R}^n \to \mathbb{C}^*$.*

3. Théorème de Jordan : énoncé

Théorème 3.1. *Soit $\Gamma \subset \mathbb{C}$ une courbe de Jordan fermée.*

Alors $\mathbb{C} \setminus \Gamma$ admet exactement deux composantes connexes, une bornée et une non bornée.

La frontière ∂O de chaque composante connexe O de $\mathbb{C} \setminus \Gamma$ est exactement Γ.

Soit $\gamma : [\alpha, \beta] \to \mathbb{C}$ un paramétrage de Γ. L'indice de γ par rapport à un point de $\mathbb{C} \setminus \Gamma$ est constant sur chaque composante connexe, identiquement égal à 0 sur la composante connexe non-bornée et à ± 1 sur la composante connexe bornée.

4. Résultats préliminaires de topologie du plan

Lemme 4.1. *Soit K un compact de $\mathbb{C}$. Alors $\mathbb{C} \setminus K$ admet une unique composante connexe non bornée.*

Preuve. Soit K un compact de $\mathbb{C}$. Alors en particulier K est borné : il existe $M > 0$ tel que $K \subset \overline{D}(0, M)$. Alors $\mathbb{C} \setminus \overline{D}(0, M)$ est connexe par arcs dans $\mathbb{C} \setminus K$. Donc $\mathbb{C} \setminus K$ admet une unique composante connexe non bornée, qui contient $\mathbb{C} \setminus \overline{D}(0, M)$.

Lemme 4.2. *Soit K un compact de $\mathbb{C}$. Chaque composante connexe de $\mathbb{C} \setminus K$ est ouverte dans $\mathbb{C}$, et le nombre de composantes connexes est au plus dénombrable.*

Preuve. K est en particulier fermé dans $\mathbb{C}$, donc $\mathbb{C} \setminus K$ est ouvert. Par conséquent toutes les composantes connexes de $\mathbb{C} \setminus K$ sont ouvertes. De plus $\mathbb{C}$ est un espace topologique à base dénombrable d'ouverts : $\mathbb{Q} + i\mathbb{Q}$ est dénombrable et dense dans $\mathbb{C}$, et toute partie ouverte de $\mathbb{C}$ contient un point à coordonnées rationnelles. Les composantes connexes de $\mathbb{C} \setminus K$ étant deux-a-deux disjointes, elles sont en nombre au plus dénombrable.

Lemme 4.3. *Soit K un compact de $\mathbb{C}$. Soit O une composante connexe de $\mathbb{C} \setminus K$. Alors $\partial O \subset K$. De plus si K est non vide, alors ∂O est non vide.*

Preuve. $\overline{O} = O \cup \partial O$, donc ∂O vide implique O fermé. Or O est ouvert d'après le lemme précédent, et K non vide implique $O \neq \mathbb{C}$, donc dans ce cas ∂O est non vide.

On va montrer $\partial O \subset K$, c'est-à-dire $\partial O \cap (\mathbb{C} \setminus K) = \emptyset$.

Soit $x \in \mathbb{C} \setminus K$. O est connexe, donc toute partie A vérifiant $O \subset A \subset \overline{O}$ est connexe. En particulier, si $x \in \partial O$, alors $O \cup \{x\}$ est connexe. Or O est une composante connexe de $\mathbb{C} \setminus K$. Donc $x \in \partial O$ implique $x \in O$. Par ailleurs O est ouvert, donc $\partial O \cap O = \emptyset$.

Lemme 4.4. *La courbe de Jordan Γ est compacte, et l'indice de γ par rapport aux points de $\mathbb{C} \setminus \Gamma$ est constant sur chaque composante connexe de $\mathbb{C} \setminus \Gamma$, et nulle sur la composante connexe non bornée.*

Preuve. L'image Γ de γ est une partie compacte de $\mathbb{C}$, car γ est continue sur $[\alpha, \beta]$. Soit $M > 0$ tel que $\Gamma \subset \overline{D}(0, M)$.

Soit $z \in \mathbb{C} \setminus \overline{D}(0, M)$. Alors $H(s, t) = s\,\gamma(t)$ definit une homotopie entre 0 et γ dans $\mathbb{C} \setminus \{z\}$, donc $\mathrm{Ind}(\gamma, z) = 0$.

Par ailleurs $z \mapsto \mathrm{Ind}(\gamma, z)$ est une fonction continue de $\mathbb{C} \setminus \Gamma$ dans $\mathbb{Z}$, donc constante sur chaque composante connexe de $\mathbb{C} \setminus \Gamma$.

Lemme 4.5. *Toute courbe de Jordan fermée est homéomorphe au cercle unité.*

Preuve. Soient $\alpha < \beta$, et soit $\gamma : [\alpha, \beta] \to \mathbb{C}$ un lacet de Jordan, d'image Γ. Quitte à effectuer une translation et une homothétie, on suppose $[\alpha, \beta] =$

$[0, 2\pi]$. On note $\partial\mathbb{D}$ le cercle unité. On pose :

$$\begin{aligned} \delta : [0, 2\pi] &\to \partial\mathbb{D} \\ \theta &\mapsto e^{i\theta} \end{aligned}$$

Alors δ est continue et $\delta_{|[0,2\pi[}$ est bijective ; on pose :

$$\begin{aligned} \hat{\gamma} : \partial\mathbb{D} &\to \Gamma \\ \zeta &\mapsto \gamma({\delta_{|[0,2\pi[}}^{-1}(\zeta)) \end{aligned}$$

L'application $\hat{\gamma}$ ainsi obtenue est la composée de deux fonctions bijectives, donc bijective elle-même. On va vérifier qu'elle est bicontinue.

L'application $\hat{\gamma}$ est continue : $\delta^{-1}_{|[0,2\pi[}$ est continue sur $\partial\mathbb{D} \setminus \{1\}$ donc $\hat{\gamma}$ est continue sur $\partial\mathbb{D} \setminus \{1\}$, et $\gamma(0) = \gamma(2\pi)$, donc

$$\lim_{\zeta \to 1} \gamma(\delta^{-1}(\zeta)) = \hat{\gamma}(1).$$

De la même manière, l'application réciproque $\hat{\gamma}^{-1} = \delta \circ {\gamma_{|[0,2\pi[}}^{-1}$ est continue : ${\gamma_{|[0,2\pi[}}^{-1}$ est continue sur $\Gamma \setminus \gamma(0)$, et $\delta(0) = \delta(2\pi)$ implique

$$\lim_{z \to \gamma(0)} \delta({\gamma_{|[0,2\pi[}}^{-1}(z)) = \hat{\gamma}^{-1}(\gamma(0)).$$

Donc $\hat{\gamma}$ est un homéomorphisme du cercle unité vers Γ.

5. Théorème de Jordan pour une courbe de classe $\mathcal{C}^1$ par morceaux

On suppose désormais que le lacet de Jordan γ est une application de classe $\mathcal{C}^1$ par morceaux, que γ' ne s'annule pas sur son ensemble de définition, et que γ' admet une limite à droite et une limite à gauche en chaque point de discontinuité.

La régularité de γ permet d'étudier le voisinage des points de Γ.

On va montrer d'une part que tout point de Γ est adhérent à exactement deux composantes connexes de $\mathbb{C}\setminus\Gamma$, et que ces deux composantes connexes sont toujours les mêmes ; et d'autre part que chaque composante connexe de $\mathbb{C}\setminus\Gamma$ est adhérente à Γ, ce qui permettra de conclure.

Lemme 5.1. *Soit $g : [a,b] \to \mathbb{C}$ un chemin de Jordan de classe $\mathcal{C}^1$ vérifiant*

$$\forall t \in [a,b]\,,\ g'(t) \neq 0.$$

Alors il existe $\varepsilon > 0$ tel que $\forall \rho \in [0,\varepsilon]\,,\ \exists!\, t \in [a,b]\,,\ |g(t)-g(a)| = \rho$.
De plus $g([a,b]) \cap \overline{D}(g(a),\varepsilon)$ est l'image de la courbe

$$\begin{aligned} c : [0,\varepsilon] &\longrightarrow \mathbb{C} \\ \rho &\longmapsto g(a) + \rho e^{i\varphi(\rho)} \end{aligned}$$

où φ est une application continue $[0,\varepsilon] \to \mathbb{R}$ telle que $\varphi(0) = arg(g'(a))$.

On peut donc représenter g par une bijection paramétrée par le module $|g(t)-g(a)|$ au voisinage de $g(a)$.

Preuve. On suppose $a = 0$ et quitte à effectuer une similitude directe, $g(0) = 0$ et $g'(0)$ réel strictement positif.

On pose $\rho = |g|$. Alors ρ est de classe $\mathcal{C}^1$ au voisinage de 0, et $\rho'(0) = \mathrm{Re}(g'(0)) > 0$.

En effet, le module est de classe $\mathcal{C}^1$ sur $\mathbb{C}^*$, et $g(t) \underset{t\to 0}{\sim} tg'(0)$ ne s'annule qu'en 0 au voisinage de 0 ; en posant $\rho(t) = \sqrt{x^2(t)+y^2(t)}$, où x et y sont de classe C^1, on peut donc calculer :

$$\forall t \neq 0,\ \rho'(t) = \frac{x(t)x'(t)+y(t)y'(t)}{\sqrt{x^2(t)+y^2(t)}}.$$

Ainsi $\rho'(0) > 0$, donc il existe $\delta > 0$ tel que ρ est strictement croissante sur $[0,\delta]$. Donc $\forall \rho_0 \in [0,\rho(\delta)[\,, \exists!\, t \in [0,\delta[\,,\ \rho(t) = \rho_0$.

De plus $\mathrm{Im}(g)$ est compact et g est injective, donc on peut poser
$\varepsilon = \frac{1}{2}\min\{|g(t)| > 0,\ t \in [\delta,b]\}$ pour assurer $\forall t > \delta,\ |g(t)| > \varepsilon$.
Donc $\forall \rho \in [0,\varepsilon]\,,\ \exists!\, t \in [a,b]\,,\ |g(t)-g(a)| = \rho$.

Enfin le chemin g est continu, donc on peut trouver une détermination continue φ de l'argument le long de g ; de plus l'argument de $g(t)-g(a)$ tend vers l'argument de $g'(a)$ quand $t \to a$, donc on peut imposer $\varphi(0) = \arg(g'(a))$.

Lemme 5.2. *Soit $t_0 \in [\alpha, \beta]$ tel que γ' est continue en t_0. Alors tout voisinage V de $\gamma(t_0)$ contient deux points z_1 et z_2 tels que*

$$|\mathrm{Ind}(\gamma, z_2) - \mathrm{Ind}(\gamma, z_1)| = 1.$$

Preuve. On suppose, quitte à reparamétrer et effectuer une similitude directe, $t_0 = 0$, $\gamma(t_0) = 0$, et $\gamma'(t_0)$ réel strictement positif.

Le lemme 5.1 appliqué à gauche et à droite de 0 donne $\varepsilon > 0$ tel que Γ soit paramétrisable au voisinage de 0 par l'application injective

$$\begin{aligned} c : [-\varepsilon, +\varepsilon] &\longrightarrow \Gamma \\ \rho &\longmapsto g(a) + \rho e^{i\varphi(\rho)} \end{aligned}$$

où φ est continue et $\varphi(0) = 0$, et Γ vérifie $\Gamma \cap \overline{D}(0, \varepsilon) = c([-\varepsilon, +\varepsilon])$.

Puisque $\varphi(0) = 0$, on peut de plus choisir ε suffisamment petit pour que $\forall \rho \in [-\varepsilon, +\varepsilon]\,, -\frac{\pi}{2} < \varphi(\rho) < +\frac{\pi}{2}$.

On va étudier la variation de l'argument le long de Γ par rapport aux points $+i\eta$ et $-i\eta$, $0 < \eta < \varepsilon$. On note $\Delta(z)$ la variation par rapport a $z \in \overline{D}(0, \varepsilon)$; on décompose $\Delta(z) = \Delta_1(z) + \Delta_0(z)$, où $\Delta_0(z)$ est la variation le long de c, et $\Delta_1(z)$ le long de γ sur $\Gamma \setminus \overline{D}(0, \varepsilon)$.

Δ est définie et continue sur $\overline{D}(0, \varepsilon) \setminus \Gamma$, et Δ_1 se prolonge par continuité sur $\overline{D}(0, \varepsilon)$.

Donc par continuité,

$$\lim_{\substack{\eta \to 0 \\ \eta > 0}} (\Delta_1(+i\eta) - \Delta_1(-i\eta)) = 0$$

Il reste à calculer Δ_0.

On peut choisir une détermination continue de l'argument de $(c - i\eta)$ à valeurs dans $\left]-\frac{3\pi}{2}, \frac{\pi}{2}\right[$. On note arg l'argument modulo 2π.

$$\lim_{\substack{\eta \to 0 \\ \eta > 0}} \arg(c(\varepsilon) - i\eta) = \varphi(\varepsilon) \qquad [2\pi]$$

$$\lim_{\substack{\eta \to 0 \\ \eta > 0}} \arg(c(-\varepsilon) - i\eta) = \varphi(-\varepsilon) + \pi \qquad [2\pi]$$

Or φ est à valeurs dans $\left]-\frac{\pi}{2}, \frac{\pi}{2}\right[$, donc :

$$\lim_{\substack{\eta \to 0 \\ \eta > 0}} \Delta_0(+i\eta) = \varphi(\varepsilon) - \varphi(-\varepsilon) + \pi$$

De même on peut choisir une détermination continue de l'argument de $(c - (-i\eta))$ à valeurs dans $\left]-\frac{\pi}{2}, \frac{3\pi}{2}\right[$.

$$\lim_{\substack{\eta \to 0 \\ \eta > 0}} \arg(c(\varepsilon) + i\eta) = \varphi(\varepsilon) \qquad [2\pi]$$

$$\lim_{\substack{\eta \to 0 \\ \eta > 0}} \arg(c(-\varepsilon) + i\eta) = \varphi(-\varepsilon) + \pi \qquad [2\pi]$$

Donc :

$$\lim_{\substack{\eta\to 0\\ \eta>0}} \Delta_0(-i\eta) = \varphi(\varepsilon) - \varphi(-\varepsilon) - \pi$$

On peut finalement conclure

$$\lim_{\substack{\eta\to 0\\ \eta>0}} (\Delta_0(+i\eta) - \Delta_0(-i\eta)) = 2\pi.$$

Donc

$$\lim_{\substack{\eta\to 0\\ \eta>0}} (\Delta(+i\eta) - \Delta(-i\eta)) = 2\pi.$$

Or Δ est continue, et multiple de 2π, donc $\Delta(+i\eta) - \Delta(-i\eta)$ est constant pour η au voisinage de 0. On peut donc trouver une valeur de $\eta \in]0, \varepsilon[$ telle que $\Delta(+i\eta) - \Delta(-i\eta) = 2\pi$, c'est-à-dire $\mathrm{Ind}(\gamma, +i\eta) - \mathrm{Ind}(\gamma, -i\eta) = 1$.

Proposition 5.3. *Soit $t_0 \in [\alpha, \beta]$. Alors il existe $\eta_0 > 0$ tel que $\forall 0 < \eta < \eta_0$, $(\mathbb{C} \setminus \Gamma) \cap \mathring{D}(\gamma(t_0), \eta)$ admet exactement deux composantes connexes.*

De plus, chaque point de $\Gamma \cap D$ est sur la frontière de chaque composante connexe, et la valeur de $\mathrm{Ind}(\gamma)$ par rapport à l'une et l'autre composantes connexes diffère de 1.

Preuve. On suppose $\alpha < t_0 = 0 < \beta$ et $\gamma(t_0) = 0$.

Il existe $\alpha < \alpha_1 < 0$ et $0 < \beta_1 < \beta$ tels que γ est de classe $\mathcal{C}^1$ sur $[\alpha_1, 0[$ et sur $]0, \beta_1]$.

On applique le lemme 5.1 pour obtenir ε, φ et ψ tels que

$$g(\rho) := \rho e^{i\varphi(\rho)} \text{paramétrise } -\gamma \text{ à gauche de } 0$$
$$d(\rho) := \rho e^{i\psi(\rho)} \text{paramétrise } +\gamma \text{ à droite de } 0$$

où φ et ψ sont continues, et on peut supposer de plus $|\psi(0) - \varphi(0)| \leq \pi$.

Alors $\forall \rho \in [-\varepsilon, +\varepsilon]$, $\varphi(\rho) \neq \psi(\rho)$ donc par continuité on peut supposer $\varphi < \psi < \varphi + 2\pi$.

On peut alors écrire $D(0, \eta) \setminus \Gamma = C_1 \cup C_2$, avec :

$$C_1 = \{\rho e^{i\theta},\ 0 < \rho < \eta,\ \varphi(\rho) < \theta < \psi(\rho)\}$$
$$C_2 = \{\rho e^{i\theta},\ 0 < \rho < \eta,\ \psi(\rho) < \theta < \phi(\rho) + 2\pi\}$$

Alors C_1 et C_2 sont chacun connexe par arcs, et tout point $\rho e^{i\varphi(\rho)}$ ou $\rho e^{i\psi(\rho)}$ de $\Gamma \cap D(0, \eta)$ est adhérent a C_1 et C_2.

De plus d'après le lemme 5.2, les indices de γ par rapport a C_1 et C_2 diffèrent de 1.

On considère une énumeration $(O_0, O_1, ...)$ des composantes connexes de $\mathbb{C} \setminus \Gamma$.

On pose

$$\begin{aligned} V_1 &: \Gamma \longrightarrow \mathbb{N} \\ &\quad z \longmapsto \min\{n \in \mathbb{N} \,|\, z \in \overline{O_n}\} \\ V_2 &: \Gamma \longrightarrow \mathbb{N} \\ &\quad z \longmapsto \max\{n \in \mathbb{N} \,|\, z \in \overline{O_n}\} \end{aligned}$$

La proposition 5.3 implique que les fonctions V_1 et V_2 sont localement constantes sur Γ, donc constantes puisque Γ est connexe.

On a ainsi montré que tout point de Γ est adhérent à exactement deux composantes connexes de $\mathbb{C} \setminus \Gamma$, et que ces deux composantes connexes sont les mêmes pour tout point de Γ. Par ailleurs, le lemme 4.3 implique que toute composante connexe de $\mathbb{C} \setminus \Gamma$ est adhérente à Γ. Donc $\mathbb{C} \setminus \Gamma$ admet exactement deux composantes connexes.

L'indice de γ par rapport aux points de l'unique composante connexe non bornée est 0 d'après le lemme 4.1, donc l'indice de γ par rapport aux points de la composante connexe bornée est ± 1 d'après la proposition 5.3.

6. Les groupes $\mathcal{G}(K)$, $\mathcal{E}(K)$ et $\mathbf{G}(K)$

Lorsque Γ n'est plus supposée régulière, on ne peut plus étudier comme précédemment les voisinages des points de Γ. On va donner une démonstration plus générale du théorème de Jordan.

Définition 4. *Soit K un compact de $\mathbb{C}$. On introduit les groupes de fonctions suivant, qui jouent un rôle important dans la démonstration :*

$$\mathcal{G}(K) = \{f : K \to \mathbb{C}^* \mid f \text{ continue}\}$$
$$\mathcal{E}(K) = \{f \in \mathcal{G}(K) \mid f \text{ admet un logarithme continu}\}$$

munis de la multiplication $fg : x \mapsto f(x)g(x)$;

Ainsi que leur groupe quotient :

$$\mathbf{G}(K) = \mathcal{G}(K)/\mathcal{E}(K).$$

Remarque 1. *On dit qu'une fonction continue $f : X \to \mathbb{C}$ admet un logarithme continu s'il existe une fonction $g : X \to \mathbb{C}$ continue telle que pour tout $x \in X$, $f(x) = e^{g(x)}$.*

Définition 5. *Pour tout $p \in \mathbb{C} \setminus K$, on note f_p la fonction de $\mathcal{G}(K)$ définie par $f_p(z) = (z - p)$.*

Définition 6. *Soient X et Y deux espaces topologiques. On dit que deux applications continues f_0 et $f_1 : X \to Y$ sont homotopes s'il existe une application continue $H : [0,1] \times X \to Y$ telle que $H(0,.) = f_0$ et $H(1,.) = f_1$.*

Définition 7. *Soit D un disque ouvert de $\mathbb{C}$. On note ∂D un paramétrage injectif de ∂D positivement orienté relativement à D. Soit $p \in \mathbb{C}$. Soit $\psi : \partial D \to \mathbb{C}$ une application continue ne prenant pas la valeur p.*

Le degré de l'application ψ au point p est l'entier :

$$\deg(\psi, p) = \mathrm{Ind}(\psi \circ \partial D, p).$$

Lemme 6.1. *Si A est une partie convexe d'un espace vectoriel normé X, alors toute application continue $f : A \to \mathbb{C}^*$ admet un logarithme continu.*

Preuve. Pour tout $a \in A$, les $B(a,r) \cap A$ sont convexes et forment une base de voisinages de a dans A, donc A est localement connexe par arcs.

D'autre part, tout lacet γ dans A est homotope dans A à un lacet constant (étant donné $a_0 \in A$, l'application $H(s,t) = s\gamma(t) + (1-s)a_0$ est une homotopie entre γ et a_0 dans A), donc vérifie $\mathrm{Ind}(f \circ \gamma, 0) = 0$. Or, si X est localement connexe par arcs, alors toute application $f : X \to \mathbb{C}^*$ admet un logarithme continu si et seulement si pour tout lacet γ dans X, $\mathrm{Ind}(f \circ \gamma, 0) = 0$. Donc f admet un logarithme continu.

6.0.1. Dans la suite, on utilise le théorème de Tietze-Urysohn :

Théorème 6.2. *Théorème de Tietze-Urysohn : Si F est un fermé de $\mathbb{R}^n$, alors toute fonction continue $u : F \to \mathbb{C}$ peut se prolonger en une fonction continue $v : \mathbb{R}^n \to \mathbb{C}$.*

Les deux lemmes qui suivent constituent le cœur de la démonstration du théorème de Jordan.

Lemme 6.3. *On considère $(O_i)_{0\leq i<N}$, $N \in \mathbb{N} \cup \{+\infty\}$ la famille des composantes connexes bornées de $\mathbb{C} \setminus K$. Soient $\forall i$, $p_i \in O_i$, et $(n_i)_{0\leq i<N}$ une famille d'entiers naturels non tous nuls. Alors $f = \prod_i {f_{p_i}}^{n_i} \notin \mathcal{E}(K)$.*

Preuve. Supposons par contraposition que $f = \prod_i {f_{p_i}}^{n_i} \in \mathcal{E}(K)$ i.e. f admet un logarithme continu. D'après le théorème de Borsuk, f admet une extension continue $F : \mathbb{C} \longrightarrow \mathbb{C}^*$. Soit $0 \leq i < N$. Comme O_i est une composante connexe de $\mathbb{C} \setminus K$, $\partial O_i \subset K$ d'après le lemme 4.3.

On a ainsi $\forall j \neq i$, $p_j \notin \overline{O_i} = O_i \cup \partial O_i$.

On pose :

$$G_i : z \longmapsto \prod_{j\neq i} (z - p_j)^{n_j}$$

qui ne s'annule pas sur $\overline{O_i}$ et

$$\begin{aligned} F_i : \overline{O_i} &\longrightarrow \mathbb{C} \\ z &\longmapsto \frac{F(z)}{G_i(z)} \end{aligned}$$

continue.

Or $F|_K = f$ donc sur ∂O_i, $F_i(z) = \dfrac{\prod_j f_{p_j}^{n_j}}{\prod_{j\neq i} f_{p_j}^{n_j}}(z) = f_{p_i}^{n_i}(z) = (z - p_i)^{n_i}$

Soit D un disque ouvert tel que $\overline{O_i} \subset D$. On pose :

$$\begin{aligned} H_i : \overline{D} &\longrightarrow \mathbb{C}^* \\ z &\longmapsto \begin{cases} F_i(z) & \text{si } z \in \overline{O_i} \\ (z - p_i)^{n_i} & \text{si } z \in \overline{D} \setminus \overline{O_i} \end{cases} \end{aligned}$$

D'après ce qui précéde, H_i est continue. De plus $\overline{D}$ est convexe donc d'après le lemme 6.1, H_i admet un logarithme continu.

En particulier, $\operatorname{Ind}(H_i \circ \partial D, 0) = 0$. D'autre part,

$$\begin{aligned} \operatorname{Ind}(H_i \circ \partial D, 0) &= \operatorname{Ind}((z - p_i)^{n_i} \circ \partial D, 0) \\ &= \frac{1}{2\pi i} \int_{(z-p_i)^{n_i} \circ \partial D} \frac{\mathrm{d}z}{z} \\ &= n_i \frac{1}{2\pi i} \int_{\partial D} \frac{\mathrm{d}w}{w - p_i} \\ &= n_i \operatorname{Ind}(\partial D, p_i) \\ &= n_i \end{aligned}$$

Donc pour tout i, $n_i = 0$.

Lemme 6.4. *Soit $(O_i)_{i\in I}$ la famille des composantes connexes bornées de $\mathbb{C} \setminus K$. En admettant l'axiome du choix dénombrable on pose pour chaque i, $p_i \in O_i$. Alors $\mathcal{G}(K)$ est engendré par $\mathcal{E}(K)$ et par les f_{p_i}.*

Autrement dit :

$$\forall f \in \mathcal{G}(K),\ \exists (m_1, ..., m_l) \in \mathbb{Z}^l,\ \frac{f}{\prod_{i=1}^{l} {f_{p_i}}^{m_i}} \in \mathcal{E}(K).$$

Remarque 2. *Si p et q sont deux points d'une même composante connexe de $\mathbb{C} \setminus K$, alors $f_p/f_q \in \mathcal{E}(K)$.*

Preuve. Soit γ un chemin reliant p et q dans $\mathbb{C} \setminus K$, alors

$$\begin{aligned} H : (s, z) &\longmapsto f_{\gamma(s)}(z) \\ (0, z) &\longmapsto f_p(z) \\ (1, z) &\longmapsto f_q(z) \end{aligned}$$

donne f_p et f_q homotopes dans $\mathbb{C}^*$ donc f_p/f_q est homotope à $\mathbf{1}$. Et donc, d'après le théorème de Borsuk, $f_p/f_q \in \mathcal{E}(K)$.

6.0.2. *Preuve du lemme.* D'après la remarque précédente, il suffit de montrer que $\forall f \in \mathcal{G}(K)$, $\exists (q_1, ..., q_l) \in \mathbb{C} \setminus K$ et $\exists (m_1, ..., m_l) \in \mathbb{Z}^l$ tel que $\frac{f}{{f_{q_1}}^{m_1} ... {f_{q_l}}^{m_l}} \in \mathcal{E}(K)$.

K est compact donc on peut supposer $K \subset Q = [0, 1] \times [0, 1]$ le carré de côté 1.

Soit $n \in \mathbb{N}^*$.

On trace les droites $\{x = k/n\}$ et $\{y = k/n\}$ pour $0 \leq k \leq n$ de manière à quadriller Q par des carrés de côté $\frac{1}{n}$. On pose $A = \bigcup \{\text{carrés qui rencontrent } K\}$ et $Q_1, ..., Q_l$ les autres. On a alors $\forall z \in A$, $d(z, K) \leq \frac{\sqrt{2}}{n}$.

Soit $f : K \to \mathbb{C}^*$ continue. Alors, d'après le théorème de Tietze-Urysohn, f admet une extension continue $F : \mathbb{C} \simeq \mathbb{R}^2 \to \mathbb{C}^*$. Si on pose $E = \{z \in \mathbb{C} | F(z) \neq 0\}$ alors E est ouvert et $K \subset E$. On choisit alors n assez grand pour que F ne s'annule pas sur A ; c'est possible car E est ouvert et F est continue. Ainsi, f admet une extension continue $F_0 : A \to \mathbb{C}^*$. $\mathbb{C}^*$ étant connexe par arcs, il existe $g_1 : \partial Q_1 \to \mathbb{C}^*$ continue qui coincide avec F_0 sur $Q_1 \cap A$.

On note q_1 le centre de Q_1.

Remarque 3. *Si V disque ou rectangle de centre p, $\mathcal{G}(\partial V)$ est engendré pas $\mathcal{E}(\partial V)$ et f_p.*

C'est une conséquence du lemme suivant :

Lemme 6.5. *Soit $\Gamma \subset \mathbb{C}$ une courbe de Jordan fermée paramétrée par un lacet γ. Pour tout $f \in \mathcal{G}(\Gamma)$, on pose $I_\gamma(f) = \mathrm{Ind}(f \circ \gamma, 0)$. Alors $I_\gamma : \mathcal{G}(\Gamma) \to \mathbb{Z}$ est un homomorphisme surjectif, de noyau $\mathcal{E}(\Gamma)$.*

Preuve. $I_{\partial V}(f_p) = \mathrm{Ind}(\partial V, p) = 1$ donc $I_{\partial V}(f_p)$ engendre $\mathrm{Im}(\partial V) = \mathbb{Z}$; donc $\mathcal{G}(\partial V)$ est engendré par f_p et $Ker(I_{\partial V}) = \mathcal{E}(\partial V)$.

On applique cette remarque à Q_1 puis on applique le théorème de Borsuk : il existe $m_1 \in \mathbb{Z}$ tel que $\frac{g_1}{{f_{q_1}}^{m_1}} \in \mathcal{E}(\partial Q_1)$ admet un prolongement continu

$G_1 : Q_1 \to \mathbb{C}^*$. On pose alors

$$F_1 : A \cup Q_1 \longrightarrow \mathbb{C}^*$$
$$z \longmapsto \begin{cases} G_1(z) & \text{si } z \in Q_1 \\ \frac{F_0(z)}{(z-q_1)^{m_1}} & \text{si } z \in A \end{cases}$$

On réitère et on obtient l'existence de $m_1, ..., m_l \in \mathbb{Z}$ et $q_1, ..., q_l \in \mathbb{C} \setminus K$ tels que $F_l : Q = [0,1] \times [0,1] \to \mathbb{C}^*$ soit un prolongement continu de $\frac{f}{{f_{q_1}}^{m_1} ... {f_{q_l}}^{m_l}}$.

Comme Q est convexe, F_l admet un logarithme continu : en particulier, $\frac{f}{{f_{q_1}}^{m_1} ... {f_{q_l}}^{m_l}} \in \mathcal{E}(K)$.

De plus $\forall i$, $\frac{f_{p_i}}{f_{q_i}} \in \mathcal{E}(K)$. Donc $\frac{f}{\prod {f_{p_i}}^{m_i}} \in \mathcal{E}(K)$.

7. Démonstration du théorème dans le cas général

On montre ici un résultat plus général que le théorème de Jordan :

Si K_1 et K_2 sont deux parties compactes homéomorphes du plan complexe, alors $\mathbb{C} \setminus K_1$ et $\mathbb{C} \setminus K_2$ admettent exactement le même nombre de composantes connexes.

Une courbe de Jordan fermée étant homéomorphe au cercle unité (lemme 4.5), ce résultat implique directement le théorème de Jordan.

On va montrer d'une part que si K_1 et K_2 sont deux compacts du plan homéomorphes alors $\mathbf{G}(K_1)$ et $\mathbf{G}(K_2)$ sont isomorphes, et d'autre part que si K est un compact de $\mathbb{C}$ alors le groupe $\mathbf{G}(K)$ est isomorphe a $\mathbb{Z}^N$, où N est le nombre de composantes connexes bornées, au plus dénombrable, de $\mathbb{C} \setminus K$.

7.1. **Si K_1 et K_2 sont deux compacts homéomorphes, alors $\mathbf{G}(K_1)$ et $\mathbf{G}(K_2)$ sont isomorphes.** Soit $\theta : K_1 \to K_2$ un homéomorphisme. Alors l'application $f \mapsto f \circ \theta$ est un isomorphisme de $\mathcal{G}(K_1)$ sur $\mathcal{G}(K_2)$ qui envoie $\mathcal{E}(K_1)$ sur $\mathcal{E}(K_2)$, donc définit un isomorphisme de $\mathbf{G}(K_1)$ sur $\mathbf{G}(K_2)$.

7.2. **$\mathbf{G}(K)$ est isomorphe a $\mathbb{Z}^N$.** On note $N \in \mathbb{N} \cup \{+\infty\}$ le nombre de composantes connexes bornées de $\mathbb{C} \setminus K$, au plus dénombrables d'après le lemme 4.2. On considère le groupe additif $\mathbb{Z}^N$; par convention, $\mathbb{Z}^\infty$ est l'ensemble des suites nulles à partir d'un certain rang.

Théorème 7.1. *Le groupe $\mathbf{G}(K)$ est isomorphe à $\mathbb{Z}^N$.*

Preuve. On note $(O_i)_{0 \leq i < N}$ la famille des composantes connexes bornées de $\mathbb{C} \setminus K$ et on choisit pour tout i un point $p_i \in O_i$. On admet ici l'axiome du choix dénombrable.

On pose pour tout $p \in \mathbb{C}$, $f_p : z \mapsto z - p$, et $\mathbf{f_p}$ son image par la projection de $\mathcal{G}(K)$ sur $\mathbf{G}(K)$.

On pose :

$$\Phi : \mathbb{Z}^N \longrightarrow \mathbf{G}(K)$$
$$(n_i)_{0 \leq i < N} \longmapsto \prod_i \mathbf{f_{p_i}}^{n_i}$$

L'application Φ est bien définie ; par convention, un éventuel produit vide est égal a $\mathbf{1}$, et les indices i sont pris dans l'ensemble fini $\{0 \leq i < N \mid n_i \neq 0\}$.

On constate que Φ est un homomorphisme de groupes. Il reste à montrer qu'il est bijectif. C'est en fait une conséquence directe des lemmes 6.3 et 6.4 démontrés plus tôt.

7.2.1. *Injectivité du morphisme Φ.* Soit $\mathbf{n} = (n_i)_{0 \leq i < N} \in \mathbb{Z}^N$ dans le noyau de Φ, c'est-à-dire tel que $\Phi(\mathbf{n}) = 0$, ou de manière équivalente : $\prod_i {f_{p_i}}^{n_i} \in \mathcal{E}(K)$. Par contraposée du lemme 6.3, on a pour tout $0 \leq i < N$, $n_i = 0$.

Donc Φ est injective.

7.2.2. *Surjectivité du morphisme* Φ. Soit $f \in \mathcal{G}(K)$. D'après le lemme 6.4, il existe $l < N$ et $(m_1, ..., m_l) \in \mathbb{Z}^l$ tels que $\mathbf{f} = \prod_{i=1}^{l} \mathbf{f_{p_i}}^{m_i}$.

Donc Φ est surjective. Donc bijective.

Lemme 7.2. $\mathbb{Z}^{N_1} \simeq \mathbb{Z}^{N_2}$ *implique* $N_1 = N_2$.

Preuve. Une famille (finie ou infinie) de $\mathbb{Z}^N$ est indépendante dans $\mathbb{Z}^N$ si et seulement si elle est libre dans l'espace vectoriel $\mathbb{Q}^N$, donc une famille indépendante de $\mathbb{Z}^N$ est au plus de cardinal N. Enfin, la base canonique est une famille indépendante de $\mathbb{Z}^N$: donc le cardinal maximum d'une famille indépendante de $\mathbb{Z}^N$ est exactement N.

Par ailleurs, soient N_1 et $N_2 \in \mathbb{N} \cup \{+\infty\}$, et soit $\varphi : \mathbb{Z}^{N_1} \to \mathbb{Z}^{N_2}$ un isomorphisme de groupes. L'image d'une famille indépendante de $\mathbb{Z}^{N_1}$ par φ est une famille indépendante de $\mathbb{Z}^{N_2}$ de même cardinal, et réciproquement avec φ^{-1}. Par conséquent, $N_1 = N_2$.

On a ainsi démontré que si K_1 et K_2 sont deux compacts homéomorphes de $\mathbb{C}$, alors $\mathbb{C} \setminus K_1$ et $\mathbb{C} \setminus K_2$ ont même nombre de composantes connexes bornées.

Par ailleurs, si $\mathbb{D} \subset \mathbb{C}$ est le cercle unité, alors $\mathbb{C} \setminus \mathbb{D}$ admet exactement deux composantes connexes, $\{z \in \mathbb{C} \,|\, |z| < 1\}$ et $\{z \in \mathbb{C} \,|\, |z| > 1\}$. Donc si Γ est une courbe de Jordan fermée, donc un compact homéomorphe à $\mathbb{D}$ d'après le lemme 4.5, alors $\mathbb{C} \setminus \Gamma$ admet exactement deux composantes connexes.

7.3. **La frontière de chaque composante connexe est égale à Γ.** Soit O une composante connexe de $\mathbb{C} \setminus \Gamma$. D'après le lemme 4.3, $\partial O \subset \Gamma$. Supposons par l'absurde que cette inclusion est stricte, c'est-à-dire qu'il existe $x \in \Gamma$ tel que $\partial O \subset \Gamma \setminus \{x\}$.

$\Gamma \setminus \{x\}$ est homéomorphe à $\mathbb{R}$, et ∂O est compact, donc ∂O est homéomorphe à un compact de $\mathbb{R}$. On peut donc appliquer les résultats précédant : $\mathbb{C} \setminus \partial O$ est connexe.

Par ailleurs O est ouvert et fermé dans $\mathbb{C} \setminus \partial O$, et $O \neq \mathbb{C} \setminus \partial O$ puisque $O \subsetneq \mathbb{C} \setminus \Gamma$, donc $\mathbb{C} \setminus \partial O$ n'est pas connexe.

Donc $\partial O = \Gamma$.

2 Apprentissage hors murs : Stage de modélisation en biologie.

RAPPORT DE STAGE

JULIE HÉMONT

Apprentissage hors murs dans le cadre du Magistère 1 de Mathématiques :
Modélisation en biologie.
20 Juin - 15 Juillet 2016

Table des matières

Table des figures

1. Introduction

Dans le cadre de l'apprentissage hors mur du magistère de mathématiques, j'ai effectué mon stage de trois semaines à l'Institut de Génétique et Microbiologie de l'Université Paris-Sud. J'ai travaillé sous la direction de Jean LEHMANN au sein de l'équipe de Daniel GAUTHERET.

J'ai choisi ce stage après avoir suivi l'UE Maths et Bio de Michel LAURENT et sous les conseils de quelques étudiants ; dans le but de conforter mon envie de faire le M2 Maths SV à l'avenir.

L'objectif de ce stage était de poursuivre l'étude qu'avait effectué Chloé MILSONNEAU l'année précédente. Elle s'était appuyée sur un article présentant une étude du codage lors d'une polymérisation.

Je me suis appuyée sur ses résultats pour observer une corrélation puis pour comprendre le comportement du système de polymérisation lorsque l'on s'en éloigne.

J'ai ensuite abordé l'optimisation de la vitesse de codage avec l'influence du L1Stalk.

Toutes les courbes ont été obtenues sous Scilab. Chloé Milsonneau avait choisi de coder en Python.

2. Introduction : problèmes soulevés par Chloé Milsonneau et corrélation observée

2.1. Étude d'une vitesse de polymérisation dans un cas simplifié de traduction.

On étudie un cas simplifié de traduction avec un codon, un transporteur et un acide aminé. On détermine la vitesse de polymérisation d'une protéine avec le schéma simplifié.

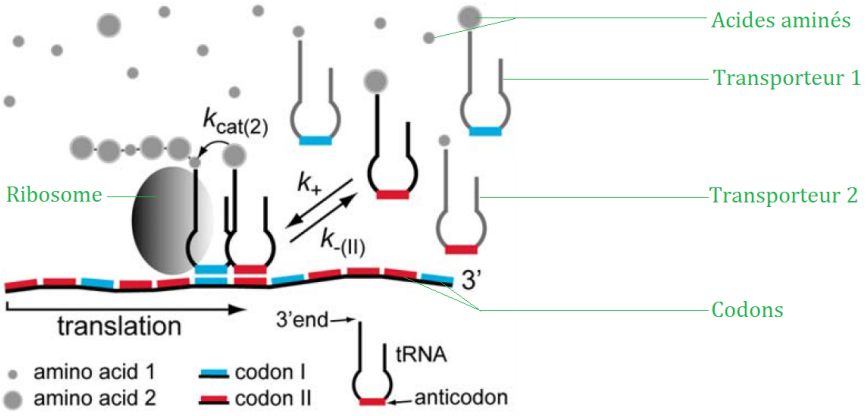


Figure 1. Schéma représentatif du problème

On notera par la suite :

— S_i les transporteurs d'acides aminés, $[S_i]$ leur concentration,

— P_i l'acide aminé i ajouté à la chaîne déjà polymérisée,

— E l'ensemble codon + chaîne d'acides aminés.

On étudie la vitesse v_0 de création de la protéine en fonction de ses paramètres tels que la constante de dissociation k_- entre le codon et le transporteur ou la concentration en transporteurs $[S_1]$.

Dans la suite, toutes les vitesses sont renormalisées par $[E_0]$, concentration de l'ensemble codon + chaîne d'acides aminés.

On précise l'unité des éléments manipulés :

— $[S_1]$ en $mol.L^{-1}$

— k_- en s^{-1}

— k_{cat} en s^{-1}

— k_+ en $(mol/L)^{-1}.s^{-1}$

— v_0 normalisé en s^{-1}

2.2. Évaluation de la vitesse.

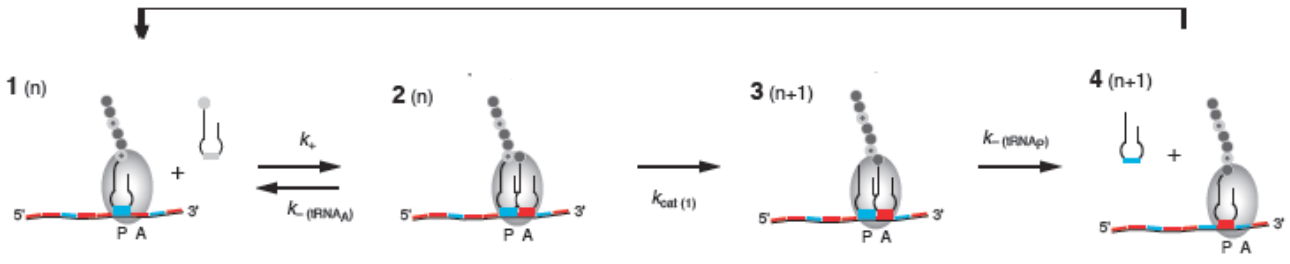


Figure 2. Schéma de réaction

L'équation de la réaction est la suivante :

$$E + S_1 \underset{k_+}{\overset{k_-}{\leftrightarrows}} ES_1 \xrightarrow{k_{cat}} EP_1 \xrightarrow{k_-} E + P_1$$

On remarque que cette équation suggère intuitivement qu'un k_- optimal existe : la constante k_- intervient dans les deux sens de l'équation. Si k_- est grand, la réaction n'aura pas le temps de se faire que le transporteur sera déjà dissocié du codon. Inversement, si k_- est très petit, le transporteur dont l'acide aminé sera rattaché à la chaine protéinique ne se dissociera pas assez vite pour que la réaction puisse se poursuivre.

On fixe dans la suite $k_+ = 10^6 (mol/L)^{-1} s^{-1}$. Cette valeur est justifiée par des mesures expérimentales (GrosJean et al.,1998).

En reprennant les calculs de Chloé, on retrouve que la vitesse de ploymérisation est :

$$v_0 = \frac{k_- k_+ k_{cat} [S_1]}{(k_{cat} + k_-)(k_- + k_+ [S_1])}.$$

On regarde comment cette vitesse varie en fonction de k_- mais aussi l'impact de la concentration en transporteurs $[S_1]$.

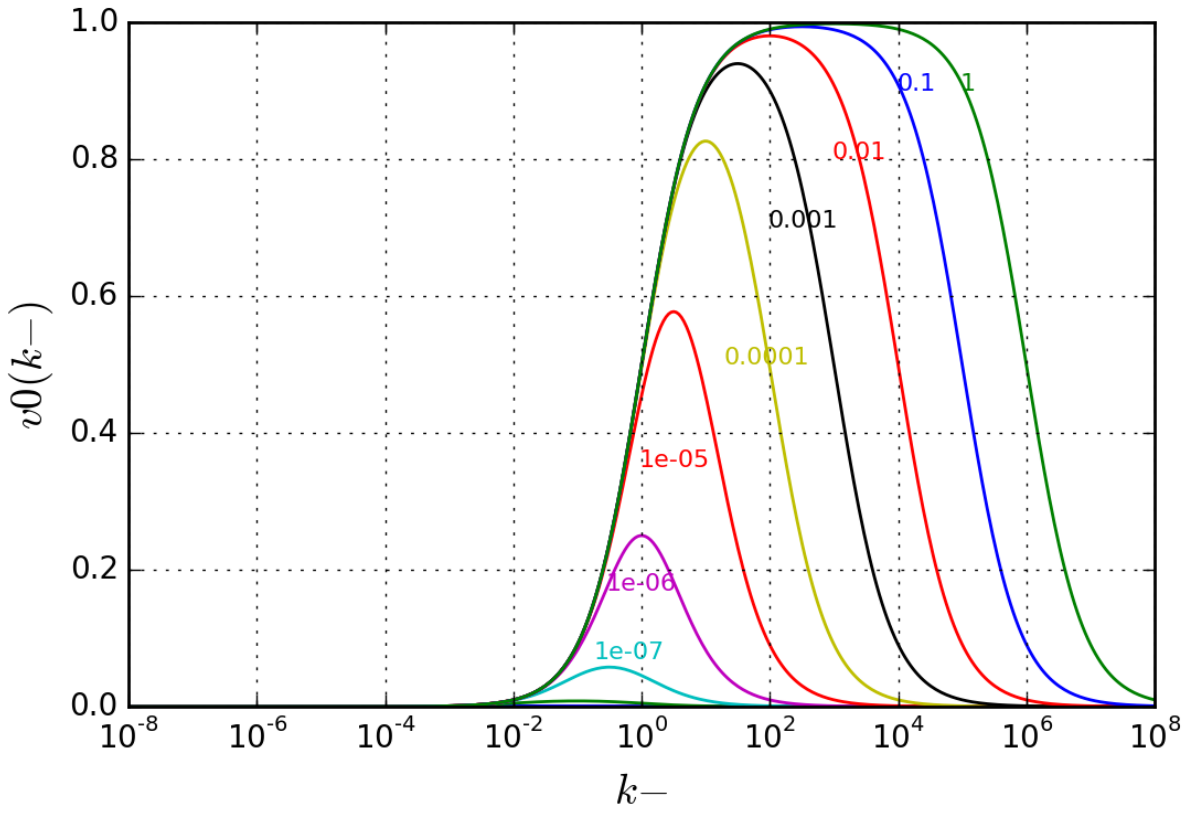


FIGURE 3. v_0 en fonction de k_- pour plusieurs valeurs de $[S_1]$, k_{cat} est fixé à $1 s^{-1}$

À k_{cat} fixé, $v_0(k_-)$ admet un maximum en $k_{-_{v_{0max}}} = \sqrt{[S_1] k_+ k_{cat}}$. On a alors :

$$v_{0max} = \frac{[S_1] k_+ k_{cat}}{(\sqrt{[S_1] k_+} + \sqrt{k_{cat}})^2}.$$

2.3. **Robustesse.**
On cherche à voir la robustesse du système. Cette grandeur permet de caractériser l'optimisation du système sous des concentrations en transporteurs $[S_1]$ fluctuantes. On lit sur la figure 3 que pour tout concentration $[S_1]$, pour $k_- = 1s^{-1}$, on se situe toujours à au moins 1/2 fois le maximum.

On définit, pour tout k_-, $\alpha(k_-)$, la robustesse du système, la valeur pour laquelle on a toujours $v_0(k_-) \geq \alpha(k_-)v_{0max}$ quelque soit la valeur de $[S_1]$. Chloé a utilisé que

$$\alpha(k_-) = \min_{[S_1]} \left(\frac{v_0(k_-)}{v_{0max}}([S_1]) \right).$$

Elle a alors tracé α en fonction de k_- pour $k_{cat} = 1s^{-1}$ puis pour $k_{cat} = 100s^{-1}$. De ses résultats (Figure 4), on voit apparaitre un maximum pour α lorsque $k_- = k_{cat}$; c'est un état pour lequel la vitesse est optimisée. En effet, on est toujours, quelque soit la concentration $[S_1]$, au plus proche du maximum.

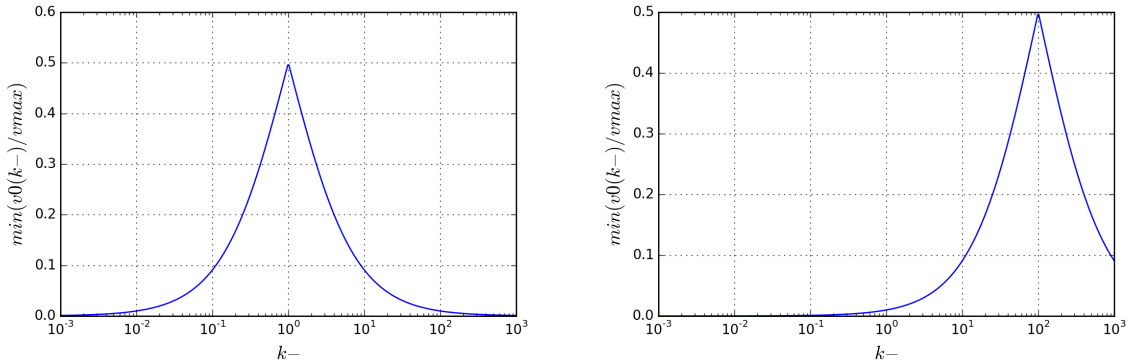


FIGURE 4. Robustesse du système pour $k_{cat} = 1s^{-1}$ et pour $k_{cat} = 100s^{-1}$

On peut s'interroger sur ce maximum qui décrit un optimum pour k_- et k_{cat}. D'autant plus qu'une corrélation entre l'énergie entre codon et anticodon et le volume de l'acide aminé correspondant existe.

2.4. **Corrélation observable entre l'énergie entre codon et anticodon et le volume de l'acide aminé.**

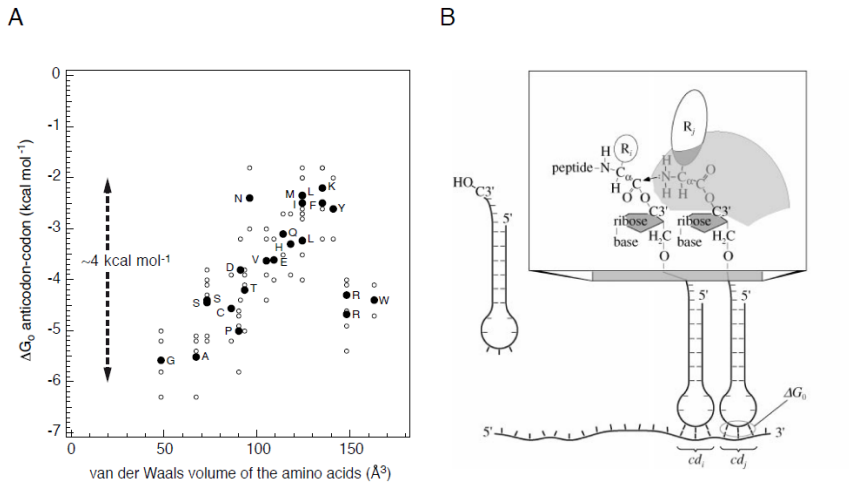


FIGURE 5. Corrélation observée

Cette corrélation observée sur la figure 5 nous indique un lien entre le k_- et le k_{cat} de chaque codon auquel est associé un acide aminé. En effet, l'énergie entre codon et anticodon est intimement liée au k_- et il a été observé que plus le volume de l'acide aminé était grand, plus le k_{cat} associé était grand.

3. Sous $[S_1]$ fluctuant

3.1. **Variations de la vitesse en fonction de la concentration** $[S_1]$**.**

On regarde l'évolution de la vitesse lorsque $[S_1]$ est fluctuant. On observe, sur la figure 6, une convergence vers 0 de la vitesse v_0 lorsque $[S_1] \longrightarrow 0$ avec la même asymptote pour les 4 couples. En effet, on rappelle que

$$v_0(k_-, k_{cat}) = \frac{k_- k_+ k_{cat}[S_1]}{(k_{cat} + k_-)(k_- + k_+[S_1])}.$$

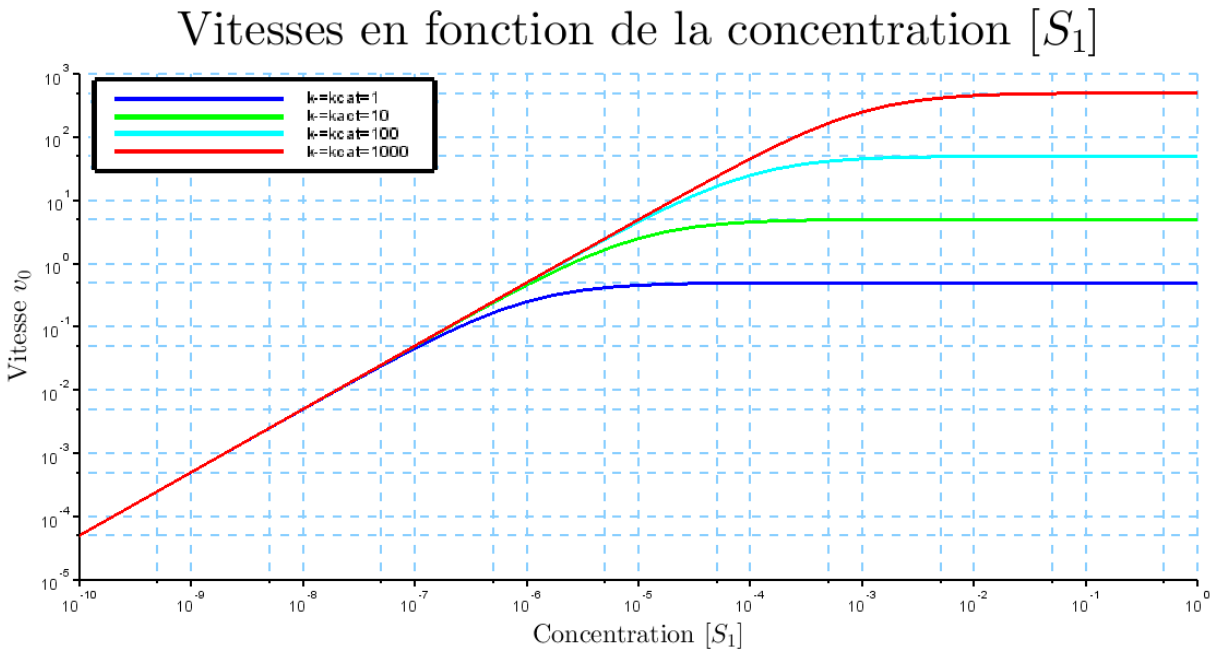


FIGURE 6. Vitesse en s^{-1} en fonction de la concentration $[S_1]$ en $mol.L^{-1}$

On peut vérifier que

$$log(v_0([S_1])) \underset{[S_1]\to 0}{\sim} log([S_1]) + log\left(\frac{k_+}{2}\right).$$

Ce qui donne l'équation de cette asymptote commune : $y = x + b$ avec $b = log(k_+/2)$.

Remarque sur ce qui se passe lorsque $[S_1] \to +\infty$:
Si elle n'est pas commune aux différentes valeurs de $k_- = k_{cat}$, on a bien une vitesse limite.

$$v_0([S_1]) \underset{[S_1]\to+\infty}{\longrightarrow} \frac{k_{cat}}{2}$$

3.2. **Trouver un** k_- **optimal.**

On rappelle que, pour tout k_-, on définit $\alpha(k_-)$ comme étant la valeur pour laquelle on a toujours $v_0(k_-, k_{cat}) \geq \alpha(k_-) v_{0max}(k_{cat})$ quelque soit la valeur de $[S_1]$. On souhaite être proche de v_{0max} et ce indépendamment de la concentration $[S_1]$. Le k_- optimal peut être défini comme étant le k_- pour lequel on a un maximum pour α.

3.2.1. *Que vaut* α *? : Expression de la fonction* $\alpha(k_-)$*.*

On peut comprendre $\alpha(k_-)$ comme le

$$\underset{[S_1]}{min}\left(\frac{v_0(k_-, k_{cat})}{v_{0max}(k_{cat})}([S_1])\right)$$

Il vaut en réalité une limite.

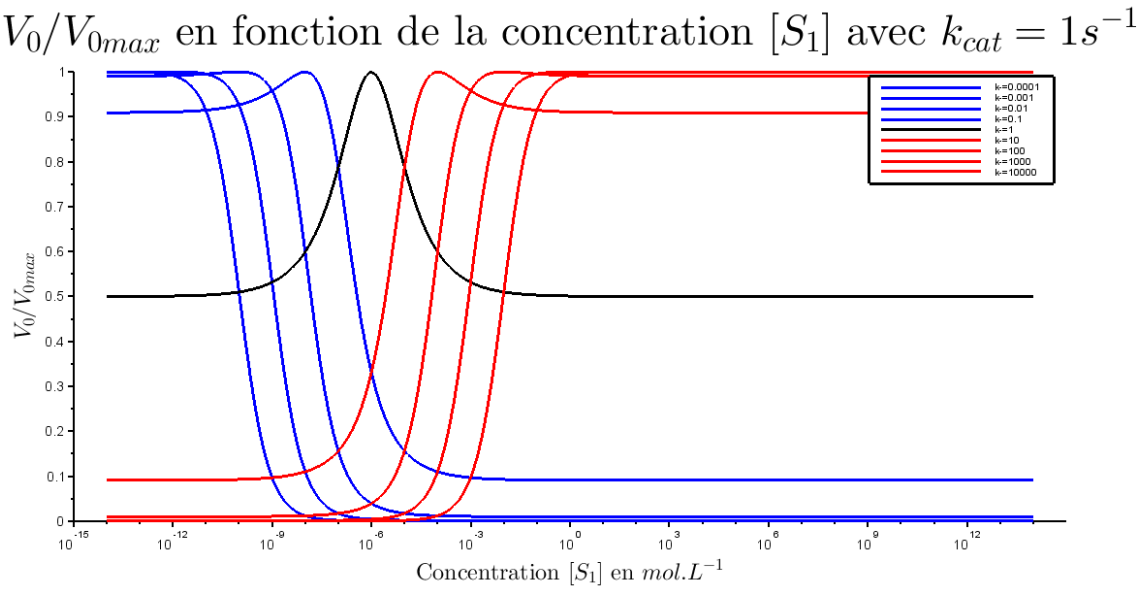


FIGURE 7. Le min est une limite, dont on connait l'expression.

En effet, interprétons la figure 7 :

(1) en rouge, $k_- > k_{cat}$; le minimum sur $[S_1]$ est atteint pour $[S_1] \to 0$.

(2) en bleu, $k_- < k_{cat}$; le minimum sur $[S_1]$ est atteint pour $[S_1] \to +\infty$.

(3) en noir, $k_- = k_{cat}$; le minimum sur $[S_1]$ est atteint pour $[S_1] \to 0$ et pour $[S_1] \to +\infty$.

Cette limite est connue.

En effet,

$$\begin{aligned}\frac{v_0(k_-, k_{cat})}{v_{0max}(k_{cat})}([S_1]) &= \frac{k_-(\sqrt{[S_1]k_+} + \sqrt{k_{cat}})^2}{(k_{cat} + k_-)(k_- + k_+[S_1])} \\ &= \frac{k_-k_+[S_1] + 2k_-\sqrt{[S_1]k_+} + k_-k_{cat}}{(k_{cat} + k_-)(k_- + k_+[S_1])}\end{aligned}$$

Lorsque $k_- \geq k_{cat}$:

$$\alpha(k_-) = \lim_{[S_1]\to 0} \frac{v_0(k_-, k_{cat})}{v_{0max}(k_{cat})}([S_1]) = \frac{k_{cat}}{k_{cat} + k_-}$$

Lorsque $k_- \leq k_{cat}$:

$$\alpha(k_-) = \lim_{[S_1]\to +\infty} \frac{v_0(k_-, k_{cat})}{v_{0max}(k_{cat})}([S_1]) = \frac{k_-}{k_{cat} + k_-}$$

On a bien, lorsque $k_- = k_{cat}$, les deux expressions qui donnent le même résultat : $\alpha = \frac{1}{2}$. α est donc continue.

On a donc une expression pour la robustesse :

$$\alpha : k_- \longmapsto \begin{cases} \frac{k_{cat}}{k_{cat}+k_-} & \text{si } k_- \geq k_{cat} \\ \frac{k_-}{k_{cat}+k_-} & \text{si } k_- \leq k_{cat} \end{cases}$$

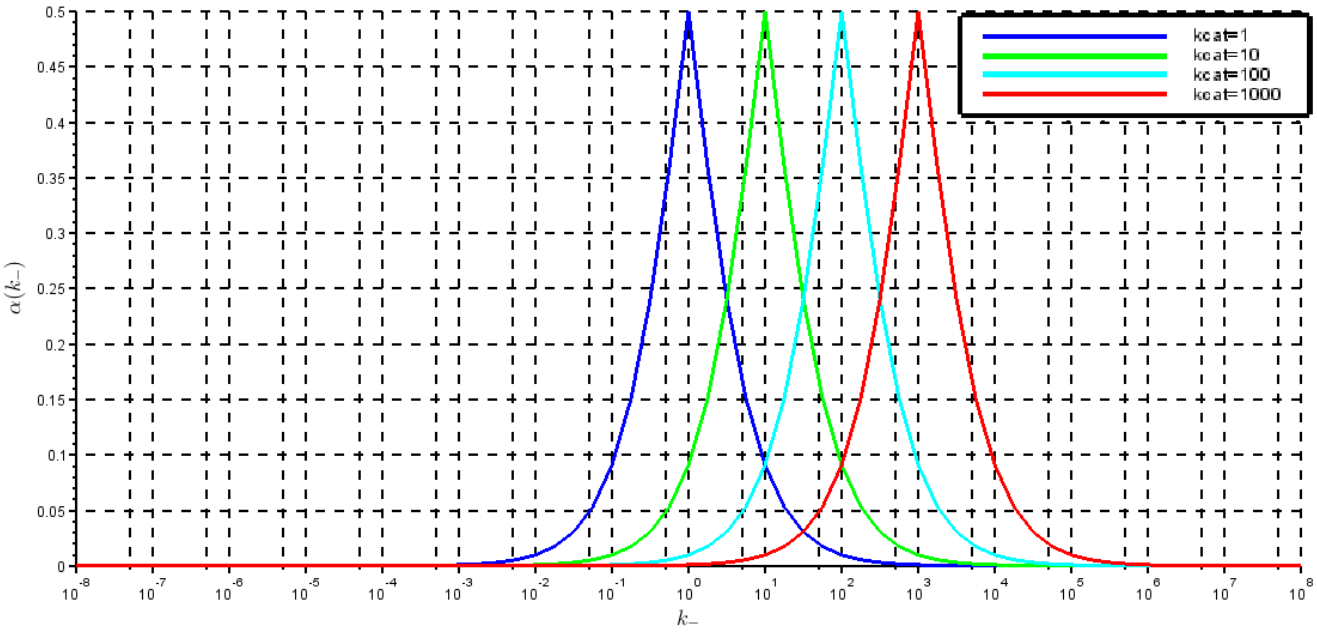


FIGURE 8. $\alpha(k_-)$ pour différentes valeurs de k_{cat}

3.2.2. *Valeur de k_- optimale.*
Pour que le système soit optimisé, il faut que $\alpha(k_-)$ soit maximal. C'est le cas lorsque $k_- = k_{cat}$.

3.2.3. *Corrélation.*

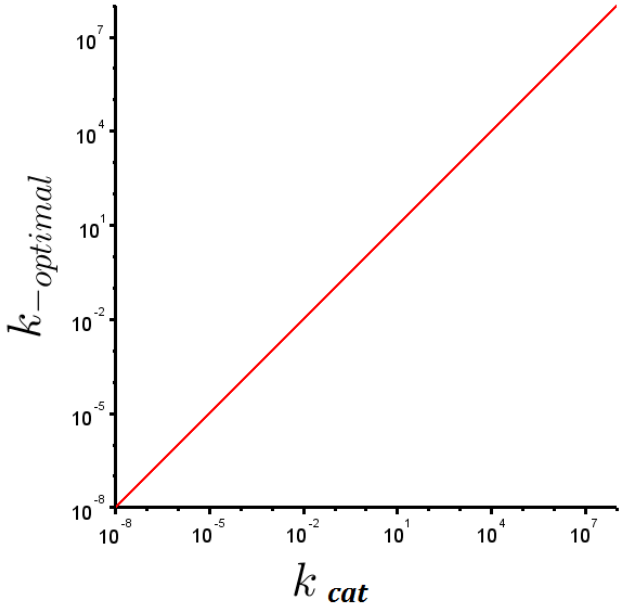


FIGURE 9. $k_{-optimal}$ en fonction de k_{cat}

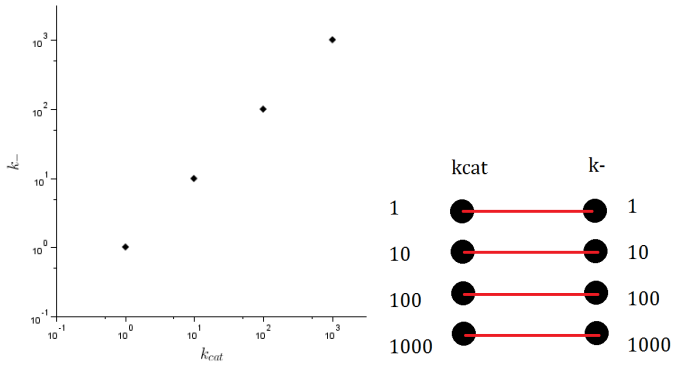
On observe une droite : k_- est optimal lorsqu'il vaut exactement k_{cat}. On retrouve la corrélation remarquable de la figure 5.

Que ce passe-t-il lorsque l'on n'associe plus k_- et k_{cat} de manière idéale ?

4. Observations pour une concentration $[S_1]$ donnée

4.1. Cas simplifié avec uniquement 4 valeurs de k_- et 4 valeurs de k_{cat}.

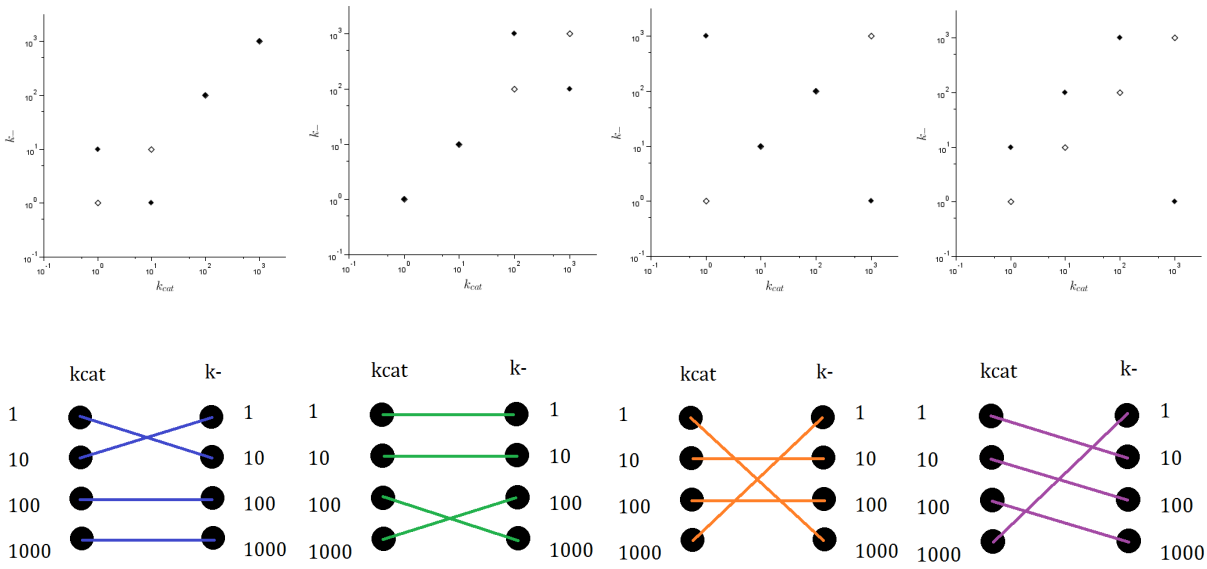
On va se limiter au cas où on possède 4 k_- et 4 k_{cat}. On observait la corrélation suivante :



On va former des couples qui représentent l'association entre l'acide aminé (k_{cat}) et le codon (k_-). On choisit d'attribuer aux valeurs k_- et k_{cat} les ordres de grandeur suivants : 1, 10, 100 et 1000.

Ces valeurs sont justifiées : sur la corrélation observée sur la figure 5, la variation de l'énergie condon-anticodon de $4kcal.mol^{-1}$ correspond à une variation du k_- entre 1 et 1000.

Questionnement : quelles sont les conséquences sur la vitesse de polymérisation si on s'éloigne de la corrélation ? On cherche à comprendre ce qui se passe si on s'éloigne du cas rouge où pour tous les couples $k_- = k_{cat}$. À chaque k_-, on associe un k_{cat}. On peut alors effectuer 24 permutations. Par exemple :



4.2. Valeurs moyennes des vitesses pour toutes les permutations.

Pour toutes les permutations que existent, c'est à dire 24 pour ce cas précis, on calcule, pour chaque couple formé, les valeurs des vitesses v_0, données par :

$$v_0(k_-, k_{cat}) = \frac{k_- k_+ k_{cat}[S_1]}{(k_{cat} + k_-)(k_- + k_+[S_1])}$$

On fixe $[S_1] = 10^{-6} mol.L^{-1}$. Cette valeur de concentration est intéressante car c'est une valeur tout à fait envisageable dans la cellule ; les vitesses ne sont pas trop faibles et les écarts entres les vitesses n'est pas trop élevé. Pour chaque couple donné k_-/k_{cat}, on calcule cette vitesse v_0. On obtient donc 4 vitesses pour chaque permutation.

4.2.1. *Observations des 4 vitesses v_0 pour les 4 couples formés pour toutes les permutations.*
Les 24 permutations sont :

Numéro de la permutation .	Valeurs prises par k_- en s^{-1}			
	$k_{cat} = 1s^{-1}$	$k_{cat} = 10s^{-1}$	$k_{cat} = 100s^{-1}$	$k_{cat} = 1000s^{-1}$
1	1000	100	10	1
2	1000	100	1	10
3	1000	10	100	1
4	1000	10.	1.	100.
5	1000	1.	100.	10.
6	1000	1.	10.	100.
7	100.	1000	10.	1.
8	100.	1000	1.	10.
9	100.	10.	1000	1.
10	100.	10.	1.	1000.
11	100.	1.	1000	10.
12	100.	1.	10.	1000.
13	10.	1000	100.	1.
14	10.	1000	1.	100.
15	10.	100.	1000	1.
16	10.	100.	1.	1000.
17	10.	1.	1000	100.
18	10.	1.	100.	1000.
19	1.	1000	100.	10.
20	1.	1000	10.	100.
21	1.	100.	1000	10.
22	1.	100.	10.	1000.
23	1.	10.	1000	100.
24	**1.**	**10.**	**100.**	**1000.**

Attention ! La permutation numéro 24 est en fait le cas où $k_- = k_{cat}$ pour tous les couples.

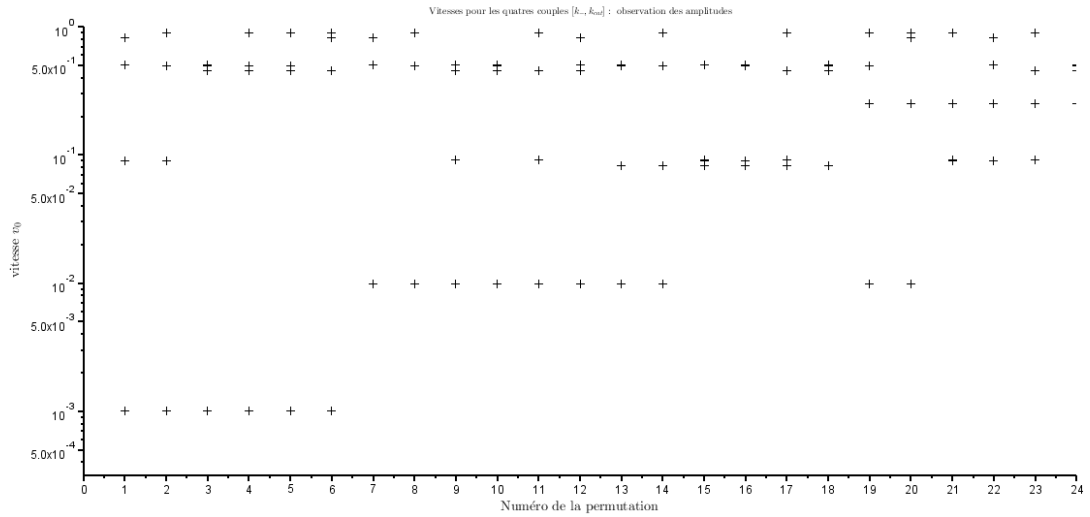


FIGURE 10. Vitesses v_0 pour les 4 couples, pour les 24 permutations

On voit que certaines vitesses, pour certains couples, sont très basses : on pourrait se trouver dans un cas de figure où une des vitesses est trop faible pour assurer la polymérisation. Pour certains couples, on remarque que la vitesse de polymérisation dépend massivement de la composition du message. Les vitesses les plus faibles pourraient même stopper complètement la polymérisation. On introduit une valeur de vitesse seuil sous laquelle la polymérisation est impossible.

4.2.2. *Observations des vitesses moyennes pour toutes les permutations.*
On calcule alors la valeur moyenne des vitesses pour chaque permutation. En calculant la vitesse moyenne, on calcule la vitesse d'un message constitué des 4 codons en proportions égales.

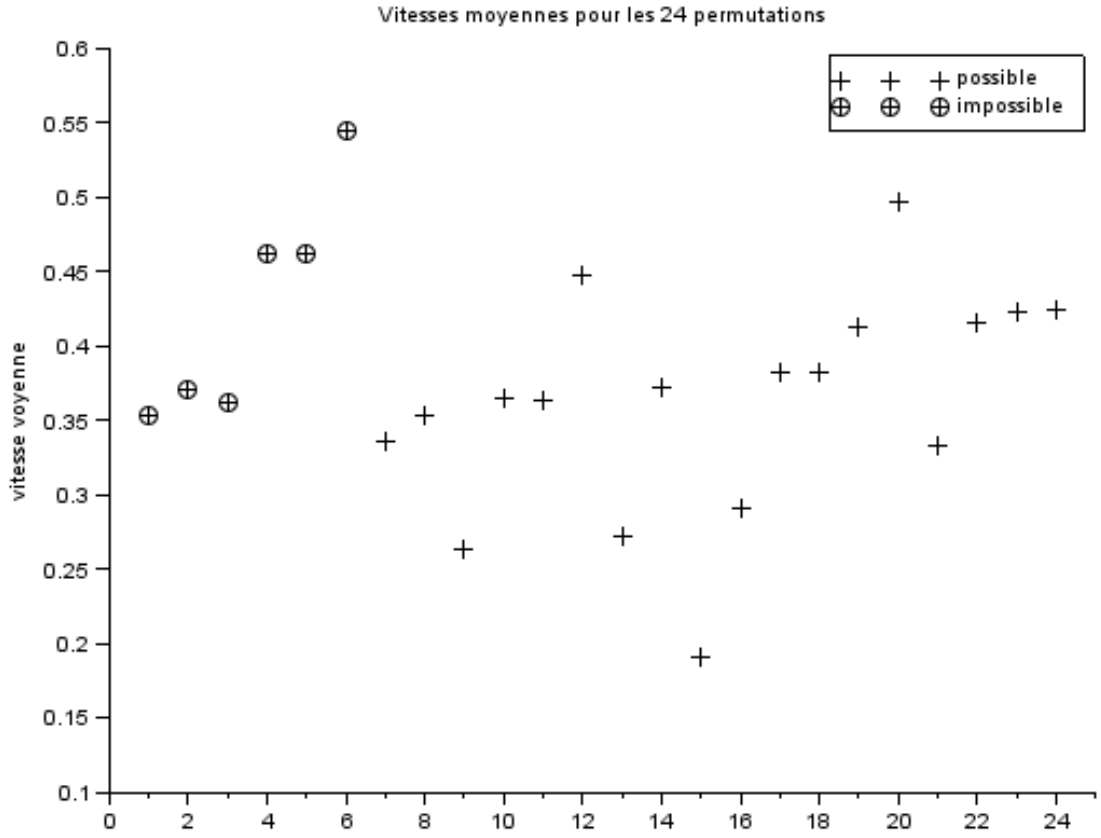


FIGURE 11. Vitesses moyennes, pour les 24 permutations

On voit que pour les permutations 1 à 8, les vitesses moyennes ne sont pas excessivement faibles mais un couple au moins a une vitesse inférieure à la vitesse de seuil $v_{0seuil} = 10^{-3}s^{-1}$.

Qu'en est-il dans le cas où on a simplement effectué un échange entre deux valeurs ?

4.3. **Valeurs moyennes des vitesses dans le cas où on effectue une transposition.**
On regarde maintenant ce qui se passe si on effectue un échange entre deux valeurs ; permutation qui semble la moins "grave".

On compte 6 échanges possibles :

N°	Échange
1	$k_- = k_{cat}$
2	$1 \leftrightarrow 10$
3	$10 \leftrightarrow 100$
4	$100 \leftrightarrow 1000$
5	$1 \leftrightarrow 100$
6	$10 \leftrightarrow 1000$
7	$1 \leftrightarrow 1000$

On remarque sur la figure 12 que pour l'échange $1 \leftrightarrow 1000$, la polymérisation est interrompue par un des couples.

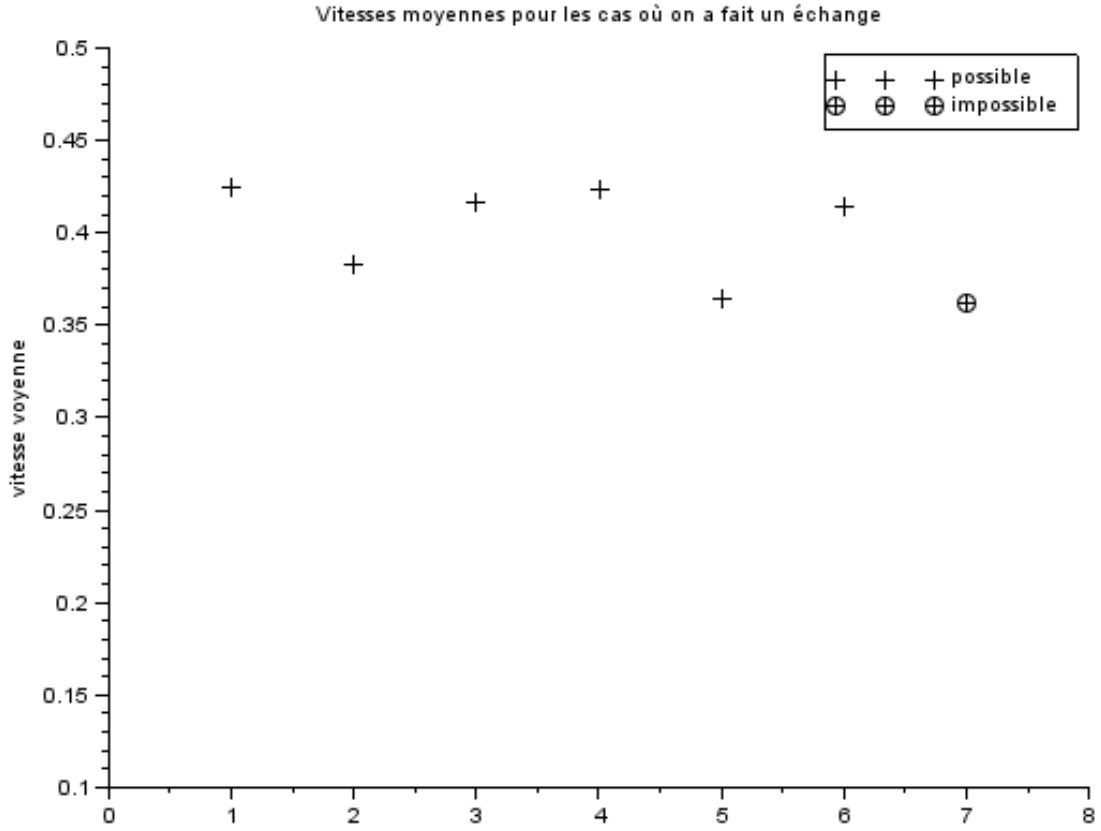


FIGURE 12. Vitesses moyennes, pour les transpositions

On voit donc sur la figure 12 que la vitesse moyenne n'est pas un bon indicateur pour qualifier l'optimisation du système : la vitesse de polymérisation dépend de la composition du message et le calcul de la vitesse moyenne ne prend pas en compte l'aspect biologique (avec une vitesse trop faible pour 1 seul codon, la traduction s'arrête).

4.4. **Classement des 4 vitesses pour les 4 couples formés pour toutes les permutations.**
On pourrait s'intéresser à un nouvel aspect : la stabilité face aux variations de composition du message.

En effet, on souhaiterait que la vitesse de polymérisation soit indépendante du code, c'est à dire du message lui-même. On aimerait voir que les vitesses de tous les couples k_-/k_{cat} sont proches les unes des autres. On va donc classer les permutations par amplitudes de vitesse croissantes. C'est à dire que l'on va regarder $v_{max} - v_{min}$ pour chaque permutation.

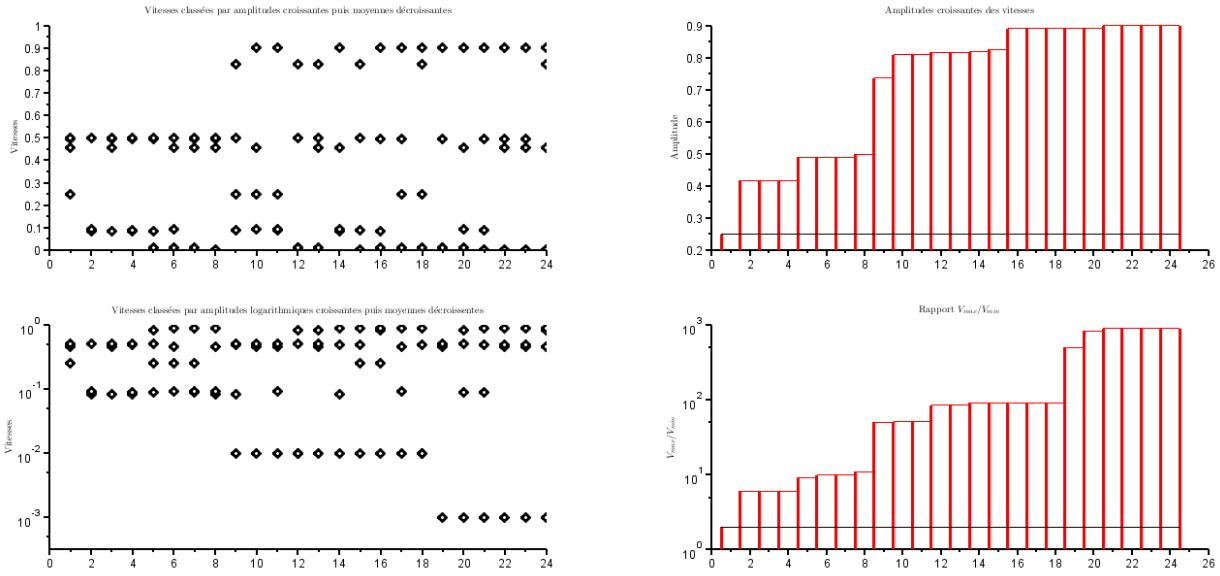


FIGURE 13. Vitesses v_0, amplitudes et rapport v_{max}/v_{min} pour les 4 couples, pour les 24 permutations, classées par amplitude croissante

On voit très nettement que dans la pemière situation de la figure 13 (celle où $k_- = k_{cat}$), les vitesses sont relativement proches tandis que certaines permutations ont un écart très important entre la vitesse minimale (parfois excessivement faible) et la vitesse maximale.

On peut aussi étudier le rapport v_{max}/v_{min}. C'est ce que l'on observe sur le quatrième graphique de la figure 13. Ce graphique nous permet notamment de voir que pour la situation où $k_- = k_{cat}$, on a $v_{max} \simeq v_{min}$ (rapport proche de 1) tandis que dans le pire des cas, on a $v_{max} \simeq 1000 v_{min}$. On a, dans ce cas de figure, un écart de 3 ordres de grandeur ; avec v_{min} très faible (proche de 0).

4.5. **Et pour d'autres valeurs de $[S_1]$?**

On regarde ce qui se produit pour des valeurs extrèmes de $[S_1]$. On se place dans deux cas particuliers :

— $[S_1] = 10^{-7} mol.L^{-1}$: l'asymptote est atteinte pour tous les couples ;
— $[S_1] = 10^{-1} mol.L^{-1}$: on est à saturation pour tous les couples.

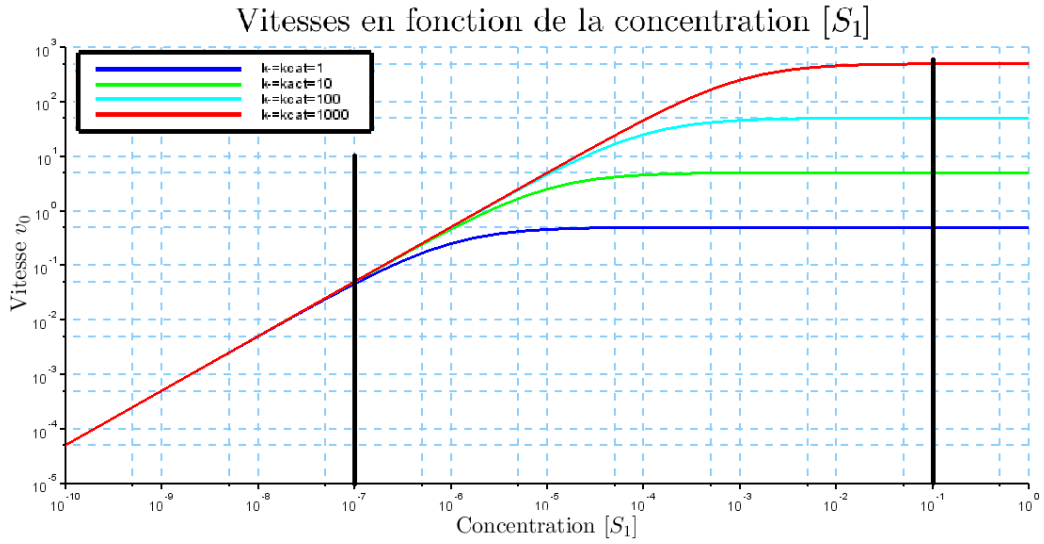


FIGURE 14. Séléction des valeurs pertinentes de $[S_1]$

4.5.1. $[S_1] = 10^{-7} mol.L^{-1}$.

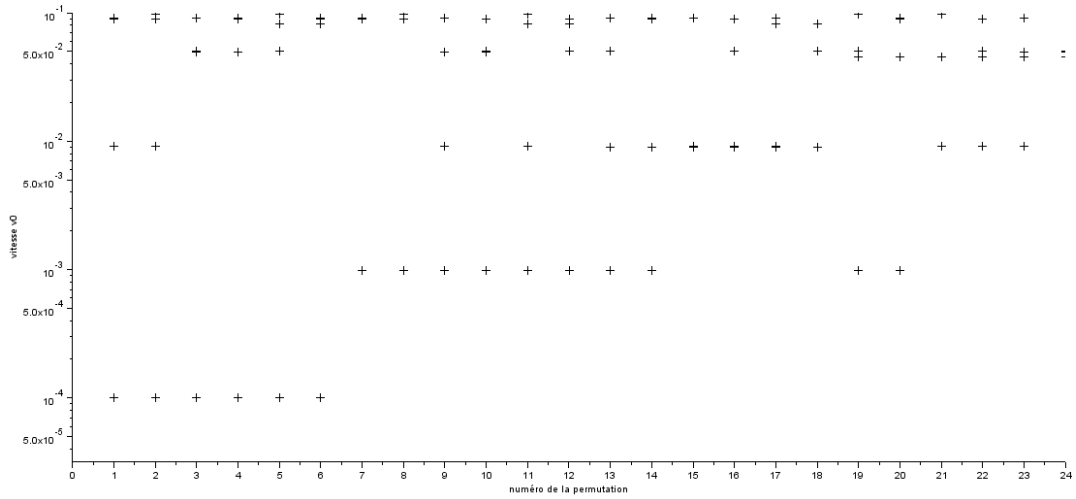


FIGURE 15. Vitesses v_0 pour les 4 couples, pour les 24 permutations à $[S_1] = 10^{-7} mol.L^{-1}$

Le cas idéal, en 24 sur la figure 15, est cette fois ci à une amplitude de vitesses encore plus faible. On classe les vitesses.

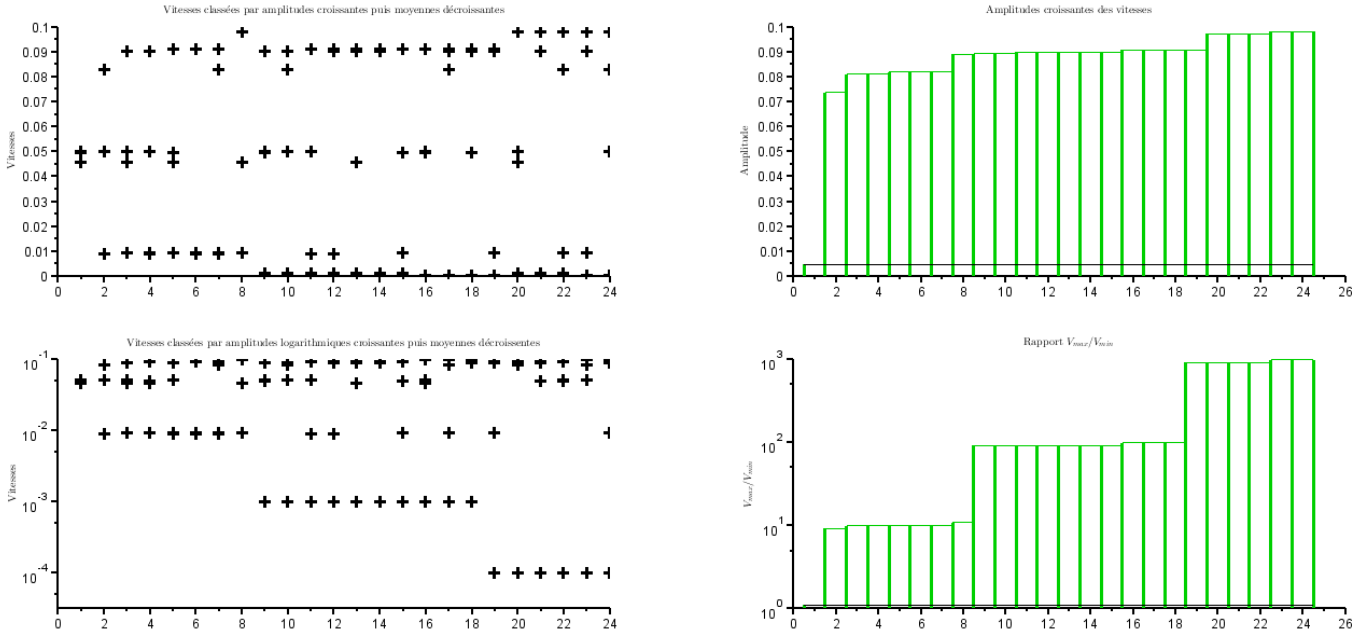


FIGURE 16. Vitesses v_0, amplitudes et rapport v_{max}/v_{min} pour les 4 couples, pour les 24 permutations, classées par amplitude croissante, $[S_1] = 10^{-7} mol.L^{-1}$

On observe que la moindre permutation augmente considérablement l'amplitude entre les vitesses. On voit sur la figure 16 que l'amplitude atteint des paliers. Dès que l'on effectue une permutation, on a au mieux $v_{max} = 10 v_{min}$, et au pire, $v_{max} = 1000 v_{min}$.

Que ce passe-t-il en hautes concentrations ?

4.5.2. $[S_1] = 10^{-1} mol.L^{-1}$.

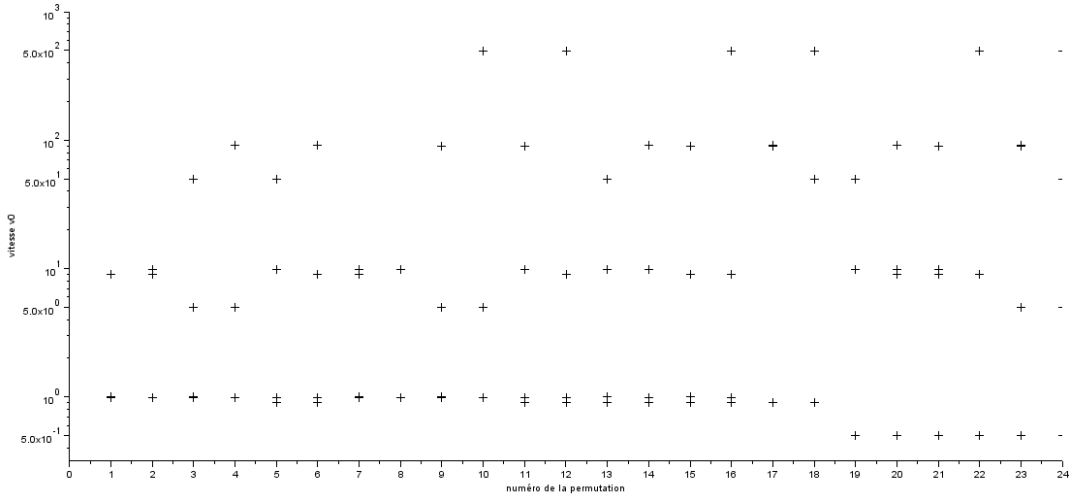


FIGURE 17. Vitesses v_0 pour les 4 couples, pour les 24 permutations à $[S_1] = 10^{-1} mol.L^{-1}$

Pour une forte concentration $[S_1]$, on est maintenant dans un cas "inversé" :
C'est lorsque $k_- = k_{cat}$ que l'amplitude entre la vitesse maximale et la vitesse minimale est la plus élevée. Cependant, il faut remarquer que même dans le "meilleur" des cas où les vitesses sont les plus proches possibles, on a un facteur 10 entre v_{min} et v_{max}. Il n'y a aucune permutation qui permet l'homogénéité des vitesses en grandes concentrations.

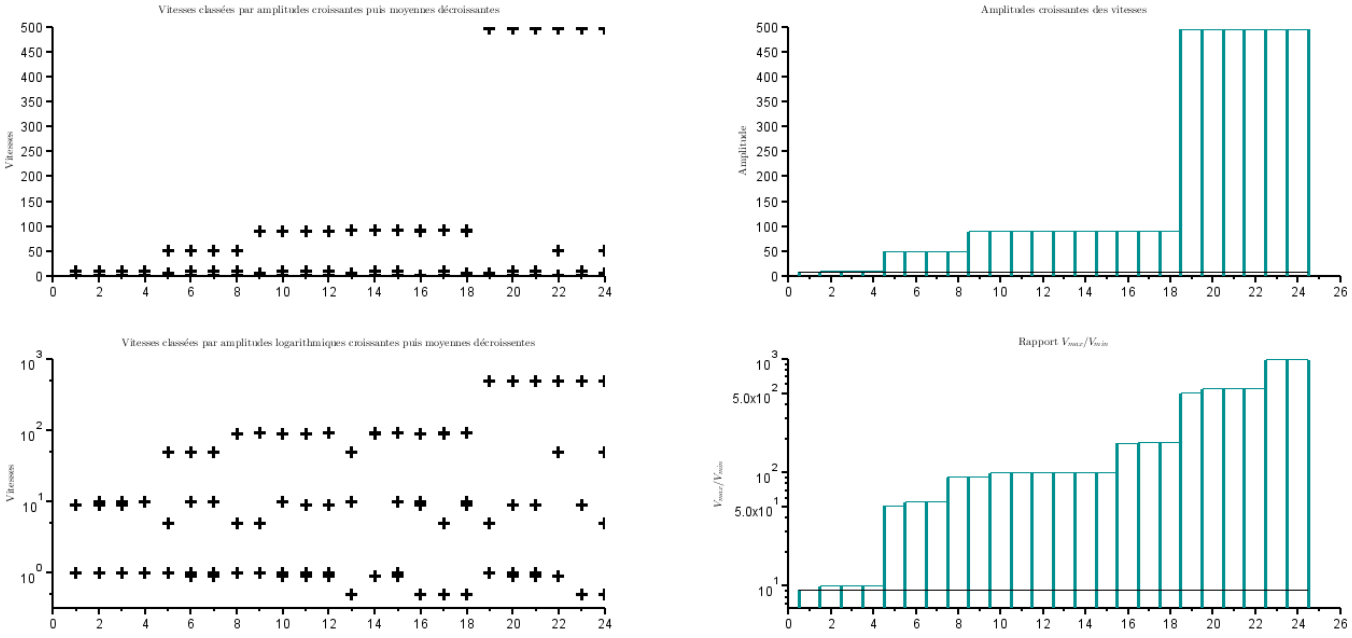


FIGURE 18. Vitesses v_0, amplitudes et rapport v_{max}/v_{min} pour les 4 couples, pour les 24 permutations, classées par amplitude croissante, $[S_1] = 10^{-1} mol.L^{-1}$

Pour comprendre ce résultat, il faut regarder les variations de vitesse avec $[S_1]$ pour plusieurs permutations remarquables. (Figure 19)

4.6. **Variations de la vitesse en fonction de la concentration $[S_1]$ pour 3 permutations remarquables.**

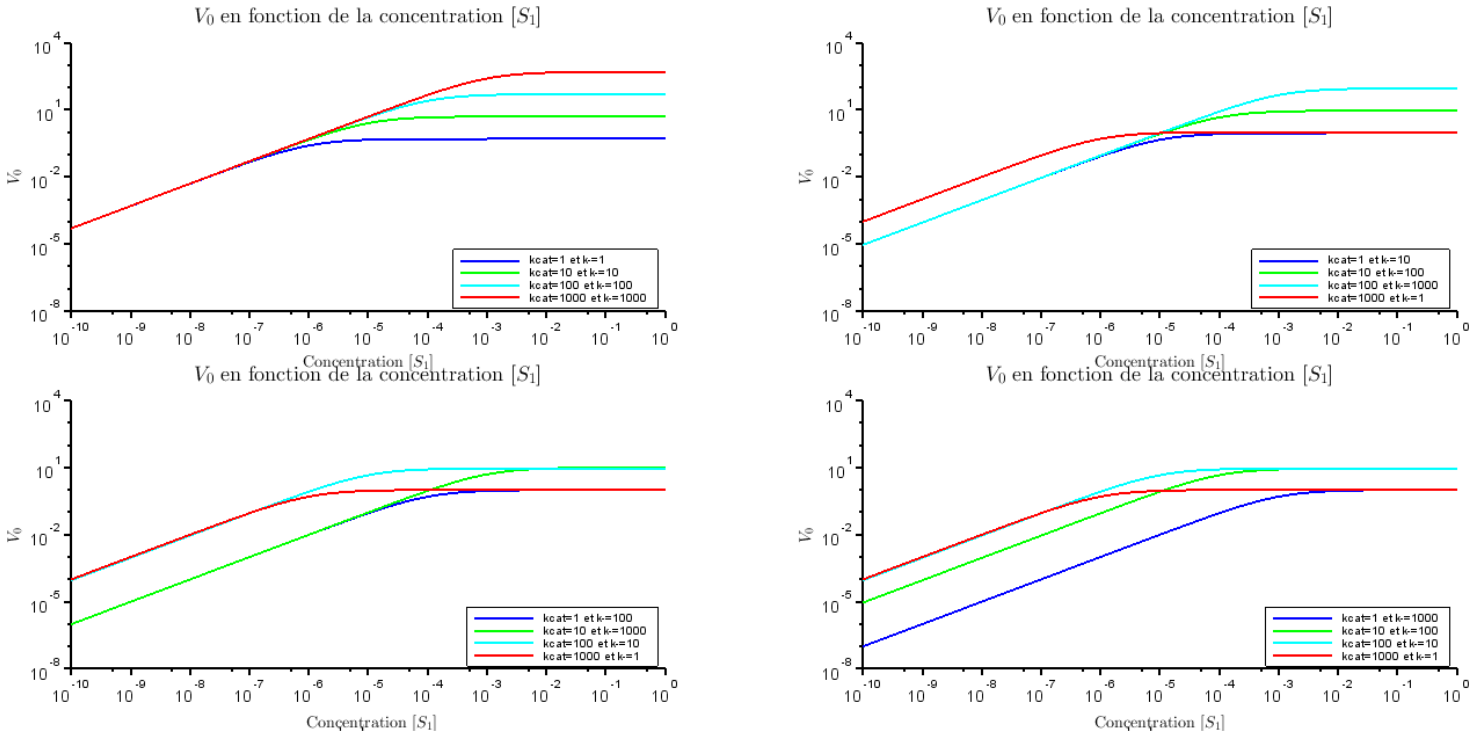


FIGURE 19. Vitesse en fonction de la concentration $[S_1]$ pour 3 permutations remarquables

Lorsqu'on effectue des permutations, on observe que l'on n'a plus cette asymptote commune comme sur la figure 6.

En effet, si on pose, pour chaque couple

$$r = \frac{k_-}{k_{cat}}$$

$$v_0([S_1]) = \frac{r k_{cat} k_+ [S_1]}{(1+r)(r k_{cat} + k_+ [S_1])}$$

L'équation de l'asymptote devient :

$$log(v_0([S_1])) \underset{[S_1]\to 0}{\sim} log([S_1]) + log\left(\frac{k_+}{1+r}\right).$$

$y = x + b$ avec $b = log\left(\frac{k_+}{1+r}\right)$. L'asymptote ne dépend en fait que du rapport r entre k_- et k_{cat}. Ce qu'il faut retenir est que la corrélation que l'on observe peut être un cas $k_- = k_{cat}$ mais celà peut également être un cas où r ne vaut pas 1.

On note n le nombre de couples que l'on forme. Alors on pose

$$\forall i \in [1, n],\ r_i = \frac{k_{-i}}{k_{cati}}$$

Si pour tous les couples i, $r_i = r$ une constante, alors on a une asymptote commune. On a conservation de l'homogénéité des vitesses. On sait que les constantes de vitesse k_{-i} et k_{cati} sont dépendantes des conditions de température, de concentration en ions ($[Mg^{2+}]$, $[K^+]$, etc), de pH... Il se peut que ces constantes de vitesses ne soient pas sensibles de la même manière aux variations de ces conditions.

Pourtant, il est raisonnable de penser que tous les k_{-i} vont se comporter d'une même manière et tous les k_{cati} d'une autre. Par exemple, les k_- sont plus dépendants de la concentration en ions que les k_{cat}.

Il est possible d'observer alors une corrélation, comme sur la figure 20, qui ne soit pas exactement $\forall i \in [1,n], k_{-i} = k_{cati}$ mais $\forall i \in [1,n], k_{-i} = rk_{cati}$; corrélation qui conserve l'homogénéité des vitesses à faibles concentrations en transporteurs $[S_1]$ malgré les variations de conditions de température, pH et concentration en ions. On ne maximise alors plus la robustesse α du système mais on préserve l'indépendance de la vitesse face aux variations de composition du message.

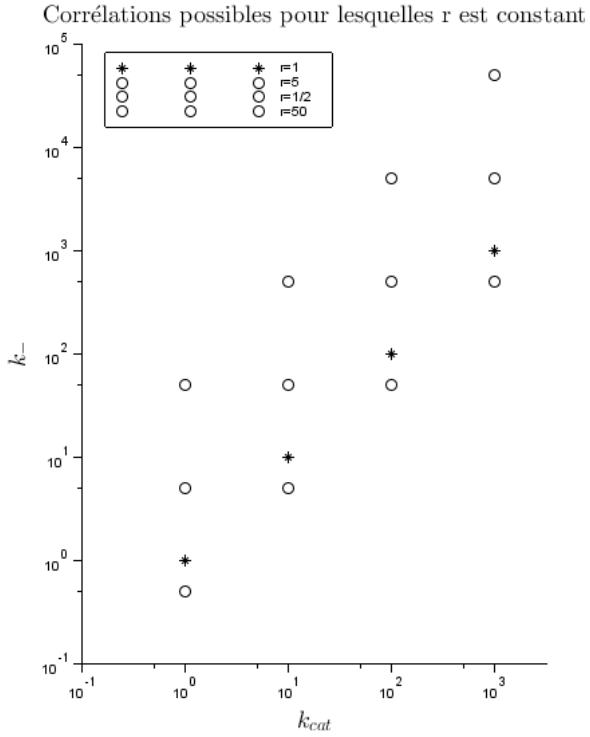


FIGURE 20. Exemples de corrélations à rapport constant

Ainsi, si, pour un système donné, le rapport $r = \frac{k_-}{k_{cat}}$ est constant (le même pour tous les couples), on a une asymptote commune dans les faibles concentrations qui témoigne de l'homogénéité des vitesses de polymérisation. On a fait la conjecture que $r = 1$, i.e. $k_- = k_{cat}$, puisque dans ce cas précis, on maximise la robustesse α, et on est donc, dans un environnement où $[S_1]$ est fluctuant, toujours au plus proche du maximum.

Après cette étude en 0, étudions ce qui se passe lorsque $[S_1] \to +\infty$:

$$v_0([S_1]) \underset{[S_1]\to+\infty}{\longrightarrow} \frac{k_- k_{cat}}{k_- + k_{cat}} = \frac{r}{1+r} k_{cat}$$

Remarque :
On observe un changement de pente autour de la concentration critique $[S_1] = 10^{-6} mol.L^{-1}$ pour $k_- = k_{cat} = 1s^{-1}$ et autour de $[S_1] = 10^{-5} mol.L^{-1}$ pour $k_- = k_{cat} = 10s^{-1}$, etc. Cette valeur critique se justifie en considérant l'expression de la vitesse comme la fonction de transfert d'un passe haut :

$$v_0 = \frac{k_- k_{cat}}{k_{cat} + k_-} \frac{\frac{k_+}{k_-}[S_1]}{1 + \frac{k_+}{k_-}[S_1]}$$

Rappelons que l'expression d'un passe haut est :

$$H(\omega) = K \frac{\frac{\omega}{\omega_0}}{1 + \frac{\omega}{\omega_0}}$$

L'allure d'un passe haut est celui de la figure 21.

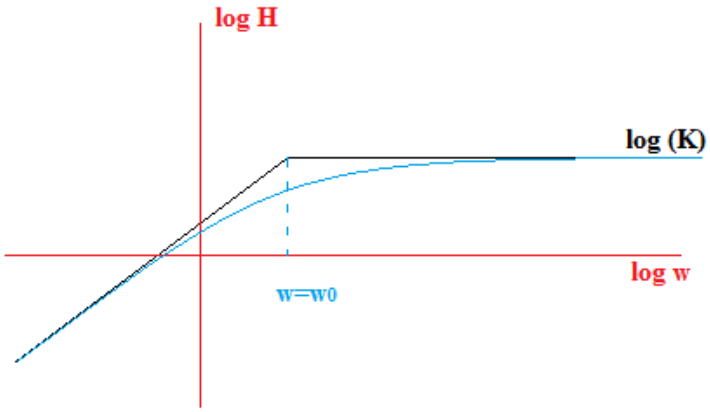


FIGURE 21. Allure d'un passe haut

Cette valeur critique de $[S_1]$ pour laquelle on observe un changement d'allure vaut donc

$$[S_1]_{critique} = \frac{k_-}{k_+}$$

Pour conclure sur ce qui précède, on a donc une uniformité de vitesse pour des faibles valeurs de $[S_1]$ lorsque $k_- = k_{cat}$; uniformité que l'on ne retrouve pas pour d'autres permutations.

5. Variation de λ

5.1. **Apparition du facteur λ dans l'équation de réaction.**

On peut se demander ce que fait la vitesse lorsque l'on modifie l'équation de réaction de la manière suivante :

$$E + S_1 \underset{k_+}{\overset{k_-}{\leftrightarrows}} ES_1 \xrightarrow{k_{cat}} EP_1 \xrightarrow{\lambda k_-} E + P_1$$

La présence de ce λ peut se justifier par la présence du L1 Stalk sur le ribosome. La figure 22 permet de mieux comprendre la situation.

Source : `httpwww.nature.comnaturejournalv443n7112fig_tabnature05126_F4.html`

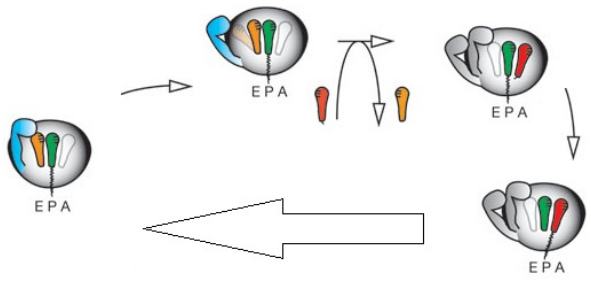


Figure 22. Schématisation du rôle du L1 Stalk

L'expression de la vitesse est alors modifiée :

$$v_0 = \frac{\lambda k_- k_+ k_{cat}[S_1]}{(k_{cat} + k_-)(\lambda k_- + k_+[S_1])}.$$

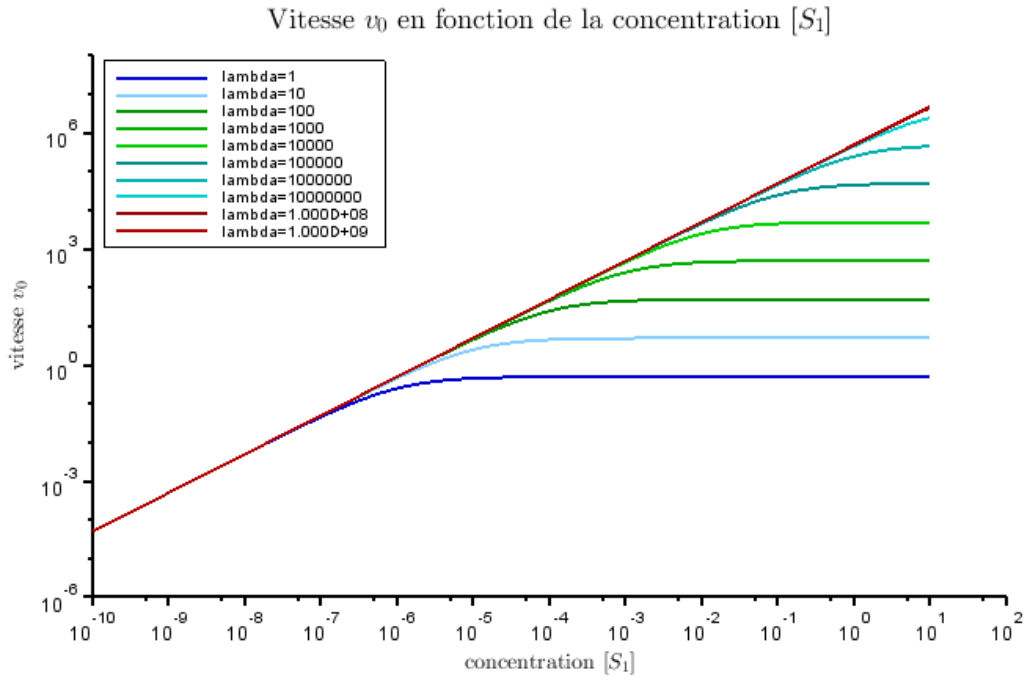


Figure 23. Vitesses v_0 en fonction de la concentration pour $k_- = k_{cat} = 1s^{-1}$ pour plusieurs valeurs de λ

On distingue, sur la figure 23, que changer le lambda ne modifie pas l'asymptote de convergence vers 0 dans les faibles concentrations. En revanche, on augmente la concentration critique et la vitesse à saturation : la valeur limite de la vitesse lorsque $[S_1] \to +\infty$.

5.2. **Recherche d'une optimisation.**

On comprend bien que ce sont les couples qui ont les plus faibles valeurs de vitesse limite quand $[S_1] \to +\infty$ qui ont le plus besoin d'augmenter ce λ. En effet, ce sont les couples avec les vitesses les plus lentes qui limitent la vitesse de polymérisation ; c'est donc ces couples là qui doivent être optimisés. On peut ainsi justifier l'idée d'un λ dépendant de k_- .

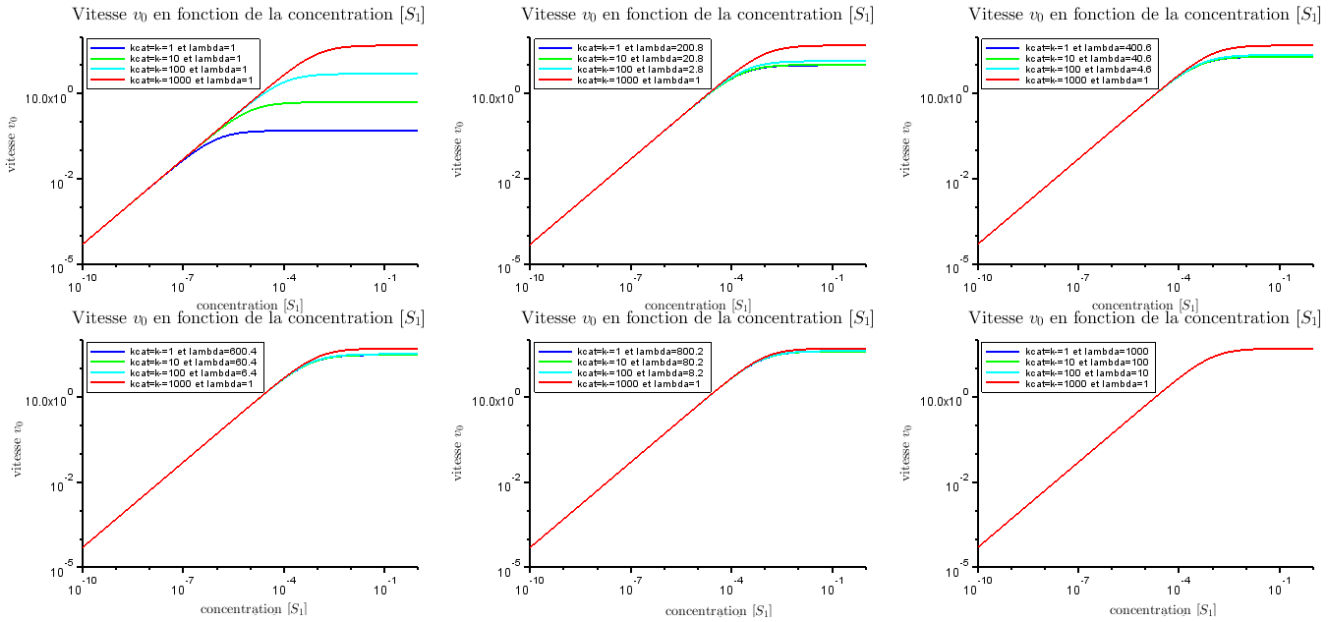


FIGURE 24. Vitesses v_0 en fonction de la concentration pour 4 couples $k_- = k_{cat}$ pour plusieurs valeurs de λ

À concentration $[S_1]$ fixée, il est clair que lorsque λ augmente, la vitesse v_0 augmente. C'est un résultat visible sur la figure 23. Pour une valeur judicieusement choisie de λ, on a convergence lorsque $[S_1] \to +\infty$ vers une même valeur, commune aux 4 couples $k_- = k_{cat}$.

5.3. **Que vaut λ ?**

On veut que pour tous les couples $(k_- = k_{cat})$, la limite de v_0 quand $[S_1] \to +\infty$ soit la même.
Or $\forall (k_- = k_{cat}), v_0 \underset{[S_1]\to+\infty}{\longrightarrow} \lambda\frac{k_-}{2}$. On voudrait également que cette limite soit maximale, valant la vitesse limite maximale des 4 couples ; c'est à dire la vitesse limite du $k_{-max} = \underset{4couples}{max}(k_-)$.
Lorsque λ est optimal, on a donc

$$\forall k_-, \underset{[S_1]\to+\infty}{lim}(v_0(k_{-max})) = \underset{[S_1]\to+\infty}{lim}(v_0(k_-))$$

Ainsi

$$\forall k_-, \lambda_{optimal}(k_-) = \frac{k_{-max}}{k_-}$$

C'est le cas du dernier graphique de la figure 24 : on a optimisé la vitesse de manière à ce que la vitesse de polymérisation soit indépendante de la composition du message avec une vitesse la plus grande possible.

On peut imaginer que le L1 Stalk oscille à la même fréquence que celle avec laquelle le transporteur le plus rapide se dissocie du codon. De cette manière, tous les couples codon/acide aminé se polymérisent à la même vitesse :

$$v_{0commune} = \frac{k_{-max} k_+ [S_1]}{2(k_{-max} + k_+ [S_1])}.$$

6. Conclusion

Les résultats de Chloé m'ont permis de confirmer que la vitesse de polymérisation a un maximum lorsqu'on fait varier k_-. Sous concentration $[S_1]$ fluctuante, on est au plus proche du maximum lorsque $k_- = k_{cat}$ ce qui permet d'appuyer la corrélation observée entre l'énergie codon-anticodon et le volume de l'acide aminé. Si on se place à $[S_1] = 10^{-6} mol/L$, on est dans un cas optimisé où on a :

— une vitesse de polymérisation élevée : rapidité de la polymérisation ;
— un homogénéité des vitesses des 4 couples : invariance de la vitesse de polymérisation devant la composition du message.

Si on s'éloigne de la corrélation, la vitesse de polymérisation va être dépendante de la composition du message : des écarts importants de vitesse entre les différents codons apparaissent. Enfin, cette étude révèle un rôle potentiel du L1 Stalk : l'uniformisation des vitesses à hautes concentrations (homogénéité des vitesses de dissociation après la catalyse).

Lors de ce stage, j'ai pu observer le fonctionnement d'un laboratoire de recherche. J'ai eu l'occasion de participer aux lab-meeting au cours desquels j'ai, par deux fois, présenté mes résultats. C'est lors de ces présentations que j'ai compris l'importance de la communication. Pour moi, ce fut un réel défi de "faire parler mes résultats". J'ai ainsi pris conscience que des résultats ne valent rien si on n'en fait pas une histoire. J'ai du faire appel à l'aspect biologique du problème pour interpréter mes courbes et les faire comprendre aux chercheurs présents.

J'ai également pu perféctionner mes capacités de codage sur LaTeX et Scilab.

Ce stage a confirmé mon envie d'enseigner. En effet, j'ai aimé partager ce que je savais dans le but de faire avancer cette énigme biologique. L'aspect chimique m'a également intriguée.

Je tiens à remercier toutes les personnes qui ont contribué au succès de ce stage et qui m'ont aidé dans mes recherches, notamment mon maître de stage, M. Jean LEHMANN qui a su partager ses connaissances et me donner de son temps. Je remercie toute l'équipe du laboratoire pour son accueil et pour l'enseignement qu'elle m'a apporté.

3 Projet de recherche de M1 : Étude des théorèmes fondamentaux de l'intégration en dimension n.

Projet M1 : Intégration en dimension n.

Christian Marguerite & Julie Hémont

Table des matières

1 Introduction.

Il s'agira d'étudier l'intégration en dimension n pour Kurzweil et Henstock puis pour Mawhin. On s'intéressera dans un premier temps à la formule de Gauss-Green :

$$\int_\Omega \text{div v} = \int_{\partial\Omega} \text{v} \cdot n,$$

qui relie l'intégrale de la divergence d'un champ de vecteur de classe C^1 défini sur un volume à son flux à travers le bord. Il s'agit d'une généralisation en dimension n du théorème fondamental classique donné par :

$$\int_a^b F' = F(b) - F(a),$$

pour toute fonction réelle F de classe $\mathcal{C}^1$ sur $[a, b]$. On verra d'abord l'insuffisance de la théorie de Kurzweil et Henstock pour démontrer ce théorème de la divergence pour un champ de vecteurs supposé simplement différentiable. Cependant, pour la théorie de Mawhin, la formule de Gauss-Green est valide.

En revanche, on montrera que la théorie de Kurzweil et Henstock s'avère efficace pour obtenir le Théorème de Fubini.

En fait, la non-validité de la formule de Gauss-Green généralisée pour la théorie de Kurzweil et Henstock est due à l'incompatibilité des énoncés des théorèmes dans leur version générale. En effet, on prouvera qu'on ne peut pas avoir, lorsqu'on construit une théorie d'intégrale, à la fois le Théorème de Gauss-Green et le théorème de Fubini.

2 Définitions et premières propriétés.

Soit $n \in \mathbb{N}^*$. On introduit, sur $\mathbb{R}^n$, les notations suivantes :

1. Pour tout $x \in \mathbb{R}^n$, pour tout $i \in \{1, \cdots, n\}$, on note x_i la $i^{\text{ème}}$ coordonnée de x ;
2. La norme infinie sur $\mathbb{R}^n$ est notée $|\cdot|_\infty$ et pour tout $x \in \mathbb{R}^n$,on a

$$|x|_\infty = \max_{1 \le i \le n} |x_i|;$$

3. La norme 2 sur $\mathbb{R}^n$ est notée $\|\cdot\|_2$ et pour tout $x \in \mathbb{R}^n$,on a

$$\|x\|_2 = \sqrt{\sum_{i=1}^n {x_i}^2};$$

4. La boule de centre $x \in \mathbb{R}^n$ et de rayon r est notée

$$\mathcal{B}_\infty(x, r) = \{y \in \mathbb{R}^n : |x - y|_\infty \le r\}.$$

2.1 Pavés, pavés ouverts et semi-pavés.

Définition 2.1. *Soit $E \subset \mathbb{R}^n$. Une* jauge δ sur E *est une application à valeurs dans $\mathbb{R}_+^*$:*

$$\begin{aligned} \delta : E &\longrightarrow \mathbb{R}_+^* \\ a &\longmapsto \delta(a). \end{aligned}$$

La notion de *jauge* permet d'avoir un contrôle sur les éléments que l'on manipule.

Définition 2.2. *Un ensemble $K \subset \mathbb{R}^n$ est un* pavé de $\mathbb{R}^n$ *lorsque K est le produit cartésien d'intervalles fermés de $\mathbb{R}$. En d'autres termes,*

$$K = \prod_{i=1}^n K_i = K_1 \times K_2 \times \ldots \times K_n$$

où $\forall 1 \le i \le n$, $K_i = [a_i, b_i] \subset \mathbb{R}$ sont des intervalles fermés de $\mathbb{R}$.

On définit de même un *pavé ouvert* et un *semi-pavé* de $\mathbb{R}^n$.

Définition 2.3. *Un* pavé ouvert de $\mathbb{R}^n$ *est un ensemble* $J \subset \mathbb{R}^n$ *tel que*

$$J = \prod_{i=1}^{n} J_i$$

où $\forall 1 \leq i \leq n$, $J_i =]a_i, b_i[\subset \mathbb{R}$ *sont des intervalles ouverts de* $\mathbb{R}$.

Définition 2.4. *Un* semi-pavé de $\mathbb{R}^n$ *est un ensemble* $I \subset \mathbb{R}^n$ *tel que*

$$I = \prod_{i=1}^{n} I_i$$

où $\forall 1 \leq i \leq n$, $I_i =]a_i, b_i] \subset \mathbb{R}$ *sont des intervalles semi-ouverts à gauche de* $\mathbb{R}$.

De même, on introduit la notion de *cube* en dimension n.

Définition 2.5. *On dit qu'un pavé* $K = \prod_{i=1}^{n} [a_i, b_i]$ *est un* n-cube *lorsque*

$$b_1 - a_1 = b_2 - a_2 = \ldots = b_n - a_n.$$

Définition 2.6. *Soit* $(K^{(i)})_{i \in \mathbb{N}}$ *une suite de pavés de* $\mathbb{R}^n$. *On dit que la suite est une suite de pavés emboîtés lorsque pour tout* $i \in \mathbb{N}$,

$$K^{(i+1)} \subset K^{(i)}.$$

Si $n = 1$, *on dit qu'il s'agit d'une suite de segments emboîtés.*

Lemme 2.1 (Théorème des segments emboîtés.)**.** *L'intersection de segments emboîtés de* $\mathbb{R}$ *est non vide.*

Démonstration. Soit $([a_j, b_j])_{j \in \mathbb{N}}$ une suite de segments emboîtés. Supposons que $(b_j - a_j)_{j \in \mathbb{N}}$ tende vers 0. Autrement, le résultat est évident. Soit pour tout $j \in \mathbb{N}$, $x_j \in [a_j, b_j]$. Alors montrons que la suite $(x_j)_{j \in \mathbb{N}}$ est de Cauchy. Soit $\varepsilon > 0$. Il existe un rang $N \in \mathbb{N}$ tel que $\forall n \geq N$, $|a_n - b_n| < \varepsilon$. Alors $\forall m, n \geq N$,

$$|x_m - x_n| \leq |a_N - b_N| < \varepsilon.$$

Donc $(x_j)_{j \in \mathbb{N}}$ est de Cauchy dans $\mathbb{R}$ donc converge. Notons x sa limite. Alors $x \in \bigcap_{j \in \mathbb{N}} [a_j, b_j]$. □

Proposition 2.2 (Théorème des pavés emboîtés.)**.** *Soit* $\left(K^{(i)}\right)_{i \in \mathbb{N}}$ *une suite de pavés emboîtés. Alors*

$$\bigcap_{i \in \mathbb{N}} K^{(i)} \neq \emptyset.$$

Démonstration. Soit $\left(K^{(i)}\right)_{i \in \mathbb{N}} = \left(\prod_{j=1}^{n} {K_j}^{(i)} \right)_{i \in \mathbb{N}}$ une suite de pavés emboîtés. Ce résultat est une conséquence directe du théorème des segments emboîtés dans $\mathbb{R}$. En effet, si on suppose que $\bigcap_{i \in \mathbb{N}} {K_j}^{(i)} \neq \emptyset$ pour chaque $1 \leq j \leq n$, alors $\forall 1 \leq j \leq n$, $\exists x_j \in \bigcap_{i \in \mathbb{N}} {K_j}^{(i)}$ et alors

$$x := (x_1, \ldots, x_n) \in \bigcap_{i \in \mathbb{N}} K^{(i)}.$$

□

Remarque 2.1. *Si* $A \subset \mathbb{R}^m$ *et* $B \subset \mathbb{R}^p$, *alors*

$$\operatorname{int}(A \times B) = \operatorname{int} A \times \operatorname{int} B$$

et

$$\operatorname{adh}(A \times B) = \operatorname{adh} A \times \operatorname{adh} B.$$

De cette remarque, on déduit donc que si K est le pavé $K = \prod_{i=1}^{n} [a_i, b_i]$, si J est le pavé ouvert $J = \prod_{i=1}^{n}]a_i, b_i[$ et si I est le semi-pavé $I = \prod_{i=1}^{n}]a_i, b_i]$, alors

$$\text{int}(K) = \text{int}(J) = \text{int}(I) = J$$

et

$$\text{adh}(K) = \text{adh}(J) = \text{adh}(I) = K.$$

Définition 2.7. *Soit $I = \prod_{i=1}^{n}]a_i, b_i]$, un semi-pavé de $\mathbb{R}^n$. On définit le* diamètre de I, *noté $d(I)$ par*

$$d(I) = \max_{1 \leq i \leq n} (b_i - a_i).$$

Définition 2.8. *Soit $I = \prod_{i=1}^{n}]a_i, b_i]$, un semi-pavé de $\mathbb{R}^n$. On définit la* mesure *de I, notée $\mu(I)$ par*

$$\mu(I) = (b_1 - a_1)(b_2 - a_2) \cdots (b_n - a_n) = \prod_{i=1}^{n} (b_i - a_i).$$

2.2 Partitions et lemme de Cousin.

Définition 2.9. *Soit $E \subset \mathbb{R}^n$. On dit qu'une famille $(E_\alpha)_{\alpha \in A}$ est une* partition de E *lorsque*

1. $\forall \alpha \in A,\ \forall \beta \in A,\ \alpha \neq \beta \Rightarrow E_\alpha \cap E_\beta = \emptyset$;
2. $E = \bigcup_{\alpha \in A} E_\alpha$.

On dit aussi que $(E_\alpha)_{\alpha \in A}$ partitionne E.

Pour la suite, on s'intéresse à partitionner les semi-pavés. Soit $I \subset \mathbb{R}^n$ un semi-pavé. On peut partitionner I.

Définition 2.10. *Une* P-partition *de I est une famille finie*

$$\Pi = \left\{ \left(\left(x^j, I^j \right) \right)_{1 \leq j \leq m} \right\} = \left\{ \left(x^1, I^1 \right), \left(x^2, I^2 \right), \ldots, \left(x^m, I^m \right) \right\}$$

telle que

1. $\left(I^j \right)_{1 \leq j \leq m}$ *est une partition de I ;*
2. $\forall 1 \leq j \leq m,\ x^j \in I^j$.

Définition 2.11. *Soit δ une jauge sur $\overline{I}$. Soit $\Pi = \left\{ \left(\left(x^j, I^j \right) \right)_{1 \leq j \leq m} \right\}$ une P-partition de I. On dit que Π est une* P-partition δ-fine de I *lorsque $\forall 1 \leq j \leq m,\ I^j \subset \mathcal{B}_\infty \left(x^j, \delta \left(x^j \right) \right)$.*

Remarque 2.2. $I^j \subset \mathcal{B}_\infty \left(x^j, \delta \left(x^j \right) \right) \Leftrightarrow \overline{I^j} \subset \mathcal{B}_\infty \left(x^j, \delta \left(x^j \right) \right)$

Théorème 2.3 (Lemme de Cousin)**.** *Soit I une semi-pavé de $\mathbb{R}^n$. Pour tout jauge δ sur $\overline{I}$, il existe une P-partition δ-fine de I.*

Démonstration. Soit $I = \prod_{i=1}^{n}]a_i, b_i]$. Par l'absurde, supposons que δ soit une jauge sur $\overline{I}$ telle que I n'admette pas de P-partition δ-fine. On découpe chaque intervalle de la manière suivante :

$$\forall 1 \leq i \leq n,\ ]a_i, b_i] = \left] a_i, \frac{a_i + b_i}{2} \right] \sqcup \left] \frac{a_i + b_i}{2}, b_i \right].$$

On obtient ainsi $\mathcal{P}$, une partition de I de semi-pavés dont les longueurs des côtés ont été divisées par deux.

Si tous les semi-pavés de $\mathcal{P}$ possédaient une P-partition δ-fine, alors I aussi. Donc il existe un semi-pavés de $\mathcal{P}$ qui n'admet pas de P-partition δ-fine. Notons $I^{(1)}$ ce semi-pavé.

On partitionne de la même manière $I^{(1)}$ et on obtient de nouveau une partition de $I^{(1)}$ qui est composée de semi-pavés de côtés divisés pas 4 par rapport à ceux de I. On peut y trouver un semi-pavé $I^{(2)}$ qui ne

possède pas de P-partition δ-fine. On itère le procédé jusqu'à obtenir une suite de semi-pavés emboîtés $\left(I^{(n)}\right)_{n\in\mathbb{N}}$ n'admettant pas de P-partition δ-fine et telle que

$$\forall k \in \mathbb{N},\ d(I^{(k)}) \leq \frac{d(I)}{2^k}.$$

Par le théorème des segments emboîtés, $\bigcap_{k\in\mathbb{N}} \overline{I^{(k)}} \neq \emptyset$, donc il existe $c \in \bigcap_{k\in\mathbb{N}} \overline{I^{(k)}}$. Comme δ est une jauge sur $\overline{I}$, $\delta(c) > 0$. Il existe donc $N \in \mathbb{N}$ tel que $\frac{d(I)}{2^N} \leq \delta(c)$. Alors, comme $c \in \overline{I^{(N)}}$, on a

$$\forall x \in I^{(N)},\ |x-c|_\infty \leq \frac{d(I)}{2^N} \leq \delta(c).$$

Ainsi, $I^{(N)} \subset \mathcal{B}_\infty(c, \delta(c))$ et $\left(c, I^{(N)}\right)$ est une P-partition δ-fine de $I^{(N)}$. Contradiction. □

Définition 2.12. *On dit que deux demi-pavés* $I^{(1)} = \prod_{i=1}^{n} \left]a_i^{(1)}, b_i^{(1)}\right]$ *et* $I^{(2)} = \prod_{i=1}^{n} \left]a_i^{(2)}, b_i^{(2)}\right]$ *sont* semblables *lorsque*

$$\frac{b_1^{(1)} - a_1^{(1)}}{b_1^{(2)} - a_1^{(2)}} = \frac{b_2^{(1)} - a_2^{(1)}}{b_2^{(2)} - a_2^{(2)}} = \cdots = \frac{b_n^{(1)} - a_n^{(1)}}{b_n^{(2)} - a_n^{(2)}}.$$

Définition 2.13. *Soit I un semi-pavé de $\mathbb{R}^n$. Une P-partition* $\Pi = \left\{\left(\left(x^j, I^j\right)\right)_{1\leq j\leq m}\right\}$ *de I est dite* régulière *lorsque les I^j sont des semi-pavés semblables à I*

Proposition 2.4 (Corollaire de la démonstration du lemme de Cousin.)**.** *Soient I un semi-pavé de $\mathbb{R}^n$ et δ une jauge sur $\overline{I}$. Alors il existe une P-partition δ-fine régulière de I.*

Définition 2.14. *Soient I un semi-pavé de $\mathbb{R}^n$,* $\Pi = \left\{\left(\left(x^j, I^j\right)\right)_{1\leq j\leq m}\right\}$ *une P-partition de I et $f : \overline{I} \longrightarrow \mathbb{R}^p$ une fonction définie sur $\overline{I}$. On définit la* somme de Riemann *de f associée à Π par*

$$S(I, f, \Pi) := \sum_{j=1}^{m} f(x^j)\mu(I^j).$$

Définition 2.15. *Soit I un semi-pavé de $\mathbb{R}^n$. Soit $f : \mathbb{R}^n \longrightarrow \mathbb{R}^p$ une fonction définie sur $\overline{I}$. On dit que f est* intégrable au sens de Kurzweil et Henstock *sur $\overline{I}$ s'il existe $J \in \mathbb{R}^p$ tel que $\forall \varepsilon > 0$, il existe une jauge δ sur $\overline{I}$ telle que pour toute P-partition δ-fine Π de I,*

$$\|S(I, f, \Pi) - J\|_2 \leq \varepsilon.$$

Dans ce cas, J est unique.

Démonstration. Supposons que J et J' vérifient :
$\forall \varepsilon > 0$, il existe une jauge δ sur $\overline{I}$ telle que pour toute P-partition δ-fine Π de I,

$$\|S(I, f, \Pi) - J\|_2 \leq \varepsilon$$

et $\forall \varepsilon > 0$, il existe une jauge δ sur $\overline{I}$ telle que pour toute P-partition δ-fine Π de I,

$$\|S(I, f, \Pi) - J'\|_2 \leq \varepsilon.$$

Soit $\varepsilon > 0$. Alors il existe δ et δ' telles que si Π est une P-partition δ-fine de I et Π' est une P-partition δ'-fine de I, alors

$$\|S(I, f, \Pi) - J\|_2 \leq \frac{\varepsilon}{2}$$

et

$$\|S(I, f, \Pi') - J'\|_2 \leq \frac{\varepsilon}{2}.$$

Posons $\delta'' : x \longmapsto \min(\delta(x), \delta'(x))$. Alors si Π'' est une P-partition δ''-fine, Π'' est aussi δ-finie et δ'-fine. Donc

$$\|J - J'\|_2 \leq \|J - S(I, f, \Pi'')\|_2 + \|S(I, f, \Pi'') - J'\|_2 \leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

Donc $\|J - J'\|_2 = 0$. D'où $J = J'$. □

3 Théorème fondamental : Formule de Gauss-Green.

3.1 Définitions et notations.

Définition 3.1. *Soit* $A = \prod_{i=1}^{n} [a_i^-, a_i^+] \subset \mathbb{R}^n$ *un pavé.*

On appelle $i^{\text{ème}}$ projection de A *le sous-ensemble*

$$A_{(i)} := [a_1^-, a_1^+] \times \cdots \times \widehat{[a_i^-, a_i^+]} \times \cdots \times [a_n^-, a_n^+] \subset \mathbb{R}^{n-1},$$

où $\widehat{[a_i^-, a_i^+]}$ *indique que le facteur* $[a_i^-, a_i^+]$ *est manquant dans le produit cartésien.*

On se basera sur des exemples de cubes ou de carrés pour visualiser ces notions en dimension 2 et 3. Pour le cube A on obtient :

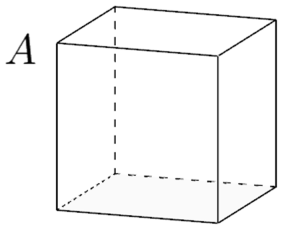


FIGURE 1 – Cube A.

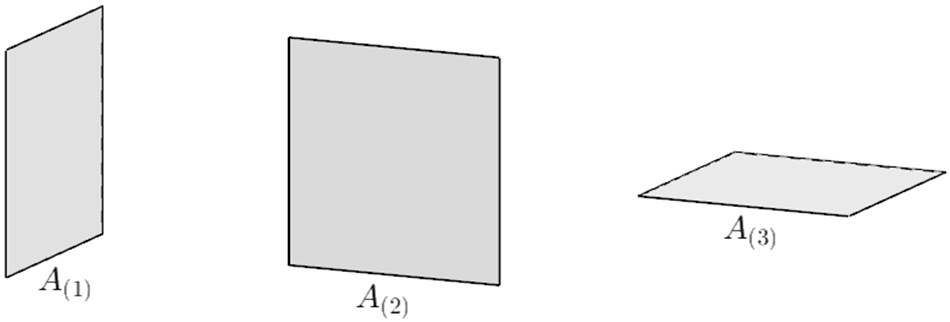


FIGURE 2 – Faces projectives de A.

On introduit de façon analogue, pour chaque $1 \leq i \leq n$*, deux faces de A :*
— *la face positivement orientée :*

$$A_i^+ := [a_1^-, a_1^+] \times \cdots \times \{a_i^+\} \times \cdots \times [a_n^-, a_n^+] \subset \mathbb{R}^n;$$

— *la face négativement orientée :*

$$A_i^- := [a_1^-, a_1^+] \times \cdots \times \{a_i^-\} \times \cdots \times [a_n^-, a_n^+] \subset \mathbb{R}^n.$$

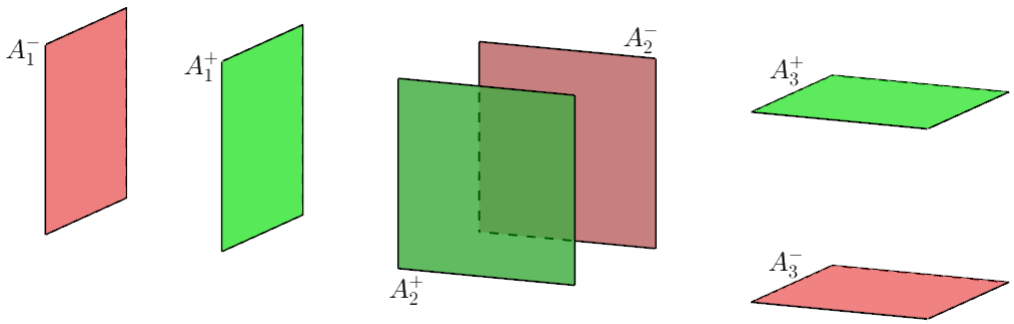


FIGURE 3 – Faces orientées du cube.

Définition 3.2. *On appelle* champ de vecteurs*, toute application* $\mathrm{v} : A \subset \mathbb{R}^n \longrightarrow \mathbb{R}^n$.

On lui associe de même deux champs de vecteurs orientés, $v_i^{\pm}$*, définis sur* $A_{(i)}$ *pour chaque* $1 \leq i \leq n$ *par :*

$$\begin{array}{rll} v_i^{\pm} : & A_{(i)} \subset \mathbb{R}^{n-1} & \longrightarrow \mathbb{R}^n \\ & (\xi_1, \ldots, \widehat{\xi_i}, \ldots, \xi_n) & \longmapsto \mathrm{v}(\xi_1, \ldots, a_i^{\pm}, \ldots, \xi_n) \end{array}$$

On définit enfin deux vecteurs normaux $n_i^{\pm}$ *de la manière suivante :*

$$n_i^{\pm} := (0, \ldots, 0, \pm 1, 0, \ldots, 0)$$

où la composante non nulle ± 1 *apparaît en* $i^{ème}$ *position.*

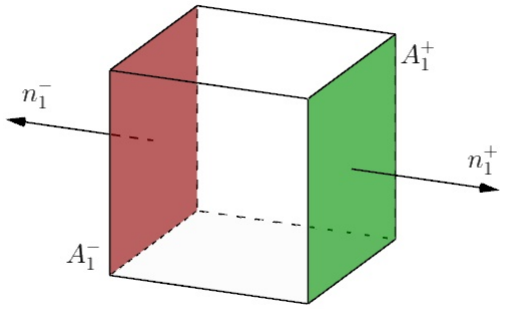


FIGURE 4 – Vecteurs normaux du cube A.

Définition 3.3. *Pour un champ de vecteurs* $\mathrm{v} : A \subset \mathbb{R}^n \to \mathbb{R}^n$ *on définit sa* divergence *par :*

$$\mathrm{div}\ \mathrm{v} = \sum_{i=1}^{n} \frac{\partial v_i}{\partial x_i}$$

Remarque 3.1. *Si on définit la* matrice Jacobienne *d'un champ de vecteurs* $\mathrm{v} : A \subset \mathbb{R}^n \to \mathbb{R}^n$ *en* $x \in \mathbb{R}^n$ *par :*

$$Jac_x \mathrm{v} = \begin{pmatrix} \dfrac{\partial v_1}{\partial x_1} & \cdots & \dfrac{\partial v_1}{\partial x_n} \\ \vdots & & \vdots \\ \dfrac{\partial v_n}{\partial x_1} & \cdots & \dfrac{\partial v_n}{\partial x_n} \end{pmatrix}$$

Alors

$$\mathrm{div}\ \mathrm{v} = \mathrm{Tr}(Jac_x \mathrm{v})$$

3.2 Conjecture et tentative de démonstration avec la théorie de Kurzweil et Henstock

Il s'agit ici de tenter d'obtenir la Formule de Gauss-Green en théorie de Kurzweil et Henstock.

Conjecture 3.1. *Soit A un pavé de $\mathbb{R}^n$ et soit $\mathrm{v} : A \subset \mathbb{R}^n \to \mathbb{R}^n$ un champ de vecteurs différentiable sur A, alors on a :*

$$\int_A \operatorname{div} \mathrm{v} = \int_{\partial A} \mathrm{v} \cdot n \tag{1}$$

où l'intégrale de la divergence de A se fait au sens de Kurzweil et Henstock, et où l'intégrale de droite représente le flux de v à travers le bord de A et est définie ci-dessous.

Définition 3.4 (Flux d'un champ de vecteurs)**.** *Soient $A \subset \mathbb{R}^n$ un pavé de dimension n et $\mathrm{v} : A \subset \mathbb{R}^n \to \mathbb{R}^n$ un champ de vecteurs sur A. Le* flux *de v à travers le bord de A est la valeur de l'application* Fv*, définie par :*

$$\begin{array}{rlll} \mathrm{Fv}: & \mathcal{P}(\mathbb{R}^n) & \longrightarrow \mathbb{R}^n & \\ & A & \longmapsto \int_{\partial A} \mathrm{v} \cdot n := \sum_{i=1}^{n} \left[\int_{A_i^+} \mathrm{v} \cdot n_i^+ + \int_{A_i^-} \mathrm{v} \cdot n_i^- \right] \end{array}$$

où l'on a posé, pour chaque $1 \leq i \leq n$:

$$\int_{A_i^\pm} \mathrm{v} \cdot n_i^\pm := \pm \int_{A_{(i)}} v_i^\pm$$

et où l'intégrale de droite, qui est une intégrale de Riemann classique à $n-1$ variables, s'interprète comme le flux de v à travers la face $A_i^\pm$.

Exemple 3.1. En dimension 2, on a ainsi :

$$\begin{aligned} \int_{\partial A} \mathrm{v} \cdot n &= \int_{A_1^+} \mathrm{v} \cdot n_1^+ + \int_{A_1^-} \mathrm{v} \cdot n_1^- + \int_{A_2^+} \mathrm{v} \cdot n_2^+ + \int_{A_2^-} \mathrm{v} \cdot n_2^-, \\ &= \int_{A_{(1)}} v_1^+ - \int_{A_{(1)}} v_1^- + \int_{A_{(2)}} v_2^+ - \int_{A_{(2)}} v_2^-, \\ &= \int_{A_{(1)}} (v_1^+ - v_1^-) + \int_{A_{(2)}} (v_2^+ - v_2^-). \end{aligned}$$

Ce que l'on observe parfaitement sur cette figure :

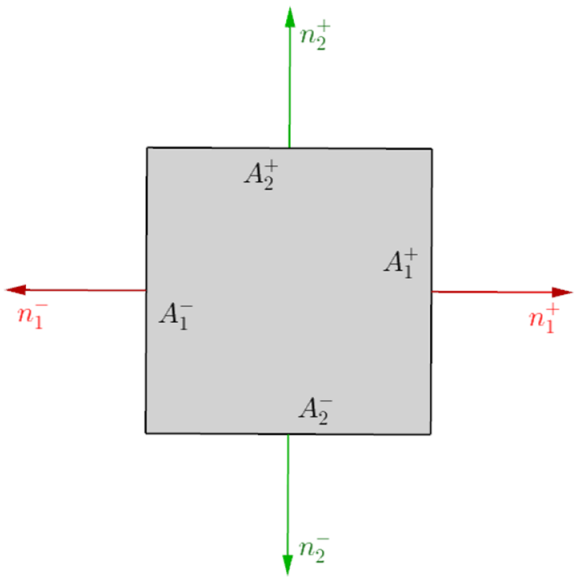


FIGURE 5 – Observation du flux passant au travers d'un pavé A de dimension 2.

Définissons la notion de *périmètre* qui nous sera utile par la suite.

Définition 3.5. *Soit* $A = \prod_{i=1}^{n} [a_i^-, a_i^+]$ *un pavé de* $\mathbb{R}^n$. *On définit le* périmètre *de A, noté* $\|A\|$, *par la formule suivante :*

$$\|A\| := 2\sum_{i=1}^{n} \mu[A_{(i)}].$$

On rappelle que l'on peut définir le *diamètre* de A, noté $d(A)$, de la façon suivante :

$$d(A) := \sup_{(x,y)\in A^2} |x-y|_\infty.$$

Définition 3.6. *Soit* $I \subset \mathbb{R}$ *un intervalle. On définit la* longueur de I *par*

$$l(I) = \sup(I) - \inf(I).$$

Proposition 3.2. *Soit* $\mathrm{v} : \mathbb{R}^n \to \mathbb{R}^n$ *un champ de vecteurs continu. Alors :*

1. $\mathrm{v} \longmapsto \int_{\partial A} \mathrm{v}\cdot n$ *est une fonctionnelle linéaire sur les champs de vecteurs continus ;*
2. $\int_{\partial A} \mathrm{v}\cdot n \leq \|A\| \sup_{x\in\partial A} |\mathrm{v}(x)|_\infty$.

Démonstration. 1. Immédiat.

2. Par définition de v, on a :

$$\begin{aligned}
\int_{\partial A} \mathrm{v}\cdot n &= \sum_{i=1}^{n} \int_{A_{(i)}} (v_i^+ - v_i^-), \\
&\leq 2\sum_{i=1}^{n} \int_{A_{(i)}} \sup_{x\in\partial A} |\mathrm{v}(x)|_\infty, \\
&= 2\sum_{i=1}^{n} \mu[A_{(i)}] \sup_{x\in\partial A} |\mathrm{v}(x)|_\infty, \\
&\leq \|A\| \sup_{x\in\partial A} |\mathrm{v}(x)|_\infty.
\end{aligned}$$

□

3 lemmes nous seront utiles pour établir une formule de Gauss-Green :

Lemme 3.3 (Additivité du flux d'un champ de vecteurs)**.** *Soit* $\mathrm{v} : A \subset \mathbb{R}^n \to \mathbb{R}^n$ *un champ de vecteurs continu. Alors le flux de* v *est une fonction additive,* i.e. *on a :*

$$\mathrm{Fv}(A) = \sum_{C\in P} \mathrm{Fv}(C),$$

pour toute partition P *de* A *par des pavés.*

Démonstration. Par additivité. □

La formule (1) est classique pour des champs de vecteurs affines.

Lemme 3.4 (Formule de Gauss-Green pour un champ de vecteurs affine)**.** *Soit* $\mathrm{w} : A \subset \mathbb{R}^n \to \mathbb{R}^n$ *un champ de vecteurs affine (*i.e. *de la forme* $\mathrm{w}(x) = M\cdot x + b$ *où* $M \in M_n(\mathbb{R})$ *est une matrice carrée d'ordre* n *et* $b \in \mathbb{R}^n$ *un vecteur). Alors on a :*

$$\int_A \operatorname{div} \mathrm{w} = \int_{\partial A} \mathrm{w}\cdot n.$$

Démonstration. Posons

$$\mathrm{w}(x) = M\cdot x + b = \begin{pmatrix} m_{1,1} & \cdots & m_{1,n} \\ \vdots & & \vdots \\ m_{n,1} & \cdots & m_{n,n} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} + \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} x_i m_{1,i} + b_1 \\ \vdots \\ \sum_{i=1}^{n} x_i m_{n,i} + b_n \end{pmatrix}.$$

On remarque que

$$Jac_x \mathrm{w} = \begin{pmatrix} \dfrac{\partial w_1}{\partial x_1} & \cdots & \dfrac{\partial w_1}{\partial x_n} \\ \vdots & & \vdots \\ \dfrac{\partial w_n}{\partial x_1} & \cdots & \dfrac{\partial w_n}{\partial x_n} \end{pmatrix} = \begin{pmatrix} m_{1,1} & \cdots & m_{1,n} \\ \vdots & & \vdots \\ m_{n,1} & \cdots & m_{n,n} \end{pmatrix} = M$$

et donc $\operatorname{div} \mathrm{w} = \operatorname{Tr}(Jac_x \mathrm{w}) = \operatorname{Tr}(M)$. L'intégrale de la divergence devient alors :

$$\int_A \operatorname{div} \mathrm{w} = \int_A \operatorname{Tr}(M) = \operatorname{Tr}(M)\mu(A).$$

Calculons le flux de w à travers le bord de A :

$$\begin{aligned}
\int_{\partial A} \mathrm{w} \cdot n &:= \sum_{i=1}^{n} \left[\int_{A_i^+} \mathrm{w} \cdot n_i^+ + \int_{A_i^-} \mathrm{w} \cdot n_i^- \right], \\
&= \sum_{i=1}^{n} \left[\int_{A_i^+} {}^t\mathrm{w} \cdot \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \int_{A_i^-} {}^t\mathrm{w} \cdot \begin{pmatrix} 0 \\ \vdots \\ 0 \\ -1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \right], \\
&\text{par définition de } w_i^{\pm} \text{ appliquée en } (\xi_1, \ldots, \widehat{\xi_i}, \ldots, \xi_n) \in A_{(i)} \text{ on a :} \\
\int_{\partial A} \mathrm{w} \cdot n &= \sum_{i=1}^{n} \int_{A_{(i)}} \left[\sum_{\substack{j=1 \\ j \neq i}}^{n} \xi_j m_{i,j} + a_i^+ m_{i,i} + b_i - \left(\sum_{\substack{j=1 \\ j \neq i}}^{n} \xi_j m_{i,j} + a_i^- m_{i,i} + b_i \right) \right], \\
&= \sum_{i=1}^{n} \int_{A_{(i)}} m_{i,i} \left[a_i^+ - a_i^- \right], \\
&= \sum_{i=1}^{n} m_{i,i} \left[a_i^+ - a_i^- \right] \mu(A_{(i)}), \\
&= \sum_{i=1}^{n} m_{i,i} \mu(A), \\
&= \operatorname{Tr}(M)\mu(A).
\end{aligned}$$

Ce qui conclut la démonstration. □

Enfin, le dernier Lemme qui nous sera utile, est un résultat sur les champs de vecteurs différentiables.

Lemme 3.5. *Soit* $\mathrm{v} : A \subset \mathbb{R}^n \to \mathbb{R}^n$ *un champ de vecteurs différentiable en tout point de* A*. Alors pour tout* $\varepsilon > 0$ *et tout* $x \in A$*, il existe* $\delta > 0$ *tel que l'on ait :*

$$\left| \operatorname{div} \mathrm{v}(x)\mu(B) - \int_{\partial B} \mathrm{v} \cdot n \right| \leq \varepsilon d(B) \|B\|,$$

pour tout pavé B *de* $\mathbb{R}^n$ *vérifiant* $B \subseteq A \cap \mathcal{B}_\infty(x, \delta)$*.*

Démonstration. Fixons $\delta > 0$ et $x \in A \subset \mathbb{R}^n$; on souhaite construire un champ de vecteurs affine ayant la même divergence que v pour pouvoir utiliser le Lemme 3.4. On pose :

$$\begin{array}{rccl} \mathrm{w} : & \mathbb{R}^n & \longrightarrow & \mathbb{R}^n \\ & y & \longmapsto & \mathrm{w}(y) := \mathrm{v}(x) + \mathrm{v}'_x(y - x) \end{array}$$

où v'_x désigne la différentielle (ou dérivée totale) de v au point x (i.e. $\mathrm{v}'_x(y-x)$ correspond au produit matriciel entre $\mathrm{Jac}_{\mathrm{x}}\, \mathrm{v}(x)$ et le vecteur colonne $(y-x)$). Ainsi construit, w est un champ de vecteurs affine.

Montrons que pour tout $y \in \mathbb{R}^n$, on a $\operatorname{div} \mathrm{w}(y) = \operatorname{div} \mathrm{v}(x)$.
Soit $y \in \mathbb{R}^n$, on calcule :

$$\begin{aligned}
\frac{\partial \mathrm{w}_i}{\partial y_i}(y) &= \frac{\partial (\mathrm{v}'_x)^i}{\partial y_i}(y), \\
&= \frac{\partial Jac^i_x\, \mathrm{v}(x)}{\partial y_i}(y-x), \text{ où } Jac^i_x\, \mathrm{v}(x) \text{ est la } i^{\text{ème}} \text{ ligne de la matrice Jacobienne de v en } x \\
&= \begin{pmatrix} \frac{\partial \mathrm{v}_i}{\partial x_1}(x) & \cdots & \frac{\partial \mathrm{v}_i}{\partial x_n}(x) \end{pmatrix} \cdot \frac{\partial}{\partial y_i} \begin{pmatrix} y_1 - x_1 \\ \vdots \\ y_n - x_n \end{pmatrix}, \\
&= \begin{pmatrix} \frac{\partial \mathrm{v}_i}{\partial x_1}(x) & \cdots & \frac{\partial \mathrm{v}_i}{\partial x_n}(x) \end{pmatrix} \cdot \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \\
&= \frac{\partial \mathrm{v}_i}{\partial x_i}(x).
\end{aligned}$$

Et on a ainsi :

$$\operatorname{div} \mathrm{w}(y) = \sum_{i=1}^{n} \frac{\partial \mathrm{w}_i}{\partial y_i}(y) = \sum_{i=1}^{n} \frac{\partial \mathrm{v}_i}{\partial x_i}(x) = \operatorname{div} \mathrm{v}(x).$$

De plus, v est différentiable en x. Il existe donc $\delta > 0$ tel que pour tout $y \in A \cap \mathcal{B}_\infty(x, \delta)$ on ait :

$$\mathrm{v}(y) = \mathrm{v}(x) + \mathrm{v}'_x(y-x) + o(y-x)$$

On trouve donc si $x \in B \subseteq A \cap \mathcal{B}_\infty(x, \delta)$ et pour tout $y \in B$:

$$|\mathrm{v}(y) - \mathrm{w}(y)|_\infty = |o(y-x)|_\infty \leq \varepsilon\, |y-x|_\infty \leq \varepsilon \sup_{(x,y)\in B^2} |y-x|_\infty = \varepsilon d(B)$$

D'après le Lemme 3.4, on obtient finalement les inégalités suivantes :

$$\begin{aligned}
\left| \operatorname{div} \mathrm{v}(x)\mu(B) - \int_{\partial B} \mathrm{v} \cdot n \right| &= \left| \operatorname{div} \mathrm{w}(y)\mu(B) - \int_{\partial B} \mathrm{v} \cdot n \right|, \\
&= \left| \int_B \operatorname{div} \mathrm{w} - \int_{\partial B} \mathrm{v} \cdot n \right|, \\
&\text{comme w est affine on applique le Lemme 3.4 :} \\
&= \left| \int_{\partial B} \mathrm{w} \cdot n - \int_{\partial B} \mathrm{v} \cdot n \right|, \\
&= \left| \int_{\partial B} (\mathrm{w} - \mathrm{v}) \cdot n \right|, \\
&\leq \int_{\partial B} |\mathrm{w} - \mathrm{v}|_\infty \cdot |n|, \\
&\leq \varepsilon d(B) \|B\|,
\end{aligned}$$

ce qui achève la démonstration de ce lemme. □

Tâchons à présent de démontrer la Conjecture énoncée ci-dessus.

Tentative de démonstration de la Conjecture 3.1. Comme $\mathrm{v} : A \subset \mathbb{R}^n \to \mathbb{R}^n$ est un champ de vecteurs différentiable alors il est continu, donc le flux Fv est bien défini et par le Lemme 3.3, il est également additif.

Soit $\varepsilon > 0$; il nous faut montrer l'existence d'une jauge δ sur A telle que pour toute P-partition δ-fine Π de A, on ait :

$$\left| \int_A \operatorname{div} \mathrm{v} - \int_{\partial A} \mathrm{v} \cdot n \right| = |S(A, \operatorname{div} \mathrm{v}, \Pi) - \mathrm{Fv}(A)| \leq \varepsilon.$$

Fixons $x \in A$. Comme v est différentiable en tout point de A, alors d'après le Lemme 3.5, il existe une jauge $\delta(x) > 0$ tel que pour tout pavé $B \subseteq A \cap \mathcal{B}_\infty(x, \delta(x))$, on ait :

$$|\operatorname{div} \mathrm{v}(x)\mu(B) - \mathrm{Fv}(B)| \leq \varepsilon d(B)\|B\|. \tag{2}$$

Ceci définit ainsi une jauge δ sur A. Fixons une P-partition δ-fine Π de A $\Pi := \{(x^j, A^j) : 1 \leq j \leq m\}$ (existence assurée par le Lemme de Cousin 2.3). On calcule alors :

$$\begin{aligned}
|S(A, \operatorname{div} \mathrm{v}, \Pi) - \mathrm{Fv}(A)| &= \left| \sum_{j=1}^{m} \left[\operatorname{div} \mathrm{v}(x^j)\mu(A^j) - \mathrm{Fv}(A^j) \right] \right|, \\
&\leq \sum_{j=1}^{m} \left| \operatorname{div} \mathrm{v}(x^j)\mu(A^j) - \mathrm{Fv}(A^j) \right|, \text{ puis par l'inégalité (2) :} \\
&\leq \varepsilon \sum_{j=1}^{m} d(A^j)\|A^j\|.
\end{aligned}$$

Nous pourrions achever la démonstration s'il existait un moyen de montrer que la somme

$$\sum_{j=1}^{m} d(A^j)\|A^j\|, \tag{3}$$

peut-être bornée par une quantité indépendante de la P-partition choisie. Or ce n'est pas toujours le cas, nous allons le voir dans un contre-exemple ; il convient d'interrompre ici notre tentative de démonstration. □

Exemple 3.2 (Contre-exemple à la majoration uniforme de (3))**.** Soit $\{A^j : 1 \leq j \leq m\}$ la partition de $[0,1] \times [0,1]$ définie pour tout $1 \leq j \leq m$ par

$$A^j := [0,1] \times \left[\frac{j-1}{m}, \frac{j}{m} \right],$$

que l'on observe sur cette figure :

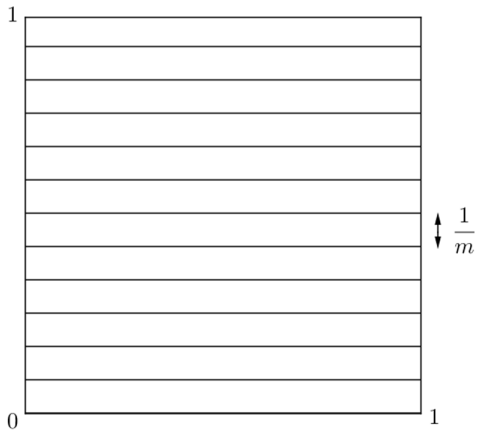


FIGURE 6 – Partition très aplatie d'un carré A.

On a alors pour tout $1 \leq j \leq m$: $d(A^j) = 1$. Fixons un entier $1 \leq j \leq m$ et calculons :

$$\|A^j\| = 2\sum_{i=1}^{2} \mu(A^j_{(i)}) = 2\left(1 + \frac{1}{m}\right) \quad \text{et} \quad \sum_{j=1}^{m} d(A^j)\|A^j\| = 2m + 2$$

Cette dépendance en m montre donc qu'il n'est pas toujours possible de contrôler le terme $\sum_{j=1}^{m} d(A^j)\|A^j\|$ en raison de cet aplatissement trop important des A^j, $1 \leq j \leq m$.

Avec ce contre-exemple on remarque qu'il n'est pas possible de conclure, en suivant l'argument esquissé précédemment, la preuve de la Conjecture 3.1.

Pour remédier à ce problème, il suffit d'intégrer *dans la définition d'intégrale* une condition garantissant qu'il soit possible de contrôler, pour une partition admissible, la quantité (3).

Cette idée, qui a été le point clef du développement ultérieur des théories non-absolues de l'intégrale en dimension $n > 1$, est due à Jean Mawhin.

3.3 Intégrale de Mawhin et démonstration de la formule de Gauss-Green

Pour développer la définition de l'intégrale de Mawhin, introduisons une terminologie qui en simplifiera l'énoncé.

Définition 3.7. *Soit A un pavé de $\mathbb{R}^n$; on définit l'*aplatissement *de A, noté $\sigma(A)$, de la façon suivante :*

$$\sigma(A) := \frac{\max\limits_{1\leq i\leq n}(a_i^+ - a_i^-)}{\min\limits_{1\leq i\leq n}(a_i^+ - a_i^-)}.$$

Remarque 3.2. *L'idée de Jean Mawhin, afin de contrôler la somme (3) dans le cadre de P-partition admissibles, est d'imposer un contrôle sur l'aplatissement des rectangles qui la constituent. Il introduit pour ce faire le concept d'*irrégularité *d'une P-partition d'un pavé A de $\mathbb{R}^n$*

Définition 3.8. *Soit $\Xi = \{(x^j, A^j) : 1 \leq j \leq m\}$ une P-partition d'un pavé A de $\mathbb{R}^n$. On définit l'*irrégularité *de Ξ, notée $\Sigma(\Xi)$, comme suit :*

$$\Sigma(\Xi) := \frac{\max\limits_{1\leq j\leq m}\sigma(A^j)}{\sigma(A)}.$$

Définition 3.9. *Soit A un pavé de $\mathbb{R}^n$. Soit $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ une fonction définie sur A. On dit que f est* intégrable au sens de Mawhin *(ou* M-intégrable*) sur A, d'intégrale $\alpha \in \mathbb{R}$, si, pour tout $\varepsilon > 0$ et pour tout $\eta \geq 1$, il existe une jauge $\delta_{\varepsilon,\eta}$ sur A telle que pour toute P-partition $\delta_{\varepsilon,\eta}$-fine Ξ de A vérifiant $\Sigma(\Xi) \leq \eta$, on ait :*

$$|S(A, f, \Xi) - \alpha| \leq \varepsilon. \tag{4}$$

*Dans ce cas on appelle l'unique réel α vérifiant la propriété précédente, l'*intégrale de f au sens de Mawhin sur A*, et on le note (M)$\int_A f$ ou simplement $\int_A f$ lorsqu'aucune confusion n'est à craindre.*

Nous allons maintenant observer quelques partitions possibles d'un pavé A de dimension 2 pour se familiariser avec les définitions, puis nous répertorierons les règles de *partitionnement* satisfaisant aux définitions pour un pavé A de dimension 2.

Exemple 3.3. On se place en dimension 2.

1. Reprenons la partition du contre-exemple 3.2 précédemment développé. On a alors $\sigma(A) = 1$ et pour tout $1 \leq j \leq m$, $\sigma(A^j) = \dfrac{1}{1/m} = m$. Donc l'irrégularité de Ξ est $\Sigma(\Xi) = \dfrac{m}{1} = m \geq 1$, ce qui signifie qu'à moins que $m = 1$ cette partition est très irrégulière au sens de Mawhin et ne permet pas de satisfaire à la définition d'intégrabilité, peu importe la fonction.
2. A contrario, la figure ci-dessous est une partition régulière :

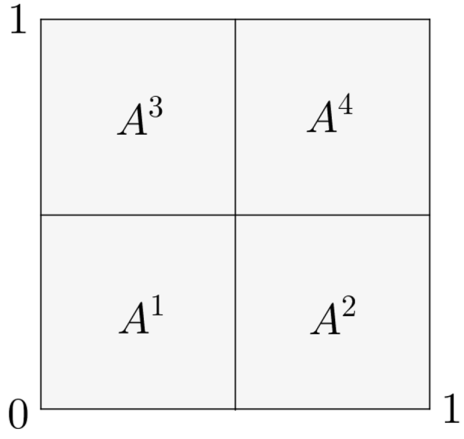


FIGURE 7 – Partition régulière d'un carré A.

En effet, on observe que $\sigma(A) = 1$, que pour tout $1 \leq j \leq 4$, $\sigma(A^j) = \dfrac{1/2}{1/2} = 1$ et donc l'irrégularité de Ξ est $\Sigma(\Xi) = \dfrac{1}{1} = 1 \leq \eta$. Ainsi cette partition est parfaite pour être parmi les partitions admissibles dans la définition de Mawhin.

Proposition 3.6. *En dimension 2 et avec η petit (i.e. très proche de 1), partant d'un pavé que l'on considérera ici comme un rectangle de largeur l et de longueur L, deux règles doivent absolument être vérifiées pour que la partition vérifie la définition d'intégrabilité au sens de Mawhin :*

- *La découpe doit toujours s'effectuer sur le côté le plus long.*
- *La mesure du côté du pavé créé doit être supérieur à $\frac{l^2}{L}$. (cf. figure ci-dessous)*

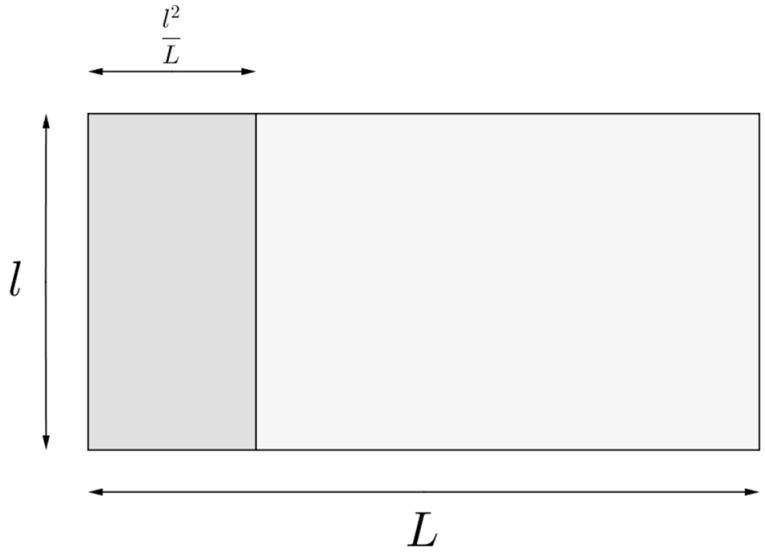


FIGURE 8 – Règles de découpe sur un pavé A en dimension 2.

Démonstration. Le premier point est assez évident, on pourra se convaincre en réalisant un dessin ou en revenant au contre-exemple avec $m = 2$ par exemple.

Pour le second point, soit A un pavé vérifiant les données de l'énoncé, on a alors $\sigma(A) = \frac{L}{l}$. Posons $x = \max_{1\leq j\leq m} \sigma(A^j)$. Pour satisfaire à la définition d'intégrabilité au sens de Mawhin, x doit vérifier l'inégalité :

$$\Sigma(\Xi) = \frac{x}{\sigma(A)} = \frac{x}{\frac{L}{l}} \leq 1,$$

i.e. $x \leq \frac{L}{l}$. Posons y la mesure du côté du pavé créé, y doit vérifier :

$$\max_{1\leq j\leq m} \sigma(A^j) = \frac{l}{y} := x \leq \frac{L}{l},$$

i.e. $y \geq \frac{l^2}{L}$, ce qui conclut la démonstration. □

Remarque 3.3. *Il résulte de la définition précédente que si f est intégrable au sens de Kurzweil et Henstock sur A, alors f est intégrable au sens de Mawhin sur A.*

Remarque 3.4. *L'intégrale de Mawhin qui impose une régularité aux P-partitions va nous permettre d'obtenir une majoration de la somme (3) posant problème dans la démonstration de la Conjecture 3.1.*

Plus précisément, on obtient dans cette nouvelle théorie le résultat suivant.

Théorème 3.7. *Si $\mathrm{v} : A \subset \mathbb{R}^n \to \mathbb{R}^n$ est un champ de vecteurs différentiable sur A, alors* div v *est intégrable au sens de Mawhin sur A et on a :*

$$(M)\int_A \operatorname{div} \mathrm{v} = \int_{\partial A} \mathrm{v}\cdot n. \tag{5}$$

Démonstration. Fixons $\varepsilon' > 0$, $\eta \geq 1$ et posons

$$\varepsilon := \frac{\varepsilon'}{2n\eta^{n-1}\sigma(A)^{n-1}\mu(A)}.$$

On associe à $\varepsilon > 0$ une jauge δ sur A exactement comme dans la tentative de preuve de la Conjecture 3.1, puis fixons une P-partition δ-fine $\Xi := \{(x^j, A^j) : 1 \leq j \leq m\}$ (existence assurée par le Lemme de Cousin 2.3) et supposons que l'on ait $\Sigma(\Xi) \leq \eta$.

On obtient, en procédant exactement comme précédemment :

$$|S(A, \text{div v}, \Xi) - \text{Fv}(A)| \leq \varepsilon \sum_{j=1}^{m} d(A^j)\|A^j\|.$$

On calcule alors, en notant, pour chaque $1 \leq i \leq n$ et chaque $1 \leq j \leq m$, A_i^j la projection sur le $i^{\text{ème}}$ axe du pavé A^j :

$$\begin{aligned}
\sum_{j=1}^{m} d(A^j)\|A^j\| &= \sum_{j=1}^{m} d(A^j) \min_{1\leq i\leq n} l(A_i^j)^{n-1} \frac{\|A^j\|}{\min\limits_{1\leq i\leq n} l(A_i^j)^{n-1}}, \\
&\leq \sum_{j=1}^{m} \mu(A^j) \frac{\|A^j\|}{\min\limits_{1\leq i\leq n} l(A_i^j)^{n-1}}, \\
&= \sum_{j=1}^{m} \mu(A^j) \frac{2\sum\limits_{i=1}^{n} \mu(A_{(i)}^j)}{\min\limits_{1\leq i\leq n} l(A_i^j)^{n-1}}, \\
&\leq 2n \sum_{j=1}^{m} \mu(A^j) \frac{\max\limits_{1\leq i\leq n} l(A_i^j)^{n-1}}{\min\limits_{1\leq i\leq n} l(A_i^j)^{n-1}}, \\
&= 2n \sum_{j=1}^{m} \mu(A^j) \sigma(A^j)^{n-1}.
\end{aligned}$$

Or il vient pour chaque $1 \leq j \leq m$:

$$\sigma(A^j) \leq \max_{1\leq j\leq m} \sigma(A^j) := \Sigma(\Xi)\sigma(A),$$

on obtient donc :

$$\sigma(A^j)^{n-1} \leq (\Sigma(\Xi)\sigma(A))^{n-1}.$$

Avec ces inégalités, on peut fournir la majoration recherchée :

$$\begin{aligned}
|S(A, \text{div v}, \Xi) - \text{Fv}(A)| &\leq \varepsilon \sum_{j=1}^{m} d(A^j)\|A^j\|, \\
&\leq 2n\varepsilon \sum_{j=1}^{m} \mu(A^j)(\Sigma(\Xi)\sigma(A))^{n-1}, \\
&= 2n\varepsilon \Sigma(\Xi)^{n-1}\sigma(A)^{n-1}\mu(A), \\
&\leq \left(2n\eta^{n-1}\sigma(A)^{n-1}\mu(A)\right)\varepsilon := \varepsilon'.
\end{aligned}$$

Ceci étant vrai pour tout ε', le théorème est démontré. □

4 Théorème de Fubini pour l'intégrale de Kurzweil et Henstock.

4.1 Contexte.

On va chercher ici à permuter deux intégrales. Dans ce chapitre, on se place sur $\mathbb{R}^n$ et on dira simplement qu'une fonction est intégrable lorsqu'elle est intégrable au sens de Kurzweil et Henstock.

On va commencer par montrer le théorème d'interversion des intégrales de Fubini pour les semi-pavés de $\mathbb{R}^n$. Soit $I = J \times K$ un semi-pavé de $\mathbb{R}^n$, avec $J \subset \mathbb{R}^q$ et $K \subset \mathbb{R}^s$ des semi-pavés et avec $n = q + s$.

Définition 4.1. *Soit $N \subset \mathbb{R}^n$. N est dite* n-négligeable *ou* de n-mesure nulle *lorsque $\forall \varepsilon > 0$, il existe une suite $\left(A^j\right)_{j\in\mathbb{N}}$ de pavés de $\mathbb{R}^n$ telle que*

$$N \subset \bigcup_{j\in\mathbb{N}} A^j \text{ et } \sum_{j\in\mathbb{N}} \mu\left(A^j\right) \leq \varepsilon.$$

Proposition 4.1. *Soit I un semi-pavé de $\mathbb{R}^n$ et soit $f : \mathbb{R}^n \longrightarrow \mathbb{R}^p$ une fonction définie sur $\overline{I}$. Si $\forall \varepsilon > 0$, il existe une jauge δ sur $\overline{I}$ telle que, pour toute P-partition δ-fine Π de I et toute P-partition δ-fine $\widetilde{\Pi}$ de I, on a*

$$\left|S(I, f, \Pi) - S(I, f, \widetilde{\Pi})\right| \leq \varepsilon,$$

alors f est intégrable sur $\overline{I}$.

Définition 4.2. *Soit I un semi-pavé de $\mathbb{R}^n$ et soit $f : \mathbb{R}^n \longrightarrow \mathbb{R}^p$ une fonction définie sur $\overline{I}$. On appelle* Condition de Cauchy *la condition suffisante d'intégrabilité de f sur $\overline{I}$ de la Proposition précédente (4.1) : "$\forall \varepsilon > 0$, il existe une jauge δ sur $\overline{I}$ telle que, pour toute P-partition δ-fine Π de I et toute P-partition δ-fine $\widetilde{\Pi}$ de I, $\left|S(I, f, \Pi) - S(I, f, \widetilde{\Pi})\right| \leq \varepsilon$."*

Démonstration de la Proposition 4.1. Construisons tout d'abord un candidat pour la valeur de l'intégrale de f sur $\overline{I}$. En prenant $\varepsilon = 1$ dans la condition de Cauchy, on peut trouver une jauge δ_1 sur $\overline{I}$ telle que, pour toute P-partition δ_1-fine Π de I et toute P-partition δ_1-fine $\widetilde{\Pi}$ de I,

$$\left|S(I, f, \Pi) - S(I, f, \widetilde{\Pi})\right| \leq 1.$$

De même, pour $\varepsilon = 1/2$, il existe une jauge δ_2 sur $\overline{I}$; que l'on peut choisir telle que, pour tout $x \in \overline{I}$, $\delta_2(x) \leq \delta_1(x)$, pour laquelle, pour toute P-partitions δ_2-fine Π et $\widetilde{\Pi}$ de I,

$$\left|S(I, f, \Pi) - S(I, f, \widetilde{\Pi})\right| \leq 1/2.$$

On peut ainsi itérer le procédé, pour tout $k \in \mathbb{N}^*$, avec $\varepsilon = 1/k$. On obtient une suite $(\delta_k)_{k\in\mathbb{N}^*}$ de jauges sur $\overline{I}$ telle que pour tout $k \in \mathbb{N}^*$, δ_k est une jauge sur $\overline{I}$ telle que, pour toute P-partition δ_k-fine Π de I et toute P-partition δ_k-fine $\widetilde{\Pi}$ de I,

$$\left|S(I, f, \Pi) - S(I, f, \widetilde{\Pi})\right| \leq 1/k, \tag{6}$$

et pour tout $x \in \overline{I}$,

$$\delta_{k+1}(x) \leq \delta_k(x).$$

Fixons, pour chaque $k \in \mathbb{N}^*$ une P-partition δ_k-fine Π_k de I et montrons que la suite $(S(I, f, \Pi_k))_{k\in\mathbb{N}^*}$ est de Cauchy dans $\mathbb{R}^p$.

Comme pour tout $x \in I$, la suite $(\delta_k(x))_{k\in\mathbb{N}^*}$ est décroissante, si $k \leq p$, alors toute P-partition δ_p-fine est aussi δ_k-fine. Soit $(k, p) \in (\mathbb{N}^*)^2$ tel que $1 \leq k \leq p$. Alors Π_k et Π_p sont des P-partitions δ_k-fines et donc, par construction,

$$|S(I, f, \Pi_k) - S(I, f, \Pi_p)| \leq 1/k. \tag{7}$$

Soit $\varepsilon > 0$. Soit $m \in \mathbb{N}^*$ tel que $\frac{1}{m} \leq \varepsilon$. Alors, pour tout $p \geq k \geq m$,

$$|S(I, f, \Pi_k) - S(I, f, \Pi_p)| \leq \frac{1}{k} \leq \frac{1}{m} \leq \varepsilon.$$

Donc $(S(I, f, \Pi_k))_{k\in\mathbb{N}^*}$ est une suite de Cauchy dans $\mathbb{R}^p$ et donc converge (dans $\mathbb{R}^p$). On note J sa limite. On a donc, en faisant tendre p vers l'infini dans (7), que

$$\forall k \in \mathbb{N}^*, |S(I, f, \Pi_k) - J| \leq \frac{1}{k}.$$

Si Π est une P-partition δ_m-fine de I, alors, par (6), pour tout $p \geq m$,

$$|S(I, f, \Pi) - S(I, f, \Pi_p)| \leq \frac{1}{m} \leq \varepsilon.$$

Dès lors, si l'on fait tendre p vers l'infini, on obtient

$$|S(I, f, \Pi) - J| \leq \varepsilon.$$

On a ainsi montré que $\forall \varepsilon > 0$, il existe $m \in \mathbb{N}^*$ et une jauge δ_m sur $\overline{I}$ telle que, pour toute P-partition δ_m-fine Π de I, $|S(I, f, \Pi) - J| \leq \varepsilon$.

C'est la définition d'intégrabilité de f sur $\overline{I}$. □

La condition de Cauchy nous permettra de prouver l'intégrabilité de fonctions sans connaître la valeur de leur intégrale.

4.2 Théorème de Fubini.

On rappelle que l'on se place sur $\mathbb{R}^n = \mathbb{R}^q \times \mathbb{R}^s$ et que $I = J \times K$ sont des semi-pavés respectivement de $\mathbb{R}^n$, $\mathbb{R}^q$ et $\mathbb{R}^s$.

Lemme 4.2. *Soit f une fonction intégrable sur $\overline{I}$. Posons*

$$T = \{y \in \overline{J} : f(y,\cdot) \text{ n'est pas intégrable sur } \overline{K}\}.$$

Alors T est de q-mesure nulle et donc

$$F : y \longmapsto \int_{\overline{K}} f(y,z)\mathrm{d}z$$

est définie presque partout sur $\overline{J}$.

Démonstration.

$$T = \{y \in \overline{J} : f(y,\cdot) \text{ n'est pas intégrable sur } \overline{K}\}.$$

On va réécrire l'ensemble T. Par la condition de Cauchy pour $f(y,\cdot)$, $f(y,\cdot)$ n'est pas intégrable sur $\overline{K}$ $\iff$ il existe $\varepsilon > 0$ tel que pour toute jauge δ_K sur $\overline{K}$, il existe deux P-partitions δ_K-fines notées Π_K et $\widetilde{\Pi}_K$ de K telles que $S(K, f(y,\cdot), \Pi_K) - S(K, f(y,\cdot), \widetilde{\Pi}_K) > \varepsilon$. Donc
$T = \{y \in \overline{J}$ tel qu'il existe $\varepsilon > 0$ tel que, pour toute jauge δ_K sur $\overline{K}$, il existe deux P-partitions δ_K-fines Π_K et $\widetilde{\Pi}_K$ de K telles que $S(K, f(y,\cdot), \Pi_K) - S(K, f(y,\cdot), \widetilde{\Pi}_K) > \varepsilon\}$.

On va construire une union croissante d'ensembles inclus dans T de q-mesure nulle. Pour tout $i \in \mathbb{N}^*$, posons
$T_i := \{y \in \overline{J}$ tel que pour toute jauge δ_K sur $\overline{K}$, il existe deux P-partitions δ_K-fines Π_K et $\widetilde{\Pi}_K$ de K telles que $S(K, f(y,\cdot), \Pi_K) - S(K, f(y,\cdot), \widetilde{\Pi}_K) > \frac{1}{i}\}$.

Alors $\forall i \in \mathbb{N}^*,\ T_i \subset T_{i+1},\ et\ T = \bigcup_{i\in\mathbb{N}^*} T_i$. Il suffit donc de montrer que pour tout $i \in \mathbb{N}^*$, T_i est de q-mesure nulle.

Soit $i \in \mathbb{N}^*$ et soit $\varepsilon > 0$. Montrons qu'il existe une jauge δ_J sur $\overline{J}$ telle que, pour toute P-partition δ_J-fine, notée Π_J, de J, $S(J, 1_{T_i}, \Pi_J) \leq \varepsilon$. Par la condition de Cauchy appliquée à ε/i pour l'intégrabilité de f sur $\overline{I}$, il existe une jauge δ sur $\overline{I}$ telle que pour toutes P-partitions δ-fines de I, Π et $\widetilde{\Pi}$,

$$|S(I, f, \Pi) - S(I, f, \widetilde{\Pi})| \leq \frac{\varepsilon}{i}.$$

Pour tout $y \in \overline{J}$, $\delta(y,\cdot)$ est une jauge sur $\overline{K}$. On cherche à construire une jauge δ_J sur $\overline{J}$. Soit $y \in \overline{J}$. Alors

— Si $y \in T_i$, par définition de T_i, il existe deux P-partitions $\delta(y,\cdot)$-fines de K,

$$\Pi_K^y := \{({z^j}_y, {K^j}_y) : 1 \leq j \leq m_y\} \text{ et } \widetilde{\Pi}_K^y := \{({\widetilde{z^j}}_y, {\widetilde{K^j}}_y) : 1 \leq j \leq \widetilde{m}_y\}$$

telles que

$$S(K, f(y,\cdot), \Pi_K^y) - S(K, f(y,\cdot), \widetilde{\Pi}_K^y) \geq \frac{1}{i}.$$

On pose alors

$$\delta_J(y) := \min\left[\min_{1\leq j\leq m_y} \delta(y, {z^j}_y), \min_{1\leq l\leq \widetilde{m}_y} \delta(y, {\widetilde{z^l}}_y)\right].$$

— Si $y \in \overline{J}\backslash T_i$, posons

$$\Pi_K^y := \{({z^j}_y, {K^j}_y) : 1 \leq j \leq m_y\}$$

une P-partition $\delta(y,\cdot)$-fine quelconque de K, qui existe par le lemme de Cousin, et posons

$$\delta_J(y) := \min_{1\leq j\leq m_y} \delta(y, {z^j}_y).$$

Nous avons ainsi défini une jauge δ_J sur $\overline{J}$. Soit $\Pi_J = \left(\left(y^h, J^h\right)\right)_{1\leq h\leq m}$ une P-partition δ_J-fine de J, qui existe par le lemme de Cousin. Il reste à montrer que $S(J, 1_{T_i}, \Pi_J) \leq \varepsilon$. Posons

$$\Pi := \{\left(\left(y^h, {z^j}_{y^h}\right), J^h \times {K^j}_{y^h}\right) \ tel\ que\ 1 \leq j \leq m_{y^h}, 1 \leq h \leq m\}$$

et

$$\widetilde{\Pi} := \{\left(\left(y^h, {\widetilde{z^j}}_{y^h}\right), J^h \times {\widetilde{K^j}}_{y^h}\right) \ tel\ que\ 1 \leq j \leq \widetilde{m}_{y^h}, 1 \leq h \leq m\}.$$

Alors, par construction, Π et $\widetilde{\Pi}$ sont deux P-partitions δ-fines de $I = J \times K$. Alors

$$|S(I,f,\Pi) - S(I,f,\widetilde{\Pi})| \leq \frac{\varepsilon}{i}.$$

Or

$$\begin{aligned}
|S(I,f,\Pi) - S(I,f,\widetilde{\Pi})| &= \left| \sum_{h=1}^{m} \mu\left(J^h\right) \left[\sum_{j=1}^{m_{y^h}} f(y^h, z^j{}_{y^h}) \mu(K^j{}_{y^h}) - \sum_{j=1}^{\widetilde{m}_{y^h}} f(y^h, \widetilde{z^j}{}_{y^h}) \mu(\widetilde{K^j}{}_{y^h}) \right] \right|, \\
&\geq \left| \sum_{\{1 \leq h \leq m \,:\, y^h \in T_i\}} \mu(J^h) \left[\sum_{j=1}^{m_{y^h}} f(y^h, z^j{}_{y^h}) \mu(K^j{}_{y^h}) - \sum_{j=1}^{\widetilde{m}_{y^h}} f(y^h, \widetilde{z^j}{}_{y^h}) \mu(\widetilde{K^j}{}_{y^h}) \right] \right|, \\
&= \left| \sum_{\{1 \leq h \leq m \,:\, y^h \in T_i\}} \mu(J^h) \left[S(K, f(y^h,\cdot), \Pi_K^{y^h}) - S(K, f(y^h,\cdot), \widetilde{\Pi}_K^{y^h}) \right] \right|, \\
&= \sum_{\{1 \leq h \leq m \,:\, y^h \in T_i\}} \mu(J^h) \left[S(K, f(y^h,\cdot), \Pi_K^{y^h}) - S(K, f(y^h,\cdot), \widetilde{\Pi}_K^{y^h}) \right], \\
&\geq \frac{1}{i} \sum_{\{1 \leq h \leq m \,:\, y^h \in T_i\}} \mu(J^h).
\end{aligned}$$

De plus,

$$\sum_{\{1 \leq h \leq m \,:\, y^h \in T_i\}} \mu(J^h) = \sum_{h=1}^{m} 1_{T_i}(y^h) \mu(J^h) = S(J, 1_{T_i}, \Pi_J).$$

Donc

$$\frac{1}{i} S(J, 1_{T_i}, \Pi_J) \leq |S(I,f,\Pi) - S(I,f,\widetilde{\Pi})| \leq \frac{\varepsilon}{i}.$$

Donc $0 \leq S(J, 1_{T_i}, \Pi_J) \leq \varepsilon$. □

Lemme 4.3. *Soit f une fonction intégrable sur $\overline{I}$. Alors, la fonction F définie pour presque tout $y \in \overline{J}$ par*

$$F(y) = \int_{\overline{K}} f(y,z) \mathrm{d}z$$

est intégrable sur $\overline{J}$ et

$$\int_{\overline{J}} F = \int_{\overline{I}} f.$$

Autrement dit,

$$\int_{\overline{J}} \left[\int_{\overline{K}} f(y,z) \mathrm{d}z \right] \mathrm{d}y = \int_{\overline{J} \times \overline{K}} f.$$

Démonstration. Soit $\varepsilon > 0$. f étant intégrable sur $\overline{I}$, il existe une jauge δ_1 sur $\overline{I}$ telle que pour toute P-partition δ_1-fine, Π, de I,

$$\left| S(I,f,\Pi) - \int_{\overline{I}} f \right| \leq \frac{\varepsilon}{4}.$$

D'autre part, il existe une jauge δ_2 sur $\overline{I}$ telle que pour toutes P-partitions δ_2-fines, Π et $\widetilde{\Pi}$, de I,

$$\left| S(I,f,\Pi) - S(I,f,\widetilde{\Pi}) \right| \leq \frac{\varepsilon}{4}.$$

Posons donc $\delta := \min(\delta_1, \delta_2)$. Alors, si Π et $\widetilde{\Pi}$ sont des P-partitions δ-fines de I, on a

$$\left| S(I,f,\Pi) - \int_{\overline{I}} f \right| \leq \frac{\varepsilon}{4}$$

et

$$\left| S(I,f,\Pi) - S(I,f,\widetilde{\Pi}) \right| \leq \frac{\varepsilon}{4}.$$

On note toujours F une extension de F à $\overline{J}$. Soit $y \in T$ et soit $\overline{\Pi}_K^y = \left\{ \left(\overline{z^j}_y, \overline{K^j}_y \right) \text{ tel que } 1 \leq j \leq \overline{m}_y \right\}$ une P-partition $\delta(y,\cdot)$-fine de K. On pose alors

— $\widetilde{\delta}_J(y) = \min\limits_{1\leq j\leq m_y} \delta(y, \overline{z^j}_y)$,

— $Q_1 = \left\{ y \in T \text{ tel que } |F(y)| + |S(K, f(y,\cdot), \overline{\Pi}_K^y)| \leq 1 \right\}$,

— pour tout $k \in \mathbb{N}^*$, $Q_k = \left\{ y \in T \text{ tel que } k-1 < |F(y)| + |S(K, f(y,\cdot), \overline{\Pi}_K^y)| \leq k \right\}$.

Alors $T = \bigcup\limits_{k\in\mathbb{N}^*} Q_k$, avec $Q_k \cap Q_l = \emptyset$ si $k \neq l$. Pour tout $k \in \mathbb{N}^*$, et pour tout $y \in \mathbb{R}^q$, on a alors $1_{Q_k}(y) \leq 1_T(y)$, et donc, pour toute P-partition Π_J de J, on a $0 \leq S(J, 1_{Q_k}, \Pi_J) \leq S(J, 1_T, \Pi_J)$. Par le lemme précédent, pour tout $k \in \mathbb{N}^*$, il existe une jauge ${\delta_J}^k$ sur $\overline{J}$ telle que pour toute P-partition ${\delta_J}^k$-fine de J, notée $\Pi_J = ((y^i, J^i))_{1\leq i\leq m}$, on a

$$S(J, 1_T, \Pi_J) \leq \frac{\varepsilon}{k.2^{k+2}}.$$

On a alors

$$0 \leq \sum_{\{1\leq i\leq m \,:\, y^i\in Q_k\}} \mu(J^i) = S(J, 1_{Q_k}, \Pi_J) \leq S(J, 1_T, \Pi_J) \leq \frac{\varepsilon}{k.2^{k+2}}.$$

Si $y \in T$, alors il existe un et un seul $k \in \mathbb{N}^*$ tel que $y \in Q_k$; on pose alors pour tout $y \in T$, $\delta_J(y) = \min\left\{ \widetilde{\delta}_J(y), {\delta_J}^k(y) \right\}$. Soit maintenant $y \in \overline{J}\backslash T$ et soit $\widetilde{\Pi}_K^y = \left\{ (\widetilde{z^j}_y, \widetilde{K^j}_y) : 1 \leq j \leq \widetilde{m}_y \right\}$ une P-partition $\delta(y,\cdot)$-fine de K. Puisque $f(y,\cdot)$ est intégrable sur $\overline{K}$ et admet $F(y)$ comme intégrale, on peut choisir une P-partition $\widehat{\Pi}_K^y = \left\{ (\widehat{z^l}_y, \widehat{K^l}_y) : 1 \leq l \leq \widehat{m}_y \right\}$, $\delta(y,\cdot)$-fine de K telle que

$$|S(K, f(y,\cdot), \widehat{\Pi}_K^y) - F(y)| \leq \frac{1}{2}\, |S(K, f(y,\cdot), \widetilde{\Pi}_K^y) - F(y)|. \tag{8}$$

Posons $\delta_J(y) = \min\left\{ \min\limits_{1\leq j\leq \widetilde{m}_y} \delta(y, \widetilde{z^j}_y), \min\limits_{1\leq l\leq \widehat{m}_y} \delta(y, \widehat{z^l}_y) \right\}$, ce qui achève de définir une jauge δ_J sur $\overline{J}$.

Soit maintenant $\Pi_J = \left((y^i, J^i)\right)_{1\leq i\leq m}$ une P-partition δ_J -fine de J. Pour tout $i \in \{1, \cdots, m\}$, on est exactement dans l'un des cas suivants :

— Si $y^i \in T$, posons

$$\Pi_K^{y^i} = \left\{ (z^j{}_{y^i}, K^j{}_{y^i}) : 1 \leq j \leq m_{y^i} \right\} = \overline{\Pi}_K^{y^i}$$
$$\breve{\Pi}_K^{y^i} = \left\{ (\breve{z^j}{}_{y^i}, \breve{K^j}{}_{y^i}) : 1 \leq j \leq \breve{m}_{y^i} \right\} = \overline{\Pi}_K^{y^i}.$$

— Si $y^i \in \overline{J}\backslash T$ et $S(K, f(y^i,\cdot), \widetilde{\Pi}_K^{y^i}) - F(y^i) > 0$, posons

$$\Pi_K^{y^i} = \left\{ (z^j{}_{y^i}, K^j{}_{y^i}) : 1 \leq j \leq m_{y^i} \right\} = \widetilde{\Pi}_K^{y^i},$$
$$\breve{\Pi}_K^{y^i} = \left\{ (\breve{z^j}{}_{y^i}, \breve{K^j}{}_{y^i}) : 1 \leq j \leq \breve{m}_{y^i} \right\} = \widehat{\Pi}_K^{y^i}.$$

— Si $y^i \in \overline{J}\backslash T$ et $S(K, f(y^i,\cdot), \widetilde{\Pi}_K^{y^i}) - F(y^i) \leq 0$, posons

$$\Pi_K^{y^i} = \left\{ (z^j{}_{y^i}, K^j{}_{y^i}) : 1 \leq j \leq m_{y^i} \right\} = \widehat{\Pi}_K^{y^i},$$
$$\breve{\Pi}_K^{y^i} = \left\{ (\breve{z^j}{}_{y^i}, \breve{K^j}{}_{y^i}) : 1 \leq j \leq \breve{m}_{y^i} \right\} = \widetilde{\Pi}_K^{y^i}.$$

Par construction, on a alors que

$$\Pi = \left\{ \left(\left(y^i, z^j{}_{y^i}\right), J^i \times K^j{}_{y^i} \right) \;:\; 1 \leq j \leq m_{y^i},\ 1 \leq i \leq m \right\}$$

et

$$\breve{\Pi} = \left\{ \left(\left(y^i, \breve{z^j}{}_{y^i}\right), J^i \times \breve{K^j}{}_{y^i} \right) \;:\; 1 \leq j \leq \breve{m}_{y^i},\ 1 \leq i \leq m \right\}$$

sont des P-partitions δ-fines de I. On a donc

$$|S(I, f, \Pi) - S(I, f, \breve{\Pi})| \leq \frac{\varepsilon}{4}.$$

De plus,

$$S(I, f, \Pi) = \sum_{i=1}^{m} S(K, f(y^i,\cdot), \Pi_K^{y^i})\mu(J^i)$$

et

$$S(I,f,\breve{\Pi}) = \sum_{i=1}^{m} S(K, f(y^i,\cdot), \breve{\Pi}_K^{y^i})\mu(J^i).$$

D'autre part,

$$\begin{aligned}|S(I,f,\Pi) - S(J,F,\Pi_J)| &= |S(I, 1_T\ f, \Pi) + S(I, 1_{\overline{J}\setminus T}\ f, \Pi) - S(J, 1_T\ F, \Pi_J) - S(J, 1_{\overline{J}\setminus T}\ F, \Pi_J)|,\\ &\leq |S(I, 1_T\ f, \Pi) - S(J, 1_T\ F, \Pi_J)| + |S(I, 1_{\overline{J}\setminus T}\ f, \Pi) - S(J, 1_{\overline{J}\setminus T}\ F, \Pi_J)|.\end{aligned}$$

On majore alors séparément sur T et sur $\overline{J}\setminus T$.

Sur T :

$$\begin{aligned}|S(I, 1_T\ f, \Pi) - S(J, 1_T\ F, \Pi_J)| &= \left| \sum_{\{1\leq i\leq m\ :\ y^i\in T\}} [S(K, f(y^i,\cdot), \Pi_K^{y^i}) - F(y^i)]\mu(J^i) \right|,\\ &\leq \sum_{k=1}^{\infty} \left\{ \sum_{\{1\leq i\leq m\ :\ y^i\in Q_k\}} \left[|S(K, f(y^i,\cdot), \Pi_K^{y^i})| + |F(y^i)| \right] \mu(J^i) \right\},\\ &\text{par définition de } Q_k,\\ &\leq \sum_{k=1}^{\infty} \left\{ \sum_{\{1\leq i\leq m\ :\ y^i\in Q_k\}} k\mu(J^i) \right\},\\ &\leq \varepsilon \sum_{k=1}^{\infty} \frac{1}{2^{k+2}} = \frac{\varepsilon}{4}.\end{aligned}$$

Sur $\overline{J}\setminus T$:

$$\left| S(I, 1_{\overline{J}\setminus T}\ f, \Pi) - S(J, 1_{\overline{J}\setminus T}\ F, \Pi_J) \right| = \left| \sum_{\{1\leq i\leq m\ :\ y^i\in \overline{J}\setminus T\}} [S(K, f(y^i,\cdot), \Pi_K^{y^i}) - F(y^i)]\mu(J^i) \right|.$$

— Si $y^i \in \overline{J}\setminus T$ et $S(K, f(y^i,\cdot), \widetilde{\Pi}_K^{y^i}) - F(y^i) > 0$, alors par (8) on a

$$S(K, f(y^i,\cdot), \widehat{\Pi}_K^{y^i}) - F(y^i) \leq \frac{1}{2}\ [S(K, f(y^i,\cdot), \widetilde{\Pi}_K^{y^i}) - F(y^i)],$$

i.e.

$$S(K, f(y^i,\cdot), \breve{\Pi}_K^{y^i}) - F(y^i) \leq \frac{1}{2}\ [S(K, f(y^i,\cdot), \Pi_K^{y^i}) - F(y^i)],$$

ce qui donne

$$2[S(K, f(y^i,\cdot), \breve{\Pi}_K^{y^i}) - F(y^i)] \leq [S(K, f(y^i,\cdot), \Pi_K^{y^i}) - F(y^i)],$$

puis

$$2[S(K, f(y^i,\cdot), \breve{\Pi}_K^{y^i}) - F(y^i)] + S(K, f(y^i,\cdot), \Pi_K^{y^i}) \leq [S(K, f(y^i,\cdot), \Pi_K^{y^i}) - F(y^i)] + S(K, f(y^i,\cdot), \Pi_K^{y^i}),$$

et donc

$$0 < S(K, f(y^i,\cdot), \Pi_K^{y^i}) - F(y^i) \leq 2[S(K, f(y^i,\cdot), \Pi_K^{y^i}) - S(K, f(y^i,\cdot), \breve{\Pi}_K^{y^i})].$$

— Si $y^i \in \overline{J}\setminus T$ et $S(K, f(y^i,\cdot), \widetilde{\Pi}_K^{y^i}) - F(y^i) \leq 0$, alors par (8) on a

$$F(y^i) - S(K, f(y^i,\cdot), \widehat{\Pi}_K^{y^i}) \leq \frac{1}{2}\ [F(y^i) - S(K, f(y^i,\cdot), \widetilde{\Pi}_K^{y^i})],$$

i.e.

$$F(y^i) - S(K, f(y^i,\cdot), \Pi_K^{y^i}) \leq \frac{1}{2}\ [F(y^i) - S(K, f(y^i,\cdot), \breve{\Pi}_K^{y^i})],$$

puis

$$\begin{aligned}&[F(y^i) - S(K, f(y^i,\cdot), \Pi_K^{y^i})] - \frac{1}{2}\ S(K, f(y^i,\cdot), \breve{\Pi}_K^{y^i})\\ &\qquad \leq \frac{1}{2}\ [F(y^i) - S(K, f(y^i,\cdot), \breve{\Pi}_K^{y^i})] - \frac{1}{2}\ S(K, f(y^i,\cdot), \breve{\Pi}_K^{y^i}),\end{aligned}$$

et donc

$$\frac{1}{2}\,[F(y^i) - S(K, f(y^i,\cdot), \breve{\Pi}_K^{y^i})] \leq S(K, f(y^i,\cdot), \Pi_K^{y^i}) - S(K, f(y^i,\cdot), \breve{\Pi}_K^{y^i}),$$

et dès lors,

$$\begin{aligned}|S(K, f(y^i,\cdot), \Pi_K^{y^i}) - F(y^i)| &\leq \frac{1}{2}\,[F(y^i) - S(K, f(y^i,\cdot), \breve{\Pi}_K^{y^i})],\\ &\leq S(K, f(y^i,\cdot), \Pi_K^{y^i}) - S(K, f(y^i,\cdot), \breve{\Pi}_K^{y^i}).\end{aligned}$$

Donc, finalement, si $y^i \in \overline{J}\backslash T$, on a

$$|S(K, f(y^i,\cdot), \Pi_K^{y^i}) - F(y^i)| \leq 2[S(K, f(y^i,\cdot), \Pi_K^{y^i}) - S(K, f(y^i,\cdot), \breve{\Pi}_K^{y^i})].$$

Or, par construction, si $y^i \in T$, on a $\breve{\Pi}_K^{y^i} = \overline{\Pi}_K^{y^i} = \Pi_K^{y^i}$; donc

$$\begin{aligned}S(I, f, \Pi) - S(I, f, \breve{\Pi}) &= S(I, 1_{\overline{J}\backslash T}\ f, \Pi) - S(I, 1_{\overline{J}\backslash T}\ f, \breve{\Pi}),\\ &= \sum_{\{1\leq i\leq m\ :\ y^i\in\overline{J}\backslash T\}} [S(K, f(y^i,\cdot), \Pi_K^{y^i}) - S(K, f(y^i,\cdot), \breve{\Pi}_K^{y^i})]\mu(J^i).\end{aligned}$$

Ainsi,

$$\begin{aligned}|S(I, 1_{\overline{J}\backslash T}\ f, \Pi) - S(J, 1_{\overline{J}\backslash T}\ F, \Pi_J)| &= \left|\sum_{\{1\leq i\leq m\ :\ y^i\in\overline{J}\backslash T\}} [S(K, f(y^i,\cdot), \Pi_K^{y^i}) - F(y^i)]\mu(J^i)\right|,\\ &\leq \sum_{\{1\leq i\leq m\ :\ y^i\in\overline{J}\backslash T\}} \left|S(K, f(y^i,\cdot), \Pi_K^{y^i}) - F(y^i)\right|\mu(J^i),\\ &\leq \sum_{\{1\leq i\leq m\ :\ y^i\in\overline{J}\backslash T\}} 2[S(K, f(y^i,\cdot), \Pi_K^{y^i}) - S(K, f(y^i,\cdot), \breve{\Pi}_K^{y^i})]\mu(J^i),\\ &= 2[S(I, f, \Pi) - S(I, f, \breve{\Pi})],\\ &\leq \frac{\varepsilon}{2}.\end{aligned}$$

Donc

$$|S(I, f, \Pi) - S(J, F, \Pi_J)| \leq \frac{\varepsilon}{4} + \frac{\varepsilon}{2} = \frac{3\varepsilon}{4},$$

et alors

$$\begin{aligned}\left|\int_{\overline{I}} f - S(J, F, \Pi_J)\right| &\leq \left|\int_{\overline{I}} f - S(I, f, \Pi)\right| + |S(I, f, \Pi) - S(J, F, \Pi_J)|\,,\\ &\leq \frac{\varepsilon}{4} + \frac{3\varepsilon}{4} = \varepsilon.\end{aligned}$$

□

Lemme 4.4. *Soit f une fonction intégrable sur $\overline{I}$. Posons*

$$S = \{z \in \overline{K} : f(\cdot, z)\ \textit{n'est pas intégrable sur}\ \overline{J}\}.$$

Alors S est de s-mesure nulle ; la fonction G définie pour presque tout $z \in \overline{K}$ par

$$G(z) = \int_{\overline{J}} f(\cdot, z) = \int_{\overline{J}} f(y, z)\mathrm{d}y$$

est intégrable sur $\overline{K}$ et

$$\int_{\overline{K}} G = \int_{\overline{I}} f.$$

Autrement dit

$$\int_{\overline{K}} \left[\int_{\overline{J}} f(y, z)\mathrm{d}y\right] \mathrm{d}z = \int_{\overline{I}} f.$$

Démonstration. On inverse les rôles de y et z dans la démonstration du Lemme 4.3. □

Théorème 4.5 (Théorème de Fubini pour les pavés)**.** *Soit f une fonction intégrable sur $\overline{I}$. Alors*

$$\int_{\overline{K}} \left[\int_{\overline{J}} f(y,z)\mathrm{d}y \right] \mathrm{d}z = \int_{\overline{J}} \left[\int_{\overline{K}} f(y,z)\mathrm{d}z \right] \mathrm{d}y.$$

On va pouvoir montrer qu'une intégrale sur un pavé de $\mathbb{R}^n$ se ramène à un calcul de n intégrales successives sur un intervalle fermé.

Corollaire 4.6. *Soit $f : \mathbb{R}^n \longrightarrow \mathbb{R}^p$ une fonction intégrable sur un pavé $\overline{I} = \overline{I_1} \times \cdots \times \overline{I_n}$. Alors, pour toute permutation $\{i_1, \cdots, i_n\}$ de $\{1, \cdots, n\}$,*

$$\int_{\overline{I}} f = \int_{\overline{I_{i_n}}} \left[\int_{\overline{I_{i_{n-1}}}} \left[\cdots \left[\int_{\overline{I_{i_1}}} f(x_1, \cdots, x_n)\mathrm{d}x_{i_1} \right] \cdots \right] \mathrm{d}x_{i_{n-1}} \right] \mathrm{d}x_{i_n}.$$

Démonstration. On applique le théorème de Fubini à différentes décompositions de $\overline{I}$. □

On souhaite étendre le théorème à une partie bornée de $\mathbb{R}^n$. Fixons les notations. Soit $C \subset \mathbb{R}^n = \mathbb{R}^q \times \mathbb{R}^s$ une partie bornée de $\mathbb{R}^n$. Alors on pose

$$A := \{y \in \mathbb{R}^q \ : \ \exists z \in \mathbb{R}^s \ : \ (y,z) \in C\},$$

$$B := \{z \in \mathbb{R}^s \ : \ \exists y \in \mathbb{R}^q \ : \ (y,z) \in C\}.$$

Ce sont les projection orthogonales de C respectivement sur $\mathbb{R}^q$ et $\mathbb{R}^s$. Pour tout $y \in A$, posons

$$B(y) = \{z \in \mathbb{R}^s \ : \ (y,z) \in C\}$$

et pour tout $z \in B$, posons

$$A(z) = \{y \in \mathbb{R}^q \ : \ (y,z) \in C\}.$$

Remarque 4.1. $C = \{(y,z) \ : \ y \in A, \ z \in B(y)\} = \{(y,z) \ : \ z \in B, \ y \in A(z)\}$.

Remarque 4.2. *Pour tout $x = (y,z) \in \mathbb{R}^n$, $1_C(x) = 1_C(y,z) = 1_A(y)1_{B(y)}(z) = 1_{A(z)}(y)1_B(z)$.*

On a besoin de définir l'intégrabilité sur C.

Définition 4.3. *Soit $f : C \subset \mathbb{R}^n \longrightarrow \mathbb{R}^p$ une application. On dit que f est* intégrable sur C *lorsqu'il existe un semi-pavé $I \subset \mathbb{R}^n$ tel que $C \subset I$ et*

$$\widetilde{f} \ : \quad \begin{array}{lll} I \subset \mathbb{R}^n & \longrightarrow \mathbb{R}^p & \\ x & \longmapsto \begin{cases} f(x) & si\ x \in A, \\ 0 & si\ x \in I \backslash A. \end{cases} \end{array}$$

est intégrable sur $\overline{I}$.

Théorème 4.7 (Théorème de Fubini pour une partie quelconque bornée de $\mathbb{R}^n$.)**.** *Soit f une fonction intégrable sur $C \subset \mathbb{R}^n$ borné. Posons*

$$\widetilde{A} = \{y \in A \ : \ f(y,\cdot) \text{ n'est pas intégrable sur } B(y)\},$$

$$\widetilde{B} = \{z \in B \ : \ f(\cdot,z) \text{ n'est pas intégrable sur } A(z)\}.$$

Alors

- *$\widetilde{A}$ est de q-mesure nulle ;*
- *$\widetilde{B}$ est de s-mesure nulle ;*
- *La fonction définie presque partout sur A par*

$$F \ : \ y \longmapsto \int_{B(y)} f(y,z)\mathrm{d}z$$

  est intégrable sur A ;
- *La fonction définie presque partout sur B par*

$$G \ : \ z \longmapsto \int_{A(z)} f(y,z)\mathrm{d}y$$

  est intégrable sur B ;

— *On a*

$$\int_A F = \int_B G = \int_C f,$$

autrement dit,

$$\int_A \left[\int_{B(y)} f(y,z)\mathrm{d}z \right] \mathrm{d}y = \int_B \left[\int_{A(z)} f(y,z)\mathrm{d}y \right] \mathrm{d}z = \int_C f.$$

Démonstration. Soit $I = J \times K \subset \mathbb{R}^n = \mathbb{R}^q \times \mathbb{R}^s$ un semi-pavé tel que $C \subset I$. On prolonge f à $\overline{I}$ et on note $\widetilde{f}$ ce prolongement. On a alors pour presque tout $x \in \overline{I}$,

$$f(x) = 1_C(x)\widetilde{f}(x)$$

avec $1_C.\widetilde{f}$ qui est une fonction intégrable sur $\overline{I}$. Posons l'ensemble

$$T := \left\{ y \in \overline{J} \ : \ 1_C(y,\cdot)\widetilde{f}(y,\cdot) \text{ n'est pas intégrable sur } \overline{K} \right\}.$$

Par les Remarques 4.1 et 4.2,

$$\begin{aligned} T &= \left\{ y \in \overline{J} \ : \ 1_A(y)1_{B(y)}(\cdot)\widetilde{f}(y,\cdot) \text{ n'est pas intégrable sur } \overline{K} \right\} \\ &= \left\{ y \in \overline{J} \ : \ 1_A(y)\widetilde{f}(y,\cdot) \text{ n'est pas intégrable sur } B(y) \right\} \\ &= \{ y \in A \ : \ f(y,\cdot) \text{ n'est pas intégrable sur } B(y) \} = \widetilde{A}. \end{aligned}$$

Par le Théorème de Fubini pour les pavés, T est de q-mesure nulle. Donc $\widetilde{A}$ est de q-mesure nulle. On trouve par le même raisonnement que $\widetilde{B}$ est de s-mesure nulle. De plus, la fonction définie sur presque tout $y \in \overline{K}$ par

$$\begin{aligned} \widetilde{F} \ : \ y \longmapsto \int_{\overline{K}} 1_C(y,z)\widetilde{f}(y,z)\mathrm{d}z &= 1_A(y) \int_{\overline{K}} 1_{B(y)}(z)\widetilde{f}(y,z)\mathrm{d}z, \\ &= 1_A(y) \int_{B(y)} \widetilde{f}(y,z)\mathrm{d}z, \\ &= 1_A(y) \int_{B(y)} f(y,z)\mathrm{d}z, \\ &= 1_A(y)F(y), \end{aligned}$$

est intégrable sur $\overline{J}$ et

$$\int_{\overline{J}} \left[1_A(y) \int_{B(y)} f(y,z)\mathrm{d}z \right] \mathrm{d}y = \int_{\overline{I}} f.$$

Autrement dit, F est intégrable sur A et

$$\int_A \left[\int_{B(y)} f(y,z)\mathrm{d}z \right] \mathrm{d}y = \int_C f.$$

De même,

$$\int_B \left[\int_{A(y)} f(y,z)\mathrm{d}y \right] \mathrm{d}z = \int_C f.$$

□

5 Incompatibilité des deux théorèmes

On va montrer ici que la définition de l'intégrale au sens de Mawhin ne permet pas d'obtenir un théorème de Fubini général. Plus généralement, on va montrer qu'il y a, quelque soit la théorie de l'intégrale, une incompatibilité des théorème de la divergence et du théorème de Fubini.

5.1 Notion de théorie d'intégrale.

Pour pouvoir parler d'intégrale, il faut une théorie de l'intégrale. Ce concept peut sembler confus ; il s'agira ici d'en donner une définition précise.

Définition 5.1. *On définit une* théorie d'intégrale uni-dimensionnelle *comme suit :*
Pour chaque intervalle $I \subset \mathbb{R}$, on se donne un espace $\mathbb{I}(I)$ (qui contient l'espace $C(I)$ des fonctions réelles continues sur I) de fonctions à valeurs réelles définies sur I et dites intégrables sur I. *On se donne ainsi une fonction linéaire positive*

$$\begin{array}{rccc} \int_I & : & \mathbb{I}(I) & \longrightarrow \mathbb{R} \\ & & f & \longmapsto \int_I f, \end{array}$$

telle que :

(A) *pour tout intervalle I, pour tout $\left(I^1, \cdots, I^m\right)$ intervalles presque disjoints deux à deux tels que $I = \bigcup_{j=1}^{m} I^j$ et pour tout f intégrable sur I, pour chaque $1 \leq j \leq m$, la restriction de f à I^j est intégrable sur I^j et*

$$\int_I f = \sum_{j=1}^{m} \int_{I^j} f;$$

(H) *pour tout intervalle I, pour tout $f : I \longrightarrow \mathbb{R}$ intégrable sur I, si on pose, pour tout $c \in \mathrm{int}(I)$,*

$$I_c^- := I \cap]-\infty, c] \text{ et } I_c^+ := I \cap [c, +\infty[,$$

on a

$$\int_I f = \lim_{c \to \max I} \int_{I_c^-} f = \lim_{c \to \min I} \int_{I_c^+} f;$$

(N) *pour tout intervalle I, la fonction constante égale à 1 est intégrable sur I et on a*

$$\int_I 1 = l(I) = \max I - \min I.$$

On définit de même une théorie d'intégrale bi-dimensionnelle *comme suit :*
Pour chaque pavé $A \subset \mathbb{R}^2$, on se donne un espace $\mathbb{I}(A)$ (qui contient l'espace $C(A)$ des fonctions réelles continues sur A) de fonctions à valeurs réelles définies sur A et dites intégrables sur I. *On se donne ainsi une fonction linéaire positive*

$$\begin{array}{rccc} \iint_A & : & \mathbb{I}(A) & \longrightarrow \mathbb{R} \\ & & f & \longmapsto \iint_A f, \end{array}$$

telle que :

(A) *pour tout pavé A, pour tout $\left(A^1, \cdots, A^m\right)$ pavés presque disjoints deux à deux tels que $A = \bigcup_{j=1}^{m} A^j$ et pour tout f intégrable sur A, pour chaque $1 \leq j \leq m$, la restriction de f à A^j est intégrable sur A^j et*

$$\iint_A f = \sum_{j=1}^{m} \iint_{A^j} f.$$

Définition 5.2. *On se donne une théorie d'intégrale bi-dimensionnelle. On dit*

(D) *qu'elle* intègre toutes les divergences *lorsque, pour tout pavé $A \subset \mathbb{R}^2$ et pour tout champ de vecteurs différentiable $\mathrm{v} : A \longrightarrow \mathbb{R}^2$, la fonction* div v *est intégrable sur A ;*

(F) *qu'elle* vérifie la condition de Fubini *lorsqu'il existe une théorie d'intégrale uni-dimensionnelle telle que, pour tout pavé $A = I \times J$ où I et J sont des intervalles de $\mathbb{R}$ et pour toute fonction $f : A \longrightarrow \mathbb{R}$ intégrable sur A, l'ensemble*

$$\{x \in I \ : \ f(x, \cdot) \text{ n'est pas intégrable sur } J\}$$

est 1-négligeable.

Observons que l'on souhaite montrer que les conditions (D) et (F) sont incompatibles. Pour ce faire, on a besoin d'introduire la notion de *suite équidistribuée.*

5.2 Suite équidistribuée.

Définition 5.3. *Une suite* $(a_n)_{n\in\mathbb{N}^*}$ *est* équidistribuée mod 1 *lorsque* $\forall (a,b)\in[0,1]^2$ *tel que* $0\leq a<b\leq 1$,

$$\lim_{N\to\infty}\frac{\operatorname{card}\{a_n\ :\ 1\leq n\leq N,\ a_n\in[a,b[\}}{N}=b-a.$$

Théorème 5.1 (Critère de Weyl.)**.** *Une suite* $(a_n)_{n\in\mathbb{N}^*}$ *est équidistribuée* mod 1 *si et seulement si* $\forall k\in\mathbb{N}^*$,

$$\lim_{N\to\infty}\frac{1}{N}\sum_{n=1}^{N}\mathrm{e}^{2\mathrm{i}\pi k a_n}=0.$$

Démonstration. Admis. □

Montrons un premier résultat.

Proposition 5.2. *Soit* $(a_n)_{n\in\mathbb{N}^*}$ *une suite d'entiers distincts. Alors, pour presque tout* $x\in[0,1]$*, la suite* $(a_n x)_{n\in\mathbb{N}^*}$ *est équidistribuée* mod 1.

Démonstration. Soit k un entier non nul. Pour tout $N\geq 1$ et pour tout $0\leq x\leq 1$, on définit

$$S(N,x):=\frac{1}{N}\sum_{n=1}^{N}\mathrm{e}^{2\mathrm{i}\pi k a_n x}.$$

Alors

$$\begin{aligned}|S(N,x)|^2&=S(N,x)\overline{S(N,x)},\\&=\frac{1}{N^2}\sum_{m=1}^{N}\sum_{n=1}^{N}\mathrm{e}^{2\mathrm{i}\pi k(a_m-a_n)x}.\end{aligned}$$

et donc

$$\begin{aligned}\int_0^1|S(N,x)|^2\mathrm{d}x&=\frac{1}{N^2}\sum_{m=1}^{N}\sum_{n=1}^{N}\underbrace{\int_0^1\mathrm{e}^{2\mathrm{i}\pi k(a_m-a_n)x}\mathrm{d}x}_{=\delta_{m,n}},\\&=\frac{1}{N^2}\sum_{m=1}^{N}\sum_{n=1}^{N}\delta_{m,n},\\&=\frac{1}{N}.\end{aligned}$$

Ainsi,

$$\int_0^1|S(N^2,x)|^2\mathrm{d}x=\frac{1}{N^2}.$$

D'où

$$\sum_{N=1}^{\infty}\int_0^1\left|S(N^2,x)\right|^2\mathrm{d}x=\sum_{N=1}^{\infty}\frac{1}{N^2}<+\infty.$$

On en déduit que le terme général de la série tend vers 0,

$$\lim_{N\to\infty}\int_0^1\left|S(N^2,x)\right|^2\mathrm{d}x=0,$$

Par théorème de convergence dominée et unicité de la limite,

$$\lim_{N\to\infty}\int_0^1\left|S(N^2,x)\right|^2\mathrm{d}x=\int_0^1\lim_{N\to\infty}\left|S(N^2,x)\right|^2\mathrm{d}x=0$$

Ce qui entraîne que, pour presque tout $x\in[0,1]$,

$$\lim_{N\to\infty}S(N^2,x)=0.$$

Soit $N \geq 1$. Il existe $m \in \mathbb{N}^*$ tel que $m^2 \leq N < (m+1)^2$. Alors, on obtient les inégalités suivantes :

$$\begin{aligned}
|S(N,x)| &= \left| \frac{m^2}{N} \frac{1}{m^2} \sum_{n=1}^{N} \mathrm{e}^{2\mathrm{i}\pi k a_n x} \right|, \\
&\leq \frac{m^2}{N} \left[\left| \frac{1}{m^2} \sum_{n=1}^{m^2} \mathrm{e}^{2\mathrm{i}\pi k a_n x} \right| + \left| \frac{1}{m^2} \sum_{n=m^2+1}^{N} \mathrm{e}^{2\mathrm{i}\pi k a_n x} \right| \right], \\
&\leq \frac{m^2}{N} \left[\left| S\left(m^2, x\right) \right| + \frac{1}{m^2} \sum_{n=m^2+1}^{m^2+2m} \underbrace{\left| \mathrm{e}^{2\mathrm{i}\pi k a_n x} \right|}_{=1} \right], \\
&\leq \left| S(m^2, x) \right| + \frac{2m}{N}, \\
&\leq \left| S(m^2, x) \right| + \frac{2}{\sqrt{N}} \underset{N \to \infty}{\longrightarrow} 0 \text{ pour presque tout } x \in [0,1].
\end{aligned}$$

Par le critère de Weyl, la suit $(a_n x)_{n \in \mathbb{N}^*}$ est équidistribuée mod 1 pour presque tout $x \in [0,1]$. □

Remarque 5.1. *Si une suite $(a_n)_{n \in \mathbb{N}^*}$ est équidistribuée* mod 1, *alors l'ensemble $\{a_n \ : \ n \in \mathbb{N}^*\}$ est dense dans* $[0,1]$.

5.3 Théorème de Pfeffer.

Théorème 5.3 (Théorème de Pfeffer)**.** *Pour toute théorie d'intégrale bi-dimensionnelle, il existe une fonction $h \ : \ \mathbb{R}^2 \longrightarrow \mathbb{R}$ et un pavé $A = I \times J \subset \mathbb{R}^2$ où I et J sont des intervalles de $\mathbb{R}$ tels que*

(i) *il existe un champ de vecteur différentiable* $\mathrm{v} \ : \ A \subset \mathbb{R}^2 \longrightarrow \mathbb{R}^2$ *tel que* $h = \operatorname{div} \mathrm{v}$ *;*

(ii) *pour toute théorie de l'intégrale uni-dimensionnelle, l'ensemble*

$$\{x \in I \ : \ h(x, \cdot) \text{ est intégrable sur } J\}$$

est 1*-négligeable.*

Démonstration. Construisons dans un premier temps un champ de vecteurs différentiable.

On va travailler avec $J := [0,1]$. Posons

— Pour tout $k \in \mathbb{N}$, $J_k := \left[\frac{1}{2^{k+1}}, \frac{1}{2^k}\right]$;

— Pour tout $k \in \mathbb{N}$, une fonction indéfiniment dérivable

$$\begin{aligned}
g_k \ : \ \mathbb{R} &\longrightarrow J = [0,1] \\
t &\longmapsto \begin{cases} 0 & \text{si } t \leq \frac{4}{3}\frac{1}{2^{k+1}}, \\ 1 & \text{si } t > \frac{5}{3}\frac{1}{2^{k+1}}; \end{cases}
\end{aligned}$$

— Une fonction

$$\begin{aligned}
f \ : \quad \mathbb{R}^2 \quad &\longrightarrow \mathbb{R} \\
(x,y) \quad &\longmapsto \begin{cases} y^2 \sin x & \text{si } y \geq 1, \\ y^2 \left[g_k(y) \sin(8^k x) + (1 - g_k(y)) \sin(8^{k+1} x) \right] & \text{si } y \in J_k, \ k \in \mathbb{N}, \\ 0 & \text{si } y \leq 0. \end{cases}
\end{aligned}$$

Par construction, f est différentiable sur $\mathbb{R}^2 \backslash (\mathbb{R} \times \{0\})$. Montrons que pour tout $a \in \mathbb{R}$, f est différentiable en $(a,0)$. Soit $(x,y) \in \mathbb{R}^2$ tel que $|(x,y)|_\infty < 1$.

— Supposons qu'il existe $k \in \mathbb{N}$ tel que $y \in J_k$. Alors, on a

$$\begin{aligned}
\frac{|f(a+x, 0+y) - f(a,0)|}{|(x,y)|_\infty} &= \frac{y^2 |g_k(y) \sin(8^k(a+x)) + (1 - g_k(y)) \sin(8^{k+1}(a+x))|}{\max(|x|,|y|)}, \\
&\leq \frac{\left(\max(|x|,|y|)\right)^2 \left[|g_k(y)| + |1 - g_k(y)| \right]}{\max(|x|,|y|)}, \\
&\leq \max(|x|,|y|) \left[g_k(y) + 1 - g_k(y) \right], \\
&\leq |(x,y)|_\infty.
\end{aligned}$$

— Supposons que $y \leq 0$. Alors $f(a+x,y)=0$. Or $f(a,0)=0$ donc

$$\frac{|f(a+x,0+y)-f(a,0)|}{|(x,y)|_\infty} \leq |(x,y)|_\infty.$$

En faisant tendre $|(x,y)|_\infty$ vers 0, il s'ensuit que f est différentiable en $(a,0)$, de différentielle nulle. Ainsi, f est différentiable sur $\mathbb{R}^2$.

On pose donc

$$\begin{array}{rccc} \mathrm{v} := (f,0) & : & \mathbb{R}^2 & \longrightarrow \mathbb{R}^2 \\ & & (x,y) & \longmapsto (f(x,y),0). \end{array}$$

v est alors un champs de vecteurs différentiable sur $A=[0,2\pi]\times[0,1]$. Posons

$$h := \operatorname{div} \mathrm{v} = \frac{\partial f}{\partial x}.$$

De plus, étant donnée une théorie d'intégrale uni-dimensionnelle sur $[0,1]$, on pose

$$S := \{x \in [0,2\pi] \ : \ h(x,\cdot) \text{ est intégrable sur } [0,1]\}.$$

On veut montrer que S est 1-négligeable. Fixons $x \in S$. Alors par définition $h(x,\cdot)$ est intégrable sur $[0,1]$. On a donc

$$\begin{aligned} \int_{[0,1]} h(x,\cdot) &= \lim_{N\to\infty} \int_{[\frac{1}{2^{N+1}},1]} h(x,\cdot), \\ &= \lim_{N\to\infty} \sum_{k=0}^{N} \int_{J_k} h(x,\cdot), \\ &= \sum_{k=0}^{\infty} c_k(x). \end{aligned}$$

où pour tout $k \in \mathbb{N}$,

$$c_k(x) := \int_{J_k} h(x,\cdot) = 8^k \cos(8^k x) \int_{J_k} y^2 g_k(y)\mathrm{d}y + 8^{k+1}\cos(8^{k+1}x) \int_{J_k} y^2\left[1-g_k(y)\right]\mathrm{d}y.$$

Donc la série $\sum_{k\in\mathbb{N}} c_k(x)$ est convergente. Il reste donc à montrer le lemme suivant :

Lemme 5.4. *Il existe un ensemble 1-négligeable $N \subset [0,2\pi]$ tel que pour tout $\alpha \in [0,2\pi]\backslash N$, la série $\sum_{k\in\mathbb{N}} c_k(\alpha)$ est divergente.*

Démonstration du Lemme. Par la Proposition 5.2 et par la Remarque 5.1, il existe une partie $N \subset [0,2\pi]$ 1-négligeable telle que, pour tout $\alpha \in [0,2\pi]\backslash N$, la suite $\left((8^k\alpha) \mod 2\pi\right)_{k\in\mathbb{N}}$ est dense dans $[0,2\pi]$. Soit $\alpha \in [0,2\pi]\backslash N$. Il existe donc une suite croissante $(k_i)_{i\in\mathbb{N}}$ d'entiers telle que pour tout $i \in \mathbb{N}$,

$$0 \leq (8^{k_i}\alpha) \mod 2\pi \leq \frac{\pi}{24}.$$

On a donc pour tout $i \in \mathbb{N}$,

$$0 \leq \max\{(8^{k_i}\alpha) \mod 2\pi, (8^{k_i+1}\alpha) \mod 2\pi\} \leq 8\cdot\frac{\pi}{24} = \frac{\pi}{3},$$

et donc $\min\{\cos(8^{k_i}\alpha), \cos(8^{k_i+1}\alpha)\} \geq \frac{1}{2}$. Par définition de $c_{k_i}(\alpha)$, on a ainsi

$$\begin{aligned} c_{k_i}(\alpha) &= 8^{k_i}\cos(8^{k_i}\alpha)\int_{J_{k_i}} y^2 g_{k_i}(y)\mathrm{d}y + 8^{k_i+1}\cos(8^{k_i+1}\alpha)\int_{J_{k_i}} y^2\left[1-g_{k_i}(y)\right]\mathrm{d}y, \\ &\geq \frac{8^{k_i}}{2}\int_{J_{k_i}} y^2 g_{k_i}(y)\mathrm{d}y + \frac{8^{k_i+1}}{2}\int_{J_{k_i}} y^2\left[1-g_{k_i}(y)\right]\mathrm{d}y, \\ &\geq \frac{8^{k_i}}{2}\int_{J_{k_i}} y^2\left[g_{k_i}(y)+1-g_{k_i}(y)\right]\mathrm{d}y, \\ &= \frac{8^{k_i}}{2}\int_{J_{k_i}} y^2\mathrm{d}y, \\ &\geq \frac{8^{k_i}}{2}\left(\frac{1}{2^{k_i+1}}\right)^2 l(J_{k_i}) = \frac{8^{k_i}}{2}\frac{1}{2^{2k_i+2}}\left(\frac{1}{2^{k_i}}-\frac{1}{2^{k_i+1}}\right) = \frac{1}{16}. \end{aligned}$$

Donc la suite $(c_k(\alpha))_{k\in\mathbb{N}}$ ne peut pas converger vers 0. D'où la divergence de la série $\sum_{k\in\mathbb{N}} c_k(\alpha)$. □

On observe alors que $S \subset N$ et donc que, par conséquent, S est 1-négligeable. La preuve du théorème est complète. □

La fonction construite dans le Théorème 5.3 permet de montrer que les conditions (D) - i.e. toute divergence d'un champ de vecteurs différentiable est intégrable - et (F) - condition préalable à l'obtention d'un théorème de Fubini - sont incompatibles pour une théorie d'intégrable bi-dimensionnelle donnée.

Supposons en effet qu'une théorie d'intégrale bi-dimensionnelle vérifie la condition (D) ; il est clair, alors, que la fonction h construite dans le Théorème 5.3, est intégrable sur A pour cette théorie d'intégrale bi-dimensionnelle ; en revanche, la condition (ii) du même Théorème 5.3 montre que la condition de Fubini (F) n'a aucune chance d'être vérifiée.

Si, au contraire, on suppose qu'une théorie d'intégrale bi-dimensionnelle vérifie la condition de Fubini (F), alors il est exclu, par la condition (ii) du Théorème 5.3 que la fonction h que l'on construit, soit intégrable sur A pour cette théorie d'intégrale bi-dimensionnelle.

On retiendra de ceci qu'une théorie d'intégrale bi-dimensionnelle permettant d'obtenir une version suffisamment générale du théorème de la divergence ; assurant l'intégrabilité sur un rectangle de la divergence de tout champ de vecteurs différentiable sur ce rectangle, est incompatible avec l'obtention d'un énoncé général de type Fubini pour cette même intégrale bi-dimensionnelle.

Références

[1] Clément Kesselmark, Laurent Moonens, *Les théorèmes fondamentaux du calcul intégral.* Gazette des mathématiciens, n° 141, juillet 2014.

[2] Jean Mawhin, *Analyse. Fondements, techniques, évolution.* Accès Sciences, De Boeck Université, Bruxelles, 1997.

[3] L. Kuipers, H. Niederreiter, *Uniform distribution of sequences.* John Wiley & Sons, Southern Illinois University, 1974.

4 Projet de recherche de M2 : Application du Stochastic Block Model à des réseaux de gènes.

Université Paris-Sud

Master 2 : Mathématiques et Sciences du Vivant

Rapport de projet

RESEAUX DE GENES ET STOCHASTIC BLOCK MODEL

Etudiantes :
Julie Hémont
Perrine Lacroix

Encadrants :
Marie-Laure Martin-Magniette
Etienne Delannoy

Table des matières

Table des figures

Liste des tableaux

Remerciements

Nos premières pensées se tournent naturellement vers les deux encadrants de ce projet Marie-Laure Martin-Magniette et Etienne Delannoy. Merci pour la visite du laboratoire où nous avons pu observer les expériences sur l'Arabidopsis Thaliana qui étaient en cours et merci également pour les explications des machines qui étaient d'ailleurs très impressionnantes. Un grand merci général pour nous avoir laissé cette chance de participer à ce projet et pour votre confiance que vous n'avez pas hésité à mentionner.

Un grand merci à Marie-Laure Martin-Magniette pour son investissement professionnel et personnel. Nous sommes conscientes du temps qu'elle a pris lors de nos rendez-vous mais aussi pour les nombreux échanges de mails ou encore lors de l'élaboration de ce mémoire. Merci d'avoir pris le temps de réfléchir et d'apporter des réponses aux questions que nous nous sommes posées tout au long de ce projet. Nous lui sommes reconnaissantes d'avoir découvert et compris des modèles qui nous étaient jusqu'alors encore inconnus et de ne pas avoir hésité à nous donner des références pour comprendre d'un point de vue théorique les objets que l'on manipulait. Un énorme merci pour l'énergie consacrée dans la découverte du logiciel R dont nous nous étions encore jamais servies auparavant. Nous avons pu construire nos premiers codes et nous y avons appris de nombreux astuces de langage grâce à elle même si nous sommes conscientes qu'il y a encore beaucoup d'amélioration à faire. Enfin, nous souhaitons mentionner sa bonne humeur et son investissement dans nos futurs projets tant professionnels que personnels. Nous retiendrons le sentiment de bien-être que nous ressentions en franchissant les portes de l'IPS2.

Nous avons eu la chance d'avoir le biologiste Etienne Delannoy à nos cotés pendant ce projet. Nous le remercions de nous avoir fait découvrir le contexte biologique concerné, d'avoir pris le temps de répondre à nos questions, même les plus banales et d'avoir été présent lors de nos discussions sur les réponses à apporter grâce aux modèles mathématiques. Nous avons beaucoup appris à ses côtés sur la façon d'aborder la génétique dans les mathématiques. Nous le remercions également pour sa confiance et sa considération.

Nous remercions plus généralement l'Institut des Plantes et des Sciences de Paris Saclay pour leur accueil et leur bienveillance.

Nous souhaitons remercier Pierre Latouche qui a montré son intérêt sur le travail qui était en train d'être effectué. Merci d'avoir répondu présent lorsque nous avons souhaité vérifier que la sélection de modèle que nous avions obtenue avec le logiciel Linkage était la même que celle rendue par le STBM non limité. Nous espérons que ce mémoire lui sera utile s'il souhaite adapter son modèle dans ce contexte génétique précis.

Enfin, nous apportons nos remerciements à Christophe Giraud, enseignant encadrant des projets et à Sylvie Méléard, responsable de la formation Mathématiques pour les Sciences du Vivant. Merci de nous avoir encadré et d'avoir montré votre intérêt tout au long de l'année. Plus généralement, merci à l'université Paris-Sud, à l'école Polytechnique et à la formation MathsSV. Pour finir, nous souhaitons mentionner Estelle Kuhn et Stéphane Robin dont leurs enseignements étaient proches de ce que nous avons découvert dans ce projet. Merci de ne pas avoir hésité à répondre à nos questions en dehors des cours et d'avoir approfondi vos cours afin d'améliorer notre compréhension sur le sujet.

1. Contexte biologique

1.1. Motivations biologiques

L'annotation fonctionnelle chez *Arabidopsis* est encore à améliorer. Seuls 16% des gènes ont une fonction validée expérimentalement et il reste encore 20% de gènes sans aucune information fonctionnelle (Zaag et al, 2015, [6]). Les technologies haut-débit permettent de mesurer la transcription des gènes dans de nombreuses conditions et les analyses de co-expression sont des approches de plus en plus utilisées pour essayer d'identifier la fonction des gènes. Le principe général d'une analyse de co-expression est de déterminer des groupes de gènes qui s'expriment de manière similaire dans un grand nombre d'expériences. Ainsi, si un gène sans aucune information fonctionnelle est classé avec d'autres gènes bien caractérisés, alors il sera possible de lui prédire une fonction.

Dans la publication Zaag et al (2015) [6], une grande analyse de co-expression chez *Arabidopsis thaliana* a été réalisé sur 18 catégories de stress (figure 1) : des stress biotiques comme la présence de bactéries, de champignons, et des stress abiotiques tels que la température ou encore l'azote. Cette analyse a permis entre autres pour chaque type de stress d'obtenir un tableau où pour chaque gène est indiqué le numéro du cluster dans lequel il est classé.

Afin de savoir si la réponse de la plante est la même pour tous les stress, une analyse intégrant les résultats de 18 analyses de clustering est en cours à l'Institut des Sciences des Plantes de Paris-Saclay (IPS2). C'est dans ce contexte que nous avons réalisé notre projet, encadré par Marie-Laure Martin-Magniette, responsable de l'équipe "Génomique Networks" qui développe des approches bioinformatiques et statistiques pour l'analyse des données transcriptomiques et par Etienne Delannoy de l'équipe "Orgenllar Gene Expression", qui étudie le rôle du chloroplaste et de la mitochondrie chez la plante.

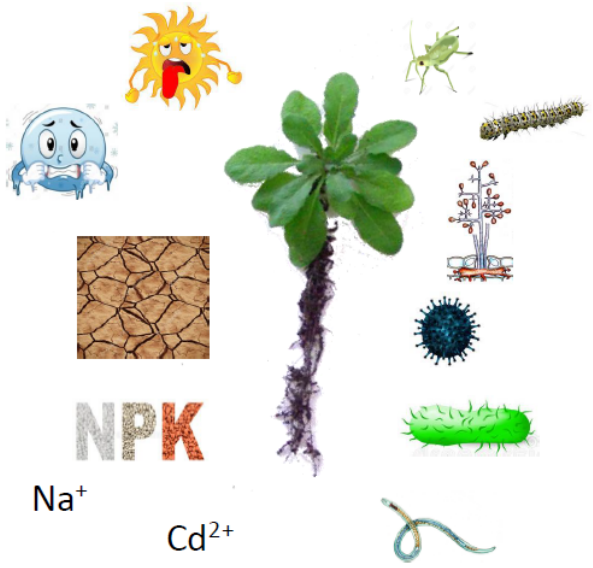


FIGURE 1 – Les différentes situations de stress subies par la plante.

Le point de départ dans notre projet était un réseau de co-expression de 4475 gènes et 56487 interactions de co-expression. Il a été construit à partir des résultats de la publication de Zaag et al. (2015), [6]. Les sommets sont les gènes et une arête entre deux gènes indique que ces 2 gènes ont été observés co-exprimés dans au moins 3 catégories de stress différentes. Les noms de ces catégories de stress sont conservées dans un label décrivant l'arête. Ce graphe est représenté sur la figure 2.

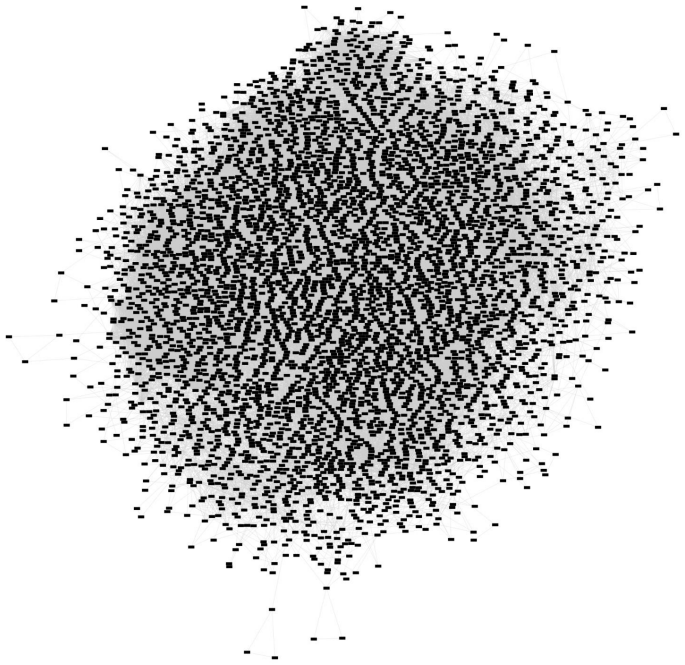


FIGURE 2 – Réseau de gènes à notre disposition

La densité d'un graphe est définie par le rapport entre le nombre d'arêtes divisé par le nombre d'arêtes possibles. La densité du réseau est de 0.006, ce qui signifie que le graphe a peu d'arêtes. La transitivité d'un graphe est définie comme la probabilité, sachant qu'on observe déjà deux arêtes entre trois sommets, d'en avoir une troisième. Pour le réseau de co-expression la transitivité est égale à 0.54. En d'autres termes, il y a peu d'arêtes sur le graphe mais lorsqu'on en a deux qui partent d'un même gène, on a de grandes chances pour former un triangle.

Ainsi, il est valide de faire l'hypothèse qu'il existe une structure cachée, qu'il faut alors expliciter. Une étude a été réalisée avant notre arrivée pour étudier la topologie de ce réseau des 4475 gènes (figure 2). Ceci a été réalisé sans tenir compte des labels des arêtes à l'aide d'un Stochastic Block Model (SBM). Le SBM diffère des algorithmes de classification modulaire qui tendent à regrouper dans un cluster les sommets qui communiquent beaucoup entre eux. Le SBM, lui, réalise une classification aboutissant à l'obtention de communautés où au sein d'une d'entre elles, les liens des sommets sont similaires à l'intérieur de la communauté et aussi avec les autres restantes. Pour différencier ces deux types d'algorithmes, nous pouvons remarquer le SBM accorde plus d'importance à

la nature des connexions et va donc regrouper dans une même communauté les noeuds ayant des topologies similaires au sein du réseau. Et c'est ce qui est intéressant dans notre projet ; on cherche à estimer les fonctions de gènes encore inconnues en utilisant la co-expression. Une attention particulière est donc portée sur la nature des connexions entre les gènes : ceux ayant un même comportement vis-à-vis des autres gènes vont avoir une fonction similaire.

Cette première analyse avec un SBM a identifié 52 communautés dont 43 communautés stables contenant 2674 gènes (figure 3). Au sein de certaines communautés, les gènes dont les fonctions sont déjà connues sont en fait impliqués dans un même processus biologique. Il est naturel de conjecturer que les gènes de ces communautés dont la fonction est encore indéterminée exercent la même fonction que cette majorité.

1.2. Problématique

Ce premier modèle donne des résultats intéressants au sens où il permet aux biologistes d'extraire de la connaissance sur des gènes encore mal annotés et d'identifier des groupes de gènes interagissant dans un même processus biologique. Cependant, cette première analyse n'exploite pas toute l'information disponible puisque que le label des arêtes n'était pas pris en compte.

L'objectif de notre projet était de reconsidérer la modélisation du réseau de co-expression en incluant le label des arêtes pour savoir comment cette information change les communautés identifiées et si cela serait plus pertinent pour extraire de l'information biologique. Statistiquement, la question est de savoir si la connaissance des stress dans lesquels une paire de gènes est co-exprimée apporte de l'information sur la topologie du réseau.

Pour répondre à cette question, nous nous sommes intéressées au Stochastic Block Model (SBM) qui étudie la topologie d'un graphe sans tenir compte de l'existence de label sur les arêtes et au Stochastic Topic Block Model (STBM) qui prend en compte le label des arêtes pour étudier la topologie. Au lieu de comparer ces 2 modèles sur le réseau des 4475 gènes, nous nous sommes focalisées sur un sous-réseau de 143 gènes, indiqué en jaune sur la figure 3, contenant suffisamment de variabilité dans le label des arêtes pour expliquer les apports d'information de la modélisation du texte sur les arêtes.

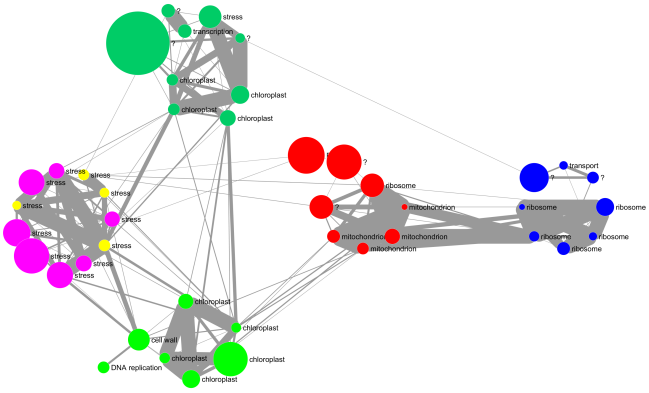


FIGURE 3 – Représentation du réseau analysé avec un SBM où les communautés ayant la même annotation fonctionnelle sont rapprochées. En jaune sont représentées les communautés de gènes sélectionnées pour appliquer nos comparaisons SBM-STBM. Au total, l'analyse portera sur les 143 gènes appartenant à ces 4 communautés.

2. Les modèles mathématiques

Dans cette section, nous allons introduire les modèles mathématiques ainsi que les notations dont nous aurons besoin. Commençons par introduire le Stochastic Block Model (SBM) qui est un modèle de clustering de graphe, qui n'utilise que la présence/absence des arêtes, puis nous présenterons le Stochastic Topic Block Model (STBM) qui tient compte du label des arêtes.

2.1. Le Stochastic Block Models

Le SBM est une modèle mathématique qui a pour but de synthétiser un graphe en créant des groupes de noeuds ayant le même comportement dans le graphe. Le résultat obtenu après estimation des paramètres d'un tel modèle est un graphe simplifié où les noeuds sont des groupes de noeuds du réseau initial et les arêtes sont pondérées : l'information portée par une arête entre le cluster i et le cluster j est la probabilité qu'il y ait une arête entre un individu du cluster i et un individu du cluster j, comme décrit sur la figure 4. Précisons ce modèle en introduisant proprement les notations.

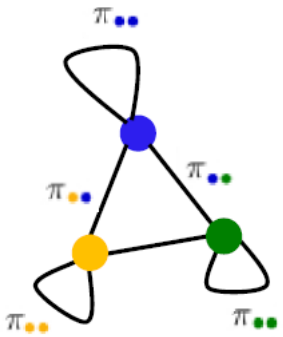


FIGURE 4 – Exemple de graphe synthétisé obtenu avec un SBM. Il y a trois communautés et les arêtes indiquent la probabilité de connexion entre chaque paire de communautés.

2.1.1. Modèle du SBM

On note M le nombre de sommets dans le graphe et A sa matrice d'adjacence. La matrice A est donc symétrique de taille $M \times M$ à coefficients dans $\{0, 1\}$ telle que pour tout couple de sommets $(i, j) \in \{1, \cdots, M\}^2$, $A_{i,j} = 1$ si et seulement si une arête est présente entre le noeud i et le noeud j.

Nous supposons qu'il existe une structure cachée dans ce graphe, c'est-à-dire qu'il existe des clusters dans ce graphe. Nous supposons que chaque sommet appartient à un cluster et à un seul et nous notons Q ce nombre de clusters. Ainsi, chaque sommet est caractérisé par son appartenance à un cluster : à chaque sommet $i \in \{1, \cdots, M\}$ on associe un vecteur latent Y_i de taille Q où $Y_{i,q} = 1$ si le sommet i est dans le cluster q et $Y_{i,q} = 0$ sinon. Nous pouvons interpréter cette classification de la manière suivante : si deux noeuds sont dans le même cluster, c'est qu'ils ont un caractère commun. Nous supposons de plus que les $(Y_i)_{1 \leq i \leq M}$ sont indépendants.

Nous construisons un modèle hiérarchique en commençant par définir la loi de $(Y_i)_{1\leq i\leq M}$. Supposons que pour tout noeud i et pour tout cluster q, la probabilité que $Y_{i,q} = 1$ vaut ρ_q. On a alors $\sum_{q=1}^{Q} \rho_q = 1$ et pour tout noeud i, $\sum_{q=1}^{Q} Y_{i,q} = 1$. Autrement dit, nous avons posé

$$\mathcal{L}(Y_i) = \mathcal{M}(1, \rho = (\rho_1, ..., \rho_Q))$$

Nous posons également une loi binomiale sur chaque coefficient de la matrice d'adjacence conditionnellement à l'appartenance des noeuds aux clusters du graphe :

$$\mathcal{L}\left(A_{i,j} | Y_{i,q}Y_{j,r} = 1\right) = \mathcal{B}(\pi_{q,r})$$

où $\pi = (\pi_{q,r})_{(q,r)\in\{1,...,Q\}^2}$ est une matrice symétrique telle que $\pi_{q,r}$ est la probabilité qu'il y ait une arête entre un noeud du cluster q et un noeud du cluster r. Ce sont les paramètres ρ et π qui caractérisent le modèle SBM, récapitulé dans le tableau 1.

Notation	**Objet**	**État**	**Lois**
M	nombre de noeuds	observé	déterministe
Q	nombre de clusters	inconnu	déterministe
Y	vecteur d'appartenance aux clusters	variable cachée	$Y_i \sim \mathcal{M}(1, \rho)$ où $\rho = (\rho_q)_{1\leq q\leq Q}$ est inconnu.
A	matrice d'adjacence	observée	Conditionnellement à $Y_{i,q}Y_{j,r} = 1$, $A_{i,j} \sim \mathcal{B}(\pi_{q,r})$ où $\pi = (\pi_{q,r})_{1\leq q,r\leq Q}$ symétrique est inconnue.

TABLE 1 – SBM : notations et lois.

Les inconnues du modèle sont π, ρ et Q. On crée une collection de modèles $\mathcal{M} = \{\text{SBM comprenant } Q \text{ clusters}\}_{Q\in\mathbb{N}}$. Nous travaillerons en deux étapes :

Étape 1. Estimation algorithmique de $\Theta = (\pi, \rho)$ en fixant Q ;

Étape 2. Sélection de modèle sur $\mathcal{M}$ pour sélectionner Q.

2.1.2. Estimation des paramètres

Dans cette section, nous supposons le nombre de clusters Q connu. Nous traiterons l'hypothèse Q inconnu dans la section suivante.

On va chercher un estimateur par maximum de vraisemblance. Calculons donc la vraisemblance de la matrice A sachant les paramètres Θ. Par indépen-

dance des coefficients de la matrice d'adjacence et par indépendance de l'appartenance des noeuds aux clusters :

$$\begin{aligned} p(A|\Theta) &= \prod_{1\leq i<j\leq M} p(A_{i,j}|\Theta) \\ &= \prod_{1\leq i<j\leq M} \sum_{q,r=1}^{Q} \mathbb{P}(Y_{i,q}Y_{j,r}=1) p(A_{i,j}|Y_{i,q}Y_{j,r}=1;\Theta) \\ &= \prod_{1\leq i<j\leq M} \sum_{q,r=1}^{Q} \rho_q\rho_r \left[{\pi_{q,r}}^{A_{i,j}} \left(1-\pi_{q,r}\right)^{1-A_{i,j}} \right]. \end{aligned}$$

Dans l'optique de la maximiser, regardons la log-vraisemblance :

$$\log p(A|\Theta) = \sum_{1\leq i<j\leq M} \log \sum_{q,r=1}^{Q} \rho_q\rho_r \left[{\pi_{q,r}}^{A_{i,j}} \left(1-\pi_{q,r}\right)^{1-A_{i,j}} \right].$$

On remarque que cette quantité n'a pas de maximum explicite puisqu'on ne sait pas travailler avec le logarithme d'une somme. On va donc s'appuyer sur un algorithme de type EM. L'idée de cet algorithme est de travailler sur l'espérance par rapport aux données de la vraisemblance complète du modèle. En utilisant la formule de Bayes, la vraisemblance complète du modèle s'écrit :

$$p(A,Y;\rho,\pi) = p(A|Y;\pi)p(Y;\rho). \tag{1}$$

D'une part, par indépendance des Y_i et de leur loi :

$$p(Y;\rho) = \prod_{i=1}^{M} p(Y_i;\rho).$$

Donc

$$\log p(Y;\rho) = \sum_{i=1}^{M}\sum_{q=1}^{Q} Y_{i,q}\log(\rho_q). \tag{2}$$

D'autre part, par l'indépendance des coefficients de la matrice d'adjacence conditionnellement aux variables latentes Y, la symétrie de la matrice π et la loi de A sachant Y, nous obtenons :

$$p(A|Y;\pi) = \prod_{i=1}^{M}\prod_{i<j}^{M}\prod_{q,r=1}^{Q} \left({\pi_{q,r}}^{A_{i,j}} (1-\pi_{q,r})^{1-A_{i,j}} \right)^{Y_{i,q}Y_{j,r}}.$$

D'où,

$$p(A|Y;\pi) = \prod_{i<j}^{M}\prod_{q,r=1}^{Q} \left({\pi_{q,r}}^{A_{i,j}} (1-\pi_{q,r})^{1-A_{i,j}} \right)^{Y_{i,q}Y_{j,r}}.$$

Ainsi,

$$\log p(A|Y;\pi) = \sum_{1\leq i<j\leq M} \sum_{q,r=1}^{Q} Y_{i,q}Y_{j,r} \log\left({\pi_{q,r}}^{A_{i,j}} (1-\pi_{q,r})^{1-A_{i,j}} \right). \tag{3}$$

Finalement, par (2) et (3) dans le log de (1) et par symétrie des lois en i et j,

$$\log p(A, Y; \rho, \pi) = \sum_{i=1}^{M} \sum_{q=1}^{Q} Y_{i,q} \log(\rho_q) + \frac{1}{2} \sum_{i,j=1,\ i \neq j}^{M} \sum_{q,r=1}^{Q} Y_{i,q} Y_{j,r} \log \left[{\pi_{q,r}}^{A_{i,j}} (1 - \pi_{q,r})^{1 - A_{i,j}} \right]$$

Dans l'étape (E) de l'algorithme EM, on a besoin d'avoir accès à la loi jointe de A et Y conditionnellement aux observations pour en prendre l'espérance. Dans notre modèle, nous n'avons pas accès à la loi de $Y_{i,q}$ et de $Y_{i,q}Y_{j,r}$ sachant la matrice d'adjacence A. Il nous faut introduire une nouvelle stratégie qui consiste à approcher la vraisemblance au lieu de la calculer, c'est l'idée de l'algorithme EM Variationnel (VEM), en supposant une hypothèse supplémentaire sur les lois des $(Y_i)_{1 \leq i \leq M}$ sachant A de manière à savoir faire les calculs.

Pour toute distribution q sur les variables cachées Y, on a

$$\log p(A; \rho, \pi) = \mathcal{L}(q(\cdot); \rho, \pi) + \mathcal{KL}\left(q(\cdot) \| p(\cdot | A; \rho, \pi)\right)$$

où

$$\mathcal{L}(q(\cdot); \rho, \pi) = \mathbb{E}\left[\log\left(\frac{p(A, Y; \rho, \pi)}{q(Y)}\right)\right] = \sum_{Y} q(Y) \log\left(\frac{p(A, Y; \rho, \pi)}{q(Y)}\right)$$

et

$$\mathcal{KL}(q(\cdot) \| p(\cdot | A; \rho, \pi))) = \mathbb{E}\left[\log\left(\frac{p(Y | A; \rho, \pi)}{q(Y)}\right)\right] = -\sum_{Y} q(Y) \log\left(\frac{p(Y | A; \rho, \pi)}{q(Y)}\right).$$

Remarquons que l'on a l'inégalité

$$\log p(A; \rho, \pi) \geq \log p(A; \rho, \pi) - \mathcal{KL}[q(\cdot) \| p(\cdot | A; \rho, \pi)]$$

et donc

$$\log p(A; \rho, \pi) \geq \mathcal{L}(q(\cdot); \rho, \pi),$$

avec égalité si et seulement si $q(Y) = p(Y | A; \rho, \pi)$.

On change donc de cible et on cherche maintenant à maximiser $\mathcal{L}(q(\cdot); \rho, \pi)$, en cherchant q au plus proche de $p(\cdot | A; \rho, \pi)$ mais que l'on ne l'atteint pas. On cherche donc à minimiser la Kullback qui est nécessairement strictement positive dès lors qu'on n'a pas la vraisemblance exacte.

La stratégie est d'approcher la distribution de Y sachant A par champs moyen. Supposons que la distribution q est factorisable et s'écrit

$$q(Y) = \prod_{i=1}^{M} q_i(Y_i) = \prod_{i=1}^{M} \prod_{q=1}^{Q} {\tau_{i,q}}^{Y_{i,q}}.$$

où $\tau_{i,q}$ est un paramètre qui modélise la probabilité que la noeud i appartienne au cluster q sachant les observations A. Cette hypothèse peut-être interprétée comme de l'indépendance des assignements des noeuds aux clusters conditionnellement aux observations.

En écrivant ceci, nous allons réduire nos calculs à une famille ne contenant pas la probabilité désirée $p(\cdot|A,\rho,\pi)$. L'erreur d'approximation que l'on fait est la distance de Kullback.

L'objectif est donc double : il faut à la fois maximiser $\mathcal{L}(q(\cdot);\rho,\pi)$ par rapport à θ à distribution q fixée et minimiser la Kullback entre q et $p(\cdot|A,\rho,\pi)$ par rapport à q. A l'itération k, on va donc prendre la distribution q dans l'ensemble

$$\mathcal{Q} = \left\{ q \ ; \ q(Y) = \prod_{i=1}^{M} q_i(Y_i) = \prod_{i=1}^{M} \prod_{q=1}^{Q} \tau_{i,q}{}^{Y_{i,q}} \right\}$$

qui maximise $\mathcal{L}(q(\cdot);\theta^{(k)})$ puis on va maximiser par rapport à θ.

L'algorithme d'inférence VEM à l'itération (h+1) est donc :

(VE)

$$\hat{q} = \arg\max_{q\in\mathcal{Q}} \mathcal{L}\left(q(\cdot);\rho^{(h)},\pi^{(h)}\right) = \arg\min_{q\in\mathcal{Q}} \mathcal{KL}\left(q(\cdot)\|p\left(\cdot|A;\rho^{(h)},\pi^{(h)}\right)\right),$$

avec $\hat{q}$ définie par les $\hat{\tau}_{i,q}$, définis eux-mêmes par

$$\hat{\tau}_{i,q} = \rho_q^{(h)} \prod_{i<j} \prod_{r=1}^{Q} p(A_{i,j};\pi_{q,r}^{(h)})^{\hat{\tau}_{j,r}} \tag{4}$$

(VM)

$$\left(\rho^{(h+1)},\pi^{(h+1)}\right) = \arg\max_{(\rho,\pi)} \mathcal{L}\left(\hat{q}(\cdot);\rho,\pi\right).$$

2.1.3. Sélection de modèle

Pour l'instant, on a travaillé à Q connu. Mais en réalité, lorsqu'on observe un graphe, on ne le connaît pas.

Notons $\mathcal{M}$ la collection des modèles des SBM avec Q clusters, $Q \in \mathbb{N}$. On cherche le modèle $m \in \mathcal{M}$ tel que $\mathbb{P}(m|A)$ est maximale. Cette probabilité est proportionnelle à $\mathbb{P}(A|m)$ sous l'hypothèse que la loi sur $\mathcal{M}$ est uniforme (ce qui est notre cas ici car on ne favorise pas un modèle par rapport à un autre). On considère donc que les modèles de la collection $\mathcal{M}$ sont équiprobables. On maximise

$$m_{opt} = \arg\max_{m\in\mathcal{M}} \mathbb{P}(A|m)$$

.

Cependant, $\mathbb{P}(A|m)$ est difficile à calculer et on utilise une approximation qui aboutit à choisir le modèle suivant le critère BIC et le modèle choisi est :

$$m_{BIC} = \arg\min_{m\in\mathcal{M}} \ \log\mathbb{P}(A|m,\hat{\pi},\hat{\rho}) - pen(m) = \arg\min_{m\in\mathcal{M}} BIC(m)$$

où $pen(m) = \frac{\nu_m}{2}\log(M)$ avec ν_m le nombre de paramètres du modèle m. Dans notre cas, ν_m est presque proportionnel à Q.

L'objectif est ici de classer les noeuds du graphe dans des clusters. Or, BIC trouve la distribution la plus "lisse", la plus proche de la vraie densité mais ne permet pas toujours de bien trouver les clusters. On introduit donc le critère ICL. Ce critère permet d'intégrer un terme d'entropie, qui quantifie le désordre.

$$ICL(m) = BIC(m) - H(m)$$

où $H(m) = -\sum_i \sum_k \tau_{i,k} \log(\tau_{i,k})$ est un terme d'entropie qui pénalise en prenant en compte les probabilités conditionnelles. On remarque que si on classe de façon certaine tous les noeuds, c'est à dire que $\forall i, \exists k$ tel que $\tau_{i,k} = 1$, alors le terme d'entropie vaut 0. On recherche donc

$$m_{ICL} = \underset{m \in \mathcal{M}}{\arg\min}\ ICL(m).$$

L'avantage de ce critère ICL est qu'on classe de façon plus certaine les noeuds dans les clusters. Remarquons que ce critère a tendance à renvoyer un nombre plus petit de clusters que le critère BIC.

On renvoie à Daudin et al. (2008),[2], pour l'expression exacte du critère ICL du SBM.

2.1.4. Classification

Une fois que nous avons trouvé le nombre de clusters Q et estimé Θ, nous pouvons classer les noeuds dans les clusters par la règle du maximum a posteriori (MAP). Cela consiste à calculer la probabilité conditionnelle d'appartenance définie par :

$$\tau_{i,q} = \mathbb{P}(Y_{i,q} = 1|A)$$

et de classer le noeud i dans le cluster $q*$ tel que

$$\tau_{i,q*} = \max_{q=1,\dots,Q} \tau_{i,q}.$$

Ces probabilités $\tau_{i,q}$ sont données Équation (4), que l'on a déjà utilisé à l'étape VE de l'algorithme d'inférence.

2.2. Le STBM : un modèle inspiré du SBM avec de l'information sur les arêtes.

Le STBM est une extension du SBM. Nous ajoutons au modèle existant du texte sur les arêtes. Nous avons travaillé sur un modèle introduit initialement pour modéliser des interactions sur les réseaux sociaux. Par exemple, on a une arête entre deux individus lorsqu'ils se déclarent amis et on ajoute sur les arêtes les posts partagés. Nous devons ajouter, à la modélisation d'apparitions d'arêtes, la modélisation d'apparitions de documents sur les arêtes.

Nous reprenons donc les notations déjà introduites pour le SBM. Nous considérons toujours un graphe non-orienté à M noeuds, décrit par une matrice d'adjacence A de taille $M \times M$ et symétrique. On rappelle qu'alors pour tous noeuds i et j du graphe, $A_{i,j} = 1$ s'il existe une arête entre i et j et $A_{i,j} = 0$ sinon.

2.2.1. Modèle du STBM

Nous exposons ici le modèle présenté par Bouveyron et al. (2017)[1]. Dans ce nouveau modèle, nous souhaitons intégrer une information portée par les arêtes. Chaque arête est caractérisée par un ensembles de $D_{i,j}$ documents, notés $W_{i,j} = \left(W_{i,j}^d\right)_{1 \leq d \leq D_{i,j}}$. Chaque document $d \in \{1, \cdots, D_{i,j}\}$ est constitué de $N_{i,j}^d$ mots, notés $W_{i,j}^d = \left(W_{i,j}^{d,n}\right)_{1 \leq n \leq N_{i,j}^d}$. Ainsi, $W_{i,j}^d$ est le $d^{\text{ème}}$ document présent

sur l'arête reliant i et j et $W_{i,j}^{d,n}$ est son $n^{\text{ème}}$ mot. D'autre part, nous supposons que ces mots appartiennent à un vocabulaire v de V mots.

Remarquons que, dans le cas d'un graphe non-orienté sans self-loop, la matrice A est symétrique et pour tout noeud i, $A_{i,i} = 0$. On a de plus $W_{j,i} = W_{i,j}$. Ce sera notre cas lorsque nous appliquerons ce modèle au réseau de co-expression.

Rappelons que nous avons à nouveau des variables latentes

$$Y = (Y_{i,1}, \cdots, Y_{i,Q})_{1 \leq i \leq M}$$

qui traduisent les appartenances à des clusters. On a donc pour tout noeud i et tout cluster q, $Y_{i,q} = 1_{i \in q}$.

On reprend le modèle du SBM pour modéliser la construction des arêtes du graphe. On suppose toujours que

$$Y_i \sim \mathcal{M}(1, \rho = (\rho_1, ..., \rho_Q))$$

et

$$(A_{i,j}|Y_{i,q}Y_{j,r}) \sim \mathcal{B}(\pi_{q,r}).$$

Il nous reste donc à modéliser la construction des documents présents sur les arêtes. On va faire l'hypothèse que les documents sont construits selon des sujets qui peuvent être vus comme des ensembles de mots du vocabulaire que l'on s'est donné, et ces mots sont présents dans les sujets en différentes proportions. On travaille avec un couple (i, j) pour lequel on a une arête entre i et j. Autrement dit, on a $A_{i,j} = 1$ et on a des documents portés par l'arête. Notons $Z_{i,j}^{d,n}$ le sujet du $n^{\text{ème}}$ mot du $d^{\text{ème}}$ document. On suppose qu'il y a K sujets. On a donc $Z_{i,j}^{d,n} \in \{1, \cdots, K\}$ et on tire un sujet selon les groupes auxquels les noeuds i et j appartiennent

$$\left(Z_{i,j}^{d,n}|A_{i,j}Y_{i,q}Y_{jr} = 1\right) \sim \mathcal{M}(1, \theta_{q,r}),$$

où $\theta_{q,r} = (\theta_{q,r,k})_{1 \leq k \leq K}$. Autrement dit, si on a une arête entre i et j, si le noeud i appartient au cluster q et le noeud j appartient au cluster r, alors le mot $W_{i,j}^{d,n}$ a une probabilité $\theta_{q,r,k}$ d'appartenir au sujet k.

Une fois le sujet choisi, on choisit un mot selon les proportions qui caractérisent le sujet. On pose donc pour tout sujet $k \in \{1, \cdots, K\}$,

$$\left(W_{i,j}^{d,n}|Z_{i,j}^{d,n} = k\right) \sim \mathcal{M}(1, \beta_k),$$

avec $\beta_k = (\beta_{k,v})_{1 \leq v \leq V}$. On a donc, conditionnellement au fait que le mot $W_{i,j}^{d,n}$ appartienne au sujet k, une probabilité $\beta_{k,v}$ que le mot choisi soit le mot v. On peut remarquer que cette probabilité ne dépend plus que de k. On retrouve la probabilité d'apparition du mot v sur une arête par la formule de Bayes :

$$\mathbb{P}(W_{i,j}^{d,n} = v|A_{i,j}Y_{i,q}Y_{j,r} = 1) = \sum_{k=1}^{K} \mathbb{P}(W_{i,j}^{d,n} = v, Z_{i,j}^{d,n} = k|A_{i,j}Y_{i,q}Y_{j,r} = 1).$$

Or

$$\begin{aligned}&\mathbb{P}(W_{i,j}^{d,n} = v, Z_{i,j}^{d,n} = k|A_{i,j}Y_{i,q}Y_{j,r} = 1)\\ &\quad = \mathbb{P}(W_{i,j}^{d,n} = v|A_{i,j}Y_{i,q}Y_{j,r} = 1, Z_{i,j}^{d,n} = k)\mathbb{P}(Z_{i,j}^{d,n} = k|A_{i,j}Y_{i,q}Y_{j,r} = 1).\end{aligned}$$

Donc

$$\mathbb{P}(W_{i,j}^{d,n} = v | A_{i,j} Y_{i,q} Y_{j,r} = 1) = \sum_{k=1}^{K} \theta_{q,r,k} \beta_{k,v}.$$

On fera de la sélection de modèle sur Q et K dans le modèle décrit.
On complète le tableau que l'on a réalisé pour le SBM en le tableau 2.

Notation	Objet	État	Lois
M	nombre de noeuds	observé	déterministe
Q	nombre de clusters	inconnu	déterministe
Y	vecteur d'appartenance aux clusters	variable cachée	$Y_i \sim \mathcal{M}(1, \rho)$ où $\rho = (\rho_q)_{1 \leq q \leq Q}$ est inconnu.
A	matrice d'adjacence	observée	Conditionnellement à $Y_{i,q}Y_{j,r} = 1$, $A_{i,j} \sim \mathcal{B}(\pi_{q,r})$ où $\pi = (\pi_{q,r})_{1 \leq q,r \leq Q}$ symétrique est inconnue.
V	nombre de mots de vocabulaire	observé	déterministe
K	nombre de sujets	inconnu	déterministe
Z	sujets	variable cachée	Conditionnellement à $A_{i,j}Y_{i,q}Y_{j,r} = 1$, $Z_{i,j}^{d,n} \sim \mathcal{M}(1, \theta_{q,r})$ où $\theta_{q,r} = (\theta_{q,r,k})_{1 \leq k \leq K}$ est inconnu.
$D_{i,j}$	nombre de documents sur l'arête entre i et j	observé	déterministe
$N_{i,j}^d$	nombre de mots dans le document d sur l'arête entre i et j	observé	déterministe
W	mots	observé	Conditionnellement à $A_{i,j}Y_{i,q}Y_{j,r} = 1$, $Z_{i,j}^{d,n} = k$, $W_{i,j}^{d,n} \sim \mathcal{M}(1, \beta_k)$ où $\beta_k = (\beta_{k,v})_{1 \leq v \leq V}$ est inconnu.

TABLE 2 – STBM : notations et lois.

2.2.2. Estimation des paramètres

Nous traitons ici les points clés de l'inférence du modèle du STBM. L'inférence s'appuie sur l'article de Bouveyron et al. (2017), [1]. Dans leur modèle, à chaque paire de clusters de noeuds (q, r) le vecteur des proportions

de sujets $\theta_{q,r} = (\theta_{q,r,k})_{1\leq k\leq K}$ suit une loi *a priori* de Dirichlet de paramètre $\alpha = (\alpha_k)_{1\leq k\leq K}$, et ce de manière indépendante. Dans la suite, α est fixé à $(1, \cdots, 1)$ pour obtenir la loi uniforme non informative. Comme pour l'estimation des paramètres pour le SBM, on travaille ici à Q fixé. On travaille également à K fixé. On renvoie à la partie suivante pour la sélection de ces deux paramètres. Notre but est de maximiser

$$\log p(A, W, Y|\rho, \pi, \beta) = \log \sum_Z \int_\theta p(A, W, Y, Z, \theta|\rho, \pi, \beta) d\theta.$$

Comme pour le SBM, on va vouloir appliquer un algorithme de type VEM. Nous travaillons donc sur la vraisemblance complète donnée par :

$$p(A, W, Y, Z, \theta|\rho, \pi, \beta) = \underbrace{p(W, Z, \theta|A, Y, \beta)}_{\text{Apport de l'information sur les arrêtes}} \times \underbrace{p(A, Y|\rho, \pi)}_{\text{Vraisemblance du SBM}}$$

Par la formule de Bayes, $p(W, Z, \theta|A, Y, \beta) = p(W|A, Z, \beta)p(Z|A, Y, \theta)p(\theta)$. Les variables latentes $Z_{i,j}^{d,n}$ étant indépendantes,

$$p(Z|Y, A, \theta) = \prod_{i=1}^{M} \prod_{j=1, j\neq i}^{M} \left[\prod_{d=1}^{D_{i,j}} \prod_{n=1}^{N_{i,j}^d} \prod_{q=1}^{Q} \prod_{r=1}^{Q} p(Z_{i,j}^{d,n}|\theta)^{Y_{i,q}Y_{j,r}} \right]^{A_{i,j}}.$$

Sachant Z, les mots $W_{i,j}^{d,n}$ sont supposés indépendants donc

$$p(W|A, Z, \beta) = \prod_{1\leq i\neq j\leq M} \left[\prod_{d=1}^{D_{i,j}} \prod_{n=1}^{N_{i,j}^d} p(W_{i,j}^{d,n}|Z_{i,j}^{d,n}, \beta) \right]^{A_{i,j}}.$$

En notant $Z_{i,j}^{d,n,k}$ la variable valant 1 si $Z_{i,j}^{d,n} = k$ et 0 sinon,

$$p(W|A, Z, \beta) = \prod_{1\leq i\neq j\leq M} \left[\prod_{d=1}^{D_{i,j}} \prod_{n=1}^{N_{i,j}^d} \prod_{k=1}^{K} p(W_{i,j}^{d,n}|\beta_k)^{Z_{i,j}^{d,n,k}} \right]^{A_{i,j}}.$$

La différence entre l'inférence du SBM et l'inférence du STBM est qu'on est dans une approche bayésienne. On a une loi *a priori* sur θ : pour chaque paire de clusters (q, r), $\theta_{q,r} = \left(\theta_{q,r}^k\right)_{1\leq k\leq K}$ suit une loi de Dirichlet de paramètre $\alpha = (\alpha_k)_{1\leq k\leq K}$

$$\theta_{q,r} \sim \mathcal{D}(\alpha)$$

et ce de manière indépendante. On a donc

$$p(\theta) = \prod_{q,r=1}^{Q} \mathcal{D}(\theta_{q,r}; \alpha)$$

Comme pour le SBM, on commence par décomposer la vraisemblance en faisant apparaître la Kullback. Pour toute distribution R de (Z, θ),

$$\log p(A, W, Y|\rho, \pi, \beta) = \mathcal{L}(R(\cdot); Y, \rho, \pi, \beta) + \mathcal{KL}\left(R(\cdot) \| p(\cdot|A, W, Y, \rho, \pi, \beta)\right)$$

où

$$\mathcal{L}(R(\cdot); Y, \rho, \pi, \beta) = \sum_Z \int_\theta R(Z, \theta) \log \frac{p(A, W, Y, Z, \theta; \rho, \pi, \beta)}{R(Z, \theta)} d\theta$$

et

$$\mathcal{KL}(R(\cdot) \| p(\cdot|A; \rho, \pi))) = -\sum_Z \int_\theta R(Z, \theta) \log \frac{p(Z, \theta|A, W, Y; \rho, \pi, \beta)}{R(Z, \theta)} d\theta.$$

Afin de pouvoir calculer $\mathcal{L}$, on approche la distribution de (Z, θ) sachant A par une distribution de (Z, θ), R, factorisable. On maximise donc $\mathcal{L}$ en R sur l'ensemble

$$\mathcal{R} = \left\{ R \ ; \ R(Z, \theta) = R_\theta(\theta) \prod_{i \neq j; A_{i,j}=1}^{M} \prod_{d=1}^{D_{i,j}} \prod_{n=1}^{N_{i,j}^d} R_{i,j,d,n}\left(Z_{i,j}^{d,n}\right) \right\}.$$

Notons d'autre part que l'on peut encore simplifier le calcule de $\mathcal{L}$:

$$\begin{aligned}
\mathcal{L}(R; Y, \rho, \pi, \beta) &= \sum_Z \int_\theta R(Z, \theta) \log \frac{p(A, W, Y, Z, \theta; \rho, \pi, \beta)}{R(Z, \theta)} d\theta \\
&= \sum_Z \int_\theta R(Z, \theta) \log \frac{p(A, Y; \rho, \pi) p(W, Z, \theta|A, Y; \beta)}{R(Z, \theta)} d\theta \\
&= \sum_Z \int_\theta R(Z, \theta) \log \frac{p(W, Z, \theta; |A, Y; \beta)}{R(Z, \theta)} d\theta + \underbrace{\log p(A, Y|\rho, \pi)}_{\text{log-vraisemblance du SBM}}
\end{aligned}$$

Cette décomposition permet de faire apparaître la log-vraisemblance du SBM et montre que les inférences sur β et sur (ρ, π) sont indépendantes à chaque étape. On utilisera l'algorithme présenté dans la section 2.1.2 pour maximiser $\log p(A, Y|\rho, \pi)$ et un algorithme de type VBEM (Variationnal Bayes EM) pour maximiser

$$\overline{\mathcal{L}}(R(\cdot); Y, \beta) = \sum_Z \int_\theta R(Z, \theta) \log \frac{p(W, Z, \theta; |A, Y; \beta)}{R(Z, \theta)} d\theta$$

sur $\mathcal{R}$.

On ne développera pas davantage l'inférence du modèle du STBM car elle a été implémentée sur le logiciel Linkage. On renvoie à la section 3 pour l'application de ce logiciel à un jeu de données réelles.

2.2.3. Sélection de modèle

Le critère de sélection de modèle sur le STBM est également un critère ICL mais qui porte sur deux paramètres, Q le nombre de clusters et K le nombre de sujets. On réfère à l'Appendice A.7 de Bouveyron et al.(2017),[1], pour la preuve de l'expression explicite du critère ICl pour le modèle du STBM.

3. Application aux données

Nous rappelons que les données à notre disposition sont représentées sur la figure 2 sur lequel nous disposons déjà d'une application d'un SBM (figure 3) qui permet d'apporter des connaissances sur la fonction de certains gènes, jusqu'alors inconnue. Comme expliqué dans la partie 1, nous avons voulu inclure l'information sur les arêtes. Une fois le nouveau modèle écrit (paragraphe 2.2.1), il nous restait à appliquer l'algorithme du STBM afin d'obtenir un nouveau clustering qui tient compte des labels. Ceci nous mettant dans la capacité de pouvoir comparer les deux graphes de communautés après SBM et après STBM. L'objectif étant de comprendre comment l'information sur les arêtes modifie le résultat du SBM.

Pour appliquer un STBM sur notre réseau, nous avons fait appel au logiciel en ligne **Linkage** crée par Pierre Latouche. Nous nous sommes heurtées à un premier problème. En effet, ce logiciel disponible en ligne, est en réalité une version limitée de l'algorithme STBM puisqu'il ne rend pas plus de 10 communautés. En d'autres termes, cet algorithme en ligne teste pour des nombres de communautés allant de 2 à 10 et renvoie le meilleur graphe parmi tous les clustering qu'il fait. Cependant, le SBM sur le gros réseau a rendu un graphe avec 52 communautés. Il nous semblait alors improbable que l'apport des informations sur les arêtes nous fasse passer d'un clutering à 52 communautés à un clustering à moins de 10 communautés. Ainsi, si nous testions notre gros jeu de données sur Linkage, l'interprétation et la comparaison avec le résultat du SBM n'auraient pas de sens car le maximum ne serait pas atteint.

Nous avons donc décider de travailler avec un plus petit réseau dont l'algorithme Linkage renvoyait un nombre de communautés inférieur à 10. nous certifiant que celui du STBM non simplifié trouverait ce même chiffre. Le choix du petit réseau a été fait à partir de la figure 3 : il est formé des gènes appartenant aux 4 communautés différentes représentées en jaune. Ces communautés sont assez reliées entre elles et en même temps, le fait de prendre les gènes de communautés différentes nous offrent de la variabilité. Si nous arrivions déjà à établir des différences entre les deux clustering avec l'ajout de l'information des arêtes sur notre petit réseau, nous serions optimistes quant aux conclusions pouvant être émises sur le gros réseau de co-expression de gènes. Une étude plus poussée quant à la pertinence de ce choix de réseau est décrite au paragraphe suivant.

3.1. Pertinence du choix du petit réseau

Notre petit réseau est constitué de 143 noeuds et 3894 arêtes. Grâce au **logiciel R**, nous avons pu calculer la densité de ce graphe qui est de 0.19, ainsi que sa transitivité qui, elle, est de 0.56. Notre petit réseau a donc peu d'arêtes mais la probabilité d'en avoir une troisième sachant qu'on en observe déjà deux entre 3 sommets, est élevée ; ce qui présuppose une structure dans ce sous-graphe.

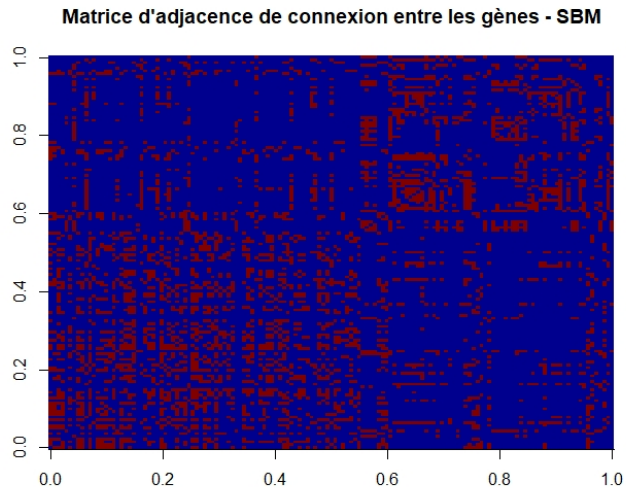


FIGURE 5 – Matrice d'adjacence du SBM. Plus la couleur est bleue, moins les gènes sont connectés. Plus la couleur est rouge, plus les gènes sont connectés

Nous représentons sur la figure 5 la matrice d'adjacence des gènes sans l'information sur les arêtes. Cette matrice est creuse et fortement portée sur la diagonale. Ceci nous indique de la variabilité dans les connexions entre les gènes. L'application d'un SBM est ainsi pertinente. Il nous faut également justifier le choix de ce petit réseau au niveau du support des arêtes. En effet, si celles-ci portent toutes la même information, cette dernière ne serait pas pertinente pour l'application d'un STBM qui classerait comme s'il y avait absence de texte (Il ne pourrait pas distinguer des communautés à l'aide de l'information des arêtes, celle-ci étant identique partout).

	gene1	gene2	stress_partages
1	AT4G26530	AT3G20470	TEMPERATURE;NITROGEN.ROOT;NECROTROPHIC.BACTERIA;FUNGI;VIRUS
2	AT4G26530	AT3G50740	TEMPERATURE;NITROGEN.ROOT;DROUGHT;VIRUS
3	AT4G26530	AT2G05540	TEMPERATURE;NITROGEN.ROOT;BIOTROPHIC.BACTERIA;FUNGI
4	AT4G26530	AT5G61590	TEMPERATURE;NITROGEN.ROOT;BIOTROPHIC.BACTERIA
5	AT4G26530	AT2G16660	TEMPERATURE;NITROGEN.ROOT;DROUGHT;VIRUS
6	AT4G26530	AT1G74670	TEMPERATURE;UV;NITROGEN.ROOT;DROUGHT;STIFENIA;VIRUS
7	AT4G26530	AT3G18080	TEMPERATURE;DROUGHT;RHODOCOCCUS
8	AT4G26530	AT3G10260	OTHER-ABIOTIC;TEMPERATURE;NECROTROPHIC.BACTERIA;STIFENIA
9	AT4G26530	AT2G38940	TEMPERATURE;NITROGEN.ROOT;FUNGI
10	AT4G26530	AT1G26800	TEMPERATURE;NITROGEN.ROOT;DROUGHT
11	AT4G26530	AT4G28240	TEMPERATURE;NITROGEN.ROOT;DROUGHT
12	AT4G26530	AT1G76180	TEMPERATURE;FUNGI;VIRUS
13	AT4G26530	AT5G20630	UV;BIOTROPHIC.BACTERIA;VIRUS
14	AT4G26530	AT2G40330	TEMPERATURE;DROUGHT;FUNGI;VIRUS
15	AT4G26530	AT3G32990	TEMPERATURE;SALT;FUNGI;STIFENIA
16	AT4G26530	AT3G28270	TEMPERATURE;NITROGEN.ROOT;BIOTROPHIC.BACTERIA;VIRUS
17	AT4G26530	AT1G69530	TEMPERATURE;STIFENIA;VIRUS
18	AT4G26530	AT3G14210	TEMPERATURE;NITROGEN.ROOT;DROUGHT;BIOTROPHIC.BACTERIA;VIRUS
19	AT4G26530	AT5G02760	TEMPERATURE;BIOTROPHIC.BACTERIA;VIRUS
20	AT4G26530	AT1G71030	NITROGEN.ROOT;HEAVY.METAL;RHODOCOCCUS;VIRUS

FIGURE 6 – Lecture des 20 premières lignes du jeu de données. La première colonne correspond au gène AT4G26530, la deuxième colonne correspond à 20 autres gènes qui co-expriment avec le premier et la troisième colonne à l'information portée par l'arête reliant le gène 1 aux 20 autres.

Nous présentons, sur la figure 6, les 20 premières lignes de notre petit jeu

de données. Nous observons une forte variabilité sur l'information de 20 arêtes, toutes reliées au gène AT4G26530. Certes, le stress température est ici présent sur 18 arêtes mais le stress virus n'apparaît que pour une bonne moitié des arêtes. Nous observons également une variabilités entre l'apparition des autres stress. Par exemple, Rhodococcus apparaît 2 fois, la première arête est portée aussi par les stress température et sécheresse, tandis que la deuxième est portée par les stress azote, métal lourd et virus. Ainsi, si l'information sur les arêtes apportent quelque chose, nous le verrons sur ce petit réseau.

Nous sommes prêtes à appliquer nos deux algorithmes sur le petit réseau. L'ensemble des codes effectués lors de notre étude se trouve en annexe A.

3.2. Estimation du SBM sur le réseau des 143 gènes

L'application du SBM sur notre petit réseau est effectué grâce au package "blockmodels" du **logiciel R** avec le tye Bernoulli. Nous lui donnons un fichier à 2 colonnes indiquant les noeuds entre lesquels il y a une arête.

Comme présenté au paragraphe 2.1.1, l'algorithme est divisé en deux étapes bien distinctes : la première étant à Q (nombre de communautés) fixé et elle consiste en l'estimation des paramètres ρ et π, la seconde a pour but de déterminer Q. Nous retrouvons ces deux étapes en pratique : l'algorithme estime ρ et π en commençant par $Q = 2$, puis $Q = 3$, etc... et pour chacun, il mémorise l'ICL obtenu. Du fait de la stochasticité de l'algorithme, il calcule les paramètres ρ et π et l'ICL plusieurs fois pour chaque Q. Il relève ensuite, pour chaque Q, le maximum des ICL et tant que ce dernier augmente entre les différents Q, il continue ses estimations. Mais dès lors qu'il commence à diminuer, il est probable que le Q correspondant au maximum des ICL obtenu à ce stade, soit le meilleur pour la classification. Notons $Q_{\max}$ ce paramètre potentiel. L'algorithme continue le même travail jusqu'à $1.5 \times Q_{\max}$. Si aucun ICL n'est au dessus du maximum correspondant à $Q_{\max}$, il considère qu'il a trouvé le meilleur Q, sinon, il recommence avec le nouveau $Q_{\max}$. Une fois $Q_{\max}$ trouvé, il crée le graphe en rangeant les gènes grâce aux $\tau_{i,q}$ calculé avec la formule de Bayes et aux ρ et π correspondant.

3.2.1. Courbe ICL et obtention des meilleurs paramètres du modèle

ICL SBM
−4967.131
−3921.806
−3544.097
−3422.891
−3330.021
−3279.324
−3245.908
−3220.750
−3226.482
−3238.645
−3254.218
−3294.777

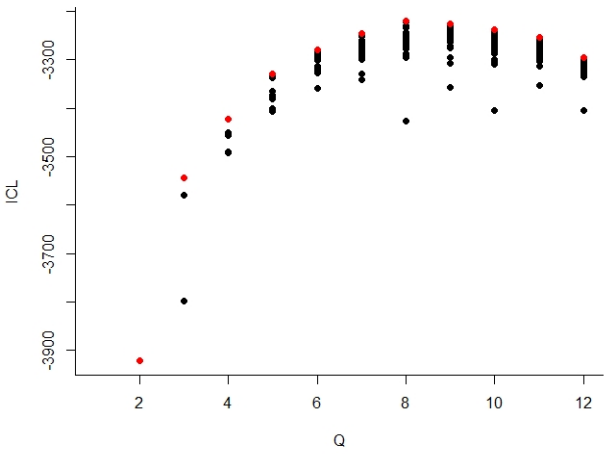


FIGURE 7 – Courbe de toutes les valeurs de critère ICL obtenues lors du lancement de l'algorithme SBM

Nous avons tracé à la figure 7 la courbe de tous les ICL obtenus en lançant le package de R sur le réseau des 143 gènes. Ici, les points rouges correspondent au maximum des ICL pour chaque Q, tandis que les points noirs correspondent aux autres valeurs des ICL. Les valeurs des points rouges sont résumés dans le tableau. L'algorithme a ainsi retenu $Q_{max} = 8$. Observons qu'il a continué de tourner jusqu'à $Q = 12$ et $12 = 8 \times 1.5$.

La figure 8 est un zoom de la précédente où ne sont représentés que les points rouges.

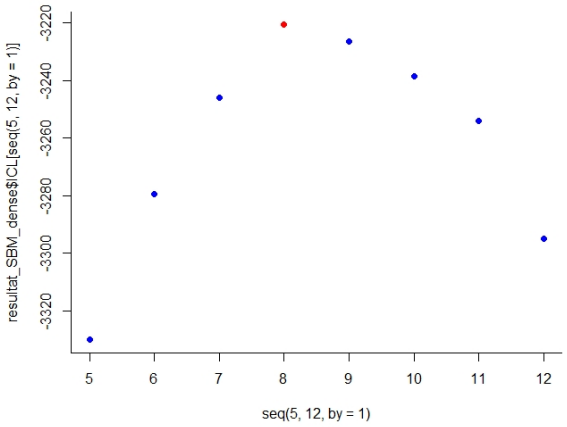


FIGURE 8 – Mise en évidence du meilleur choix du nombre de communautés

Le SBM nous donne ainsi un graphe à 8 communautés. Nous pouvons en extraire les paramètres correspondant à cette structure afin de l'analyser.

3.2.2. Analyse du réseau

L'application du SBM nous offre une classification des gènes en 8 communautés. La première idée est de visualiser cette classification sous forme de graphe. Après un algorithme de force avec le package "igraph" du logiciel R, nous obtenons le graphe suivant :

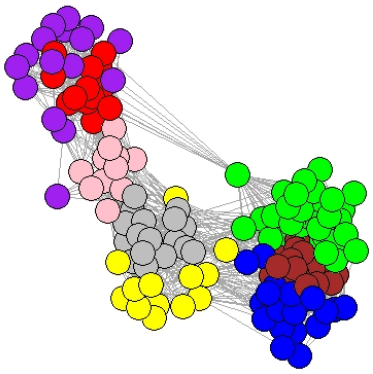


FIGURE 9 – Graphe des communautés obtenus par le SBM. Chaque couleur représente une communautés.

Nous sommes tentées d'étudier le nouveau réseau à l'échelle des communautés, c'est-à-dire, exploiter les probabilités de connexion des gènes au sein des communautés et entre les communautés, ceci grâce au paramètre π. Soulevons qu'il ne s'agit pas d'en faire l'étude à l'échelle des gènes.

Nous illustrons les probabilités de connexions sur la figure 10 : la diagonale est fortement foncée, ce qui signifie que les gènes au sein d'une communauté communiquent beaucoup plus entre eux qu'avec les gènes des autres cluster. Remarquons également qu'une communauté est fortement reliée à une ou deux autres communautés et très faiblement vers celles restantes. Il semblerait donc qu'il y ait également des ressemblances fortes à l'échelle des communautés. D'autres résultats seront évoqués dans la partie 3.4 lorsque la comparaison entre les deux modèles sera réalisée.

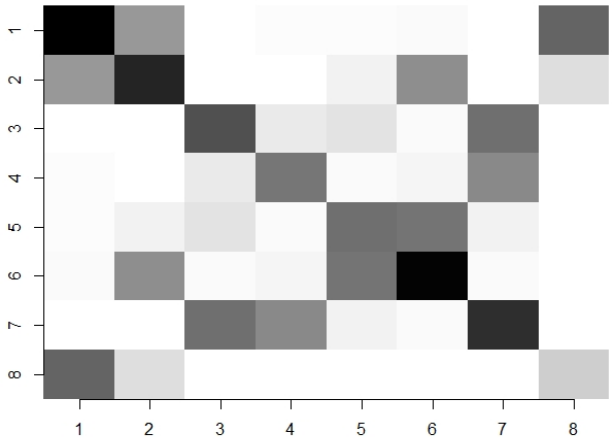


FIGURE 10 – Probabilité de connexions intra et extra clusters. En allant du blanc au noir, les connexions sont de plus en plus fortes.

3.3. Application du STBM sur le réseau des 143 gènes

Nous avons soumis notre jeu de données au logiciel **linkage** afin d'obtenir une classification selon l'algorithme du STBM. Cette fois-ci, notre fichier à 3 colonnes : les deux premières pour les noeuds reliés par une arête et la troisième pour les différents stress portés par cette connexion.

Tout comme le SBM, l'algorithme est divisé en deux étapes majeures. La première estime les paramètres ρ, π, θ, β à K et Q fixés et la deuxième sélectionne le meilleur modèle. Toujours du fait de la stochasticité du problème, il évalue plusieurs fois les paramètres à Q et K fixés et retient les ICL maximales obtenus.

3.3.1. Courbe ICL et obtention des meilleurs paramètres du modèle

Avec l'utilisation du logiciel Linkage, nous sommes limitées à des STBM avec moins de 10 communautés mais c'est pourquoi nous avons considéré un petit réseau et nous espérons que Linkage trouvera moins de 10 communautés. Par ailleurs, nous ne connaissons pas le critère d'arrêt que Pierre Latouche a choisi pour son algorithme. C'est la raison pour laquelle nous avons relancé l'estimation plusieurs fois.

La forme des courbes ICL en 3D nous donne une idée sur l'optimalité du paramètre retenu par l'algorithme. En effet, si cette courbe décroît en les deux paramètres après avoir atteint un maximum, nous serons amenées à penser qu'elle ne recroîtra plus par la suite.

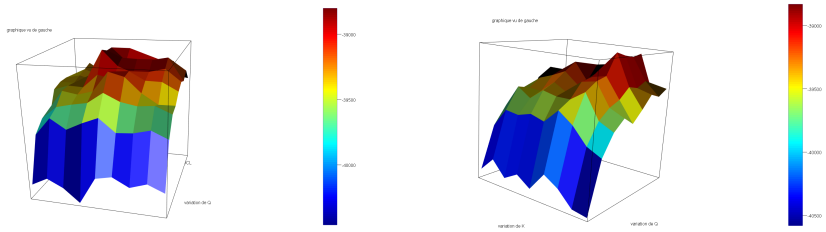


FIGURE 11 – Deux exemples de tracé de courbe d'ICL. Chaque ICL est en réalité le maximum des ICL parmi les simulations de linkage à Q et K fixés.

Sur la figure 11 sont représentées les courbes ICL de deux simulations Linkage qui ne prennent en compte que les maximums des ICL à Q et K fixés parmi toutes les simulations. Nous observons un maximum correspondant à $Q = 7$ et $K = 7$ sur la figure de gauche, tandis que le maximum de la courbe de droite est atteint pour $Q = 7$ et $K = 8$. Contrairement à l'algorithme du SBM, l'étape de la sélection de modèle diffère d'une soumission à une autre, ce qui est assez problématique. Nous nous sommes limitées, pour ce rapport, à lancer 8 fois Linkage et à conserver l'estimation du jeu de données qui contenait le maximum des ICL parmi tous ceux calculés lors de nos 8 simulations.

Nous représentons sur la figure 12 la courbe lissée des ICL : chaque point a pour valeurs le maximum des ICL parmi tous les ICL correspondant à ce (Q, K) de nos 8 jeux de données.

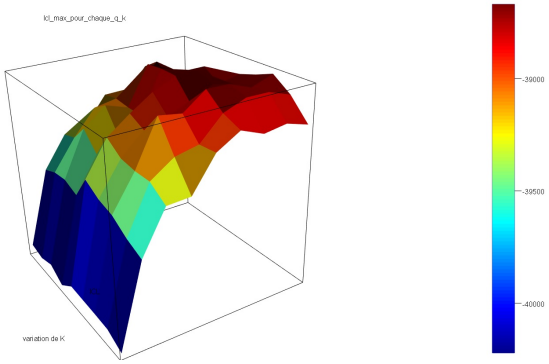


FIGURE 12 – Courbe lissée des ICL sur nos 8 soumissions. Pour chaque couple (K, Q), la valeur du point représente le maximum des ICL de ce couple parmi toutes les simulations.

Le lissage nous permet d'observer la décroissance de la courbe au delà du maximum qui est ici atteint pour $(Q = 7, K = 7)$ nous laissant penser que si nous pouvions prospecter plus, le STBM renverrait ce même paramètre. Notons également que $7 \times 1.5 = 10$. Ainsi, si le critère d'arrêt de linkage est le même que celui du SBM, nous sommes assurés d'avoir trouvé le couple de paramètre optimal.

Dans la suite, nous nous fixons le jeu de données qui a donné le maximum des ICL. Précisions que la courbe ICL correspondant à sa simulation est celle à gauche de la figure 11 et retenons que $Q_{max} = 7$ et $K_{max} = 7$.

3.3.2. Matrice d'adjacence

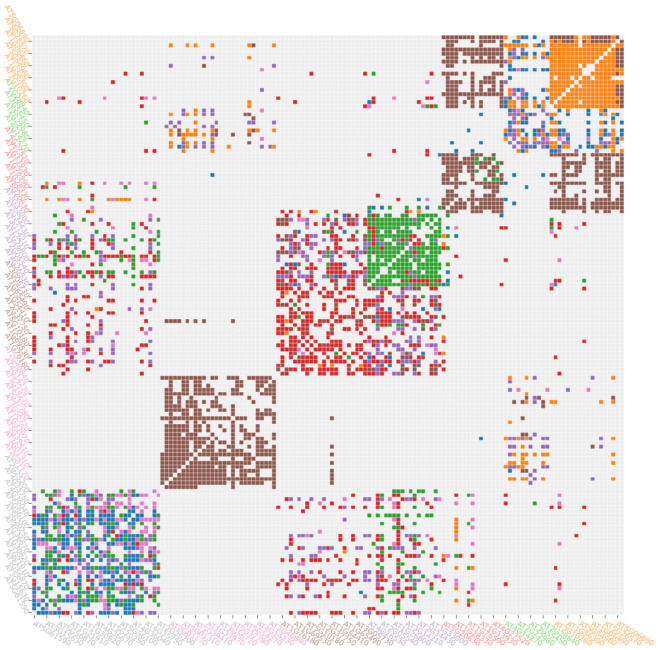


FIGURE 13 – Matrice d'adjacence du STBM. Une absence de couleur correspond à une absence de connexion entre les gènes. Les différentes couleurs représentent les différentes natures des connexions en fonction des stress que portent les arêtes. Ainsi, les entrées de même couleur correspondent à un pourcentage similaire de partage de chaque stress.

La figure 13 représente la matrice d'adjacence du réseau en prenant en compte les stress sur les arêtes. Nous retrouvons, sur cette matrice, la structure creuse et l'importance des connexions autour de la diagonale que nous avions déjà obtenues pour la matrice d'adjacence du réseau sans information sur les arêtes (figure 5). Mais, désormais, nous pouvons relever aussi une variété dans les couleurs. Ces dernières modélisent la nature des connexions. La couleur rouge représente un partage porté par une certaine nature de connexions entre deux gènes qui diffère de celui représenté par la couleur bleue par exemple. Il existe donc une forte diversité quant à la nature des connexions entre les gènes. Cette matrice d'adjacence offre beaucoup plus d'information que celle obtenue sans tenir compte de l'information sur les arêtes. En effet, nous voyons ici que la communauté 2 (en partant du bas sur l'axe vertical) n'a qu'un seul sujet de discussion puisque les connexions sont toutes de couleur marron alors que la communauté 1 est plus versatile car elle partage plusieurs sujets représentés par des couleurs différentes. Cette discussion nous permet de justifier l'utilité du STBM dans notre situation : au sein d'une communauté, on observe ou non une diversité au niveau de la nature des topics échangés ce qui nous permet d'apporter des caractéristiques nouvelles à chaque cluster.

Après avoir conservé les estimations des différents paramètres calculées par

Linkage pour ce jeu de données, nous nous sommes définitivement détachées de ce logiciel et nous disposons des fichiers suivants :

1. "dictionnary.txt" : l'ordre des stress.
2. "labels.txt" : l'ordre des gènes.
3. "tdm.spmat.txt" : les arêtes entre les gènes du petit réseau.
4. "clusters.csv" : 7 lignes pour les 7 clusters et chaque ligne est constituée de tous les gènes appartenant au cluster correspondant.
5. "edges.with.topics.csv" : la première colonne est constitué du gène 1, la seconde du gène 2 lui étant relié, la troisème du numéro du topic dominant sur l'arête de connexion, la quatrième colonne du poids de l'arête.
6. "nodes.with.clusters.csv" : Une ligne par gène. La première colonne est constituée du nom du gène et la seconde du chiffre d'appartenance à son cluster associé.
7. "topics.csv" : une ligne par topics et dans chaque ligne est représentée les 18 probabilités des stress formant ce topic.
8. "cluster.txt" : vecteur de taille le nombre de gènes composé des chiffres caractérisant l'appartenance aux clusters pour les différents gènes.
9. "crit.txt" : une seule valeur : l'ICL pour le couple (Q, K) retenu.
10. "PI.txt" : vecteurs de taille $Q \times Q$: le poids des arêtes reliant les communautés.
11. "rho.txt" : vecteur de taille Q : la probabilité d'appartenir aux clusters.
12. "thetaQR.txt" : vecteur de taille $Q \times Q \times K$: les probabilités, pour chaque connexion entre les communautés, de parler des 7 différents topics.
13. "topics.txt" : vecteur de taille $18 \times K$: la proportion des 18 stress dans chaque topics.
14. "topics.per.edges.txt" : La même chose que le fichier "thetaQR.txt" mais chaque topic est décrit par ses pourcentages de stress.

3.3.3. Analyse du réseau

Nous utilisons de nouveau le logiciel R pour interpréter les résultats. Le STBM nous a renvoyé 7 communautés et 7 topics. Représentons dans un premier temps, le graphe issu de la classification en 7 cluster. (figure 14)

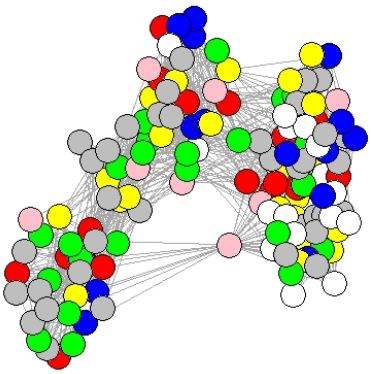


FIGURE 14 – Graphe des communautés obtenus par le STBM. Chaque couleur représente une communauté

Cette représentation est satisfaisante en première étude mais nous limite assez rapidement. En effet, puisque nous avons pris en compte l'information sur les arêtes et que nous savons que l'algorithme a renvoyé des communautés reliées par des arêtes parlantes, nous souhaiterions comprendre la nature des liens des gènes au sein d'une communauté ou entre deux d'entre elles. Nous allons donc axer l'étude du STBM sur les arêtes.

3.3.4. Étude des topics

Chacune des arêtes qui relie les communautés est portée par un vecteur de pourcentage de topics et chacun d'entre eux est lui-même un ensemble de pourcentage de stress. Une étude à l'échelle des connexions entre les communautés permet de connaître si celles-ci sont toutes portées par un même topic dominant, auquel cas, elles seront majoritairement portées par les mêmes stress, ou si au contraire, chaque connexion est une moyenne pondérée des topics caractéristiques des communautés qu'elle lie.

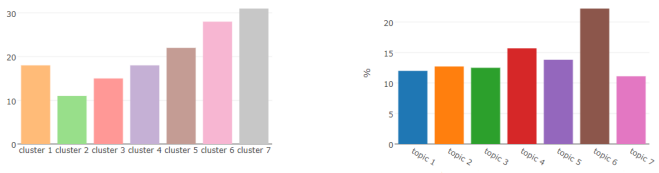


FIGURE 15 – A gauche : taille des clusters en pourcentage. A droite : taille des topics en pourcentage.

La figure 15 permet de répondre à cette question. Le topic dominant est le numéro 6 et sur un total de 100%, il apparaît à légèrement plus de 20%. Le topic 4 le suit avec environ 16% et tous les autres ont une fréquence d'apparition autour de 12%. Ce résultat ne permet pas de conjecturer à des connexions portées par un même topic dominant, mais au contraire, suggère que tous les topics ont leur rôle et aucun d'entre eux est négligeable. Nous verrons par la suite le détail des natures des connexions entre les communautés.

La dispersion des stress entre les topics attise notre curiosité. Nous représentons à la table 3 les stress apparaissant avec une probabilité supérieure à 0.1 dans chaque topic.

Topics	**Stress prépondérants**
1	temperature, virus, gamma
2	biotrophi bacteria, oomycete, gamma
3	nitrogen root, temperature, drought, oxydative stress
4	nitrogen root, temperature, heavy metal
5	nitrogen root, temperature, biotrophi bacteria, oomycete
6	fungi, virus, oxydative stress
7	temperature, virus, oomycete, gamma

TABLE 3 – Listes des stress prépondérants dans chaque topics

Une première interprétation de ce tableau serait de le lier avec le résultat précédent : les topics 6 et 4 sont les plus présents ce qui suggérerait la prédominance des stress champignon, virus, stress oxydatif ou encore température et azote. Une deuxième interprétation ferait des liens entre les stress. Nous observons que le couple "température-azote" apparaît 3 fois dans le tableau, les couples "température-virus" et "virus-gamma" 2 fois. Des conclusions biologiques pourraient rapprocher des stress partageant le même comportement au niveau de l'expression de certains gènes associés.

Après réflexions sur les interprétations que nos résultats pourraient apporter au niveau biologique, nous étions convaincues qu'il fallait passer de l'étude des topics à l'étude des stress. En effet, savoir que la connexion entre deux des 7 cluster était portée par $\alpha_1\%$ du topic 1, $\alpha_2\%$ du topic 2, etc... n'était pas exploitable puisque lors d'une expérience sur la plante, on se fixe une situation de stress et non une moyenne pondérée de situations de stress. Nous cherchons ainsi à conclure sous la forme : "la connexion entre le cluster p et le cluster q est portée par $\beta_1\%$ du stress virus, $\beta_2\%$ du stress azote, etc... Nous avons donc travaillé pour nous libérer des topics.

3.3.5. Etude à l'échelle des stress

Commençons par rappeler quelques notations de l'algorithme STBM général :

— Q est le nombre de cluster, K est le nombre de topics.
— $Y = (Y_1, ..., Y_M)$ est le vecteur d'appartenance au cluster. M étant le nombre de noeuds.
— A est la matrice d'adjacence.

— V est l'ensemble des mots.
— $\forall (i,j) \in [1,M]^2, Z_{i,j} = ((Z_{i,j})_{i,j}^{d,n})_{d,n \in (\{1,\cdots D_{i,j}\},\{1,\cdots,N_{i,j}^d\})}$ le vecteur des sujets entre les sommets i et j, d étant un document et $N_{i,j}^d$ le nombre de mots v dans le document d.
— $\mathbb{P}(Z_{i,j}^{d,n} = k | A_{i,j} = 1, Y_{i,q} = 1, Y_{j,r} = 1) = \theta_{q,r,k}$
— $\forall (i,j) \in [1,M]^2,\ \forall d \in \{1,\cdots,D_{i,j}\},\ \forall n \in \{1,\cdots,N_{i,j}^d\},\ W_{i,j}^{d,n}$ le $n$ème mot du $d$ème document.

Pour faire le lien avec notre situation, V est l'ensemble des stress, les sujets sont les topics et nous n'avons qu'un seul document. Nous voulons obtenir les probabilités de partage de stress entre les cluster. Pour tous les couples de clusters (q,r), on a :

Si $v \in V$ est un mot, disons que v est à la coordonnée t du vecteur V,

$$\mathbb{P}\left(W_{i,j}^{d,n} = v | A_{i,j} = 1, Y_{i,q} = 1, Y_{j,r} = 1\right)$$
$$= \sum_{k=1}^{K} \mathbb{P}(W_{i,j}^{d,n} = v, Z_{i,j}^{d,n} = k | A_{i,j} = 1, Y_{i,q} = 1, Y_{j,r} = 1)$$

Par la formule de Bayes, on a :

$$\mathbb{P}(W_{i,j}^{d,n} = v | A_{i,j} = 1, Y_{i,q} = 1, Y_{j,r} = 1)$$
$$= \sum_{k=1}^{K} \mathbb{P}(W_{i,j}^{d,n} = v | Z_{i,j}^{d,n} = k, A_{i,j} = 1, Y_{i,q} = 1, Y_{j,r} = 1)$$
$$\times \mathbb{P}(Z_{i,j}^{d,n} = k | A_{i,j} = 1, Y_{i,q} = 1, Y_{j,r} = 1)$$

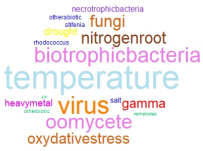
Il est donc possible, à partir des topics, d'obtenir les probabilités d'échange de stress sur chaque arête. Ceci revient à appliquer un simple produit matriciel, fruit de ces égalités.

Nous sommes donc prêtes à l'étude des connexions entre les communautés à l'échelle des stress.

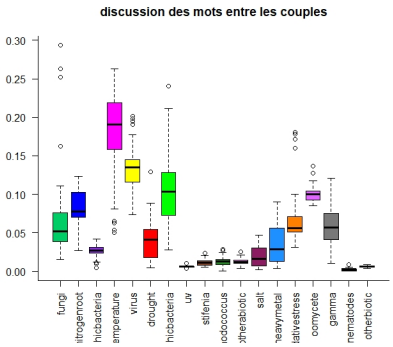
Tout d'abord, faisons comme pour les topics, c'est-à-dire, trouvons les stress dominants.

Nous représentons à la figure 16 un nuage de mots illustrant les stress prédominants dans l'ensemble des connexions intra et extra communautés ainsi que 18 diagrammes en boîte pour chaque stress.

Nous observons que les stress température, virus, bactéries biotrophiques ou encore l'azote sont les stress prépondérants liant les clusters. Cette étude nous renvoie une idée globale des connexions. Désormais, il est temps de caractériser les clusters par leurs liens avec les autres. Nous allons donc approfondir l'étude à l'échelle des communautés.



(a) Nuage de mots sur l'ensemble des échanges entre les communautés.



(b) Diagramme en boîte représentant les discussions de stress entre les communautés.

FIGURE 16 – Illustrations des stress dominants dans les échanges entre les communautés.

Pour la suite, fixons un ordre des stress (voir le tableau ci dessous).

Ordre des stress
fungi
nitrogenroot
necrotrophicbacteria
température
virus
drought
biotrophicbacteria
uv
stifenia
rhodococcus
otherabiotic
salt
heavymetal
oxydativestress
oomycete
gamma
nematodes
otherbiotic

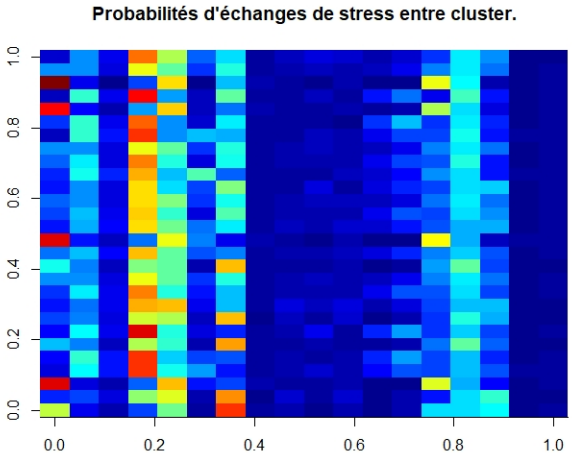


FIGURE 17 – Probabilité des échanges de stress entre chaque couple de clusters (q, r) où $q \leq r$. Les 18 colonnes représentent les 18 stress (voir l'ordre des stress dans le tableau ci-contre, la première colonne représentant le stress fungi), et chaque ligne représente un couple de cluster : la 1ère ligne du bas pour "1-1, ensuite "1-2", ..., "1-7","2-2", etc....

Nous avons représenté à la figure 17 les probabilités d'échange de chaque stress par couple de communautés. Là encore, deux interprétations sont pos-

sibles. Une lecture verticale nous permet de constater une variabilité des stress. Par exemple,

- le stress "champignon" est fortement présent sur les connexions "cluster1 - cluster 3", "cluster3 - cluster 3", "cluster 5 - cluster 6" ou encore "cluster 6 - cluster 6", nous le retrouvons à probabilité plus faible sur d'autres connexions et il semble disparaître sur les restantes.
- Le stress "température" semble prendre une part non négligeable dans toutes les connexions au contraire des stress "uv, "stifenia", "rhodococcus", "autres stress biotiques", "sel" qui sont peu partagés.
- Le stress "bactérie biotrophique" intervient dans les connexions avec le cluster 1 et de façon faible en dehors de cette communauté.
- Enfin, "oomocete" était assez visible dans le nuage de mot (figure 16) et cette nouvelle figure nous permet de conclure qu'il ne prend pas de part imposante dans les connexions mais apparaît dans toutes de façon non négligeable.

La pertinence du choix du réseau s'observe une nouvelle fois : il y a hétérogénéité dans les pourcentages de stress, chacun ayant un rôle bien précis. Une lecture horizontale permet de confirmer l'intérêt du STBM : les connexions sont caractérisées par un poids de pourcentage de stress, qui varie fortement d'un couple de clusters à un autre. En effet, la connexion "cluster1 - cluster7" est fortement portée par le stress "température", puis avec probabilité plus faible par les stress "azote", "virus" et "oomocyte" ; tandis que la connexion "cluster 1 - cluster 3" est marquée par le partage du stress "champignon" puis par les stress "virus", "stress oxydatif". Remarquons qu'il n'y a pas beaucoup d'échange entre les communautés 3 et 6, ceci suggère qu'en plus du fait que les gènes se ressemblent au sein d'une communauté, il existe des similarités plus ou moins fortes entre plusieurs communautés et nous avons connaissance des situations de stress pour lesquelles nous observons ces ressemblances.

Il est cependant difficile de pouvoir conclure à l'importance de chaque stress vu individuellement dans notre petit réseau. Pour cela, observons les fréquences de chacun d'entre eux sur les différentes connexions.

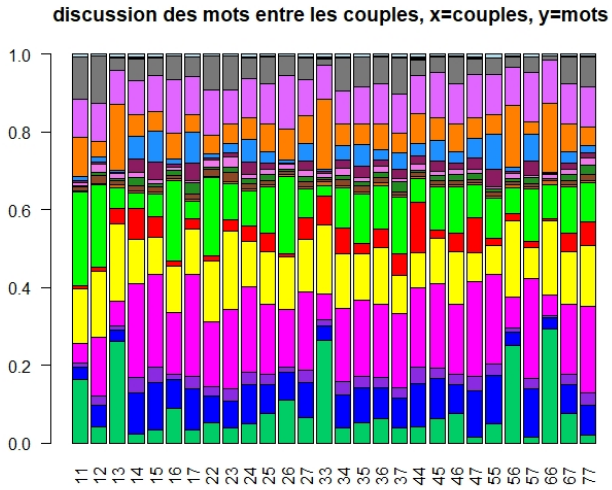


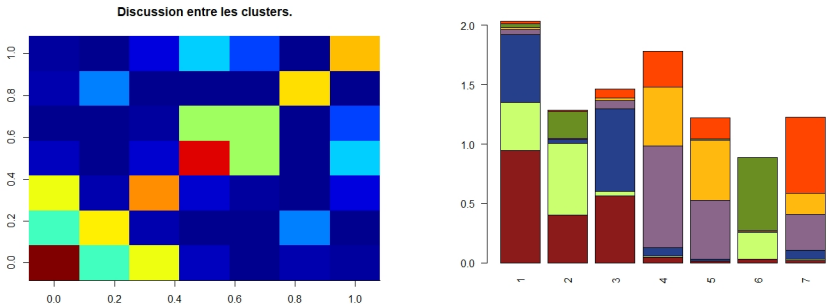
FIGURE 18 – Caractérisation des connexions entre les communautés au travers des partages de stress. Ne sont représentées que les couples (q, r) où $q \leq r$. Les 18 couleurs représentent les 18 stress dont l'ordre est celui du tableau ci-dessus. La première couleur verte du bas représente le stress "champignon" et ainsi de suite.

Il est frappant de voir la dominance de la couleur rose correspondant au stress "température" : il apparaît dans toutes les connexions et souvent de manière très marquée. La couleur jaune représentant le stress "virus" est présente à quasiment même fréquence (assez élevée aussi) dans tous les liens. La couleur verte ("bactéries biotrophiques") est variable : parfois imposante, parfois négligeable. C'est le cas aussi pour la première couleur verte du bas ("champignon") : ce stress est prépondérant dans les connexions "cluster 1 - cluster 3", "cluster 3 - cluster 3", "cluster 5 - cluster 6", "cluster 6 - cluster 6" mais est négligeable pour les autres. D'autres couleurs sont homogènes comme le rose pâle, le marron, le jaune ou le bleu, ce qui présupposent que les stress correspondant ne jouent pas de rôle dans la classification : ils sont importants pour expliquer une connexion vue individuellement mais n'intervient pas lorsqu'il s'agit d'expliquer la fabrication du clustering puisque les fréquences sont égales sur toutes les connexions.

Remarquons que l'interprétation globale a ses limites : on a un graphique représentant 7 communautés et 49 échanges de 18 stress, ce qui est beaucoup. De plus, lors de son expérience, le biologiste ne va pas s'intéresser à l'ensemble des gènes du petit réseau mais à un sous-ensemble : ceux appartenant à un même cluster par exemple (puisqu'ils semblent avoir une similarité dans leur fonction biologique). Il est donc intéressant de changer d'échelle et de se placer au sein d'une communauté pour pouvoir la comprendre. Pour que cela soit pertinent, intéressons-nous aux communautés ayant beaucoup d'échanges. D'après la figure 19 de gauche, obtenue grâce aux valeurs estimées de π, nous obtenons des connexions portées par la diagonale : les échanges sont donc dominants au sein des communautés mais nous en retrouvons des non négligeables entre commu-

nautés différentes : par exemple les couples 1-3 ou 4-5. La figure de droite illustre la taille de chaque arête. Observons que la hauteur des bâtons concernant les communautés 1,3 et 4 respectivement est élevée : ces communautés échangent beaucoup et en effet, les couleurs rouge, bleu et violette sont plus imposantes que les autres.

Nous allons donc étudier plus en détail ces communautés et ces connexions.



(a) Probabilités d'échanges intra et extra clusters. L'abscisse de gauche à droite correspond aux communautés 1,2,...,7, l'ordonnée de bas en haut correspond aux communautés 1,2,...,7.

(b) Illustration des poids des arêtes par communautés. La hauteur totale représente le poids total de l'arête et au sein de chaque bâton, les couleurs représentent le poids de l'échange avec la communauté en question.

FIGURE 19 – Illustration de la variabilité des arêtes entre les communautés.

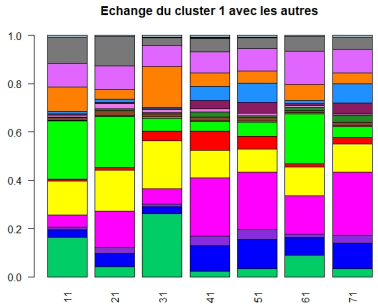
Les graphiques (a), (b) et (c) de la figure 20 illustrent les échanges de stress des communautés 1, 3 et 4 respectivement avec toutes les autres communautés.

Pour la communauté 1, le stress "champignon" est fortement présent dans les échanges à l'intérieur de celle-ci mais aussi pour la liaison avec le cluster 3. Nous retrouvons cette présence importante nulle part ailleurs. Le stress "température" est quand à lui marqué par les liens avec les clusters 4,5,6 et 7. Ainsi, nous pouvons caractériser la communauté 1 comme celle regroupant les gènes ayant une fonctionnalité biologique forte dans les situations de stress dues aux champignons, aux virus ou aux bactéries biotrophiques. En étudiant la connaissance de celle-ci avec la communauté 5, nous concluons que l'ensemble des gènes issus de ces deux clusters se ressemblent en situation de stress "température" et "virus".

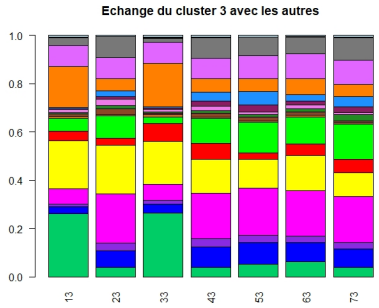
En revanche, lorsque nous portons notre regard sur la communauté 4, elle semblerait échanger de manière équivalente ses stress : la palette de couleur est homogène au travers de chaque connexion. Ainsi, au sein d'elle, les gènes s'expriment en situation de stress "température", et en combinant les gènes d'une autre communauté avec la numéro 4, nous obtiendrons les mêmes conclusions. Donc, des différences sont perçues en se concentrant sur les échanges d'une communauté choisie mais aussi en les comparant deux à deux : les échanges de la communauté 1 semble similaire à ceux de la communauté 3 alors qu'ils sont complètement différents de ceux de la communauté 4.

Enfin, nous avons représenté le nuage de mots des échanges de stress de la liaison 1-3. Contrairement au nuage de mot total (figure 16) où la dominance

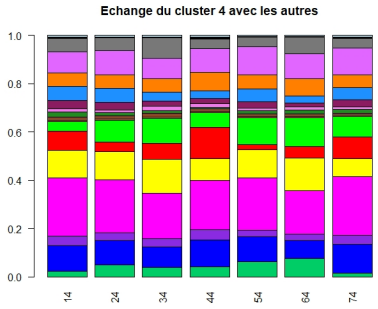
s'observait au travers des stress "température", "virus" ou encore "bactéries biotrophiques", la connexion se repose essentiellement sur les stress "champignons", "virus" ou "stress oxydatif".



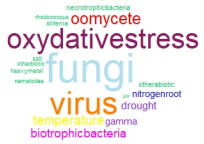
(a) Échange de stress de la communauté 1 avec les autres



(b) Échange de stress de la communauté 3 avec les autres



(c) Échange de stress de la communauté 4 avec les autres



(d) Nuage de mots entre les communautés 1 et 3

FIGURE 20 – Échange de stress : étude à échelle des clusters

En conclusion de cette partie, la variabilité des tailles des arêtes et des poids de chaque stress sur chacune d'entre elles confirme l'intérêt du STBM : nous avons relevé une importante hétérogénéité de ces poids, ce qui suggère que la classification s'est faite à travers les liens entre les gènes mais aussi à travers la nature des connexions. Ainsi, chaque communauté semble être caractérisée par des gènes ayant des fonctionnalités biologiques semblables mais aussi partageant une expression génétique similaire lorsque la plante est confrontée aux différentes situations de stress. Les mêmes conclusions sont apportées en combinant deux communautés entre elles.

Cependant, à ce stade, seules des conjectures peuvent être relevées : "les stress varient et les lignes et les colonnes des diagrammes obtenus ne sont pas uniformes, donc, a priori, l'information sur les arêtes apportent quelque chose". Et afin de comprendre le clustering issu du STBM et d'en déduire l'importance des stress, il est nécessaire de comparer les résultats de classification obtenus par le SBM et le STBM.

3.4. Comparaison des résultats obtenus avec le SBM et avec le STBM

Notre objectif est de détecter et de comprendre les changements dûs à l'apport des informations sur les arêtes. Nous allons donc prendre les résultats du SBM comme référence et comparer à eux ceux issus du STBM.

Alors que nous serions tentées de penser que l'ajout de l'information des arêtes divisera les communautés du SBM en sous catégories de gènes se rassemblant par leur nature de connexions, le phénomène inverse se produit : nous avons perdu une communauté en passant du SBM au STBM. Nous avons alors cherché une explication. Le STBM doit classer en prenant en compte la nature des connexions. Par notre intuition, celle-ci sépare des gènes mais elle peut également rapprocher certains d'entre eux qui n'étaient pas forcément semblables aux yeux du SBM : c'est le cas en considérant par exemple un groupe de gènes peu reliés entre eux mais partageant les mêmes stress.

Ainsi, ces situations ont tendance à regrouper des gènes : la classification donnera moins de groupe. Un compromis entre notre intuition et cette explication est donc mis en jeu dans le STBM : certaines connexions éloignent des gènes tandis que d'autres les rapprochent.

Nous sommes donc en possession de 8 communautés pour le SBM et de 7 du côté du STBM. L'une des premières choses que nous souhaiterions connaître est l'existence ou non d'un lien entre ces clusters. Nous cherchons à connaître si le passage de 8 à 7 est le fruit d'une explosion d'une communauté en deux, ou au contraire, si le STBM a renvoyé un clustering complètement différent de celui du SBM.

3.4.1. Matrice de contingence

Afin de comparer les deux clustering, nous allons utiliser deux outils : la **matrice de contingence** et l'indice **AdjustingRandIndex**.

Matrice de contingence

Nous cherchons à créer la matrice de contingence entre les groupes obtenus par le SBM et ceux obtenus par le STBM. Cette méthode rend compte de la dépendance entre les deux partitions. En effet, pour chaque paire (communauté SBM,communauté STBM), elle dénombre les gènes qui appartiennent à ces deux clusters.

	cluster_dans_STBM						
cluster_dans_SBM	1	2	3	4	5	6	7
1	1	2	0	3	3	1	3
2	0	0	1	0	2	3	5
3	3	1	1	3	2	3	7
4	11	4	3	2	3	6	7
5	1	2	3	3	2	1	1
6	1	2	0	2	5	6	3
7	2	3	0	2	2	3	3
8	0	3	1	0	4	1	7

FIGURE 21 – Matrice de contingence entre SBM et STBM

La matrice que nous obsevons à la figure 21 n'est pas creuse, ce qui indique qu'il est impossible d'identifier les communautés du STBM avec celles du SBM. Par exemple, nous observons que toutes les communautés du SBM contiennent au moins un gène de la communauté 5 du STBM, il en est de même pour les communautés 6 et 7. Ainsi, aucun lien ne semble apparent entre les deux classifications et l'information sur les arêtes modifie donc fortement les classes de fonctionnalité biologique formées.

AdjustedRandIndex

L'**AdjustedRandIndex (ARI)** est un indice qui évalue la proximité de deux partitions qui contiennent toutes les deux les mêmes éléments mais peuvent différer en nombre de classe. Donnons sa forme littérale :
Notons $z = (z_1, ..., z_H)$ et $z^{'} = (z_1^{'}, ..., z_{H'}^{'})$ les deux partitions de I éléments et définissons

- a est le nombre de paires appartenant au même groupe dans z.
- b est le nombre de paires appartenant au même groupe dans z mais n'appartenant pas au même groupe dans $z^{'}$.
- c est le nombre de paires appartenant au même groupe dans $z^{'}$ mais n'appartenant pas au même groupe dans z.
- d est le nombre de paires appartenant à des clusters différents à la fois dans z et dans $z^{'}$.

Alors, par définition,

$$ARI(z, z^{'}) = \frac{2(ad - bc)}{b^2 + c^2 + 2ad + (a + d)(b + c)}.$$

Notons que le $ARI(z, z^{'})$ est toujours inférieur à 1, qu'il vaut 1 si les partitions sont identiques à permutations près et qu'il est négatif si la comparaison des deux partitions est pire que de les avoir créer toutes les deux totalement aléatoirement.

Cet indice est de 0.0101746 dans notre étude, ce qui est assez faible mais cohérent avec la matrice de contingence obtenue plus haut : aucune correspondance n'est possible entre les communautés du SBM et celles du STBM.

Superposition des deux classifications

Pour illustrer ce phénomène d'explosion des communautés, nous avons superposé les graphes issus des deux classifications. La figure 22 illustre la répartition des communautés du STBM au sein de chaque communauté du SBM. Elle représente le graphe de du réseau à 143 noeuds agencé pour illustrer la clusterisation issue du SBM : les gènes d'une même communauté SBM sont représentés le long d'un cercle et le coloriage que nous avons effectué caractérise l'appartenance des sommets à la communauté correspondante dans le STBM.

L'hétérogénéité des couleurs dans chaque communauté du SBM nous empêche d'appairer les résultats des deux classifications. Il est toutefois possible d'effectuer des études à l'échelle des clusters du SBM, c'est-à-dire, pour un cluster provenant du SBM, connaître la répartition de ses gènes dans les différentes communautés du STBM. Par exemple Les communautés 3, 4 et 6 du SBM en effectuant la numérotation de la gauche vers la droite et de haut en bas

contiennent au moins un gène de chaque communauté du STBM. En revanche, la deuxième communauté ne contient que 4 communautés parmi les 7 du STBM.

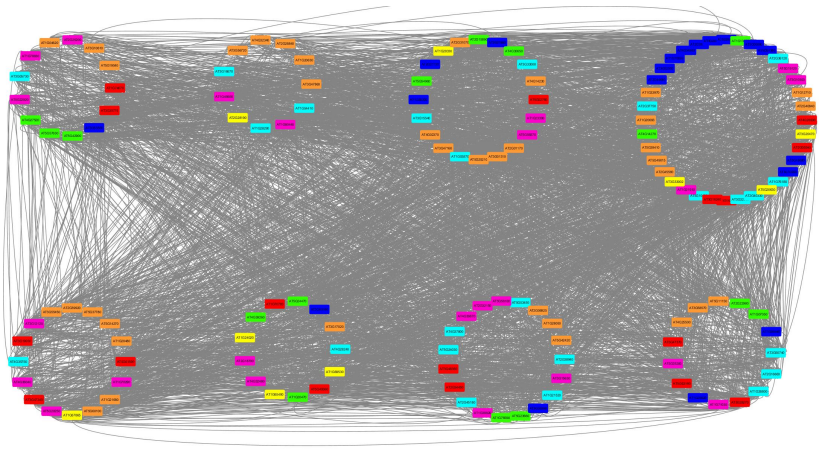


FIGURE 22 – Graphe du petit réseau d'étude sur lequel est représentée la répartition des communautés issues du STBM dans chaque communauté obtenue par le SBM. Les 7 couleurs représentent les 7 clusters du STBM et les 8 cercles illustrent les 8 clusters du SBM.

3.4.2. Différences des clustering

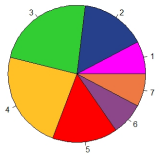
Sur la figure 23 est illustrée la fréquence des clusters STBM dans chaque communauté du SBM. Insistons sur le fait que comme aucune correspondance n'est possible entre les communautés du SBM et celles du STBM, les couleurs peuvent être choisies aléatoirement et il n'y a pas de conséquence à les changer d'une figure à l'autre. Cette figure offre une meilleure visualisation des répartitions des clusters. Ainsi, nous pouvons constater que la communauté 2, séparée en 4 couleurs, est majoritairement portée par la couleur orange et la violette. La communauté 7 est quant à elle portée par 6 communautés avec un poids pratiquement égal pour chacune d'elle. Dans un second temps, interprétons les couleurs. Nous observons ici que la communauté du STBM représentée en rouge est présente de manière significative dans toutes les communautés du SBM, comme c'est le cas aussi pour celles illustrées en violet ou en orange. Les couleurs rose, et bleu sont présentes sur 7 camemberts. En revanche, les couleurs jaune, vertes sont absentes sur les communautés 2 et 8 pour la jaune et 1,6 et 7 pour la verte. Ainsi, certaines communautés semblent être plus explosées que d'autres même si les autres sont séparées en au moins 5 communautés. Ainsi, il n'y a donc pas de structure permettant d'expliquer le clustering du STBM à partir du SBM.

(a) Communauté 1 du SBM

(b) Communauté 2 du SBM

(c) Communauté 3 du SBM

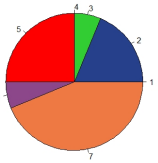
(d) Communauté 4 du SBM



(e) Communauté 5 du SBM

(f) Communauté 6 du SBM

(g) Communauté 7 du SBM



(h) Communauté 8 du SBM

FIGURE 23 – Description des 8 communautés du SBM en fonction des communautés du STBM

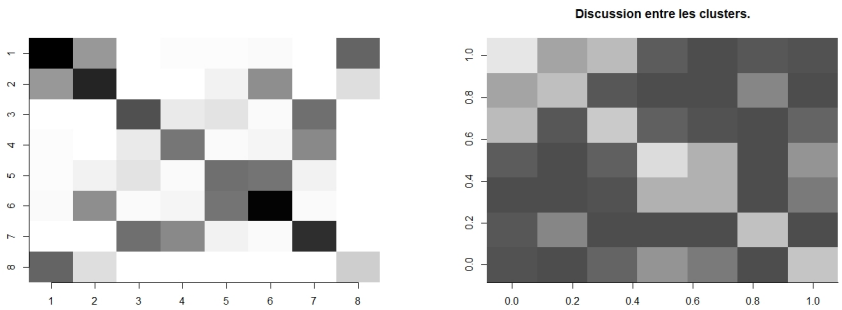
Tout ceci nous certifie que l'application de l'algorithme STBM a bouleversé complètement la structure obtenue par le SBM et qu'il n'est pas possible de comparer les communautés du SBM à celles du STBM.

Étudions maintenant ce qu'il en est à l'échelle des connexions entre les communautés. En effet, il est intéressant de déterminer quelle classification entre le SBM et le STBM a été le plus facile à effectuer dans le sens d'attribution d'un

gène à un cluster d'appartenance.

3.4.3. Etude du poids des arêtes

Nos intuitions ont engendré deux scénarios possible :
— soit l'ajout des stress a permis une meilleure classification dû au fait que l'information disponible était plus riche et donc la distinction entre des gènes et des connexions étaient plus faciles.
— Soit au contraire, avoir plus d'informations empêche une classification bien distincte puisque le nombre de critère à discriminer est plus grand.



(a) Probabilités de connexions intra et extra clusters pour le SBM. En allant du blanc au noir, les connexions sont de plus en plus fortes.

(b) Probabilités de connexions intra et extra clusters pour le STBM. En allant du blanc au noir, les connexions sont de plus en plus faibles.

FIGURE 24 – Probabilités de connexions intra et extra clusters.

Après vérification, il s'agit du deuxième phénomène qui s'est produit en passant du SBM au STBM. En effet, nous avons comparé les probabilités de connexions entre les communautés du SBM et du STBM respectivement. La figure 24 (a) décrit les probabilités de connexions des communautés de la classification par le SBM et la figure 24 (b) représente les probabilités des liens entre les clusters de la classification issue du STBM. Notons que la légende de couleur n'est pas la même entre les deux figures : à gauche, plus la couleur est noire, plus les connexions sont fortes tandis qu'à droite, plus les connexions sont importantes, plus la couleur devient claire. Le contraste des couleurs entre la diagonale et le reste est beaucoup plus frappant chez le SBM que chez le STBM, ceci signifie que les gènes au sein d'une communauté chez le SBM sont fortement liés entre eux et le sont beaucoup moins avec d'autres gènes d'autres clusters. En revanche, bien que la diagonale porte les plus fortes probabilités, les variances de gris sont plus homogènes chez le STBM. Ainsi, les arêtes extra clusters sont de poids beaucoup plus gros chez le STBM ce qui suggère que les gènes communiquent beaucoup avec d'autres gènes n'appartenant pas à leur classe et que la distinction des communautés a été plus difficile. D'ailleurs, nous pouvons faire le lien avec les résultats des sélections de modèle obtenus par les deux algorithmes : SBM a trouvé 8 communautés tandis que STBM en a trouvé que 7. Il était déjà légitime de penser que l'apport des stress discriminerait moins les gènes entre eux et que la classification engendrerait à des chemins plus courts

entre les communautés.

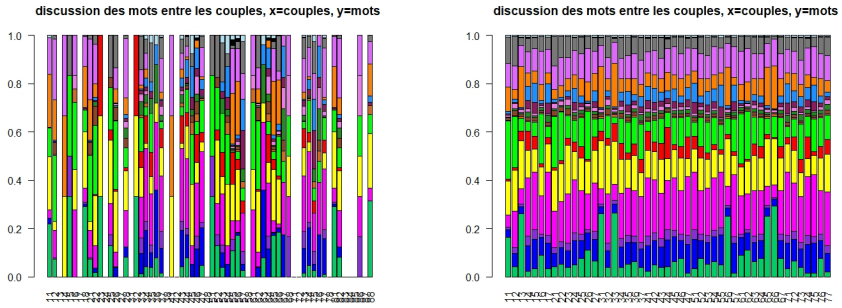
Ainsi, l'information des stress à engendrer une homogénéisation de la matrice de connexions entre les clusters dû au fait qu'en plus d'être classés par la présence ou non de liens, les gènes sont regroupés par rapport à la nature de ceux-ci. Les arêtes étant plus riches, les similarités possibles entre les gènes sont donc plus probables. Nous pourrions alors pousser l'étude à l'échelle des stress c'est-à-dire étudier la nature des arêtes entre les communautés issues des deux algorithmes. En effet, nous pourrions nous demander si l'information sur les arêtes n'est pas implicitement pris en compte dans le SBM.

3.4.4. Étude à l'échelle des stress

Nous souhaiterions, dans cette sous-session, pouvoir comparer les échanges de stress provenant du STBM avec ceux du SBM. Comme ce dernier algorithme ne prend pas en compte cette information, nous avons décidé de l'inclure a posteriori, une fois la classification réalisée. Il sera alors possible de comparer les résultats des natures de liens reliant les clusters pour les deux algorithmes.

Pour ce faire, nous travaillons à paire de clusters du SBM fixée et, afin de créer une nature de connexion pour cette paire, nous revenons à la base de données du réseau et nous ne retenons que les vecteurs des stress partagés sur les arêtes reliant deux gènes appartenant chacun à un cluster de la paire considérée.

Ceci nous permet de créer un vecteur de comptage de taille 18 où chaque coordonnée a pour valeur le nombre d'apparition du stress correspondant à cette coordonnée dans tous ces vecteurs. Après normalisation, nous obtenons un vecteur des fréquences de chaque stress pour la connexion entre les clusters de la paire. Ceci fait pour chaque paire de communautés (y compris au sein d'un même cluster), nous obtenons une classification ayant les mêmes types d'arêtes que celles issues du STBM.



(a) Modèle du SBM (b) Modèle du STBM

FIGURE 25 – Caractérisation des connexions entre les communautés au travers des partages de stress. Sont représentés tous les couples $(q, r) \in [1, Q] \times [1, Q]$ où $Q = 8$ pour le SBM et $Q = 7$ pour le STBM. Les 18 couleurs représentent les 18 stress dont l'ordre est défini dans le tableau à gauche de la figure 17. La première couleur verte du bas représente le stress "champignon" et ainsi de suite.

Nous comparons les partages de stress grâce à la figure 25. Notons que contrairement à la figure 18, nous ne pouvons pas nous contenter de représenter uniquement les paires de couple (q, r) où $q \leq r$. Ceci est expliqué car comme les communautés SBM/STBM ne peuvent pas être appairées, l'ordre est arbitraire et il est important de pouvoir distinguer facilement les 8 colonnes pour une communauté du SBM afin de pouvoir les comparer à toutes les paires de 7 colonnes correspondant à chaque cluster provenant du STBM.

Ceci rend la lecture des graphiques plus compliquée, néanmoins, nous pouvons tout de même constater une régularité dans les couleurs chez le STBM que nous n'observons pas chez le SBM. En effet, il est difficile de révéler une tendance sur la figure de gauche alors que sur celle de droite, nous pouvons affirmer que la couleur rose (représentant le stress température) est présent dans toutes les connexions, presque tout le temps de façon significative, que la couleur jaune (pour le stress virus) est homogène, tout comme la couleur violette ou encore que la couleur verte décrivant le stress champignon, n'est portée fortement que sur quelques arêtes et en analysant de plus près, elle concerne les communautés 3 et 6. Aucune structure n'est observable du côté du SBM : par exemple, si nous portons notre regard uniquement sur la première communauté, la couleur verte du stress "champignon" est présente sur les colonnes 1, 2, 5, 8, la couleur jaune de manière significative pour les colonnes 1, 4, 6 et 8 et légèrement pour la 2, quant à la couleur orange, nous ne l'observons que sur les colonnes 1, 2 et 4. Il est ainsi impossible de pouvoir caractériser les stress portés par la communautés 1. Remarquons que l'observation des stress dominants est plus compliqué avec la figure de gauche : on distingue la prépondérance des couleurs rose, verte, jaune ou encore violette mais de manière nettement plus difficile.

Observons enfin qu'un certain nombre non négligeable de connexions n'est porté par aucun stress, ce qui signifie que dans la base de données, toute arête reliant deux gènes de chaque cluster de la paire en question est inexistante. Ceci rejoint la conclusion faite à la sous-session précédente : le SBM classifie de manière beaucoup plus brutale que le STBM permettant l'absence de liens totale entre gènes de certaine paire de communautés. Enfin, notons évidemment la différence globale entre les deux figures ce qui certifie que le SBM ne prend pas en compte implicitement l'information des arêtes et justifie une fois de plus l'intérêt de l'application d'un STBM quand nous souhaitons tenir compte de l'information portée sur les arêtes.

En conclusion, l'ajout des stress a explosé totalement la classification réalisée par l'algorithme SBM. Le nouveau clustering obtenu est porté par des arêtes de poids plus élevés ce qui engendre à une distinction des classes plus difficile. Les gènes communiquent beaucoup plus au travers des communautés et ceci par des liens propre à la résolution de l'algorithme STBM. Ceci permet de conclure que les stress prennent une place incommensurable dans la classification des gènes.

4. Conclusion et Discussion

Afin de pouvoir améliorer l'annotation fonctionnelle de certains gènes chez *Arabidopsis thaliana*, nous avons cherché à résumer un réseau de co-expression assez conséquent et présentant une structure cachée ceci en créant des communautés de gènes. Un *Stochastic Block Models* a été réalisé au préalable mais ne prenant pas en compte toute l'information disponible sur le jeu de données. C'est pourquoi, nous avons cherché à inclure cette nouvelle connaissance, à savoir la nature des stress portée par les arêtes. Pour cela, nous avons appliqué un *Stochastic Topic Block Model* initialement développé pour l'étude de réseau d'échanges de mail entre individus.

L'application de cette modélisation sur un réseau de co-expression de 143 gènes a été réalisée grâce au logiciel *Linkage*. Nous nous sommes impliquées dans une analyse fine des résultats obtenus pour comprendre les modifications apportées par l'ajout de l'information des arêtes. D'abord, nous avons observé que nous sommes passés de 8 communautés avec le SBM à 7 communautés avec le STBM, puis en étudiant à l'échelle des clusters, nous avons trouvé que le STBM a totalement explosé la structure bâtie par le SBM et que les connexions du STBM entre les communautés étaient beaucoup plus fortes. Enfin, une étude à l'échelle des stress nous a permis de justifier l'implication de cette nouvelle information dans la classification.

Notre travail a ainsi été divisé en trois grandes parties. La première consistait en la compréhension de la question biologique sur laquelle reposait notre projet. S'est ensuite suivie une importante phase de compréhension des différents modèles et des inférences sur lesquels nous avons travaillé et nous nous sommes questionnées tout au long du projet. Enfin, la dernière étape était consacrée à l'application des algorithmes sur un jeu de données réelles qui a été suivie d'une analyse et d'une comparaison, conséquentes des résultats obtenus.

Nous avons contribué à ce projet d'annotation fonctionnelle en appliquant l'algorithme STBM. Nous avons pu apporter une réponse quant à la pertinence de l'ajout de l'information des arêtes. En effet, notre encadrant biologiste Etienne Delannoy, qui ne se s'était encore jamais impliqué dans un tel modèle, a pu valider nos résultats : le résultat du STBM permet de déterminer les gènes répondant aux stress mais aussi de déterminer où ceux-ci agissent (par exemple, chloroplaste), tandis que le SBM ne permet que d'obtenir les réponses au stress. Ainsi, des conclusions ont pu être apportées sur le petit réseau, ce qui est encourageant pour une éventuelle application sur le gros réseau de co-expression puisque sa variabilité est encore plus importante que celle du petit réseau. De plus, nos questionnements et nos discussions ont suscité l'intérêt de Pierre Latouche pour combiner son algorithme et la problématique biologique, ceci engendrant la naissance d'un nouveau projet.

Nous nous sommes néanmoins heurtées à plusieurs difficultés tout au long de ce projet qui ont ralenti nos avancées. En effet, sur le plan théorique, les modèles mathématiques que nous avons étudiés nous ont été présentés en cours tard dans l'année, ce qui nous a mené à une compréhension individuelle et personnelle de la théorie. De plus, avant de travailler sur le petit réseau présenté dans ce rapport, nous nous étions basées sur un autre petit réseau sur lequel nous avons tenté d'appliquer les comparaisons SBM-STBM. Cependant, nous nous sommes rendues compte plus tard que ce choix n'était pas pertinent : il y

avait peu de variabilité dans les stress partagés et donc l'algorithme du STBM classait presque comme l'algorithme du SBM. Presque aucune discrimination n'était causée par la nature du stress. Enfin, le logiciel R nous était inconnu avant cette année, ce qui a nécessité une prise en main au préalable, ce qui a ralenti l'analyse des résultats d'application des deux algorithmes.

A ce jour, certaines de nos questions demeurent non élucidées. La première étant l'adaptation du contexte au modèle du STBM. En effet, nous nous sommes demandées si cet algorithme faisait intervenir le nombre de stress partagés entre deux gènes, en plus de la nature de ses stress. En particulier, une arête portée par un seul stress semble plus spécifique et donc plus importante dans la classification. Or, le STBM donne l'impression de privilégier les arêtes portées par plusieurs stress puisque l'information est plus grande. Par comparaison, dans le modèle de Pierre Latouche sur les échanges de mail, la question est de savoir si l'échange d'un seul long mail entre deux personnes est perçu de la même façon aux yeux de l'algorithme que plusieurs échanges de petits mails entre ces deux personnes. Nous pouvons peut être proposer l'ajout d'une couche bayésienne à ce modèle permettant de prendre en compte le nombre de mails échangés, où, dans ce cas, il serait pertinent de considérer comme proche deux individus échangeant une multitude de petits mails plutôt que deux autres n'échangeant qu'un seul long mail. Dans notre contexte biologique, nous serions tentées de vouloir privilégier les arêtes spécifiques, c'est-à-dire portées par moins de stress. Il serait peut être possible de ne plus considérer un modèle de Bernoulli pour les valeurs quantitatives mais plutôt un modèle de Poisson qui prendrait en compte le comptage des stress portés par chaque arête. Un second questionnement concerne l'application de notre jeu de données sur Linkage. En effet, du fait de la stochasticité de l'algorithme, nous l'avons lancé 8 fois et nous avons conservé celui qui comportait le maximum des ICL. Cependant, en étudiant ces 8 estimations du jeu de données, nous constatons que les nombres de clusters estimés étaient $Q = 2$, $Q = 10$, $Q = 7$, $Q = 10$, $Q = 8$, $Q = 10$, $Q = 7$ et $Q = 7$, ce qui est assez variable. Nous suggérons que, peut être, le lancé de nouvelles simulations nous donnerait un nouveau ICL maximum et ainsi une nouvelle classification qui différerait de celle sur laquelle nous avons basé toutes nos comparaisons.

A. Codes R

```
rm(list=objects())
## Download and install the package
install.packages("blockmodels")
install.packages("igraph")
install.packages("scatterplot3d")
install.packages("plot3D")
install.packages("plot3Drgl")
install.packages("fields")
install.packages("wordcloud") # générateur de word-cloud
install.packages("RColorBrewer") # Palettes de couleurs
intall.packages("mclust")  # Pour ARI
#Charger
library(igraph)
library(blockmodels)
library(scatterplot3d)
library(plot3D)
library(plot3Drgl)
library(fields)
library(wordcloud)
library(RColorBrewer)
library(mclust)


############################## SBM ################################################

setwd("C:/Users/Perrine/Documents/MSV/Projet_SBM/Pratique")
getwd

reseau <- read.table("subnet.txt",sep="",header = FALSE,row.names=NULL,col.names =
c("gene1","gene2","stress_partages"))

############################## OBTENTION DU SBM

## Construire la matrice d'adjacence
edges=data.frame(reseau$gene1,reseau$gene2)
graphe=graph_from_data_frame(edges,directed=FALSE,vertices=NULL)     #directed=FALSE car graphe non orienté.
Adj=as_adjacency_matrix(graphe,sparse=FALSE)
par(mfrow=c(1,1))
image(Adj,col=tim.colors(64), main = "Matrice d'adjacence de connexion entre les gènes - SBM")   # Symétrique

## Vérifier Adj est symétrique
isSymmetric.matrix(Adj)

## Estimer le SBM
reseau_SBM=BM_bernoulli("SBM_sym",Adj,verbosity=0,autosave='', plotting=character(0))
reseau_SBM$estimate()        # Pour chaque valeur de Q, SBM plusieurs fois et ensuite on prend le max pour ICL

###############################

resultat_SBM_dense <- reseau_SBM

############################## PREMIERES ANALYSES

## Nombre de clusters obtenu
Qmax=which.max(resultat_SBM_dense$ICL)
```

```
## voir la courbe ICL
par(mfrow=c(1,1))
plot(resultat_SBM_dense$ICL, main = "Courbe ICL issue de l'algorithme SBM")
max_icl=max(resultat_SBM_dense$ICL[seq(5,12, by = 1)])
col1 = as.numeric(max_icl > resultat_SBM_dense$ICL[seq(5,12, by = 1)]) * 4
col2 = as.numeric(max_icl <= resultat_SBM_dense$ICL[seq(5,12, by = 1)]) * 2
col = col1 + col2
plot(seq(5,12, by = 1),resultat_SBM_dense$ICL[seq(5,12, by = 1)], col = col, pch = 19)

# Proba de connexion entre les clusters:
par(mfrow=c(1,1))
resultat_SBM_dense$plot_parameters(Qmax)
resultat_SBM_dense$model_parameters[[8]]

# Proba que les gènes appartiennent aux clusters
resultat_SBM_dense$memberships[[8]]$Z   # Proba que chaque gene appartient a chaque cluster: 8 colonnes et
nombre de gènes lignes.
ou_est_gene_SBM=apply(resultat_SBM_dense$memberships[[8]]$Z,1,which.max)

################################

############################################## graphe du SBM ####################################

par(mfrow=c(1,1))
g_SBM <- graph.data.frame(edges, directed=FALSE)

vertexCol <- ou_est_gene_SBM
vertexCol[vertexCol == 1] = "red"
vertexCol[vertexCol == 2] = "pink"
vertexCol[vertexCol == 3] = "blue"
vertexCol[vertexCol == 4] = "green"
vertexCol[vertexCol == 5] = "yellow"
vertexCol[vertexCol == 6] = "grey"
vertexCol[vertexCol == 7] = "brown"
vertexCol[vertexCol == 8] = "purple"

plot(g_SBM, vertex.color= vertexCol, shape='circle', vertex.label=NA)

print(vertexCol)

V(g_SBM)$degre <- degree(g_SBM) #degre des sommets (V comme vertex)
E(g_SBM)$between <- edge.betweenness(g_SBM) #attributs des liens (E comme edges)
g_SBM$densite <- graph.density(g_SBM) #densite du graphe: ratio entre le nombre de liens présents et le nombre
de liens possibles.

plot(log(seq(1,54, by = 1)), degree.distribution(g_SBM), xlab = 'degrés', ylab = 'frequence de gène par nombre de
degré', main = "proportion de sommets ayant 0,1,2,..., etc degrés")  # proportion de sommets ayant un degré zéro,
un, deux etc
transitivity(g_SBM)  # Elle est élevée


## Ordre des gènes pour igraph
rownames(Adj)==names(degree(g_SBM))

################################
```

```
########################################## Etude des stress  ##############################

# On voudrait voir si les stress ne sont pas finalement cachés implicitement dans l'EMVB
# L'objectif est de voir si le STBM apporte vraiment de l'information par rapport au SBM.
# On cherche à retourner un boxplot de la proportion de partage de chaque stress entre toutes les communautés.

# On a 8 clusters

proba_stress_par_couple=matrix(0,18,8*8)
titre_stress = c( 'FUNGI', 'NITROGENROOT' , 'NECROTROPHICBACTERIA' , 'TEMPERATURE',
                  'VIRUS' , 'DROUGHT' , 'BIOTROPHICBACTERIA' , 'UV', 'STIFENIA' ,
                  'RHODOCOCCUS' , 'OTHERABIOTIC' , 'SALT' , 'HEAVYMETAL' ,
                  'OXYDATIVESTRESS', 'OOMYCETE' , 'GAMMA' , 'NEMATODES' , 'OTHERBIOTIC')

## Transformer les données en chaîne de caractères.
caractere_stress=lapply(reseau$stress_partages,toString)

## Séparer et Recoller les noms séparés par '.'
caractere_stress_sans_point=lapply(caractere_stress, function(x) paste(unlist(strsplit(x,".", TRUE)) , collapse = ""))

## Séparer et Recoller les noms séparés par '-'
caractere_stress_sans_tiret=lapply(caractere_stress_sans_point, function(x) paste(unlist(strsplit(x,"-")) , collapse =
""))

## Séparer les stress
caractere_stress_separes=lapply(caractere_stress_sans_tiret, function(x) unlist(strsplit(x,";")))


## On travaille avec igraph

# On sélectionne le graphe: sommets, arêtes, nombre de stress partagés
sub2_non_parlant=graph_from_data_frame(reseau,directed=FALSE)

# A chaque gène (nœud), on attribut un numéro, un label pour indiquer les communautés
sub2_communaute=set_vertex_attr(sub2_non_parlant,"communaute",value=ou_est_gene_SBM)   # on crée une
nouvelle colonne "communauté"

## Extraction des arêtes entre 2 communautés
res.freq=list()
cpt=1
for (i in 1:8)  # on fixe le cluster 1
{
  for (j in 1:8)    # On fixe le cluster 2
  {
    if (dim(table(unlist(strsplit(E(sub2_communaute)[ which(V(sub2_communaute)$communaute==i) %--%
which(V(sub2_communaute)$communaute==j)]$stress_partages,";"))))!=0)
    # ie on n'a pas qu'aucun duo de gènes de i et j partagent aucun stress.
      {
      res.freq[[cpt]]=data.frame(i,j,table(unlist(strsplit(E(sub2_communaute)[
which(V(sub2_communaute)$communaute==i) %--%
which(V(sub2_communaute)$communaute==j)]$stress_partages,";"))),nb.aretes=length(E(sub2_communaute)[
which(V(sub2_communaute)$communaute==i) %--% which(V(sub2_communaute)$communaute==j)]))
      # On construit une liste de liste
    }
    cpt=cpt+1
  }}
```

```
# table(unlist(strsplit(E(sub2_communaute)[ which(V(sub2_communaute)$communaute==i) %--%
which(V(sub2_communaute)$communaute==j)]$stress_partages,";")))
# which(V(sub2_communaute)$communaute==i): renvoie l'indice des gènes dans la communauté i
# strsplit(E(sub2_communaute)[ which(V(sub2_communaute)$communaute==i) %--%
which(V(sub2_communaute)$communaute==j)]$stress_partages,";")
#      On prend les indices des gènes se trouvant à la fois dans i et à la fois dans j
#      on donne la colonne 'stress partagés' pour ces gènes là
# table: compte le nombre de fois que chaque stress ait apparu entre les deux communautés

# "c" pour dire que c'est un character; "n" pour dire que c'est un nombre.


# On veut donner des pourcentages de partage de stress: pour chaque duo, on compte le nombre de partage total et
on divise chaque partage de stress par le pourcentage global
# On crée une matrice de taille 8*8 colonnes et 18 lignes pour le boxplot

res.freq_pourcentage=list()
proba_stress_par_couple=matrix(0,18,8*8)
titre_stress = c( 'FUNGI', 'NITROGEN.ROOT' , 'NECROTROPHIC.BACTERIA' , 'TEMPERATURE',
                  'VIRUS' , 'DROUGHT' , 'BIOTROPHIC.BACTERIA' , 'UV', 'STIFENIA' ,
                  'RHODOCOCCUS' , 'OTHER-ABIOTIC' , 'SALT' , 'HEAVY.METAL' ,
                  'OXYDATIVE.STRESS', 'OOMYCETE' , 'GAMMA' , 'NEMATODES' , 'OTHER-BIOTIC')

for (i in 1:length(res.freq))
{
  if (length(res.freq[[i]]) !=0)
  {
    res.freq_pourcentage[[i]]=data.frame(res.freq[[i]],Occ.stress=sum(res.freq[[i]]$Freq))
  res.freq_pourcentage[[i]]=data.frame(res.freq_pourcentage[[i]],pourcentage.stress=res.freq_pourcentage[[i]]$Freq
/ res.freq_pourcentage[[i]]$Occ.stress)
    for (k in 1:18)
    {
      if (is.element(titre_stress[k],unlist(res.freq_pourcentage[[i]]$Var1))==TRUE)
      {
        indice=which(res.freq_pourcentage[[i]]$Var1==titre_stress[k],arr.ind=TRUE)
        proba_stress_par_couple[k,i]=res.freq_pourcentage[[i]]$pourcentage.stress[indice]
      }
    }
  }
}


# data.frame(i,j,table(unlist(strsplit(E(sub2_communaute)[ which(V(sub2_communaute)$communaute==i) %--%
which(V(sub2_communaute)$communaute==j)]$stress_partages,";"))),nb.aretes=length(E(sub2_communaute)[
which(V(sub2_communaute)$communaute==i) %--% which(V(sub2_communaute)$communaute==j)]))
# Créer un data frame à 18*i*j lignes: une pour chaque stress de chaque paire de couples.
# Première colonne: i
# Deuxième colonne: j
# Troisième colonne: le stress
# Quatrième colonne: la condition du "if": pour chaque stress, le nombre de pair entre i et j partageant ce stress
# Cinquième colonne: le nombre d'arête formée par paire de i et j

q1=c(11:18)
q2=c(21:28)
q3=c(31:38)
```

```
q4=c(41:48)
q5=c(51:58)
q6=c(61:68)
q7=c(71:78)
q8=c(81:88)
titre_couple_mots=c(q1,q2,q3,q4,q5,q6,q7,q8)
colnames(proba_stress_par_couple)=titre_couple_mots
rownames(proba_stress_par_couple)=titre_stress

apply(proba_stress_par_couple,1,sum)
apply(t(proba_stress_par_couple)[seq(1,57,by=8),],1,sum)
apply(t(proba_stress_par_couple)[1:8,],1,sum)

par(mfrow=c(1,1))
image(t(proba_stress_par_couple), col=tim.colors(64), xlabel="couple(q,r)", ylabel= titre_stress , main =
"Probabilités d'échanges de stress entre cluster.")
boxplot(proba_stress_par_couple,las=2, main="discussion des mots entre les couples")
couleur_18_stress=colors()[c(613,26,31,451,653,552,254,52,588,139,510,459,128,91,463,200,24,401)]
barplot(proba_stress_par_couple,las=2, main="discussion des mots entre les couples, x=couples, y=mots ", col=
couleur_18_stress)

## On enlève les répétitions:
sans_repetition=1:8
for (k in 1:7)
{
  l=8*k+k+1
  c=seq(l,l+(7-k),by=1)
  sans_repetition=c(sans_repetition,c)
}

sans_repetition

proba_stress_par_couple_sans_repetition=t(proba_stress_par_couple)[sans_repetition,]

image(proba_stress_par_couple_sans_repetition, col=tim.colors(64), xlabel="couple(q,r)", ylabel= titre_stress , main
= "Probabilités d'échanges de stress entre cluster.")
barplot(t(proba_stress_par_couple_sans_repetition),las=2,col=1:8, main="discussion des mots entre les couples,
x=couples, y=mots ")

################################


########################################## On étudie les probabilités a posteriori ##################
## On veut savoir avec quelle probabilité tel gène est affecté dans telle communauté.
## Le but est de pouvoir comparer ces probabilités avec celles du STBM pour voir avec quelle méthode, on classifie
le mieux.

proba_gene_cluster=apply(resultat_SBM_dense$memberships[[8]]$Z,1,max)
plot(1:143,proba_gene_cluster,main="probabilités d'appartenance des gènes dans les
communautés",col="purple",type="o",xlab = "gènes",ylab = "probabilite a posteriori")

## Représentation globale
compte=1
loi_a_posteriori=matrix()
iteration_a_posteriori=matrix()
reste=proba_gene_cluster
while (length(reste)>0)
```

```
{
  loi_a_posteriori[compte]=reste[1]
  iteration_a_posteriori[compte]=length(which(reste==reste[1],arr.ind=TRUE))/143
  compte=compte+1
  reste=reste[-which(reste==reste[1],arr.ind=TRUE)]
}

barplot(iteration_a_posteriori,las=2,col=1:8, main="proportion des différentes loi a posteriori ")

################################
```

```
############################## STBM: Resultat de LINKAGE
#############################################

setwd("C:/Users/Perrine/Documents/MSV/Projet_SBM/Pratique")
getwd

reseau <- read.table("subnet.txt",sep="",header = FALSE,row.names=NULL,col.names =
c("gene1","gene2","stress_partages"))

###################### Pertinence du réseau ####################
pertinence <- data.frame(gene1 = reseau$gene1[seq(1,20, by = 1)], gene2 = reseau$gene2[seq(1,20, by = 1)] ,
stress_partages= reseau$stress_partages[seq(1,20, by = 1)] )
pertinence


############################## Prétraitement des données afin de pouvoir le donner à linkage

# Symétrisation du jeu de données

## Transformer les données en chaîne de caractères.
caratere_gene1=lapply(reseau$gene1,toString)
caractere_gene2=lapply(reseau$gene2,toString)
caractere_stress=lapply(reseau$stress_partages,toString)

## Séparer et Recoler les noms séparés par '.'
caractere_stress_sans_point=lapply(caractere_stress, function(x) paste(unlist(strsplit(x,".", TRUE)) , collapse = ""))

## Séparer et Recoler les noms séparés par '-'
caractere_stress_bon=lapply(caractere_stress_sans_point, function(x) paste(unlist(strsplit(x,"-")) , collapse = ""))


reseau_new_sym_espace=data.frame(gene1=c(unlist(caratere_gene1),unlist(caractere_gene2)),gene2=c(unlist(cara
ctere_gene2),unlist(caratere_gene1)),stress_partages=c(unlist(caractere_stress_bon),unlist(caractere_stress_bon)))
write.table(reseau_new_sym_espace,"petit_reseau_dense.txt",row.names=FALSE,col.names=F,quote=3,sep=",")


################################

############################## On fait linkage plusieurs fois pour conserver le meilleur job
############################## avec l'ICL le plus élevé.

## Obtention de tous les ICL.
```

```
tous.les.icl<-list()
endroit=c("3831","3832","3833","3834","3835","3836","3837","3838")
for (h in 1:8)
{
 job=endroit[h]
 res<-list()
 for (i in 2:10)
 { resk <-list()
 for (j in 2:10)
 {
  setwd(paste0('C:/Users/Perrine/Documents/MSV/Projet_SBM/Pratique/petit_reseau_',job,'/k',i,'_q',j,'/raw'))
  resk[j-1] <- scan("crit")
 }
 res[[i-1]]<-resk
 }
 tous.les.icl[[h]]<-res}

huit.icl=data.frame(unlist(tous.les.icl))   # de taille 9*9*8


## Rangeons ces données en matrice de taille 9x(9*8)
## Les 9 lignes correspondent aux 9 valeurs de clusters (de 2 à 10) et les 72 colonnes aux 9*8 nombre de topics

valeurs_ICL=matrix(0,9,9*8)
for (h in 1:8)
{
 for (i in 1:9)
 {
  j=9*(i-1)+1+81*(h-1)
  l=j+9-1
  valeurs_ICL[,9*(h-1)+i] <- t(huit.icl)[j:l]
 }
}

valeurs_ICL

# On trace les ICL max pour chaque (q,k) pour obtenir quelque chose de lisse
max_ICL=matrix(0,9,9)
for (i in 1:9)
{
 for (j in 1:9)
 {
  max_ICL[i,j] <- max(valeurs_ICL[i,seq(j,9*7+j,by=9)])
 }
}

axe_Q=2:10
axe_K=2:10
persp(axe_Q,axe_K,max_ICL, theta = -45, phi = 20, smooth = FALSE, lighting = TRUE, new = TRUE, main =
"Icl_max_pour_chaque_q_k", xlab = "variation de Q", ylab = "variation de K", zlab = "ICL")

## Image interactive
persp3Drgl(axe_K, axe_Q , max_ICL , smooth = FALSE, lighting = TRUE, new = TRUE, main =
"Icl_max_pour_chaque_q_k", xlab = "variation de Q", ylab = "variation de K", zlab = "ICL")

which.max.matrix(valeurs_ICL)
## Le max est pour k=7, q=7 sur le 2ième jeu de données.
```

```
################################

################################ Pour voir la variabilité des courbes ICL à un jeu de données fixé,
## traçons cette courbe pour un jeu de données qui n'a pas le max des max
## On pourra comparer avec celui qui contient le max des max

## Ex: 3837

## ICL STBM

res<-list()
for (i in 2:10)
{ resk <-list()
for (j in 2:10)
{
  setwd(paste0('C:/Users/Perrine/Documents/MSV/Projet_SBM/Pratique/petit_reseau_3837/k',i,'_q',j,'/raw'))
  resk[j-1] <- scan("crit")
}
res[[i-1]]<-resk
}

icl=data.frame(unlist(res))
# k=2, q=2,...,10 ; k=3, q=2,...,10


## On range les valeurs icl dans une matrice de taille KxQ
M_qlignes_kcolonnes=matrix(0,9,9)
for (i in 1:9)
{
  j=9*(i-1)+1
  l=j+9-1
  M_qlignes_kcolonnes[,i] <- t(icl)[j:l]
}


## On trace ICL en 3D (variation de k et de q)
axe_Q=2:10
axe_K=2:10
persp(axe_Q,axe_K,M_qlignes_kcolonnes,theta = -45, phi = 20, smooth = FALSE, lighting = TRUE, new = TRUE, main =
"Icl_max_pour_chaque_q_k", xlab = "variation de Q", ylab = "variation de K", zlab = "ICL")

## image interactive
persp3Drgl(axe_K, axe_Q , M_qlignes_kcolonnes , smooth = FALSE, lighting = TRUE, new = TRUE, main = "graphique
vu de gauche", xlab = "variation de Q", ylab = "variation de K", zlab = "ICL")

################################


################################ On travaille désormais avec le 2ième jeu de données: job_3832

## ICL STBM

res<-list()
for (i in 2:10)
{ resk <-list()
```

```
for (j in 2:10)
{
  setwd(paste0('C:/Users/Perrine/Documents/MSV/Projet_SBM/Pratique/petit_reseau_3832/k',i,'_q',j,'/raw'))
  resk[j-1] <- scan("crit")
}
res[[i-1]]<-resk
}

icl=data.frame(unlist(res))
# k=2, q=2,...,10 ; k=3, q=2,...,10

## On range les valeurs icl dans une matrice de taille KxQ
M_qlignes_kcolonnes=matrix(0,9,9)
for (i in 1:9)
{
  j=9*(i-1)+1
  l=j+9-1
  M_qlignes_kcolonnes[,i] <- t(icl)[j:l]
}

## On trace ICL en 3D (variation de k et de q)
axe_Q=2:10
axe_K=2:10
persp(axe_Q,axe_K,M_qlignes_kcolonnes,theta = -45, phi = 20, smooth = FALSE, lighting = TRUE, new = TRUE, main =
"Icl_max_pour_chaque_q_k", xlab = "variation de Q", ylab = "variation de K", zlab = "ICL")


## Image interactive
persp3Drgl(axe_K, axe_Q , M_qlignes_kcolonnes , smooth = FALSE, lighting = TRUE, new = TRUE, main = "graphique
vu de gauche", xlab = "variation de Q", ylab = "variation de K", zlab = "ICL")

################################

## On travaille désormais avec k=7, q=7
i=7
j=7
setwd(paste0('C:/Users/Perrine/Documents/MSV/Projet_SBM/Pratique/petit_reseau_3832/k',i,'_q',j,'/raw'))


################################ Nuages de mot et étude des échanges de stress

sujet <- scan("topics")

## On range les mots des différents topics dans une matrice de taille 18x7
M_18mots_7topics=matrix(0,18,7)
for (i in 1:7)
{
  j=18*(i-1)+1
  l=j+18-1
  M_18mots_7topics[,i] <- t(sujet)[j:l]
}

titre_stress=c('fungi' , 'nitrogenroot' , 'necrotrophicbacteria' , 'temperature' , 'virus' , 'drought' , 'biotrophicbacteria' ,
'uv' , 'stifenia' ,
'rhodococcus','otherabiotic','salt','heavymetal','oxydativestress','oomycete','gamma','nematodes','otherbiotic')
rownames(M_18mots_7topics)=titre_stress
M_18mots_7topics
```

```
# On ressort la probabilité globale d échange des mots entre clusters ( en fonction des proportions des mots dans chaque topic et la probabilité de chaque topic entre les clusters)
# On fixe k=7 et on regarde dans tous les clusters

thetaqr<-read.table('thetaQR')
(apply(thetaqr,1,sum))
dim(thetaqr)

#thetaqr: les 49 lignes représentent les 49 couples (q,r) pour Q=7. les 7 colonnes repréntent les 7 topics. Proportion d'échange des topics en fonction des couples.
#M_18mots_7topics: 18 lignes pour 18 mots. 7 colonnes pour 7 topics. Dans chaque topic, proportion des mots.

proba.mot.par.couple.de.communaute=as.matrix(thetaqr)%*%t(M_18mots_7topics)
dim(proba.mot.par.couple.de.communaute)
isSymmetric(proba.mot.par.couple.de.communaute)
#proba.mot.par.couple.de.communaute: 49 lignes pour 49 couples (q,r) pour Q=7. 18 colonnes pour les 18 mots. Pour chaque couple, proportion d'échange des mots.
q1=c(11:17)
q2=c(21:27)
q3=c(31:37)
q4=c(41:47)
q5=c(51:57)
q6=c(61:67)
q7=c(71:77)
titre_couple_mots=c(q1,q2,q3,q4,q5,q6,q7)
rownames(proba.mot.par.couple.de.communaute)=titre_couple_mots
colnames(proba.mot.par.couple.de.communaute)=titre_stress

apply(proba.mot.par.couple.de.communaute,1,sum)
apply(proba.mot.par.couple.de.communaute[seq(1,43,by=7),],1,sum)
apply(proba.mot.par.couple.de.communaute[1:7,],1,sum)


## On regarde tous les clusters en même temps
image(proba.mot.par.couple.de.communaute, col=tim.colors(64), xlabel="couple(q,r)", ylabel= titre_stress , main = "Probabilités d'échanges de stress entre cluster.")

couleur_18_stress=colors()[c(613,26,31,451,653,552,254,52,588,139,510,459,128,91,463,200,24,401)]
boxplot(proba.mot.par.couple.de.communaute,las=2, main="discussion des mots entre les couples", col = couleur_18_stress)
barplot(proba.mot.par.couple.de.communaute,las=2, main="discussion des mots entre les couples", col = couleur_18_stress)
barplot(t(proba.mot.par.couple.de.communaute),las=2, main="discussion des mots entre les couples, x=couples, y=mots ", col = couleur_18_stress)

# fungi
# nitrogenroot
# necrotrophicbacteria
# température
# virus
# drought
# biotrophicbacteria
# uv
# stifenia
# rhodococcus
# otherabiotic
```

```
# salt
# heavymetal
# oxydativestress
# oomycete
# gamma
# nematodes
# otherbiotic


## On enlève les répétitions:
sans_repetition=1:7
for (k in 1:6)
{
  l=7*k+k+1
  c=seq(l,l+(6-k),by=1)
  sans_repetition=c(sans_repetition,c)
}

sans_repetition

proba_mot_par_couple_sans_repetition=proba.mot.par.couple.de.communaute[sans_repetition,]

image(t(proba_mot_par_couple_sans_repetition), col=tim.colors(64), xlabel="couple(q,r)", ylabel= titre_stress ,
main = "Probabilités d'échanges de stress entre cluster.")
barplot(t(proba_mot_par_couple_sans_repetition),las=2, main="discussion des mots entre les couples, x=couples,
y=mots ", col = couleur_18_stress)

######### On ne regarde qu'un cluster

## Prenons en compte la probabilité de connexions entre les clusters:
# Car il est inutile de travailler avec des clusters qui n'échangent pas beaucoup.

PI<-read.table('PI')
dim(PI)

numero_cluster=1:7
rownames(PI)=numero_cluster
colnames(PI)=numero_cluster

image(t(PI),col=tim.colors(64), xlabel="q", ylabel= "r" , main = "Discussion entre les clusters.")
image(t(PI),col=grey.colors(64), xlabel="q", ylabel= "r" , main = "Discussion entre les clusters.")
boxplot(PI,las=2, main="discussion entre les clusters")

couleur_cluster=colors()[c(137,86,566,545,76,493,503)]
barplot(t(PI),las=2, col = couleur_cluster, type="pl")


#### Pour comparer avec SBM #####
PI_pour_comparaison_SBM=PI
for (k in 1:7)
{
  PI_pour_comparaison_SBM[k,]=PI[7-k+1,]
}

image(t(PI_pour_comparaison_SBM),col=tim.colors(64), xlabel="q", ylabel= "r" , main = "Discussion entre les
clusters.")
```

```
image(t(PI_pour_comparaison_SBM),col=gray.colors(64), xlabel="q", ylabel= "r" , main = "Discussion entre les clusters.")

# Les cluster 1,3 et 4 font beaucoup d'échanges.

# cluster 1
barplot(t(proba.mot.par.couple.de.communaute[seq(1,43,by=7),]),las=2,type="pl", main="Echange du cluster 1 avec les autres", col = couleur_18_stress) #echange d'un seul cluster avec tous les autres
# cluster 3
barplot(t(proba.mot.par.couple.de.communaute[seq(3,45,by=7),]),las=2,type="pl", main="Echange du cluster 3 avec les autres", col = couleur_18_stress)
# cluster 4
barplot(t(proba.mot.par.couple.de.communaute[seq(4,46,by=7),]),las=2,type="pl", main = "Echange du cluster 4 avec les autres", col = couleur_18_stress)

#matplot(t(proba.mot.par.couple.de.communaute[seq(1,43,by=7),]),las=2,col=1:7)
#matplot(t(proba.mot.par.couple.de.communaute[1:7,]),las=2,col=1:7,type="pl")

## Nuage de mots total

presence_mots=t(apply(proba.mot.par.couple.de.communaute,2,sum))
frequence_mots=presence_mots/49
wordcloud(words = titre_stress, freq=frequence_mots, colors = couleur_18_stress, min.freq=0,max.words=18, random.order=FALSE, rot.per=0)


# Nuage de mots entre les clusters
q=3
r=1
i=7*(q-1)+r

cluster_qr=proba.mot.par.couple.de.communaute[i,]
wordcloud(words = titre_stress, freq=cluster_qr, colors = couleur_18_stress, min.freq=0, max.words=18, random.order=FALSE, rot.per=0)

################################

################################ graphe du STBM

# Obtenir les graphes

par(mfrow=c(1,1))
edges=data.frame(reseau$gene1,reseau$gene2)
g_STBM <- graph.data.frame(edges, directed=FALSE)

cluster_STBM <- scan('clusters')

vertexCol <- cluster_STBM + 1
vertexCol[vertexCol == 1] = "red"
vertexCol[vertexCol == 2] = "pink"
vertexCol[vertexCol == 3] = "blue"
vertexCol[vertexCol == 4] = "green"
vertexCol[vertexCol == 5] = "yellow"
vertexCol[vertexCol == 6] = "grey"
vertexCol[vertexCol == 7] = "brown"

plot(g_STBM, vertex.color= vertexCol, shape='circle', vertex.label=NA)
```

```
print(vertexCol)

## Ordre des gènes pour igraph

setwd(paste0('C:/Users/Perrine/Documents/MSV/Projet_SBM/Pratique/petit_reseau_3832/raw'))
ordre_gene <- read.table("labels",sep="",header = TRUE,row.names=NULL)
ordre_gene_caractere=apply(ordre_gene,toString)
apply(ordre_gene,2,toString)==names(degree(g_STBM))

################################

################################ On étudie les topics

sujets=scan('topics')
# On veut regarder les stress prépondérants dans chaque topics:
stress_preponderants=which(M_18mots_7topics >= 0.1,2)
stress_preponderant_topics1=which(stress_preponderants[,2]==1)
# temperature, virus, gamma
stress_preponderant_topics2=which(stress_preponderants[,2]==2)
# biotrophibacteria, oomycete, gamma
stress_preponderant_topics3=which(stress_preponderants[,2]==3)
# nitrogenroot, temperature, drought, oxydativestress
stress_preponderant_topics4=which(stress_preponderants[,2]==4)
# nitrogenroot, temperature, heavymetal
stress_preponderant_topics5=which(stress_preponderants[,2]==5)
# nitrogenroot, temperature, biotrophibacteria, oomycete
stress_preponderant_topics6=which(stress_preponderants[,2]==6)
# fungi, virus, oxydativestress
stress_preponderant_topics7=which(stress_preponderants[,2]==7)
# temperature, virus, oomycete, gamma

################################

---

################################## Gros graphe: visualisation de notre jeu de données ####################
setwd("C:/Users/Perrine/Documents/MSV/Projet_SBM/Pratique")
getwd

gros_reseau <- read.table("Edges_3_UniqLink_29122014.txt",sep="",header = TRUE,row.names=NULL,col.names =
c("gene1","gene2","nombre_stress_partages","stress_partages"))
par(mfrow=c(1,1))

edges_gros=data.frame(gros_reseau$gene1,gros_reseau$gene2)
gros_graphe <- graph.data.frame(edges_gros, directed=FALSE)

i=5
j=7
setwd(paste0('C:/Users/Perrine/Documents/MSV/Projet_SBM/Pratique/petit_reseau_3832/k',i,'_q',j))
gene_dans_subnet <- read.table("nodes_with_clusters.txt",sep=",",header = TRUE,row.names=NULL,col.names =
c("gene","cluster_dappartenance"))

# On extrait l'ensemble des gènes du gros réseau
gene1_unique=gros_reseau$gene1[which((!duplicated(gros_reseau$gene1))==TRUE)]
```

```
gene2_unique=gros_reseau$gene2[which(is.element(gros_reseau$gene2[which((!duplicated(gros_reseau$gene2))==TRUE)],gene1_unique)==FALSE)]
genes_unique=c(unlist(lapply(gene1_unique,toString)),unlist(lapply(gene2_unique,toString)))[which(!duplicated(c(unlist(lapply(gene1_unique,toString)),unlist(lapply(gene2_unique,toString))))==TRUE)]

sommets=vector("numeric",length(genes_unique))
sommets[which(is.element(genes_unique,unlist(gene_dans_subnet[1]))==TRUE)]=1

vertexColGros <- sommets
vertexColGros[vertexColGros == 1] = "red"
vertexColGros[vertexColGros == 0] = "blue"

plot(gros_graphe, vertex.color= vertexColGros, shape='circle', vertex.label=NA)
print(vertexCol)

##############################


############################## SBM ##############################################

setwd("C:/Users/Perrine/Documents/MSV/Projet_SBM/Pratique")
getwd

reseau <- read.table("subnet.txt",sep="",header = FALSE,row.names=NULL,col.names =
c("gene1","gene2","stress_partages"))

## Construire la matrice d'adjacence
edges=data.frame(reseau$gene1,reseau$gene2)
graphe=graph_from_data_frame(edges,directed=FALSE,vertices=NULL)     #directed=FALSE car graphe non orienté.
Adj=as_adjacency_matrix(graphe,sparse=FALSE)
par(mfrow=c(1,1))
image(Adj,col=tim.colors(64))   # Symétrique

## Vérifier Adj est symétrique
isSymmetric.matrix(Adj)

## Estimer le SBM
reseau_SBM=BM_bernoulli("SBM_sym",Adj,verbosity=0,autosave='', plotting=character(0))
reseau_SBM$estimate()        # Pour chaque valeur de Q, SBM plusieurs fois et ensuite on prend le max pour ICL


resultat_SBM_dense <- reseau_SBM

## Nombre de clusters obtenu
Qmax=which.max(resultat_SBM_dense$ICL)

## voir la courbe ICL
par(mfrow=c(1,1))
plot(resultat_SBM_dense$ICL)

##############################

############################## STBM ##############################################

## On travaille avec le 1er jeu de données: job_3813
## On travaille avec k=6, q=8
```

```
i=5
j=7
setwd(paste0('C:/Users/Perrine/Documents/MSV/Projet_SBM/Pratique/petit_reseau_3832/k',i,'_q',j,'/raw'))

################################

############################## Comparaison SBM-STBM ##############################################

######################### Matrice de contingence###################################

cluster_dans_SBM=apply(resultat_SBM_dense$memberships[[8]]$Z,1,which.max)
cluster_dans_STBM <- scan("clusters")
cluster_dans_STBM = cluster_dans_STBM + 1
matrice_contigence=table(cluster_dans_SBM,cluster_dans_STBM)
colnames(matrice_contigence)=c(1,2,3,4,5,6,7)
matrice_contigence

adjustedRandIndex(cluster_dans_SBM,cluster_dans_STBM)

################################

############################## Dans le but de visualiser les appartenances SBM et STBM simultanément

setwd(paste0('C:/Users/Perrine/Documents/MSV/Projet_SBM/Pratique/petit_reseau_3832/raw'))
ordre_gene <- read.table("labels",sep="",header = TRUE,row.names=NULL)
caratere_gene1=sapply(ordre_gene,toString)

setwd(paste0('C:/Users/Perrine/Documents/MSV/Projet_SBM/Pratique'))
appartenance_SBM_STBM= data.frame(gene=c(unlist(caratere_gene1)), numero_cluster_SBM=cluster_dans_SBM,
numero_cluster_STBM=cluster_dans_STBM)
write.table(appartenance_SBM_STBM,"appartenance_SBM_STBM.txt",row.names=TRUE,col.names=TRUE,quote=3,
sep=",")

## Création de 8 camemberts pour chaque communauté de SBM divisés en les proportions des
## 7 communautés de STBM
couleur_communaute_STBM=colors()[c(450,566,448,148,553,512,586)]
for (i in 1:8)
{
  localisation_i=which(cluster_dans_SBM==i)
  localisation_STBM=cluster_dans_STBM[localisation_i]
  proportion=sapply(1:7, function(x){sum(localisation_STBM==x)})
  pie(proportion, labels = c(1,2,3,4,5,6,7), edges = 200, radius = 0.8, col = couleur_communaute_STBM)
}

################################

############################## Proba que les genes appartiennent aux clusters
#SBM
resultat_SBM_dense$memberships[[8]]$Z   # Proba que chaque gene appartient a chaque cluster: 8 colonnes et
nombre de gènes lignes.
ou_est_gene_SBM=apply(resultat_SBM_dense$memberships[[8]]$Z,1,which.max)
#cluster4
cluster4_SBM=which(ou_est_gene_SBM==4)
c=1:148
proba_appartenance_gene_cluster4_SBM=resultat_SBM_dense$memberships[[8]]$Z[cluster4_SBM]
#cluster2
cluster2_SBM=which(ou_est_gene_SBM==2)
```

```
proba_appartenance_gene_cluster2_SBM=resultat_SBM_dense$memberships[[8]]$Z[cluster2_SBM]

# STBM
CLUSTER <- read.table('clusters')
cluster7_STBM=which(CLUSTER==6)

##############################

########################################## Motivation du STBM plutôt que SBM
#### On va lancer SBM et STBM sur 20 gènes au pif et voir la matrice d'adjacence

setwd("C:/Users/Perrine/Documents/MSV/Projet_SBM/Pratique")
getwd

reseau <- read.table("subnet.txt",sep="",header = FALSE,row.names=NULL,col.names =
c("gene1","gene2","stress_partages"))

## Construire la matrice d'adjacence
edges_20=data.frame(reseau$gene1[seq(1,100, by = 5)],reseau$gene2[seq(1,100, by = 5)])
graphe_20=graph_from_data_frame(edges_20,directed=FALSE,vertices=NULL)      #directed=FALSE car graphe non
orienté.
Adj_20=as_adjacency_matrix(graphe_20,sparse=FALSE)
par(mfrow=c(1,1))
couleur_black_white=colors()[c(1,24)]
image(Adj_20,col=couleur_black_white, main = "Matrice d'adjacence de connexion entre les gènes - SBM")   #
Symétrique

# Symétrisation des jeux de données

## Transformer les données en chaîne de caractères.
caratere_gene1_20=lapply(reseau$gene1[seq(1,100, by = 5)],toString)
caractere_gene2_20=lapply(reseau$gene2[seq(1,100, by = 5)],toString)
caractere_stress_20=lapply(reseau$stress_partages[seq(1,100, by = 5)],toString)

## Séparer et Recoller les noms séparés par '.'
caractere_stress_sans_point_20=lapply(caractere_stress_20, function(x) paste(unlist(strsplit(x,".", TRUE)) , collapse =
""))

## Séparer et Recoller les noms séparés par '-'
caractere_stress_bon_20=lapply(caractere_stress_sans_point_20, function(x) paste(unlist(strsplit(x,"-")) , collapse =
""))

reseau_new_sym_espace_20=data.frame(gene1=c(unlist(caratere_gene1_20),unlist(caractere_gene2_20)),gene2=c(
unlist(caractere_gene2_20),unlist(caratere_gene1_20)),stress_partages=c(unlist(caractere_stress_bon_20),unlist(car
actere_stress_bon_20)))
write.table(reseau_new_sym_espace_20,"reseau_pour_motivation_STBM.txt",row.names=FALSE,col.names=F,quot
e=3,sep=",")

# On peut alors lancer linkage

##############################
```

Références

[1] C. Bouveyron, P. Latouche, and R. Zreik. The stochastic topic block model for the clustering of vertices in networks with textual edges. Statisctics and Computing, 2017.

[2] J.J. Daudin, F. Picard, and S. Robin. A mixture model for random graphs. Statisctics and Computing, 2008.

[3] Christine Keribin. Les méthodes bayésiennes variationnelles et leur application en neuroimagerie : une étude de l'existant. INRIA, 2009.

[4] P. Latouche. Modèles de graphes aléatoires à structure cachée pour l'analyse des réseaux. PhD thesis, Université d'Évry Val d'Essonne, 2011.

[5] Y. Vasseur. Inférence de réseaux de régulation orientés pour les facteurs de transcription d'Arabidopsis thaliana et création de groupes de co-régulation. PhD thesis, Université Paris Sud, 2017.

[6] R. Zaag, J.P. Tamby, C Guichard, Z. Tariq, G. Rigaill, E. Delannoy, J.P. Renou, S. Balzergue, T. Mary-Huard, S. Aubourg, M.L. Martin-Magniette, and V. Brunaud. Gem2net : from gene expression modeling to -omics networks, a new catdb module to investigate arabidopsis thaliana genes involved in stress response. Oxford University Press, 2015.

5 Stage de M2 : Inférence bayésienne sur des modèles de croissance de plantes hétérogènes en interaction.

Université Paris-Saclay

Rapport de stage de fin d'études
du 1er Avril au 30 Août 2019

2018 - 2019
M2 Mathématiques pour les Sciences du Vivant

Inférence bayésienne sur des modèles de croissance de plantes hétérogènes en interaction

Étudiante :
Julie Hémont

Encadrant :
Paul-Henry Cournède

Déclaration d'intégrité relative au plagiat

Je soussignée *Hémont Julie* certifie sur l'honneur :

1. Que les résultats décrits dans ce rapport sont l'aboutissement de mon travail ;
2. Que je suis l'auteur de ce rapport ;
3. Que je n'ai pas utilisé des sources ou résultats tiers sans clairement les citer et les référencer selon les règles bibliographiques préconisées.

Je déclare que ce travail ne peut être suspecté de plagiat.

Date : 01/09/2019

Signature :

Préambule

Ce rapport intitulé *Inférence bayésienne sur des modèles de croissance de plantes hétérogènes en interaction* a été réalisé dans le cadre du stage du Master 2 de *Mathématiques appliquées aux Sciences du Vivant* (MSV) de l'Université Paris-Saclay.

Ce travail a été effectué au sein de l'équipe Biomathematics du Laboratoire de Mathématiques et Informatique pour la Complexité et les Systèmes (MICS) de CentraleSupelec, sous la tutelle de Paul-Henry Cournède, directeur de ce laboratoire, du 1er avril au 30 août 2019 (5 mois).

Remerciements

Mes premières pensées se tournent naturellement vers mon encadrant Paul-Henry Cournède qui m'a ouvert la porte de son laboratoire. Merci de m'avoir accordé votre confiance.

J'ai eu la chance de travailler avec Antonin Della Noce qui a été d'un soutien sans faille. Merci pour ta patience, ton dévouement, ton humour aussi. Merci pour l'énergie que tu as consacré à me guider et parfois même à me tenir tête, c'est cette énergie qui fait avancer la pensée scientifique. Merci aussi pour ta curiosité ; travailler à tes côtés fût enrichissant.

Un grand merci de manière générale à l'équipe du laboratoire MICS pour son accueil et sa bienveillance. Merci de m'avoir donné l'opportunité de travailler avec vous, notamment durant ce séminaire à Corfou. Vous avez su partager avec moi vos expertises scientifiques mais vous m'avez également laissé une place autour du baby-foot. Merci pour les footings dont je ne remets pas, pour la comédie musicale, la découverte de l'escalade et quelques échanges de ping-pong. Merci pour tout.

Plus particulièrement, j'aimerais remercier Mahmoud, Gautier et Andreas pour leur aide lors de mes apprentissages de Julia, Fusion et GitLab. Merci pour votre temps.

Fabienne, tu as su m'aider dès mon arrivée. Ton rire manquera à mes prochaines pauses café.

Je tiens à remercier l'équipe de la station expérimentale de l'INRA à Grignon (78) qui a réalisé l'expérience dont sont issues les données réelles sur lesquelles j'ai eu la chance de pouvoir travailler.

Plus précisément, je remercie Amélie Mathieu de m'avoir fait découvrir le contexte de ces données, d'avoir pris le temps de répondre à mes questions, même les plus banales et d'avoir été présente lors de nos discussions sur les modèles mathématiques.

Je tiens à témoigner toute ma reconnaissance aux étudiants et à l'équipe enseignante et pédagogique du Master MSV, sous la responsabilité de Sylvie Méléard. Au delà de la richesse des enseignements dispensés, ce Master a été une véritable aventure humaine.

Enfin, mes chers amis, vous m'avez donné votre soutien inconditionnel une fois de plus !

Perrine, ce stage n'aurait pas été le même sans toi. Merci pour ces trajets tous les matins et tous les soirs. Merci aussi pour ton oreille attentive et tes conseils avisés. Christian, je te dois mes progrès en LaTeX. Julien, merci pour ce regard sur la vie, j'en aurai toujours besoin.

Table des matières

1 Introduction

1.1 Le laboratoire MICS

Ce stage de recherche a été effectué au sein du laboratoire de **Mathématiques et Informatique pour la Complexité et les Systèmes (MICS)**. [1] Ce laboratoire a été créé pour développer et structurer la recherche en mathématiques appliquées et en informatique de **CentraleSupélec**.

Son axe de recherche principal est la modélisation, la conception et la simulation de systèmes complexes, qu'ils soient issus du vivant, de l'industrie, des systèmes d'information ou des organisations socio-économiques.

Pour atteindre ce but, les chercheurs du laboratoire MICS effectuent leurs recherches dans plusieurs directions complémentaires, parmi lesquelles on peut citer :

- l'étude de propriétés empiriques des systèmes complexes ;
- la modélisation à partir des données ;
- la formalisation de la notion de complexité ;
- l'élaboration d'objets mathématiques représentant des systèmes complexes ;
- la simulation numérique et le calcul scientifique ;
- la visualisation et conceptualisation de données massives, non structurées.

Plus précisément, j'ai été accueillie au sein de l'équipe **Biomathematics**. Cette équipe est formellement née en Septembre 2016, elle hérite en partie de l'ancienne équipe Digiplante, qui se consacrait principalement aux systèmes de la biologie végétale, en élargissant les thématiques vers la santé.

J'ai été encadrée par Paul-Henry Cournède, directeur du laboratoire MICS. J'ai travaillé également avec Antonin Della Noce, doctorant au laboratoire MICS et Amélie Mathieu, Maître de conférences à AgroParisTech. [2]

1.2 Contextes biologique et mathématique.

Même au sein d'une même espèce, les individus présentent des variations génétiques. Pour les plantes, elles s'observent par exemple par une capacité de résistance plus ou moins grande aux maladies, ou aux insectes ravageurs. D'un autre point de vue, au sein d'une parcelle agricole, les plantes sont soumises à des conditions environnementales différentes et la qualité du sol peut varier par endroits. Cette variabilité inter-individuelle peut également être due à des variables exogènes qui sont difficilement observables ; ainsi des plantes intrinsèquement identiques dans un environnement en apparence homogène peuvent exhiber des comportements différents, pouvant être attribués à des micro-variations de l'environnement.

Un intérêt particulier est porté à cette variabilité et plus largement, aux cultures mixtes, où les plantes sont toutes différentes. En effet, les populations hétérogènes, par exemple lorsque l'on mélange différentes variétés de blé

1. Laboratoire MICS, CentraleSupélec, Université Paris-Saclay, 91190, Gif sur Yvette. Lien du site du laboratoire http://mics.centralesupelec.fr/fr/
2. INRA AgroPariTech, Route de la Ferme, 78850, Thiverval-Grignon

[Borg et al., 2018] ou différentes espèces [Tang et al., 2018], semblent présenter des avantages tels que la résistance à certaines maladies ou le transfert d'azote entre les plantes.

Ces exemples montrent la nécessité de tenir compte de la variabilité entre les plantes dans les modèles que l'on utilise. Une possibilité est d'introduire des modèles de population qui permettent de décrire un comportement général de la population tout en préservant l'idée de variabilité entre les individus. Les modèle mixtes permettent de rendre compte de la distribution des caractéristiques individuelles au sein d'une population d'individus. Inférés par exemple par des méthodes de Monte-Carlo par chaîne de Markov (MCMC) et algorithmes Espérance-Maximisation (EM) [Kuhn, 2004], ils trouvent leurs applications dans plusieurs domaines du vivant comme la pharmacodynamique [Comets et al., 2008] ou l'écologie [Bolker et al., 2009] mais également pour les modèles de croissance de plantes [Baey, 2014].

D'autre part, les plantes produisent de la biomasse par photosynthèse. La lumière, l'eau et les nutriments du sol sont des ressources vitales. Lorsque les plantes sont en forte densité, comme dans un champs, elles entrent en compétition pour les ressources. Par exemple, une plante qui débute très tôt sa croissance va faire de l'ombre aux plantes situées autour d'elle qui, sans lumière disponible, vont pousser plus lentement.

Les plantes poussent donc en interaction : la croissance d'une plante dépend de la croissance des plantes qui lui sont voisines. Si les effets de la compétition pour de la lumière ont été étudiés pour des populations homogènes [Cournede et al., 2008], l'influence de la compétition sur la croissance de plantes formant une population hétérogène est, à notre connaissance, peu étudiée.

Les modèles de croissances considérés par la suite sont des modèles dynamiques, pouvant s'écrire sous la forme, soit d'un système différentiel décrivant l'évolution de l'état du système en fonction du temps ; soit d'un modèle à incrément donnant l'état du système au temps $t + \Delta t$ en fonction de l'état au temps t et du taux de croissance associé au pas de temps Δt.

Les plantes sont représentées par des grandeurs caractéristiques telles que la taille de la tige ou la surface foliaire. Les conditions initiales et les paramètres individuels sont aléatoires, identiquement distribués selon des distributions de population. L'interaction est représentée par des facteurs de compétition, calculés entre paires d'individus.

Lors de la prise en compte de l'interaction entre les individus et lorsque la population est importante, l'estimation des paramètres d'un modèle de population hétérogène à partir de données expérimentales est difficile. Une approche bayésienne a été privilégiée dans le cadre de ce travail pour permettre d'intégrer des connaissances *a priori* et pallier au problème de la faible quantité de données usuellement rencontré en estimation fréquentiste. On souhaite donc inférer des modèles hiérarchiques bayésiens, c'est à dire retrouver la loi *a posteriori* π des paramètres du modèle sachant un ensemble d'observations. L'inférence, par méthode de type MCMC, nécessite de simuler toute la population sans possibilité de paralléliser les calculs.

Ce rapport présente dans un premier temps la problématique de l'inférence

bayésienne sur un modèle de population pour un système dynamique. Les différentes méthodes d'inférence bayésienne de type MCMC adaptées aux modèles hiérarchiques sont ensuite détaillées. Les croissances de plantes sont ensuite modélisées ; on présentera ici trois modèles, un modèle sans interaction, que l'on comparera à un modèle similaire avec compétition et un modèle adapté du modèle GreenLab pour le Colza. Les méthodes d'inférence ont été ensuite appliquées à des modèles de population hétérogène représentant des plantes en concurrence, par exemple, pour la lumière, à des données simulées puis à des données réelles provenant d'une expérience menée sur le colza [Baey et al., 2018].

2 Problème d'inférence bayésienne sur un modèle de population pour un système dynamique.

Dans cette première section, les caractéristiques des modèles utilisés par la suite sont détaillées dans le but d'établir la problématique de leur inférence. Les modèles inférés durant ce stage sont présentés dans la section 4.

Modèle d'évolution

La première caractéristique des modèles étudiés est qu'il s'agit de modèles basés sur des systèmes dynamiques. Chaque plante est décrite par son état au cours du temps. Alors, l'état du système est décrit par un système dynamique de la forme suivante :

$$\forall t > 0,\ \forall 1 \leq i \leq N,\ \begin{cases} E_i(0) = e_i^0, \\ \dfrac{dE_i(t)}{dt} = F_t\left(E_i(t), \theta_i, (E_{i'}(t), \theta_{i'})_{1 \leq i' \neq i \leq N}\right), \end{cases} \tag{1}$$

où

- N est le nombre de plantes ;
- chaque plante est indexée par $i \in \{1, \cdots, N\}$;
- E_i est l'**état** de la plante i : ce sont les caractéristiques de la plante qui varient au cours du temps comme par exemple la taille de la tige ou la surface foliaire ;
- θ_i est le vecteur des **paramètres individuels** de la plante i : ce sont les caractéristiques intrinsèques de la plante, supposées invariantes au cours du temps ;
- F_t est une fonction définissant le taux de croissance de la plante i au temps t : elle modélise l'influence de la population, représentée par la variable $(E_{i'}(t), \theta_{i'})_{1 \leq i' \neq i \leq N}$, sur le développement de la plante i.

Modèle de population

L'hétérogénéité de la population est représentée par une mesure de probabilité p_0^θ qui n'est pas réduite à une mesure de Dirac. Cette distribution donne la répartition des valeurs prises par les paramètres individuels et par les états initiaux $\left(e_i^0\right)_{1 \leq i \leq N}$. Les paramètres de cette loi, η_θ, sont qualifiés de *paramètres de population* et font partie des **hyperparamètres** du modèle. Les paramètres individuels et les états initiaux sont alors supposés distribués de manière indépendante selon $p_0^\theta(\cdot|\eta_\theta)$.

La fonction F_t est également paramétrée par des hyperparamètres : les *paramètres d'interaction*, η_F. Ces hyperparamètres sont supposés indépendants des paramètres de population.

Étant donnés des paramètres d'interaction, un ensemble de paramètres individuels $\theta = (\theta_i)_{1 \leq i \leq N}$ et une condition initiale $\left(e_i^0\right)_{1 \leq i \leq N}$, si F_t est suffisamment régulière, le système (1) a une unique solution : $\forall t > 0,\ (E_i(t))_{1 \leq i \leq N}$.

Modèle d'observation

Les observations de ce système sont supposées bruitées. Les observations, effectuées aux temps t_j pour $j \in \{1, \cdots, M\}$, sont donc la somme de la solution du système dynamique au temps t_j et d'un bruit gaussien. Autrement dit,

$$E^o_{i,j} = E_i(t_j) + \varepsilon_{i,j}$$

où $\varepsilon_{i,j} \underset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$.

Les observations $E^o = \left(E^o_{i,j}\right)_{1 \leq i \leq N,\ 1 \leq j \leq M}$ constituent une base de données.

Modèle bayésien

Le problème statistique consiste à identifier les hyperparamètres du modèle, qui donnent la distribution des paramètres individuels et la fonction donnant le taux de croissance F_t en se basant sur les observations collectées.

L'approche bayésienne est privilégiée. Des lois *a priori* sont donc ajoutées sur les hyperparamètres et sur la variance d'observation σ^2.

Modèle graphique

Le modèle construit peut être résumé par un modèle graphique comme sur la figure 2.1.

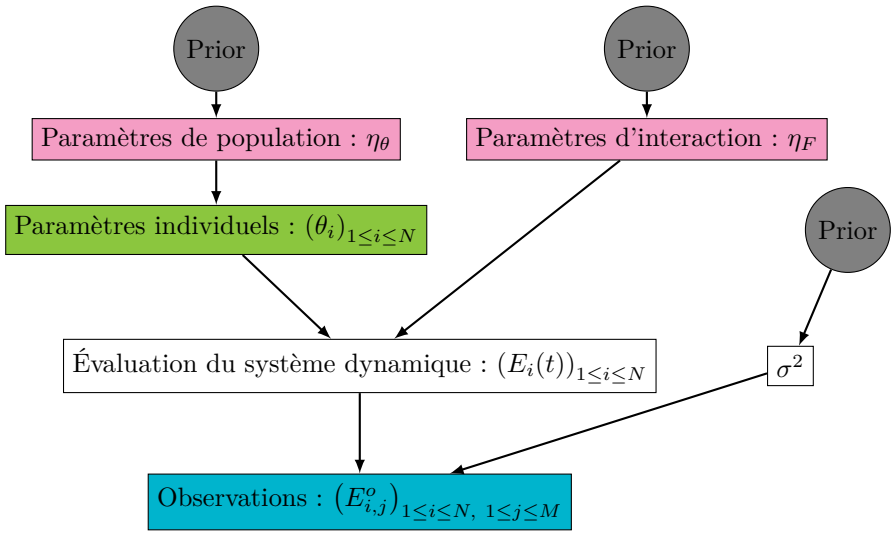


Figure 2.1 – Modèle graphique d'un modèle de population bayésien sur un système dynamique. En bleu, les données, en vert, les paramètres individuels, en rose les hyperparamètres et en gris, les loi *a priori*.

Comme il s'agit d'un modèle bayésien, l'objectif est de déterminer la loi *a posterori*, π, des paramètres du modèle, $x = \left(\eta_\theta, \eta_F, \theta, \sigma^2\right)$ sachant les données, E^o : $\pi\left(x\right) = p\left(\eta_\theta, \eta_F, \theta, \sigma^2 | E^o\right)$.

Par le théorème de Bayes, la densité *a posteriori* est proportionnelle à la densité jointe :

$$p\left(\eta_\theta, \eta_F, \theta, \sigma^2 | E^o\right) = \frac{p\left(E^o, \eta_\theta, \eta_F, \theta, \sigma^2\right)}{p\left(E^o\right)} = \frac{p\left(E^o, \eta_\theta, \eta_F, \theta, \sigma^2\right)}{\int p\left(E^o, x\right) dx} \propto p\left(E^o, \eta_\theta, \eta_F, \theta, \sigma^2\right).$$

Le calcul de la densité *a posteriori* requière le calcul d'une intégrale sur l'espace des paramètres. Ce calcul peut potentiellement être complexe et sur un espace de grande dimension, ce qui le rend infaisable.

Par la règle de Bayes et par les indépendances liées au modèle établi précédemment,

$$p\left(E^o, \eta_\theta, \eta_F, \theta, \sigma^2\right) = \underbrace{\prod_{i=1}^{N}\prod_{j=1}^{M} p\left(E^o_{i,j} | \eta_F, \theta, \sigma^2\right)}_{\text{Vraisemblance}} \times \underbrace{\prod_{i=1}^{N} p\left(\theta_i | \eta_\theta\right)}_{\text{Distribution de population}} \times \underbrace{p(\eta_\theta)p(\eta_F)p(\sigma^2)}_{prior}$$

Lorsque le modèle est complexe, par exemple pour les modèles non-linéaires, cette densité n'est pas calculable. Cependant, il existe des méthodes permettant de simuler une loi π dont on connaît l'expression analytique.

La méthode par acceptation/rejet n'est pas envisageable. En effet, la constante permettant une majoration de la densité par la densité d'une loi auxiliaire doit être connue, ce qui n'est pas possible aux vues des calculs d'intégrales liées à cette densité particulièrement complexe.

On souhaite simuler des vecteurs x suivant une distribution de probabilité π. On cherche donc à avoir une suite de K vecteurs $(x_0, x_1, \ldots, x_{K-1})$ telle que la distribution des x_i approche π. L'idée des méthodes suivantes va être d'utiliser les propriétés d'ergodicité des chaînes de Markov.

Méthodes MCMC

On rappelle ici les éléments de base sur les chaînes de Markov, nécessaires pour la suite. Pour plus de détails, consulter [Tatarinova and Schumitzky, 2015].

Soit $x = (x_0, x_1, \cdots)$ une séquence de vecteurs aléatoires à valeur dans un espace vectoriel E de dimension finie. x est une **chaîne de Markov** lorsque pour tout $A \subset E$, pour tout $k \in \mathbb{N}$,

$$\mathbb{P}(x_{k+1} \in A | x_0, \cdots, x_k) = \mathbb{P}(x_{k+1} \in A | x_k).$$

Une loi π est dite **invariante** (ou **stationnaire**) pour une chaîne de Markov x lorsque pour tout $k \in \mathbb{N}$, si x_k suit la loi π, alors x_{k+1} aussi.

Une chaîne de Markov x de loi invariante π est dite irréductible lorsque pour tout $A \subset E$ tel que $\pi(A) > 0$, pour tout $y \in E$, $\mathbb{P}(\exists k \in \mathbb{N}, x_k \in A | x_0 = y) > 0$. Une chaîne de Markov irréductible atteint donc tout sous-ensemble de mesure non nulle pour sa loi invariante π.

Soit $x = (x_0, x_1, \cdots)$ une chaîne de Markov irréductible de loi invariante π et soit f une fonction sur E à valeurs réelles telle que $\mathbb{E}_\pi[|f|] < 0$. Par le théorème ergodique,

$$\mathbb{P}\left(\frac{1}{K}\sum_{k=0}^{K} f(x_k) \underset{K\to\infty}{\longrightarrow} \mathbb{E}_\pi[f]|x_0 = y\right) = 1,$$

pour π-presque tout $y \in E$.

D'autre part, une chaîne de Markov x est dite **apériodique** lorsqu'il n'existe pas de partition mesurable $B = \{B_0, \cdots, B_{r-1}\}$ pour $r \geq 2$, telle que pour tout $k \in \mathbb{N}$, pour tout $y \in B_0$, $\mathbb{P}\left(x_k \in B_{(k \mod r)}|x_0 = y\right)$.

Le théorème suivant justifie les procédures mises en place par la suite. Soit x une chaîne de Markov irréductible et apériodique, de loi invariante π. Alors, pour π-presque tout $y \in E$, pour tout $A \subset E$,

$$\mathbb{P}(x_k \in A|x_0 = y) \underset{K\to\infty}{\longrightarrow} \pi(A). \tag{2}$$

Autrement dit, si on laisse la chaîne évoluer suffisamment longtemps, on obtient un échantillon de la réalisation de la loi π. En pratique, on écarte les premières itérations, trop dépendantes de l'initialisation de la chaîne.

Étant donnée une loi π, toute méthode consistant à produire une chaîne de Markov de loi invariante π est appelée méthode de *Monte-Carlo par chaîne de Markov* (MCMC).

3 Méthodes d'inférence de Monte-Carlo par chaîne de Markov.

L'objectif de cette partie est d'étudier des méthodes d'inférence qui ont pour but de simuler un échantillon d'une loi, appelée *loi cible*, $\pi(x)$. Lorsqu'il est impossible de générer directement un échantillon indépendant à partir de $\pi(x)$, ce qui se produit notamment quand il est difficile d'intégrer la densité $\pi(x)$ sur son support, on peut opter pour une stratégie d'*échantillonnage d'importance* pour laquelle les échantillons sont générés à partir d'une loi différente de la loi cible puis on attribue des poids à ces échantillons, ou on peut produire des échantillons dépendants basés sur une méthode de Monte-Carlo par chaîne de Markov (MCMC). L'idée est de construire une chaîne de Markov de loi stationnaire $\pi(x)$. Les algorithmes de Metropolis-Hastings et de Gibbs, ainsi que leurs variantes, pour des modèles de croissance de plantes sont développés dans la thèse de G. Viaud [Viaud, 2018].

3.1 Metropolis-Hastings

En 1953, inspiré par les travaux d'Enrico Fermi, le physicien Nicholas Constantine Metropolis propose une méthode de Monte-Carlo par chaîne de Markov, dans le but d'échantillonner selon une distribution de probabilité à densité [Metropolis et al., 1953]. Cette première version considérait le cas particulier de la distribution de Boltzmann, une des distributions les plus utilisées en physique statistique. L'algorithme, généralisé par Hastings [Hastings, 1970] en 1970 à n'importe quelle distribution, est un algorithme itératif : on construit une chaîne de Markov $(x^k)_{k\in\mathbb{N}}$ de distribution stationnaire la densité cible $\pi(x)$.

3.1.1 Formulation mathématique

L'algorithme de Metropolis-Hastings est détaillé dans l'algorithme 1. À l'itération k, l'état x^k est construit et on cherche à construire l'état x^{k+1}. On génère un *candidat* x^* selon une loi *instrumentale* ou *de proposition*, de densité $q(x^k, \cdot)$ ne dépendant que de x^k et facile à simuler comme par exemple une loi normale centrée en x^k et de variance fixée σ^2, qui a pour densité

$$q(x^k, x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x - x^k}{\sigma}\right)^2\right].$$

Là où Metropolis choisit de travailler avec une loi de proposition symétrique, Hastings a étendu l'algorithme à toute loi à densité q telle que si $q(x, y) > 0$, alors $q(y, x) > 0$. Le candidat x^* est alors soumis à une étape d'acceptation/rejet : il est accepté ($x^{k+1} = x^*$) avec une probabilité α et rejeté ($x^{k+1} = x^k$) avec probabilité $1-\alpha$, où α est appelé la *probabilité d'acceptation*. Elle est une fonction de x^* et x^k, et est choisi de manière à assurer la convergence vers la distribution cible :

$$\alpha = \alpha(x^k, x^*) = \min\left\{\frac{\pi(x^*)\ q(x^k, x^*)}{\pi(x^k)\ q(x^*, x^k)}, 1\right\}.$$

Notons que si q est une loi symétrique, alors la probabilité d'acceptation devient :

$$\alpha = \alpha(x^k, x^*) = \min\left\{\frac{\pi(x^*)}{\pi(x^k)}, 1\right\}.$$

Algorithme 1 : Algorithme de Metropolis-Hastings

Choisir une valeur initiale x^0
pour $k = 0 : K$ **faire**
  Simuler $x^* \sim q(x^k, \cdot)$
  Calculer $r(x^k, x^*) = \dfrac{\pi(x^*)\ q(x^k, x^*)}{\pi(x^k)\ q(x^*, x^k)}$
  Simuler $u \sim \mathcal{U}([0, 1])$
  Mettre à jour :

$$x^{k+1} = \begin{cases} x^* & \text{si } u \leq r(x^k, x^*) \\ x^k & \text{sinon} \end{cases}$$

fin
retourner $x^{0:K}$.

La grande souplesse de l'algorithme de Metropolis-Hastings est de ne requérir que la connaissance de la densité π à une constante multiplicative près. En effet, $r(x^k, x^*)$ est le rapport de deux évaluations de la densité ce qui permet de simplifier par la constante de normalisation, qui n'apparaît plus. Cet algorithme est donc particulièrement adapté aux densités que l'on ne sait pas intégrer, comme par exemple les densités des lois conditionnelles. En effet, si on note f la densité de la loi jointe de deux variables aléatoires X et Y et si pour un certain y_0, on a $\int f(t, y_0)\mathrm{d}t \neq 0$, alors la densité conditionnelle de X sachant $Y = y_0$ est donnée par

$$g(x \mid y_0) = \frac{f(x, y_0)}{\int f(t, y_0)\mathrm{d}t} \propto_x f(x, y_0).$$

L'algorithme de Metropolis-Hastings permet de s'affranchir du calcul de la constante de normalisation $\int f(t, y_0)\mathrm{d}t$.

Dans le cas où $q(x, y)$ ne dépend pas de x, l'algorithme est dit *indépendant*. Dans le cas où la loi de proposition est une loi normale centrée en x, l'algorithme est dit *à marche aléatoire*.

Lors de l'implémentation de cet algorithme, le choix de la loi de proposition est crucial. La loi normale présente l'avantage d'être symétrique, ce qui simplifie le calcul de la probabilité d'acceptation, et facile à simuler.

3.1.2 Metropolis-Hastings à marche aléatoire

Supposons ici que la loi cible soit définie sur $\mathbb{R}^d$, espace euclidien de dimension d. Une manière naturelle de voir une perturbation de l'état courant x^k est de lui ajouter une erreur gaussienne. Le candidat proposé est alors $x^* = x^k + \varepsilon_k$ où les $\varepsilon_k \sim \mathcal{N}(0, \sigma^2 I_{d\times d})$ sont *i.i.d.*. L'algorithme qui correspond est résumé dans l'algorithme 2.

Se pose alors le choix de cette variance d'exploration σ^2. Une étude menée par Gelman, Roberts et Gilks [Gelman et al., 1996] suggère de choisir σ de manière à maintenir le *taux d'acceptation*, défini par le nombre de fois où le candidat a été accepté sur le nombre total d'itérations, entre 25% et 35%. Des taux

Algorithme 2 : Algorithme de Metropolis-Hastings à marche aléatoire

Choisir une valeur initiale x^0
pour $k = 0 : K$ **faire**
  Simuler $\varepsilon \sim \mathcal{N}(0, \sigma^2 I_{d\times d})$ et poser $x^* = x^k + \varepsilon$
  Calculer $r(x^k, x^*) = \dfrac{\pi(x^*)}{\pi(x^k)}$
  Simuler $u \sim \mathcal{U}([0,1])$
  Mettre à jour :

$$x^{k+1} = \begin{cases} x^* & \text{si } u \leq r(x^k, x^*) \\ x^k & \text{sinon} \end{cases}$$

fin
retourner $x^{0:K}$.

d'acceptation supérieurs indiqueraient que l'on simule des échantillons de loi *trop proche* de la loi instrumentale.

3.1.3 Pourquoi l'algorithme de Metropolis-Hastings fonctionne ?

Nous vérifions ici que l'algorithme de Metropolis-Hastings fournit une règle de transition pour la chaîne de Markov qui laisse la densité cible $\pi(x)$ invariante. Une preuve détaillée est disponible dans le livre de Liu [Liu, 2008]. Notons $A(x, y)$ la fonction de transition de l'algorithme, de l'état x à l'état y. L'étape d'acceptation/rejet a pour conséquence que A est différent de q. On cherche à montrer que

$$\int \pi(x)A(x, y)dx = \pi(y).$$

Montrons tout d'abord que pour tous états x et y, on a la relation de *réversibilité* :

$$\pi(x)A(x, y) = \pi(y)A(y, x). \tag{3}$$

Pour deux états $x \neq y$, la probabilité de passer de x à y est la probabilité de proposer y, $q(x, y)$, multipliée par la probabilité que le candidat y soit accepté, $\alpha(x, y)$:

$$A(x, y) = q(x, y) \min\left\{\frac{\pi(y)\ q(x, y)}{\pi(x)\ q(y, x)}, 1\right\},$$

et par conséquent,

$$\begin{aligned}\pi(x)A(x, y) &= \pi(x)q(x, y) \min\left\{\frac{\pi(y)\ q(x, y)}{\pi(x)\ q(y, x)}, 1\right\} \\ &= \min\left\{\pi(y)\ q(x, y), \pi(x)q(y, x)\right\},\end{aligned}$$

qui est symétrique en x et y. Ainsi, en échangeant les rôles de x et y, on obtient

$$\pi(y)A(y, x) = \min\left\{\pi(x)q(y, x), \pi(y)\ q(x, y)\right\} = \pi(x)A(x, y).$$

On a bien montré la relation 3. On a alors :

$$\int \pi(x)A(x, y)dx = \int \pi(y)A(y, x)dx = \pi(y)\underbrace{\int A(y, x)dx}_{=1} = \pi(y).$$

Donc l'algorithme de Metropolis-Hastings laisse bien la densité π invariante. D'après l'équation (2), la chaîne de Markov ainsi construite, sous l'hypothèse d'être irréductible et apériodique, converge en loi vers la distribution cible π.

Cet algorithme requière la simulation d'une marche aléatoire dans l'espace des paramètres. Cette marche aléatoire ne permet pas d'explorer correctement l'espace lorsqu'on se place en grande dimension. L'échantillonnage de Gibbs permet d'inférer plus rapidement en travaillant dans des espaces de dimension réduite.

3.2 Échantillonnage de Gibbs

L'échantillonnage de Gibbs a été introduit en 1984 par les frères Geman [Geman and Geman, 1984] dans le cadre de la restauration d'images. La principale idée de cet algorithme réside dans le fait que la chaîne de Markov est construite par assemblage de composantes suivant un ensemble de directions (souvent les coordonnées), où ces composantes suivent des distributions conditionnelles.

Supposons que l'on puisse décomposer la variable aléatoire x en d composantes : $x = (x_1, \cdots, x_d)$. Dans $\mathbb{R}^d$, on peut par exemple décomposer x par coordonnées ou par blocs de coordonnées. Pour tout $1 \leq i \leq d$, on note $x_{[-i]}$ le vecteur avec toutes les composantes sauf la $i^{\text{ème}}$: $x_{[-i]} = (x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_d)$ et $\pi_i(\cdots | x_{[-i]})$ la distribution conditionnelle de la composante i sachant les autres composantes. L'état x étant construit, on choisit, de manière aléatoire ou systématique, une composante, par exemple x_i. On met alors à jour cette composante avec une nouvelle valeur x_i', échantillonnée selon la distribution conditionnelle $\pi_i(\cdot | x_{[-i]})$. Une description de l'algorithme pour un choix systématique est rédigée dans l'algorithme 3.

Algorithme 3 : Échantillonnage de Gibbs

Choisir une valeur initiale $x^{(0)}$
pour $\underline{k = 0 : K}$ **faire**
  pour $\underline{i = 1 : d}$ **faire**
    Échantillonner $x_i^{(k+1)}$ selon

$$\pi_i \left(x_i | x_1^{(k+1)}, \cdots, x_{i-1}^{(k+1)}, x_{i+1}^{(k)}, \cdots, x_d^{(k)} \right)$$

  fin
fin
retourner $x^{(0:K)}$.

Chaque mise à jour de l'algorithme laisse la distribution π invariante au sens où si $x^{(k)}$ a pour loi π, alors $x^{(k+1)}$ aussi. La chaîne de Markov ainsi construite admet π comme distribution stationnaire.

L'échantillonnage de Gibbs est particulièrement adapté à l'inférence des modèles hiérarchiques. Avec cet algorithme, on simplifie les problèmes, notamment lorsque les lois conditionnelles sont connues et faciles à simuler. Si ce n'est pas

le cas, on peut faire appel à l'algorithme de Metropolis-Hastings (algorithme 1) pour chaque loi conditionnelle dans un algorithme dit *hybride*.

3.3 Combinaison de méthodes : Metropolis-Hastings dans Gibbs

Les algorithmes de Metropolis-Hastings et d'échantillonnage de Gibbs sont combinés dans des algorithmes dits *hybrides*. Ces algorithmes sont utilisés par exemple lorsque les lois conditionnelles de la loi cible sont inconnues. L'idée est de combiner des algorithmes qui laissent invariants la distribution π. On obtient à nouveau un algorithme qui laisse lui aussi invariant la distribution π. L'algorithme Metropolis-Hastings Within Gibbs [Gilks et al., 1995] est un exemple d'algorithme hybride.

3.3.1 Metropolis-Hastings Within Gibbs

L'algorithme hybride Metropolis-Hastings Within Gibbs est un algorithme itératif, basé sur l'échantillonnage de Gibbs : on met à jour la variable d'intérêt x coordonnée par coordonnée, auquel on ajoute une étape de type acceptation-rejet de Metropolis-Hastings. Supposons que les distributions conditionnelles de π ne soient pas simulables et ne soient connues qu'à constante multiplicative près. Ajouter une étape de Metropolis-Hastings permet de palier à ces difficultés. L'algorithme 4 présente une version de l'algorithme de Metropolis-Hastings Within Gibbs (MHWG) avec une loi de proposition q quelconque.

Algorithme 4 : Metropolis-Hastings Within Gibbs

Choisir une valeur initiale $x^{(0)}$
pour $\underline{k = 0 : K}$ **faire**
  pour $\underline{i = 1 : d}$ **faire**
    Simuler $x_i^* \sim q\left(x_i^{(k)}, \cdot\right)$
    Calculer

$$r_i(x_i^{(k)}, x_i^*) = \frac{\pi\left(x_i^* | x_1^{(k+1)}, \cdots, x_{i-1}^{(k+1)}, x_{i+1}^{(k)}, \cdots, x_d^{(k)}\right) q\left(x_i^{(k)}, x_i^*\right)}{\pi\left(x_i^{(k)} | x_1^{(k+1)}, \cdots, x_{i-1}^{(k+1)}, x_{i+1}^{(k)}, \cdots, x_d^{(k)}\right) q\left(x_i^*, x_i^{(k)}\right)}$$

    Simuler $u \sim \mathcal{U}([0,1])$
    Mettre à jour :

$$x_i^{(k+1)} = \begin{cases} x_i^* & \text{si } u \leq r_i(x_i^{(k)}, x_i^*) \\ x_i^{(k)} & \text{sinon} \end{cases}$$

  fin
fin
retourner $x^{(0:K)}$.

Lorsque la densité complète du modèle $\pi(x, Obs)$ est connue, pour obtenir la densité conditionnelle de x_i sachant toutes les autres variables, notée $\pi(x_i | \cdots)$,

à constante près, il suffit d'utiliser la formule de Bayes. En effet, $\pi(x_i|\cdots) = \pi(\cdots)^{-1}\pi(x_i,\cdots)$ donc il suffit de ne conserver que les termes qui impliquent x_i dans la densité complète $\pi(x_i,\cdots) = \pi(x, Obs)$. Un exemple est traité dans la Section 5.

Un exemple classique d'algorithme de Metropolis-Hastings Within Gibbs est d'utiliser une marche aléatoire comme loi de proposition : $x_i^* \sim \mathcal{N}(x_i^{(k)}, \sigma_i^2)$. Une étape d'un Metropolis-Hastings Within Gibbs avec marche aléatoire est décrite dans l'algorithme 5. Se pose alors, comme pour l'algorithme précédent, le problème du choix des variances d'exploration σ_i^2. Une solution peut être d'ajouter un schéma adaptatif sur les variances d'exploration.

Algorithme 5 : Une étape d'un Metropolis-Hastings Within Gibbs avec marche aléatoire.

1 Supposons construit $x^{(k)}$ et $x_{1:i-1}^{(k+1)}$; construisons x_i^{k+1} :
2 Simuler $\varepsilon_i \sim \mathcal{N}(0, \sigma_i^2)$ et poser $x_i^* = x_i^k + \varepsilon_i$
3 Calculer

$$r_i(x_i^{(k)}, x_i^*) = \frac{\pi\left(x_i^*|x_1^{(k+1)}, \cdots, x_{i-1}^{(k+1)}, x_{i+1}^{(k)}, \cdots, x_d^{(k)}\right)}{\pi\left(x_i^{(k)}|x_1^{(k+1)}, \cdots, x_{i-1}^{(k+1)}, x_{i+1}^{(k)}, \cdots, x_d^{(k)}\right)}$$

4 Simuler $u \sim \mathcal{U}([0,1])$
5 Mettre à jour :

$$x_i^{(k+1)} = \begin{cases} x_i^* & \text{si } u \leq r_i(x_i^{(k)}, x_i^*) \\ x_i^{(k)} & \text{sinon} \end{cases}$$

3.3.2 Ajout d'un schéma adaptatif

Une possibilité pour améliorer les performances de l'algorithme est d'inclure un schéma adaptatif qui met à jour la variance d'exploration. De cette manière, on espère éviter une variance ni trop faible, qui implique que l'espace est exploré trop lentement, ni trop grande, qui proposeraient des candidats trop éloignés des zones de forte densité et possiblement peut manquer certaines de ces zones. Le taux d'acceptation est un bon critère pour vérifier qu'un algorithme de type Metropolis-Hastings explore correctement l'espace d'état.

On met à jour la variance d'exploration σ_i^2 de la composante i de l'échantillonnage de Gibbs d'après le schéma suivant :

$$\begin{cases} \gamma_i^{k+1} = \dfrac{1}{k+1} \\ \mu_i^{k+1} = \mu_i^k + \gamma_i^{k+1}(x_i^{k+1} - \mu_i^k) \\ \Sigma_i^{k+1} = \Sigma_i^k + \gamma_i^{k+1}\left[(x_i^{k+1} - \mu_i^{k+1})(x_i^{k+1} - \mu_i^{k+1})^T - \Sigma_i^k\right] \\ \lambda_i^{k+1} = \lambda_i^k \exp\left(\gamma_i^{k+1}[\alpha_i^k - \alpha^*]\right) \\ \sigma_i^{2\,k+1} = \lambda_i^{k+1}\Sigma_i^{k+1} \end{cases}$$

où $\alpha_i = \min\left(1, r_i(x_i^{(k)}, x_i^*)\right)$ et α^* est le taux optimal d'acceptation. Il vaut approximativement 0.234 dans le cas multidimensionnel [Bédard, 2008].

Lorsqu'on cherche à inférer en grande dimension, on se heurte à des problèmes de mauvaise exploration. De plus, par les effets de la marche aléatoire, les réalisations de la chaîne de Markov sont très corrélées. Pour pallier à ces difficultés liées à la grande dimension, l'algorithme de Monte-Carlo Hamiltonien offre une alternative à la marche aléatoire en améliorant la loi de proposition d'un algorithme de type Metropolis-Hastings.

3.4 Monte-Carlo Hamiltonien

L'objectif de cette algorithme est d'éviter une forte corrélation entre les réalisations de la chaîne de Markov afin de mieux explorer. Avec l'algorithme de Metropolis-Hastings Within Gibbs avec marche aléatoire (algorithme 2), le candidat proposé est "proche" de l'état courant de la chaîne de Markov. Avec le Monte-Carlo Hamiltonien, on espère ne pas rester dans le voisinage de la valeur courante et ainsi mieux explorer l'espace d'état de la chaîne de Markov. Cet algorithme est d'autant plus utile lorsque la dimension de l'espace à explorer est grande [Betancourt, 2017]. La méthode d'échantillonnage de Monte-Carlo par les dynamiques hamiltoniennes est détaillée dans le Chapitre 5 de *Handbook of Markov Chain Monte-Carlo*, [Neal, 2010].

3.4.1 Dynamique hamiltonienne

Analogie avec la physique - Afin de comprendre les objets de la dynamique hamiltonienne, une analogie avec la physique peut apporter une intuition. On s'intéresse au déplacement sans frottement d'une bille sur une surface. La bille possède une position q à laquelle on associe une énergie potentielle $U(q)$. La bille a également un moment $p = mv$ où m est la masse de la bille et v sa vitesse. On définit l'énergie cinétique du système par $K(p) = \frac{1}{2}mv^2 = \frac{1}{2}\frac{\|p\|^2}{m}$. L'énergie totale du système est donc donnée par $H(q,p) = U(q) + K(p)$.

Par la loi de conservation de l'énergie, l'énergie totale reste constante dans un système fermé. La dérivée par rapport au temps de $H(q,p)$ vaut donc 0. On obtient ainsi les équations du mouvement :

$$\begin{cases} \frac{dq}{dt} = \frac{\partial H}{\partial p} = \frac{\partial K}{\partial p} \\ \frac{dp}{dt} = -\frac{\partial H}{\partial q} = -\frac{\partial U}{\partial q} \end{cases} \tag{4}$$

Intuitivement, lorsque la bille rencontre une pente qui monte, sa vitesse va lui permettre de continuer à avancer en montant. Son énergie cinétique va décroître tandis que l'énergie potentielle augmente ; et ce jusqu'à ce que l'énergie cinétique soit nulle. A cet instant, la bille va s'arrêter puis redescendre. Pendant la descente, l'énergie cinétique augmente et l'énergie potentielle diminue.

Généralisation en dimension 2d - La dynamique hamiltonienne opère sur un vecteur **position** q de dimension d et un vecteur **moment** p de dimension d de manière à ce que l'espace total soit de dimension $2d$. Le système est décrit

par une fonction de q et p, le **Hamiltonien**, $H(q,p)$, différentiable. Les dérivées partielles du Hamiltonien déterminent la dynamique de q et p au cours du temps t, selon de système d'équations :

$$\forall 1 \leq i \leq d, \begin{cases} \dfrac{dq_i}{dt} = \dfrac{\partial H}{\partial p_i}; \\ \dfrac{dp_i}{dt} = -\dfrac{\partial H}{\partial q_i}. \end{cases}$$

Autrement dit, le vecteur $z = (q,p)$ vérifie

$$\frac{dz}{dt} = J \ \nabla H(z)$$

où $J = \begin{pmatrix} 0_{d\times d} & I_{d\times d} \\ -I_{d\times d} & 0_{d\times d} \end{pmatrix}$.

Pour l'algorithme de Monte-Carlo Hamiltonien (HMC), on utilise des Hamiltoniens de la forme

$$H(q,p) = U(q) + K(p) \tag{5}$$

où $U(q)$ est appelée l'**énergie potentielle** et $K(p)$ est l'**énergie cinétique**. On définit généralement $K(p) = \frac{1}{2}p^T M^{-1} p$ où M est symétrique définie positive, souvent diagonale. On appelle M la **matrice de masse**. Les équations hamiltoniennes s'écrivent alors :

$$\forall 1 \leq i \leq d, \begin{cases} \dfrac{dq_i}{dt} = \left[M^{-1}p\right]_i; \\ \dfrac{dp_i}{dt} = -\dfrac{\partial U}{\partial q_i}. \end{cases}$$

3.4.2 Propriétés de la dynamique hamiltonienne

Réversibilité en temps - La dynamique hamiltonienne est une dynamique dite **réversible** au sens où l'application T_s qui à un état au temps t, $(q(t), p(t))$, associe l'état au temps $t+s$, $(q(t+s), p(t+s))$, obtenu en suivant une dynamique hamiltonienne partant de $(q(t), p(t))$, est inversible, d'inverse T_{-s}. Lorsque l'Hamiltonien est sous la forme (5) et que K est paire ($K(-p) = K(p)$), alors l'inverse peut s'obtenir par prendre $-p$, appliquer ensuite T_s, puis reprendre $-p$.

Conservation du Hamiltonien au cours du temps - Une propriété de la dynamique hamiltonienne est que le Hamiltonien reste constant au cours du temps. En effet, d'après (4),

$$\frac{dH}{dt} = \sum_{i=1}^{d} \left[\frac{dq_i}{dt}\frac{\partial H}{\partial q_i} + \frac{dp_i}{dt}\frac{\partial H}{\partial p_i} \right] = \sum_{i=1}^{d} \left[\frac{\partial H}{\partial p_i}\frac{\partial H}{\partial q_i} - \frac{\partial H}{\partial q_i}\frac{\partial H}{\partial p_i} \right] = 0.$$

Dans l'exemple de la bille se déplaçant sur une surface, cette propriété correspond à la conservation de l'énergie totale du système.

Conservation du volume - Cette propriété ce traduit de la manière suivante : l'image d'une région R de volume V par l'application T_s sera également de

volume V. Une démonstration possible de la préservation du volume est de montrer que la divergence du champs de vecteurs est nulle [Arnold et al., 2013] :

$$\sum_{i=1}^{d}\left[\frac{\partial}{\partial q_i}\frac{dq_i}{dt}+\frac{\partial}{\partial p_i}\frac{dqp_i}{dt}\right]=\sum_{i=1}^{d}\left[\frac{\partial}{\partial q_i}\frac{\partial H}{\partial p_i}-\frac{\partial}{\partial p_i}\frac{\partial H}{\partial q_i}\right]=\sum_{i=1}^{d}\left[\frac{\partial^2 H}{\partial q_i \partial p_i}-\frac{\partial^2 H}{\partial p_i \partial q_i}\right]=0.$$

3.4.3 Utilisation des propriétés de la dynamique hamiltonienne dans des problèmes d'inférence

Pour un problème d'inférence bayésienne, on va s'appuyer sur la dynamique hamiltonienne pour explorer l'espace d'état des paramètres de manière à produire un échantillonnage de la loi *a posteriori* des paramètres sachant les observations. En d'autres termes, on va définir la position q de taille d comme étant un vecteur de paramètres. On associe à cette position une énergie liée à la densité $\pi(q)$ de la loi *a posteriori*.

Intuition avec l'exemple de la bille - Rappelons notre exemple d'une bille glissant sans frottement sur une surface. Lorsqu'on visualise le comportement de la position de la bille aux abords d'un puits de potentiel, on imagine que la bille a tendance à rester proche de ce lieu. On a donc envie de faire correspondre une surface à notre loi cible pour laquelle un puits de potentiel correspond au mode de notre loi cible. Une transformation de ce type est de prendre $-\log$ de la densité cible. Cette transformation, appliquée sur un exemple, est représentée sur la figure 3.2.

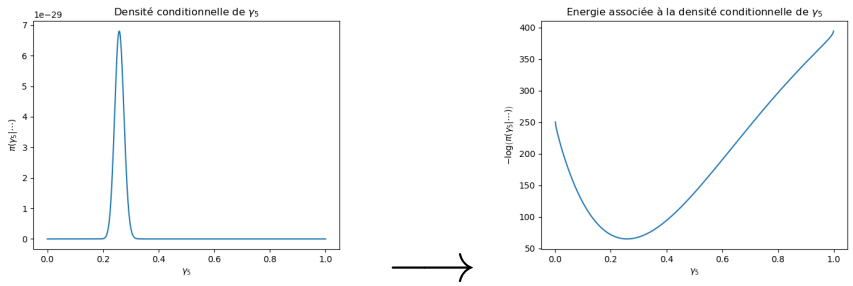


Figure 3.2 – Transformation d'une loi cible en une énergie potentielle faisant correspondre le mode avec un puits de potentiel. Exemple de la densité conditionnelle du paramètre γ_5 du modèle exposé dans la section 4.2. À gauche, la densité cible, à droite, l'énergie potentielle associée.

Probabilités et Hamiltonien - Supposons que l'on cherche à échantillonner selon une loi de densité $\pi(q) \propto \exp[-U(q)]$ avec U différentiable sur $\mathbb{R}^d$. On définit l'énergie potentielle d'une position q par $U(q) = -\log(\pi(q))$. Il nous reste à introduire un moment p, de taille d également, et une énergie cinétique $K(p)$ associée. Remarquons que si on sait simuler (q,p) de distribution

$$\pi(q,p) \propto \exp\left(-H(p,q)\right) \text{ où } H(p,q) = U(q)+K(p) = -\log\left(\pi(q)\right)+\frac{1}{2}p^T M^{-1} p,$$

alors marginalement, q est de densité $\pi(q)$ et p suit une loi normale centrée en 0 et de covariance M dont la densité associée est proportionnelle à $\exp\left(-\frac{1}{2}p^T M^{-1} p\right)$. On va introduire le moment de manière artificielle.

L'idée de l'algorithme est la suivante : étant donnée une position q à l'étape k, on construit q' à l'étape $k+1$ telle que

1. On génère un moment p selon une loi gaussienne centrée et de matrice de covariance M ;
2. On suit une dynamique hamiltonienne pendant un temps τ et partant de (q,p) pour obtenir un nouvel état (q',p').

Le temps est également artificiel. Définir la vitesse associée à un mouvement revient à définir le temps. La vitesse, qui est la dérivée de la position par rapport au temps, est liée au moment par la masse. Elle est définie par $v = M^{-1}p$.

Inférence bayésienne - En statistiques bayésiennes, la densité cible est généralement la distribution *posterior* des paramètres du modèles. Ces paramètres vont donc jouer le rôle de la position q. On peut décomposer l'expression du *posterior* pour écrire l'énergie potentielle sous la forme

$$U(q) = -\log\left[g(q)\mathcal{L}(q|Obs)\right]$$

où g est la densité de la loi *a priori* et $\mathcal{L}(q|Obs)$ est la vraisemblance liée aux observations.

3.4.4 Mise en oeuvre de l'algorithme HMC

Discrétisation de la dynamique hamiltonienne : l'algorithme de saute-moutons - Pour la mise en oeuvre algorithmique de la deuxième étape de l'algorithme HMC, la dynamique hamiltonienne doit être discrétisée en temps. Notons ε le pas de temps et L le nombre d'itérations. L et ε sont choisis tels que $\tau = \varepsilon L$. Supposons ici que M soit diagonale, d'éléments diagonaux $m_1, \cdots, m_d$ tels que

$$K(p) = \sum_{i=1}^{d} \frac{p_i^2}{2m_i}.$$

On va appliquer un algorithme de type "saute-moutons" (ou *"Leapfrog"*) qui consiste à mettre à jour la position et le moment de manière décalée. La discrétisation s'écrit pour tout $1 \leq i \leq d$:

$$\begin{aligned}
p_i(t+\varepsilon/2) &= p_i(t) - \frac{\varepsilon}{2}\frac{\partial U}{\partial q_i}(q_i(t)); \\
q_i(t+\varepsilon) &= q_i(t) + \varepsilon\frac{p_i(t+\varepsilon/2)}{m_i}; \\
p_i(t+\varepsilon) &= p_i(t+\varepsilon/2) - \frac{\varepsilon}{2}\frac{\partial U}{\partial q_i}(q_i(t+\varepsilon)).
\end{aligned}$$

On commence donc par effectuer un demi pas pour actualiser le moment. Puis on effectue des pas entiers alternativement pour la position et pour le moment. On termine avec un demi pas pour le moment de manière à terminer la trajectoire. Lors du déroulement du saute-moutons, les valeurs prises par le temps pour le moment sont décalées de $\varepsilon/2$ par rapport à celle pour la position. La figure 3.3 est un schéma décrivant sommairement le procédé.

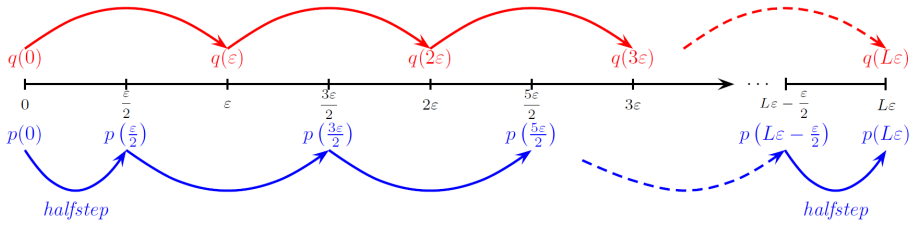


Figure 3.3 – Discrétisation du système Hamiltonien : le principe de l'algorithme saute-moutons.

Les avantages de cet algorithme sont :
- le volume est préservé ;
- par symétrie, il est réversible en temps en prenant le moment opposé et en appliquant le même nombre de pas ;

ces deux propriétés n'étant pas vérifiées par des schémas de résolution classiques du type schémas d'Euler.

Acceptation/rejet de Metropolis - Pour corriger l'erreur commise dans la non conservation du Hamiltonien lors du saute-moutons, on ajoute une étape de Metropolis-Hastings qui garantie que l'algorithme laisse la densité cible invariante. La loi de proposition de cet étape est donnée par le lancement de l'algorithme de saute-moutons. L'algorithme HMC devient, avec la position $q^{(k)} = q$ construite à l'étape k :

1. On génère un moment p selon une loi gaussienne de densité proportionnelle à $\exp(-K(p))$;
2. On lance L étapes de saute-moutons avec un pas $\varepsilon = \tau/L$ et partant de (q, p) pour obtenir un nouvel état (q', p') ;
3. On applique une étape d'acceptation/rejet de type Metropolis-Hastings *i.e.* on accepte q' pour $q^{(k+1)}$ avec probabilité

$$\begin{aligned}\alpha &= \min\{1, \exp(-H(q', p') + H(q, p))\} \\ &= \min\{1, \exp(-U(q') + U(q) - K(p') + K(p))\}.\end{aligned}$$

S'il n'est pas accepté, la position $q^{(k+1)}$ à l'étape $k+1$ vaut l'ancienne position q.

Remarquons que le choix de $K(p)$ donne $K(-p) = K(p)$, ce qui rend la loi de proposition dans l'algorithme de Metropolis symétrique.

De plus, une conséquence importante de la préservation du volume pour l'algorithme est qu'il n'est pas nécessaire de tenir compte de tout changement de volume dans la probabilité d'acceptation de l'étape de Metropolis-Hastings. Si nous avions proposé de nouveaux états en utilisant une dynamique arbitraire, non-hamiltonienne, nous aurions dû calculer le déterminant de la matrice jacobienne, ce qui pourrait bien être impossible ou lourd en calculs.

Choix de la matrice de masse - Même s'il est difficile de faire un choix optimal pour la variance d'échantillonnage des moments M, cette matrice, lorsqu'elle est

diagonale, permet de donner différentes échelles aux différents paramètres *i.e.* aux coordonnées de q. Si s_i est une échelle convenable pour q_i, alors on peut appliquer la transformation $\hat{q}_i = q_i/s_i$ en posant $M = diag(m_1, \cdots, m_d)$ où $m_i = 1/s_i^2$.

On considère désormais l'algorithme suivant :

$$\begin{aligned} p_i(t+\varepsilon/2) &= p_i(t) - \frac{\varepsilon}{2}\frac{\partial U}{\partial q_i}(q_i(t)); \\ q_i(t+\varepsilon) &= q_i(t) + \varepsilon s_i^2\ p_i(t+\varepsilon/2); \\ p_i(t+\varepsilon) &= p_i(t+\varepsilon/2) - \frac{\varepsilon}{2}\frac{\partial U}{\partial q_i}(q_i(t+\varepsilon)). \end{aligned}$$

Il ne s'agit en réalité que d'un changement d'échelle : cherchons à retrouver l'algorithme initial avec tous les m_i égaux à 1. Notons $(q^{(0)}, p^{(0)})$ l'état initial du saute-moutons, c'est à dire $(q(t), p(t))$, et $(q^{(1)}, p^{(1)})$ l'état final, $(q(t+\varepsilon), p(t+\varepsilon))$. On définit également le moment de demi-pas, $p^{(1/2)}$. On peut réécrire l'algorithme :

$$\begin{aligned} p_i^{(1/2)} &= p_i^{(0)} - \frac{\varepsilon}{2}\frac{\partial U}{\partial q_i}(q_i^{(0)}); \\ q_i^{(1)} &= q_i^{(0)} + \varepsilon s_i^2\ p_i^{(1/2)}; \\ p_i^{(1)} &= p_i^{(1/2)} - \frac{\varepsilon}{2}\frac{\partial U}{\partial q_i}(q_i^{(1)}). \end{aligned}$$

Définissons le moment normalisé $\hat{p}_i = s_i p_i$ et les pas $\varepsilon_i = s_i\varepsilon$. On peut alors écrire la mise à jour du saute-moutons pour ces variables :

$$\begin{aligned} \hat{p_i}^{(1/2)} &= \hat{p_i}^{(0)} - \frac{\varepsilon_i}{2}\frac{\partial U}{\partial q_i}(q_i^{(0)}); \\ q_i^{(1)} &= q_i^{(0)} + \varepsilon_i\ \hat{p_i}^{(1/2)}; \\ \hat{p_i}^{(1)} &= \hat{p_i}^{(1/2)} - \frac{\varepsilon_i}{2}\frac{\partial U}{\partial q_i}(q_i^{(1)}). \end{aligned}$$

On retrouve ici l'algorithme de saute-moutons où tous les m_i sont égaux à 1.

L'algorithme de Monte-Carlo Hamiltonien est détaillé dans l'algorithme 6.

Mise en oeuvre pratique : réglages des paramètres - Dans la pratique, pour implémenter un algorithme de Monte-Carlo Hamiltonien, il est nécessaire de sélectionner des valeurs adaptées pour :

— la taille du pas du saute-moutons : ε ;
— le nombre de pas du saute-moutons : L ;
— la longueur de la trajectoire simulée par le saute-moutons εL ;
— les échelles nécessaires à la construction de la matrice de masse.

Une étape de calibrage s'impose avant d'effectuer l'inférence.

Un pas ε trop large pourrait faire chuter significativement le taux d'acceptation. Les trajectoires simulées donneraient des candidats qui seraient systématiquement rejetés et la chaîne de Markov resteraient longtemps sur la même position. Réciproquement, un pas trop petit serait un gaspillage en temps de calcul.

Algorithme 6 : Monte-Carlo Hamiltonien

Choisir une valeur initial q^0
pour $\underline{k = 0 : K}$ **faire**
  Échantillonner un nouveau moment $p_0 \sim \mathcal{N}(0_{d\times d}, M)$
  Prendre $q = q^k$
  Effectuer un demi saut pour le moment :

$$p = p_0 - \frac{\varepsilon}{2}\nabla U(q)$$

  Alterner des pas entiers pour la position et le moment :
  pour $\underline{i = 1 : L}$ **faire**
    Effectuer un pas pour la position :

$$q = q + \varepsilon M^{-1} p$$

    Effectuer un pas pour le moment, sauf à la fin de la trajectoire :
    si $\underline{i < L}$ **alors**

$$p = p - \varepsilon \nabla U(q)$$

    fin
  fin
  Effectuer un demi-pas pour le moment à la fin de la trajectoire :

$$p = p - \frac{\varepsilon}{2}\nabla U(q)$$

  Calculer :

$$\begin{aligned}\log(r) &= current_U - candidat_U + current_K - candidat_K \\ &= U(q^k) - U(q) + p_0^T M^{-1} p_0/2 - p^T M^{-1} p/2\end{aligned}$$

  Simuler $u \sim \mathcal{U}(0, 1)$
  Mettre à jour :

$$q^{k+1} = \begin{cases} q \text{ si } \log(u) < \log(r) \\ q^k \text{ sinon} \end{cases}$$

fin
retourner $q^{0:K}$.

3.4.5 Gestion des problèmes de bord : le billard

Jusqu'à présent, la loi cible était supposée de la forme $\pi(q) \propto \exp[-U(q)]$. Dans cette section, il est question de traiter le cas d'une densité cible ayant un support A, avec $A = I_1 \times \cdots \times I_d$ un produit cartésien d'intervalles. La densité s'écrit donc sous la forme $\pi(q) \propto \exp[-U(q)]\,\mathbb{1}_A(q)$. Autrement dit, chaque coordonnée du vecteur position est possiblement minoré, majoré, ou les deux.

Prenons tout d'abord l'exemple d'une loi *a priori* uniforme sur un intervalle. Le support du paramètre est donc inclus dans cet intervalle. Hors du support, la densité cible vaut 0. Au bord de l'intervalle, la densité peut alors présenter une discontinuité. Or, lorsqu'on simule une trajectoire avec l'algorithme de saute-moutons, on calcule le gradient de l'énergie et on effectue un pas suivant la direction du gradient. Lorsque l'on s'approche du bord, par la forme de la densité, la position proposée peut se trouver en dehors du support. La figure 3.4 montre, sur un exemple traité dans la section 4.2, la problématique soulevée.

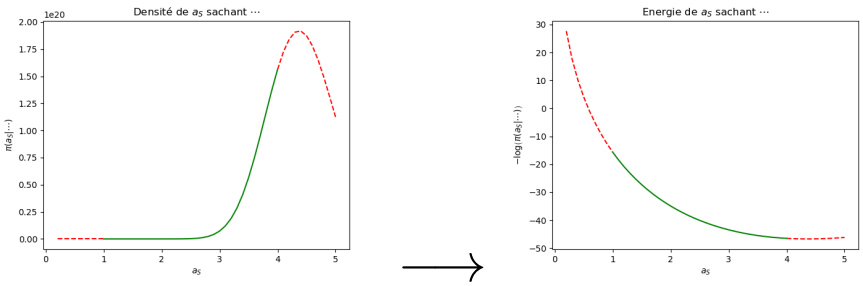


Figure 3.4 – Problématique des bords sur l'algorithme HMC : cas d'un *prior* uniforme. En vert, densité conditionnelle de a_S sachant tous les autres paramètres fixés et énergie correspondante dans le cas d'un modèle de Schneider avec compétition (Section 4.2). En rouge, la continuité de la densité et de l'énergie sans le *prior* uniforme : "ce que voit l'algorithme".

Les contraintes considérées sont de la forme $q_i \leq u_i$, $q_i \geq l_i$ ou les deux. Lorsque la position q est en dehors des bornes, la densité cible vaut 0 et donc l'énergie potentielle est infinie. L'idée est de signifier à l'algorithme la présence d'un mur infranchissable à l'emplacement du bord de l'intervalle comme sur la figure 3.6. Par exemple, pour le problème de la figure 3.4, on placerait un mur comme sur la figure 3.6.(a).

Lorsque le gradient est calculé approximativement de manière numérique par différence finie, il peut arriver que le pas ε ne soit pas assez petit pour détecter un changement brutal de pente, comme sur l'exemple de la figure 3.5. L'algorithme peut, dans cette situation également, proposer un candidat qui n'est pas dans le support de la densité cible. Le problème, bien qu'il soit de l'ordre de la précision numérique et non de la régularité de la densité cible, se traite de la même manière que dans le cas d'une loi *a priori* uniforme : on place un mur infranchissable à l'emplacement du bord. Par exemple, pour le problème de la figure 3.5, on placerait un mur comme sur la figure 3.6.(b).

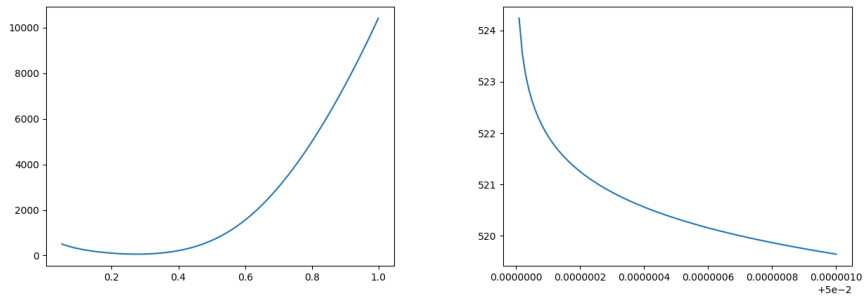
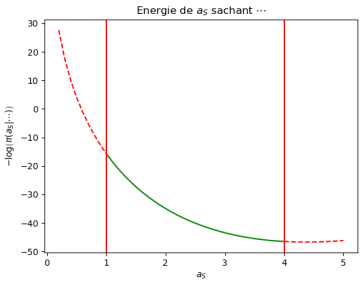
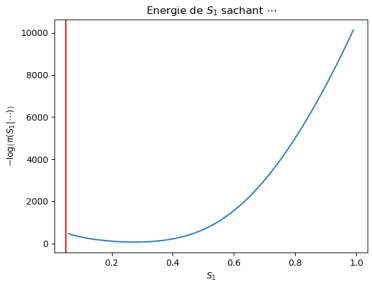


Figure 3.5 – Problématique des bords sur l'algorithme HMC : cas d'un changement brutal de pente. Énergie correspondant à la densité conditionnelle de S_1 sachant tous les autres paramètres fixés dans le cas d'un modèle de Schneider avec compétition (Section 4.2).

En reprenant l'exemple de la bille et en supposant qu'elle démarre en hauteur, elle va suivre la pente, passer par le puits de potentiel mais son énergie cinétique va lui permettre de continuer sa course jusqu'à atteindre la borne représentée par le mur. Sur ce mur, elle va rebondir sans perte d'énergie pour repartir dans la direction opposée, en s'éloignant du mur.



(a) Énergie associée à la densité conditionnelle de a_S sachant tous les autres paramètres fixés dans le cas d'un modèle de Schneider avec compétition : placement d'un mur artificiel sur les bords du support de la loi *a priori* uniforme.



(b) Énergie associée à la densité conditionnelle de S_1 sachant tous les autres paramètres fixés dans le cas d'un modèle de Schneider avec compétition : placement d'un mur artificiel aux bords du support.

Figure 3.6 – Détails sur le principe de l'algorithme billard : placement de murs artificiels.

L'algorithme 7 donne l'algorithme à insérer dans la mise à jour du saute-mouton pour prendre en compte des contraintes du type $q_i \leq u_i$ et $q_i \geq l_i$.

Algorithme 7 : Billard - à insérer entre les lignes 12 et 13 de l'algorithme 6

```
pour j = 1 : d faire
    q'_j = q_j et p'_j = p_j
    tant que la coordonnée j a des contraintes non vérifiées faire
        si la coordonnée j a une contrainte supérieure u_j et que q'_j > u_j
         alors
            q'_j = u_j − (q'_j − u_j) et p'_j = −p'_j.
        fin
        si la coordonnée j a une contrainte inférieure l_j et que q'_j < l_j
         alors
            q'_j = l_j + (q'_j − l_j) et p'_j = −p'_j.
        fin
    fin
    q_j = q'_j et p_j = p'_j
fin
```

3.5 Combinaison de méthodes : Gibbs et Monte-Carlo Hamiltonien

Pour l'inférence de son modèle hiérarchique bayésien, Radford M. Neal traite de manière différente les paramètres de bas étages (qui correspondent aux paramètres individuels dans nos modèles de population présentés dans la section 4) et les hyperparamètres [Neal, 2010]. Il opte pour un échantillonnage de Gibbs où un HMC est appliqué sur la loi conditionnelle des paramètres de bas étages sachant les hyperparamètres puis les hyperparamètres, souvent moins dépendant les uns des autres, sont mis à jour selon une étape de Metropolis-Hastings. L'algorithme 8 résume cet algorithme hybride combinant Gibbs et HMC.

Algorithme 8 : Algorithme hybride final

```
Choisir une valeur initiale
pour k = 0 : K faire
    pour les hyperparamètres faire
        Mettre à jour selon une étape de Metropolis-Hastings Within
         Gibbs avec une marche aléatoire : algorithme 5.
    fin
    Mettre à jour les paramètres individuels suivant une étape de
     Monte-Carlo Hamiltonien avec billard : algorithmes 6 et 7.
fin
```

4 Modélisation des croissances de plantes en population hétérogène.

4.1 Modèle hiérarchique de croissance de plantes, selon un modèle de Gompertz, avec indépendance entre les individus (pas d'interaction).

Ce modèle permet de se familiariser avec les objets et de cerner le principe d'inférence bayésienne. Les *individus* sont des plantes. On parlera de *population* pour désigner l'ensemble des individus.

On s'inspire de l'article [Schneider et al., 2006]. On note N la taille de la population. On regarde, chez un individu $i \in \{1, \cdots, N\}$, une grandeur caractéristique s_i, par exemple la taille de sa partie aérienne, au cours du temps $t \in \mathbb{R}^+$: $s_i(t)$. Chaque individu i possède des paramètres individuels $\theta_i = (S_i, \gamma_i)$ où S_i est la taille asymptotique de la plante et γ_i est le taux de croissance de la plante i.

On suppose que les tailles des individus suivent des dynamiques indépendantes selon une fonction de Gompertz, définie par l'équation différentielle

$$\frac{ds_i(t)}{dt} = \gamma_i \; s_i(t) \; \log\left(\frac{S_i}{s_i(t)}\right). \tag{6}$$

En imposant la condition initiale $s_i(0) = s_0 > 0$, on obtient la solution définie sur $\mathbb{R}^+$ par

$$\forall t \in \mathbb{R}^+, \; s_i(t) = S_i \; \exp\left(-\log\left(\frac{S_i}{s_0}\right) e^{-\gamma_i t}\right).$$

Une croissance typique est représentée figure 4.7.

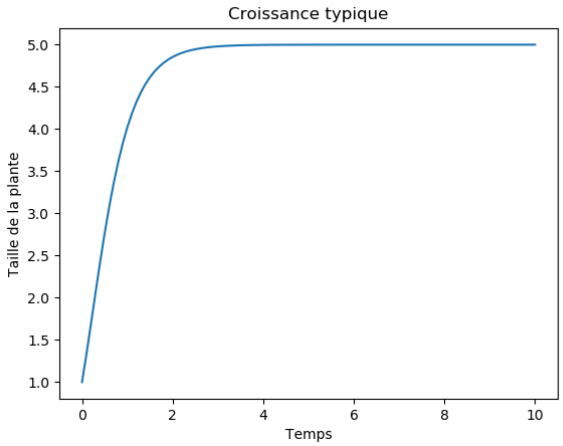


Figure 4.7 – Allure typique d'une croissance suivant une fonction de Gompertz pour $s_0 = 1$, $S_i = 5$ et $\gamma_i = 2$.

Il est clair que, dans ce modèle, les plantes poussent de manière indépendante. Il n'y a pas d'interaction entre les individus, pas de compétition pour les

ressources ni coopération.

On suppose que les paramètres individuels sont issus d'un échantillonnage *i.i.d.* selon une loi à densité p_0^θ. Il est supposé que les coordonnées de θ_i, S_i et γ_i, sont indépendantes et suivent des lois log-normales

$$\begin{cases} S_i \sim s_m + \mathcal{LN}\left(\mu_S, \sigma_S\right) \ ; \\ \gamma_i \sim \mathcal{LN}\left(\mu_\gamma, \sigma_\gamma\right). \end{cases} \tag{7}$$

Ainsi, $S_i > s_m$ et $\gamma_i > 0$ presque sûrement.

On met ce modèle sous la forme d'un modèle hiérarchique bayésien. On ajoute donc des lois *a priori* sur les paramètres que l'on qualifie *de population* : $\mu_S, \sigma_S, \mu_\gamma$ et σ_γ. On pose comme lois *a priori* des lois formant une famille conjuguée avec la loi normale ou la loi log-normale :

$$\begin{cases} \mu_S \sim \mathcal{N}(\mu, \tau) \\ \sigma_S^2 \sim \mathcal{IG}(a, b) \text{ où } \mathcal{IG} \text{ désigne la loi Inverse-Gamma} \\ \mu_\gamma \sim \mathcal{N}(\nu, \rho) \\ \sigma_\gamma^2 \sim \mathcal{IG}(c, d). \end{cases} \tag{8}$$

On suppose disposer d'une base de données. On suppose que l'on observe les tailles des N individus aux temps $(t_j)_{1\leq j\leq M}$. On suppose que ces données, notées $s^o = \left(s^o_{i,j}\right)_{1\leq i\leq N,\ 1\leq j\leq M}$, sont bruitées et ont été générées par le modèle suivant :

$$\forall 1 \leq i \leq N,\ \forall 1 \leq j \leq M,\ s^o_{i,j} = s_i(t_j) + \varepsilon_{i,j} \text{ où } \varepsilon_{i,j} \underset{i.i.d}{\sim} \mathcal{N}(0, \sigma^2).$$

On ajoute également une loi *a priori* sur la variance d'erreur d'observation σ^2 :

$$\sigma^2 \sim \mathcal{IG}(e, f). \tag{9}$$

Le modèle graphique est représenté sur la figure 4.8. L'objectif est d'effectuer une inférence bayésienne. Ce modèle graphique permet d'écrire la densité cible : la densité de la loi *a posteriori*, loi des paramètres inconnus sachant les observations. Dans l'objectif d'appliquer un algorithme de Monte-Carlo par chaînes de Markov, commençons par écrire la vraisemblance complète du modèle. Par la règle de Bayes et par les indépendances entre les individus et entre les observations, en notant $S = (S_i)_{1\leq i\leq N}$ et $\gamma = (\gamma_i)_{1\leq i\leq N}$,

$$p(s^o, S, \gamma, \mu_S, \sigma_S^2, \mu_\gamma, \sigma_\gamma^2, \sigma^2) =$$
$$\underbrace{\prod_{i=1}^{N}\prod_{j=1}^{M} p(s^o_{i,j}|S_i, \gamma_i, \sigma^2)}_{\text{Vraisemblance}} \underbrace{\prod_{i=1}^{N} p(S_i|\mu_S, \sigma_S^2) \prod_{i=1}^{N} p(\gamma_i|\mu_\gamma, \sigma_\gamma^2)}_{\text{Distribution de population}} \underbrace{p(\mu_S)p(\sigma_S^2)p(\mu_\gamma)p(\sigma_\gamma^2)p(\sigma^2)}_{prior}$$

On écrit explicitement les densités. Notons h la fonction qui à un couple de paramètres individuels (S_i, γ_i) et à un temps t associe la solution de (6) au temps t :

$$h(S_i, \gamma_i, t) = S_i \exp\left[-\log\left(\frac{S_i}{s_0}\right) e^{-\gamma_i t}\right].$$

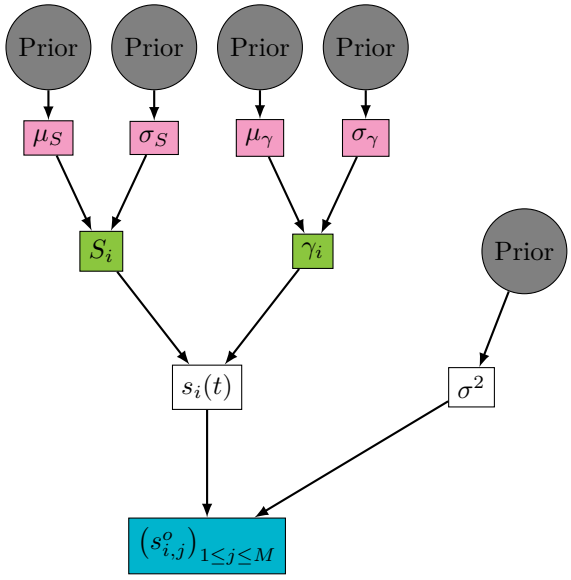


Figure 4.8 – Modèle graphique du modèle de croissance de plantes sans interactions. En gris, les lois *a priori*, en rose, les hyperparamètres, en vert, les paramètres individuels, en bleu, les données.

Alors la densité de l'observation au temps t_j de l'individu i sachant les paramètres individuels S_i et γ_i ne dépend que de $h(S_i, \gamma_i, t_j)$. Par l'expression de la densité de la loi normale,

$$\begin{aligned} p(s^o_{i,j}|S_i, \gamma_i, \sigma^2) &= p(s^o_{i,j}|h(S_i, \gamma_i, t_j), \sigma^2) \\ &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2}(s^o_{i,j} - h(S_i, \gamma_i, t_j))^2\right]. \end{aligned}$$

D'autre part, par (7) et par l'expression de la densité de la loi log-normale,

$$\begin{aligned} p(S_i|\mu_S, \sigma^2_S) &= \frac{1}{(S_i - s_m)\sqrt{2\pi\sigma^2_S}} \exp\left[-\frac{1}{2\sigma^2_S}(\log(S_i - s_m) - \mu_S)^2\right] \mathbb{1}_{S_i > s_m}; \\ p(\gamma_i|\mu_\gamma, \sigma^2_\gamma) &= \frac{1}{\gamma_i\sqrt{2\pi\sigma^2_\gamma}} \exp\left[-\frac{1}{2\sigma^2_\gamma}(\log(\gamma_i) - \mu_\gamma)^2\right] \mathbb{1}_{\gamma_i > 0}. \end{aligned}$$

Remarquons que,du fait que les individus sont supposés indépendants, les densités conditionnelles des paramètres individuels S_i et γ_i sont indépendantes des paramètres individuels des autres individus. Il est donc possible de paralléliser les calculs lors de l'inférence, comme par exemple, dans les étapes de l'algorithme Metropolis-Hastings Within Gibbs (algorithme 4)

Enfin, les densités *a priori*, d'après (8) et (9) et par les expressions des

densités des lois normale et Inverse-Gamma, s'écrivent

$$
\begin{aligned}
p(\mu_S) &= \frac{1}{\sqrt{2\pi\tau^2}} \exp\left[-\frac{1}{2\tau^2}(\mu_S - \mu)^2\right] \\
p(\sigma_S^2) &= \frac{b^a}{\Gamma(a)} \left(\frac{1}{\sigma_S^2}\right)^{a+1} \exp\left[-\frac{b}{\sigma_S^2}\right] \mathbb{1}_{\sigma_S^2>0} \\
p(\mu_\gamma) &= \frac{1}{\sqrt{2\pi\rho^2}} \exp\left[-\frac{1}{2\rho^2}(\mu_\gamma - \nu)^2\right] \\
p(\sigma_\gamma^2) &= \frac{d^c}{\Gamma(c)} \left(\frac{1}{\sigma_\gamma^2}\right)^{c+1} \exp\left[-\frac{d}{\sigma_\gamma^2}\right] \mathbb{1}_{\sigma_\gamma^2>0} \\
p(\sigma^2) &= \frac{f^e}{\Gamma(e)} \left(\frac{1}{\sigma^2}\right)^{e+1} \exp\left[-\frac{f}{\sigma^2}\right] \mathbb{1}_{\sigma^2>0}
\end{aligned}
$$

L'inférence de ce modèle sur des données simulées à l'aide de l'algorithme de Metropolis-Hastings Within Gibbs (algorithme 4) décrit dans la section 3 est traitée dans la section 5.

4.2 Modèle hiérarchique de croissance de plantes, selon un système dynamique, où les individus sont en compétition pour de la lumière.

On considère, comme pour le modèle précédent, une population de N individus. On s'intéresse, pour chaque individu $i \in \{1, \cdots, N\}$, à l'évolution d'une grandeur au cours du temps : $s_i(t)$. Chaque individu i possède des paramètres individuels : S_i est la taille maximale que la plante puisse atteindre et γ_i détermine la rapidité de la plante à atteindre sa taille maximale.

Dans ce modèle, les individus ne sont plus à croissances indépendantes. On introduit une compétition, par exemple pour de la lumière. Intuitivement, plus une plante est grande, plus elle fait de l'ombre aux plantes qui sont proches d'elle. Donc plus les plantes sont proches, plus elles sont en compétition. Il est nécessaire d'introduire la position X_i des individus. Le modèle de croissance a la forme du système dynamique suivant : pour tout $1 \leq i \leq N$,

$$
\begin{cases}
s_i(0) = s_0 \\
\dfrac{d\ s_i(t)}{dt} = \gamma_i s_i(t) \left[\log\left(\dfrac{S_i}{s_m}\right)\left(1 - \dfrac{1}{N-1}\displaystyle\sum_{i' \neq i} C\left(s_i(t), s_{i'}(t), \|X_i - X_{i'}\|\right)\right) - \log\left(\dfrac{s_i(t)}{s_m}\right)\right]
\end{cases}
$$

$$
\text{où } C\left(s_i, s_{i'}, \|X_i - X_{i'}\|\right) = \frac{\log\left(\dfrac{s_{i'}}{s_m}\right)}{2R_M\left(1 + \dfrac{\|X_i - X_{i'}\|^2}{\sigma_x^2}\right)}\left(1 + \tanh\left(\frac{\log\left(\dfrac{s_{i'}}{s_i}\right)}{\log\left(\dfrac{\sigma_S}{s_m}\right)}\right)\right) \tag{10}
$$

où s_m et R_M sont des constantes supposées connues. On pose $S_M = s_m e^{R_M}$.

C est appelée la fonction d'interaction. On remarque que si C est constante égale à 0, on retrouve le modèle précédent, sans interaction. C est bien une *fonction de compétition* au sens où elle vérifie :

— Le facteur $1 - \dfrac{1}{N-1} \sum_{i' \neq i} C\left(s_i(t), s_{i'}(t), \|x_i - x_{i'}\|\right)$ est sans dimension et compris entre 0 et 1. Ce facteur réduit la quantité de lumière disponible, représentée par le terme $\log\left(\dfrac{S_i}{s_m}\right)$. Une plante ne peut logiquement pas recevoir plus de lumière en présence d'autres individus que si elle était seule.

— Plus une plante est grande, plus elle exerce de la compétition : c'est le terme $\dfrac{\log\left(\dfrac{s_{i'}}{s_m}\right)}{2R_M}$.

— La compétition que s'exerce sur une plante i est d'autant plus importante que les autres plantes sont plus grandes que la plante i, ce qui vient du facteur $\left(1 + \tanh\left(\dfrac{\log\left(\dfrac{s_{i'}}{s_i}\right)}{\log\left(\dfrac{\sigma_S}{s_m}\right)}\right)\right)$.

— Plus une plante est loin de la plante i, moins elle exerce de compétition. Réciproquement, plus une plante est proche, plus la compétition est forte. C'est ce que traduit le facteur $\dfrac{1}{\left(1 + \dfrac{\|X_i - X_{i'}\|^2}{\sigma_x^2}\right)}$.

Il n'y a pas de croissance typique pour ce modèle. Les plantes sont en interaction donc la croissance de chaque individu dépend significativement de la croissance de chaque individu du reste de la population. Cette dépendance empêche de paralléliser les calculs lors de l'inférence.

La fonction d'interaction est paramétrée par deux paramètres de compétition : σ_x^2 et σ_S. Les positions des individus sont distribuées uniformément sur un carré de côté L : pour tout $1 \leq i \leq N$, $X_i \sim \mathcal{U}\left([0, L]^2\right)$. Les distributions des S_i et γ_i doivent être identifiées sous l'hypothèse de paramétrisation suivante :

$$\begin{cases} \dfrac{\gamma_i}{\gamma_M} \underset{iid}{\sim} \mathcal{B}(a_\gamma, b_\gamma); \\ \dfrac{S_i - s_m}{S_M - s_m} \underset{iid}{\sim} \mathcal{B}(a_S, b_S). \end{cases} \tag{11}$$

La nouvelle difficulté de ce modèle est que l'on va supposer que certains individus ne sont pas observés. On dispose donc d'une base de données d'observations sur n individus, $s^o = \left(s^o_{i,j}\right)_{1 \leq i \leq n,\ 1 \leq j \leq M}$, supposées bruitées et générées par :

$$\forall 1 \leq i \leq N,\ \forall 1 \leq j \leq M,\ s^o_{i,j} = s_i(t_j) + \varepsilon_{i,j} \text{ où } \varepsilon_{i,j} \underset{i.i.d}{\sim} \mathcal{N}(0, \sigma^2).$$

Pour simplifier l'inférence et n'observer que les difficultés liées à l'introduction de compétition, on suppose σ connu. Les paramètres a_S, b_S, a_γ et b_γ suivent des loi *a priori* uniformes sur des intervalles :

$$\begin{cases} a_S \sim \mathcal{U}([\inf_a, \sup_a]) \\ b_S \sim \mathcal{U}([\inf_b, \sup_b]) \\ a_\gamma \sim \mathcal{U}([\inf_a, \sup_a]) \\ b_\gamma \sim \mathcal{U}([\inf_b, \sup_b]) \end{cases} \tag{12}$$

Les paramètres de compétition ont comme loi *a priori* des Inverses Gamma :

$$\begin{cases} \sigma_x^2 \sim \mathcal{IG}(\alpha_x, \beta_x); \\ \sigma_S \sim \mathcal{IG}(\alpha_S, \beta_S). \end{cases} \tag{13}$$

On travaillera dans les sections suivantes sur deux modèles :

1. les positions des individus, $(X_i)_{1\leq i\leq N}$, sont toutes connues ;
2. on ne connaît que les positions des individus observés, $(X_i)_{1\leq i\leq n}$, les autres, $(X_i)_{n<i\leq N}$, sont inconnues.

La figure 4.9 représente le modèle graphique correspondant au second cas.

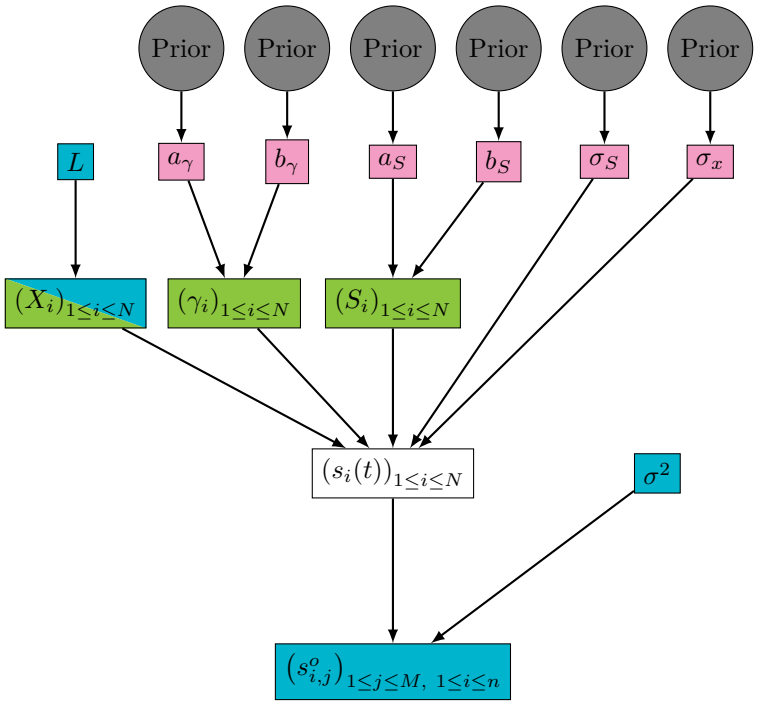


Figure 4.9 – Modèle graphique du modèle de croissance de plantes en compétition pour de la lumière. En gris, les lois *a priori*, en rose, les hyperparamètres, en vert, les paramètres individuels, en bleu, les données.

Ce modèle graphique nous permet d'écrire la densité complète. On rappelle la notation $S = (S_i)_{1\leq i\leq N}$ et $\gamma = (\gamma_i)_{1\leq i\leq N}$. Alors, la densité jointe s'écrit

$$\begin{aligned} p(s^o, S, \gamma, X, \sigma_x^2, \sigma_S, \sigma^2, a_\gamma, b_\gamma, a_S, b_S) = \\ p(s^o|S, \gamma, X, \sigma_x^2, \sigma_S, \sigma^2) \prod_{i=1}^{N} p(S_i|a_S, b_S) p(\gamma_i|a_\gamma, b_\gamma) \\ \prod_{i=n+1}^{N} p(X_i)\ p(\sigma_x^2|\alpha_x, \beta_x)\ p(\sigma_S|\alpha_S, \beta_S)\ p(a_S)\ p(b_S)\ p(a_\gamma)\ p(b_\gamma) \end{aligned}$$

Détaillons la vraisemblance des observations. En notant $f_{i,j}(S,\gamma,X,\sigma_x^2,\sigma_S)$ la solution de (10) pour l'individu i au temps t_j, avec les paramètres individuels S, γ et X et les paramètres de compétition σ_x^2 et σ_S,

$$p(s^o|S,\gamma,X,\sigma_x^2,\sigma_S,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}^{nM}} \exp\left(-\frac{\sum_{i=1}^{n}\sum_{j=1}^{M}\left(s_{i,j}^o - f_{i,j}(S,\gamma,X,\sigma_x^2,\sigma_S)\right)^2}{2\sigma^2} \right)$$

D'autre part, par (11), en notant $B(x,y) = \int_0^1 t^{x-1}(1-t)^{y-1}\mathrm{d}t$ la fonction bêta,

$$\begin{aligned}
p(S_i|a_S,b_S) =& \frac{1}{B(a_S,b_S)} \frac{(S_i - s_m)^{a_S-1}(S_M - S_i)^{b_S-1}}{(S_M - s_m)^{a_S+b_S-2}} \mathbb{1}_{s_m \leq S_i \leq S_M} \\
p(\gamma_i|a_\gamma,b_\gamma) =& \frac{1}{B(a_\gamma,b_\gamma)} \frac{\gamma_i^{a_S-1}(\gamma_M - \gamma_i)^{b_\gamma-1}}{\gamma_M^{a_\gamma+b_\gamma-2}} \mathbb{1}_{0 \leq \gamma_i \leq \gamma_M}.
\end{aligned}$$

Enfin, pour ce qui est des lois *a priori*, en notant $X_i = (x_i, y_i)$,

$$\begin{aligned}
p(\sigma_x^2|\alpha_x,\beta_x) =& \frac{\beta_x^{\alpha_x}}{\Gamma(\alpha_x)} \left(\frac{1}{\sigma_x^2}\right)^{\alpha_x+1} \exp\left(-\frac{\beta_x}{\sigma_x^2}\right) \mathbb{1}_{\sigma_x^2>0} \\
p(\sigma_S|\alpha_S,\beta_S) =& \frac{\beta_S^{\alpha_S}}{\Gamma(\alpha_S)} \left(\frac{1}{\sigma_S}\right)^{\alpha_S+1} \exp\left(-\frac{\beta_S}{\sigma_S}\right) \mathbb{1}_{\sigma_S>0} \\
p(x_i) =& \frac{1}{L}\mathbb{1}_{0<x_i<L} \ ; \ p(y_i) = \frac{1}{L}\mathbb{1}_{0<y_i<L} \\
p(a_S) =& \frac{1}{sup_a - inf_a}\mathbb{1}_{inf_a<a_S<sup_a} \\
p(a_\gamma) =& \frac{1}{sup_a - inf_a}\mathbb{1}_{inf_a<a_\gamma<sup_a} \\
p(b_S) =& \frac{1}{sup_b - inf_b}\mathbb{1}_{inf_b<b_S<sup_b} \\
p(b_\gamma) =& \frac{1}{sup_b - inf_b}\mathbb{1}_{inf_b<b_\gamma<sup_b}
\end{aligned}$$

L'inférence de ce modèle sur des données simulées à l'aide d'algorithmes décrits dans la section 3 est traitée dans la section 5.

4.3 Adaptation d'un modèle GreenLab pour du colza.

Dans cette section, on s'appuie sur le travail de Charlotte Baey qu'elle a réalisé en thèse [Baey, 2014] et publié dans l'article [Baey et al., 2018]. Dans cet article, un modèle GreenLab sur le colza a été appliqué à des données expérimentales dans le cadre d'un modèle mixte ne prenant pas en compte l'interaction entre les individus. On présente ici le modèle GreenLab modifié qui permet d'inclure de la compétition entre les individus. L'inférence sur ce modèle à partir de données expérimentales est réalisée dans la section 6.

Ce modèle est adapté au Colza en stade rosette. Il tient compte de la structure et du fonctionnement de la plante. Chaque feuille est caractérisée par son rang sur la tige principale. Les feuilles sont numérotées du pied au sommet.

On considère ici aussi une population de plantes de N individus. On notera $X_i = (x_i, y_i)$ la position de l'individu i pour $1 \leq i \leq N$.

4.3.1 Temps thermique et phyllochrone.

Les plantes sont sensibles à leur environnement. Les plantes ne sont pas sensibles au temps calendaire mais au *temps thermique*. Le temps thermique de l'expérience se calcule de la manière suivante. On relève $T(s)$ la température moyenne au jour s pour tout $0 \leq s < t$ puis on calcule le temps thermique au jour t :

$$\tau(t) = \int_0^t \max\left(T(s) - T_b, 0\right)\ ds \simeq \sum_{s=0}^{t-1} \max\left(T(s) - T_b, 0\right)$$

où T_b est la température de base, un seuil de température considéré comme constant. Dans le cas du colza, $T_b = 4,5°C$.

Le temps thermique doit dépasser une certaine valeur τ_i^{init} pour que la plante i émerge. Passé ce stade, la plante commence à intercepter de la lumière et débute donc sa production de biomasse par photosynthèse.

Le temps thermique qui sépare l'apparition de deux feuilles successives est appelé le *phyllochrone*. Il est considéré pendant le stade rosette comme constant par phases, propre à chaque individu. Dans ce modèle, il y a deux phases, l'une avant et l'une après le temps thermique de rupture τ^R. Si on note $\tau_{i,j}^{init}$ le temps thermique d'apparition de la feuille de rang j sur la plante i et $\omega_i^{(1)}$ (réciproquement $\omega_i^{(2)}$) le phyllochrone de la plante i pendant la période avant (récip. après) le temps thermique de rupture, alors le nombre de feuilles de la plante i au jour t est

$$N_i^{leaves}(t) = 1 + \left\lfloor \frac{\tau(t) - \tau_i^{init}}{\omega_i^{(1)}} \mathbb{1}_{\tau(t) \geq \tau_i^{init}} + \left(\frac{1}{\omega_i^{(1)}} - \frac{1}{\omega_i^{(2)}}\right) \left(\tau(t) - \tau^R\right) \mathbb{1}_{\tau(t) \geq \tau^R} \right\rfloor.$$

En particulier, si n_i feuilles sont apparues sur la plante i avant la rupture, alors

$$\begin{cases} \tau_{i,j}^{init} = \tau_i^{init} + (j-1)\ \omega_i^{(1)} & \text{pour tout } 1 \leq j \leq n_i; \\ \tau_{i,j}^{init} = \tau_i^{init} + (n_i - 1)\ \omega_i^{(1)} + (j - n_i)\ \omega_i^{(2)} & \text{pour tout } j \geq n_i. \end{cases}$$

4.3.2 Répartition de la biomasse.

Comme les plantes de colza sont encore au stade rosette, on suppose que la biomasse n'est située que dans les feuilles, la graine étant attribuée à la feuille 1. Les feuilles reçoivent de la biomasse lorsqu'elles sont en phase d'expansion. Cette phase a une durée thermique fixe, notée τ_e. La fonction de distribution, appelée *puits de biomasse*, dans une feuille au rang j de la plante i est supposée proportionnelle à une loi bêta, paramétrée par a, un paramètre de population et b, une constante du modèle :

$$s_{i,j}(t) = \left(\frac{\tau(t) - \tau_{i,j}^{init}}{\tau_e}\right)^{a-1} \left(1 - \frac{\tau(t) - \tau_{i,j}^{init}}{\tau_e}\right)^{b-1} \mathbb{1}_{\tau_{i,j}^{init} \leq \tau(t) \leq \tau_{i,j}^{init} + \tau_e}.$$

Le quantité de biomasse totale demandée par la plante i est donc

$$d_i(t) = \sum_{j=1}^{N_i^{leaves}(t)} s_{i,j}(t).$$

Alors, la quantité $\frac{s_{i,j}(t)}{d_i(t)}$ représente la proportion de la biomasse qui sera allouée à la feuille j de la plante i au jour t. Si on note $F_i(t)$ la biomasse accumulée durant le jour t par la plante i, alors la masse totale d'une feuille au rang $j > 1$ de la plante i au début du jour t est

$$q_{i,j}(t) = \sum_{s=1}^{t-1} F_i(s) \frac{s_{i,j}(s)}{d_i(s)}.$$

Dans le cas particulier de la première feuille, il nous faut considérer la biomasse issue de la graine, notée q_0. On a alors

$$q_{i,1}(t) = q_0 + \sum_{s=1}^{t-1} F_i(s) \frac{s_{i,1}(s)}{d_i(s)}.$$

D'autre part, la quantité de biomasse totale produite par la plante i au début du jour t vaut

$$Q_i(t) = \sum_{s=1}^{t-1} F_i(s).$$

4.3.3 Surface active.

La biomasse générée par une plante provient principalement la photosynthèse. Il est donc nécessaire de calculer la *surface active*, c'est à dire la surface de feuille capable d'effectuer de la photosynthèse. Chaque feuille a un temps de vie τ_l pendant lequel elle peut faire de la photosynthèse. Passé ce délais, elle n'est plus considérée comme active. La surface active au début du jour t se calcul par l'intermédiaire de la masse active

$$q_i^{act}(t) = \sum_{j=1}^{N_i^{leaves}(t)} q_{i,j}(t) \mathbb{1}_{\tau(t) - \tau_{i,j}^{init} < \tau_{i,j}^{l}}.$$

En divisant par la masse de feuille par unité de surface e, on obtient la surface active au début du jour t :

$$s_i^{act}(t) = \frac{q_i^{act}(t)}{e}.$$

Le calcul de la surface active au jour t permet d'obtenir la quantité de biomasse produite durant le jour t.

4.3.4 Production de biomasse.

La quantité initiale de biomasse est donnée par la masse de la graine, q_0. Après l'apparition de la première feuille, c'est à dire dès que le temps thermique vérifie $\tau(t) > \tau_{i,1}^{init}$, de la biomasse est produite grâce à la photosynthèse. La quantité de biomasse produite au jour t par la plante i, $F_i(t)$, est classiquement donnée par une loi de Beer-Lambert : pour tout $t \in \mathbb{N}^*$ tel que $\tau(t) > \tau_{i,1}^{init}$,

$$F_i(t) = u_t \ \mu \ s^{pr} \left(1 - \exp\left(-k_B \frac{s_i^{act}(t)}{s^{pr}}\right)\right),$$

où u_t est le rayonnement photo-synthétiquement actif (PAR) et traduit l'effet des conditions environnementales du jour t, μ traduit l'efficacité de la plante pour convertir la lumière en biomasse, s^{pr} traduit l'effet de la densité en plantes dans laquelle pousse la plante i, k_B est le coefficient d'absorption de la loi de Beer-Lambert et $s_i^{act}(t)$ est la surface de feuille active pour la photosynthèse au début du jour t.

Dans cette étape du modèle GreenLab, il est alors possible d'introduire de la compétition avec une fonction du même type que pour le modèle de Schneider (équation (10)). Cette fonction de compétition, pour une paire d'individus i et i', dépend de la biomasse totale de chacun des individus, Q_i et $Q_{i'}$, et de la distance entre les plantes, $\|X_i - X_{i'}\|$. La compétition exercée par la plante i' sur la plante i est :

$$C\left(Q_i, Q_{i'}, \|X_i - X_{i'}\|\right) = \frac{\tanh\left(\dfrac{Q_{i'}}{s_m}\right)}{2\left(1 + \dfrac{\|X_i - X_{i'}\|^2}{\sigma_x^2}\right)} \left(1 + \tanh\left(\frac{Q_{i'} - Q_i}{\sigma_S}\right)\right),$$

où s_m est une constante et σ_x^2 et σ_S sont des paramètres de compétition. La fonction de production au jour t devient :

$$F_i(t) = u_t \ \mu \ s^{pr} \left(1 - \exp\left(-k_B \frac{s_i^{act}(t)}{s^{pr}}\right)\right) \left[1 - \frac{1}{N-1} \sum_{i' \neq i} C\left(Q_i(t), Q_{i'}(t), \|X_i - X_{i'}\|\right)\right].$$

4.3.5 Fonction de transition.

La population de plantes que l'on considère peut-être caractérisée, au jour t, par un état E_t qui contient l'état, $e_{i,t}$, de chaque individu i tel que $1 \leq i \leq N$, au jour t. D'après ce qui précède, pour tout individu $i \in \{1, \cdots, N\}$, $e_{i,t}$ contient

- $N_i^{leaves}(t)$ le nombre de feuilles au début du jour t ;
- $Q_i(t)$ la quantité de biomasse totale au début du jour t ;
- $q_i^{act}(t)$ la masse active au début du jour t ;
- $s_i^{act}(t)$ la surface active au début du jour t ;
- $F_i(t)$ la quantité de biomasse produite durant le jour t ;
- pour chaque feuille $j \in \{1, \cdots, N_i^{leaves}(t)\}$,
  - $s_{i,j}(t)$ la fonction de puits de biomasse ;
  - $q_{i,j}(t)$ la quantité de biomasse de la feuille au début du jour t.

Pour tout jour $t \in \mathbb{N}^*$, l'équation de transition de l'état E_t à l'état E_{t+1} s'écrit

$$E_{t+1} = F_t(E_t, \text{Env}_t, x, \eta, \nu),$$

où

- Env_t regroupe les paramètres environnementaux du jour t, c'est à dire la température, $T(t)$, et le PAR, u_t ;
- $x = (\mu, a, \sigma_x, \sigma_S)$ contient les paramètres du modèle ;
- $\eta = (\tau_e, \tau_l, T_b, k_B, s^{pr}, e, q_0, s_m, \tau^{init}, \tau^R, b)$ contient les constantes du modèle ;
- $\nu = (\nu_i)_{1 \leq i \leq N}$ contient les paramètres individuels : pour tout individu i, $\nu_i = (\omega_i^{(1)}, \omega_i^{(2)})$;
- F_t est la fonction de transition.

4.3.6 Modélisation.

Comme la compétition n'intervient pas dans le nombre de feuilles, on va distinguer deux modèles :

1. Un modèle lié au nombre de feuilles sur les plantes ;
2. Un modèle lié à la production de biomasse.

Dans un premier temps, intéressons nous au modèle lié au nombre de feuilles. On fait l'hypothèse que l'on est dans un modèle de population, c'est à dire un modèle mixte. Les effets fixes sont τ^R et τ^{init} et les effets mixtes portent sur les phyllochrones. Comme les phyllochrones sont positifs, il est préférable de travailler avec le logarithme du phyllochrone. Alors, pour tout individu i,

$$\begin{cases} \log\left(\omega_i^{(1)}\right) = \log\left(\omega_{pop}^{(1)}\right) + \eta_i \text{ où } \eta_i \underset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_1^2); \\ \log\left(\omega_i^{(2)}\right) = \log\left(\omega_{pop}^{(2)}\right) + \xi_i \text{ où } \xi_i \underset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_2^2). \end{cases}$$

On suppose de plus que l'on dispose d'une base de données d'observations bruitées du nombre de feuilles à différents jours : pour tout individu i tel que $1 \leq i \leq N$ et tout temps t_j tel que $1 \leq j \leq M$,

$$N_{i,j}^o = N_i^{leaves}(t_j) + \varepsilon_{i,j} \text{ où } \varepsilon_{i,j} \underset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$$

Le modèle graphique qui correspond à ce premier modèle est représenté sur la figure 4.10.

Dans le second modèle, les paramètres liés au nombre de feuilles sont supposés connus et ne sont donc pas des paramètres du modèle. Dans ce modèle lié à la production de biomasse, les paramètres sont les paramètres de population μ et a et les paramètres de compétition σ_x et σ_S.

Les paramètres doivent rester positifs, donc les lois *a priori* choisies sont des lois Log-Normales ou des Inverse-Gamma, paramétrées selon les connaissances biologiques comme par exemple les ordres de grandeurs des observations typiques.

$$\begin{cases} \mu \sim \mathcal{LN}(m_\mu, \sigma_\mu) \\ a \sim 1 + \mathcal{LN}(m_a, \sigma_a) \\ \sigma_S \sim \mathcal{IG}(\alpha_S, \beta_S) \\ \sigma_x^2 \sim \mathcal{IG}(\alpha_x, \beta_x) \end{cases} \tag{14}$$

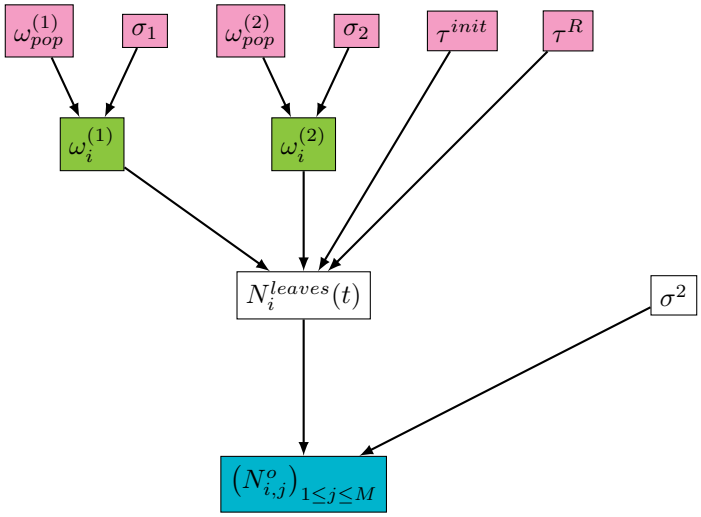


Figure 4.10 – Modèle graphique du nombre de feuilles au cours du temps du modèle GreenLab sur le colza avec compétition. En rose, les paramètres de population, en vert, les paramètres individuels, en bleu les observations

On suppose que l'on dispose d'une base de données d'observations portant sur n individus. Les données pour un individu i sont des relevés de biomasse de certaines feuilles, de rangs $N_i^{min} \leq j \leq N_i^{max}$, au temps final t_f :

$$q_{i,j}^o = q_{i,j}(t_f) + \varepsilon_{i,j}$$

où $\varepsilon_{i,j} \underset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$. On dessine alors un profil pour chaque individu. Un profil typique est tracé sur la figure 4.11.

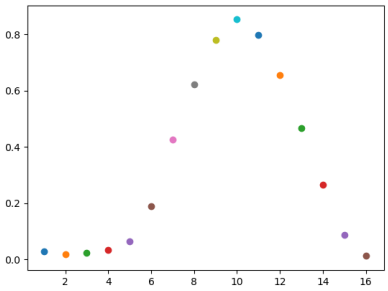


Figure 4.11 – Profil typique de la quantité de biomasse (en g) en fonction du rang de la feuille sur la tige.

On ajoute une loi *a priori* sur la variance d'observation :

$$\sigma^2 \sim \mathcal{IG}(e, f).$$

Sans variabilité individuelle sur les paramètres, l'étage des paramètres individuels n'apparaît pas dans le modèle graphique résumé sur la figure 4.12.

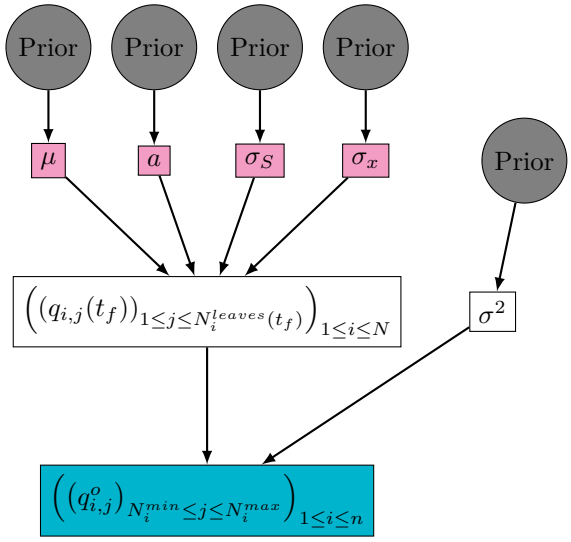


Figure 4.12 – Modèle graphique du modèle GreenLab sur le colza avec compétition et sans variabilité individuelle. En gris, les lois *a priori*, en rose, les hyperparamètres, en bleu, les données.

Ce modèle graphique nous permet d'écrire la densité complète.

$$p(q^o, \mu, a, \sigma^2, \sigma_x^2, \sigma_S) = \underbrace{\prod_{i=1}^{n} \prod_{j=N_i^{min}}^{N_i^{max}} p(q^o_{i,j}|\mu, a, \sigma^2, \sigma_x^2, \sigma_S)}_{\text{Vraisemblance}} \times \underbrace{p(\mu|m_\mu, \sigma_\mu)\ p(a|m_a, \sigma_a)\ p(\sigma_x^2|\alpha_x, \beta_x)\ p(\sigma_S|\alpha_S, \beta_S)\ p(\sigma^2)}_{Prior}$$

avec, d'après (14),

$$\begin{cases} p(\mu|m_\mu, \sigma_\mu) = \dfrac{1}{\mu\sqrt{2\pi\sigma_\mu^2}} \exp\left[-\dfrac{(\log(\mu) - m_\mu)^2}{2\sigma_\mu^2}\right] \mathbb{1}_{\mu>0}; \\ p(a|m_a, \sigma_a) = \dfrac{1}{(a-1)\sqrt{2\pi\sigma_a^2}} \exp\left[-\dfrac{(\log(a-1) - m_a)^2}{2\sigma_a^2}\right] \mathbb{1}_{a>1}; \\ p(\sigma_x^2|\alpha_x, \beta_x) = \dfrac{\beta_x^{\alpha_x}}{\Gamma(\alpha_x)} \left(\dfrac{1}{\sigma_x^2}\right)^{\alpha_x+1} \exp\left(-\dfrac{\beta_x}{\sigma_x^2}\right) \mathbb{1}_{\sigma_x^2>0}; \\ p(\sigma_S|\alpha_S, \beta_S) = \dfrac{\beta_S^{\alpha_S}}{\Gamma(\alpha_S)} \left(\dfrac{1}{\sigma_S}\right)^{\alpha_S+1} \exp\left(-\dfrac{\beta_S}{\sigma_S}\right) \mathbb{1}_{\sigma_S>0}; \\ p(\sigma^2) = \dfrac{f^e}{\Gamma(e)} \left(\dfrac{1}{\sigma^2}\right)^{e+1} \exp\left[-\dfrac{f}{\sigma^2}\right] \mathbb{1}_{\sigma^2>0}. \end{cases}$$

L'inférence de ce modèle par un algorithme de Metropolis-Hastings Within Gibbs (algorithme 4) est traitée dans la section 6.

5 Application sur des données simulées.

Dans cette section, les résultats énoncés ont été obtenus à partir de données simulées disponibles en annexe A. L'objectif est de tester les algorithmes présentés dans la section 3 sur les modèles mathématiques présentés dans la section précédente avant de les appliquer sur des données réelles (section 6).

Des calculs sont nécessaires avant d'appliquer les algorithmes. Les calculs sont également détaillés dans cette section.

5.1 Application de l'algorithme Metropolis-Hastings within Gibbs ...

5.1.1 ... à un modèle de Schneider sans compétition

Le modèle utilisé ici est celui explicité dans la section 4.1. Pour appliquer l'algorithme de Metropolis-Hastings Within Gibbs (algorithme 4), on a vu dans la section 3.3.1 qu'il nous suffisait d'écrire les densité des lois conditionnelles à constante multiplicative près en ne conservant que les termes faisant intervenir le paramètre.

On peut alors isoler les densités des lois conditionnelles à constante multiplicative près. Rappelons que la notation $p(x|\cdots)$ désigne la densité conditionnelle de x sachant tous les autre paramètres du modèle.

Pour les paramètres individuels de tout individu $1 \leq i \leq N$, on obtient :

$$\begin{aligned} p(S_i|...) &\propto_{S_i} \prod_{j=1}^{M} p(s^o_{i,j}|h(S_i,\gamma_i,t_j),\sigma^2)p(S_i|\mu_S,\sigma_S^2) \\ &\propto \frac{1}{(S_i-s_m)} \exp\left[-\frac{1}{2}\left[\frac{(\log(S_i-s_m)-\mu_S)^2}{\sigma_S^2} + \frac{\sum_{j=1}^{M}(s^o_{i,j}-h(S_i,\gamma_i,t_j))^2}{\sigma^2}\right]\right]\mathbb{1}_{S_i>s_m} \end{aligned}$$

$$\begin{aligned} p(\gamma_i|...) &\propto_{\gamma_i} \prod_{j=1}^{M} p(s^o_{i,j}|h(S_i,\gamma_i,t_j),\sigma^2)\ p(\gamma_i|\mu_\gamma,\sigma_\gamma^2) \\ &\propto \frac{1}{\gamma_i} \exp\left[-\frac{1}{2}\left[\frac{(\log(\gamma_i)-\mu_\gamma)^2}{\sigma_\gamma^2} + \frac{\sum_{j=1}^{M}(s^o_{i,j}-h(S_i,\gamma_i,t_j))^2}{\sigma^2}\right]\right]\mathbb{1}_{\gamma_i>0} \end{aligned}$$

Pour les paramètres de population, on obtient les expressions suivantes.

Pour le paramètre μ_S,

$$
\begin{aligned}
p(\mu_S|...) &\propto_{\mu_S} \prod_{i=1}^{N} p(S_i|\mu_S, \sigma_S^2)\ p(\mu_S) \\
&\propto \exp\left[-\frac{1}{2\sigma_S^2}\sum_{i=1}^{N}(\log(S_i - s_m) - \mu_S)^2 - \frac{1}{2\tau^2}(\mu_S - \mu)^2\right] \\
&\propto \exp\left[-\frac{1}{2}\left[\mu_S^2\left(\frac{N}{\sigma^2} + \frac{1}{\tau^2}\right) - 2\mu_S\left(\frac{\mu}{\tau^2} + \sum_{i=1}^{N}\frac{\log(S_i - s_m)}{\sigma_S^2}\right)\right]\right] \\
&= \mathcal{N}\left(moy = \frac{\sigma_S^2\mu + \tau^2\sum\limits_{i=1}^{N}\log(S_i - s_m)}{N\tau^2 + \sigma_S^2}, var = \frac{\sigma_S^2\tau^2}{N\tau^2 + \sigma_S^2}\right)
\end{aligned}
$$

De même,

$$
p(\mu_\gamma|...) = \mathcal{N}\left(moy = \frac{\sigma_\gamma^2\nu + \rho^2\sum\limits_{i=1}^{N}\log(\gamma_i)}{N\rho^2 + \sigma_\gamma^2}, var = \frac{\sigma_\gamma^2\rho^2}{N\rho^2 + \sigma_\gamma^2}\right)
$$

Par ailleurs,

$$
\begin{aligned}
p(\sigma_S^2|...) &\propto_{\sigma_S} \prod_{i=1}^{N} p(S_i|\mu_S, \sigma_S^2)\ p(\sigma_S^2) \\
&\propto \left(\frac{1}{\sigma_S^2}\right)^{\frac{N}{2} + a + 1} \exp\left[-\frac{\frac{1}{2}\sum\limits_{i=1}^{N}(\log(S_i - s_m) - \mu_S)^2 + b}{\sigma_S^2}\right]\mathbb{1}_{\sigma_S^2 > 0} \\
&= \mathcal{IG}\left(\frac{N}{2} + a, \frac{1}{2}\sum_{i=1}^{N}(\log(S_i - s_m) - \mu_S)^2 + b\right)
\end{aligned}
$$

De même,

$$
p(\sigma_\gamma^2|...) = \mathcal{IG}\left(\frac{N}{2} + c, \frac{1}{2}\sum_{i=1}^{N}(\log(\gamma_i) - \mu_\gamma)^2 + d\right)
$$

Enfin,

$$p(\sigma^2|...) \propto_{\sigma^2} \prod_{i=1}^{N}\prod_{j=1}^{M} p(s^o_{i,j}|h(S_i,\gamma_i,t_j),\sigma^2)\ p(\sigma^2) \tag{15}$$

$$\propto \left(\frac{1}{\sigma^2}\right)^{\frac{NM}{2}+e+1} \exp\left[-\frac{\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}(s^o_{i,j}-h(S_i,\gamma_i,t_j))^2+f}{\sigma^2}\right]\mathbb{1}_{\sigma^2>0} \tag{16}$$

$$= \mathcal{IG}\left(\frac{NM}{2}+e, \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}(s^o_{i,j}-h(S_i,\gamma_i,t_j))^2+f\right) \tag{17}$$

Les lois conditionnelles des paramètres de population sont faciles à simuler (on a choisi les lois *a priori* pour que ce soit le cas). Pour ces paramètres, on va pouvoir appliquer une étape de Gibbs classique. Pour les paramètres individuels en revanche, une étape de type acceptation/rejet de Metropolis-Hastings est nécessaire.

Résultats

L'algorithme de Metropolis-Hastings Within Gibbs (algorithme 4) a été appliqué sur des données simulées de $N = 10$ individus, observés à $M = 10$ temps différents. La figure A.30 (en annexe A) représente les données simulées dont on dispose.

On observe que les plantes ont bien un comportement typique mais aussi que la population est hétérogène. En effet, l'individu 2 pousse plus haut mais plus lentement que les autres plantes tandis que l'individu 6 a un comportement opposé, il atteint très rapidement sa taille maximale qui est plus petite que la moyenne.

Le tableau 5.1 regroupe les valeurs utilisées pour simuler le modèle.

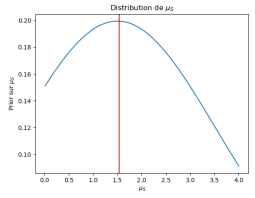
Lois *a priori*									
μ	τ	a	b	ν	ρ	c	d	e	f
1.5	2.0	3.0	1.0	0.0	1.0	2.0	1.0	4.0	1.0

Paramètres de simulation				
$\inf(S_i)$	$\sup(S_i)$	$\sup(\gamma_i)$	Taille initiale	Variance d'observation
s_m	S_M	γ_M	s_0	σ
5e-2	1.0	1.0	0.2	0.5

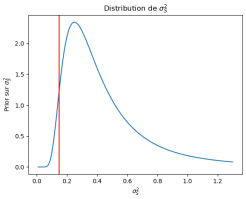
Table 5.1 – Valeurs des paramètres utilisés pour simuler le modèle de Schneider sans compétition.

Avant d'appliquer un algorithme, il faut s'assurer que le modèle est correct. Les vraies valeurs des hyperparamètres doivent être des valeurs probables pour les lois *a priori* et les densités conditionnelles doivent être régulières. Les lois *a*

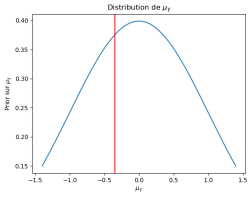
priori sont choisies de manière à avoir la vraie valeur proche du mode. C'est ce que l'on observe sur les figures 5.13(a) à 5.13(e). Les densités sont plus faciles à retrouver avec les algorithmes lorsqu'elles sont régulières et unimodales. C'est ce que l'on peut observer sur la figure 5.13(f).



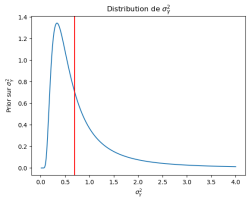
(a) Choix du prior de μ_S.



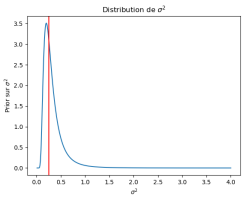
(b) Choix du prior de σ_S^2.



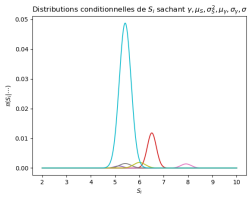
(c) Choix du prior de μ_γ.



(d) Choix du prior de σ_γ^2.



(e) Choix du prior de σ^2.



(f) Régularité des densités conditionnelles.

Figure 5.13 – Vérifications sur le modèle de Schneider sans interaction avant le lancement d'un algorithme.

Le tableau 5.2 donne les paramètres utilisés pour inférer ce modèle. Tous les paramètres sont explorés selon la même variance σ_i, c'est à dire que tous les paramètres ont pour loi de proposition la marche aléatoire gaussienne de variance σ_i.

Paramètres d'inférence	
Variance d'exploration	Nombre d'itérations
σ_i	K
0.5	100 000

Table 5.2 – Valeurs des paramètres d'inférence pour un modèle de Schneider sans interaction, inféré à l'aide d'un Metropolis-Hastings Within Gibbs.

L'algorithme de Metropolis-Hastings est initialisé sur une réalisation des lois *a priori* pour les hyperparamètres, puis, une fois donnés les hyperparamètres, les paramètres individuels initiaux sont des réalisations des distributions de population.

La figure 5.14 montre l'évolution des états pris, sur les coordonnées des hyperparamètres, par la chaîne de Markov qui est construite au cours des itérations de l'algorithme. On voit que la chaîne explore convenablement l'espace d'état. On observe également que la chaîne semble avoir convergé vers un état d'équilibre.

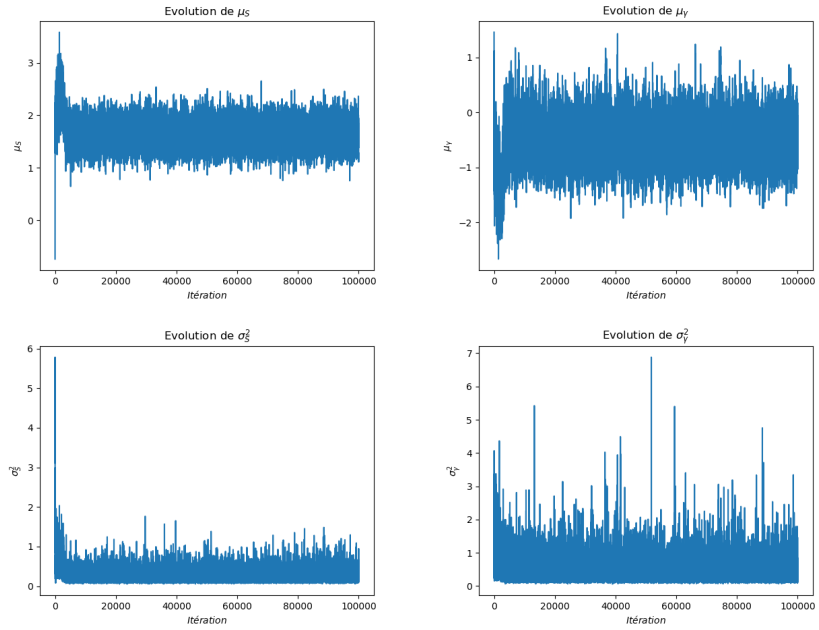


Figure 5.14 – Évolution de l'état des hyperparamètres de la chaîne de Markov, pour le modèle de Schneider sans compétition, au cours des itérations de l'algorithme de MHWG initialisé sur une réalisation de la loi *a priori*.

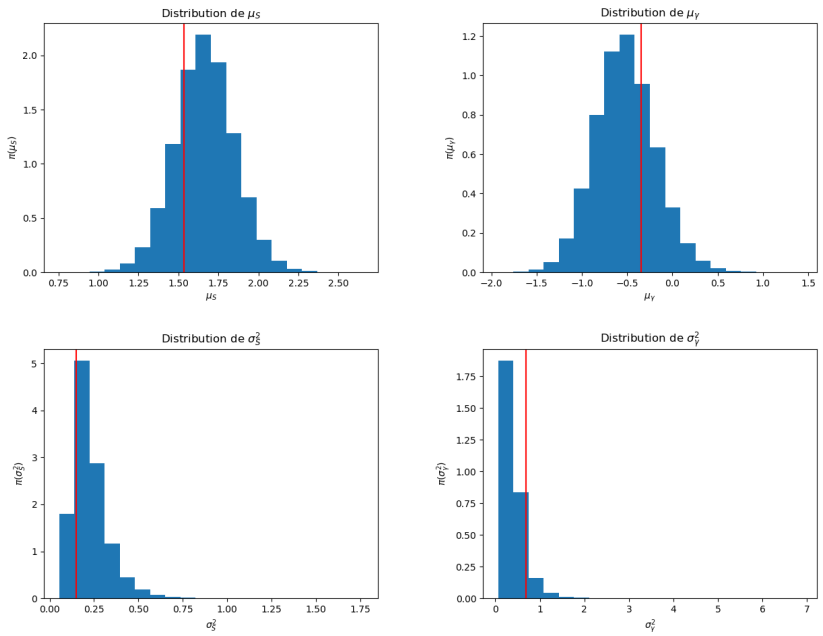


Figure 5.15 – En bleu, distributions empiriques, pour le modèle de Schneider sans compétition, des hyperparamètres de la chaîne de Markov obtenue avec l'algorithme MHWG initialisé sur une réalisation de la loi *a priori*. En rouge, la valeur utilisée pour simuler les données.

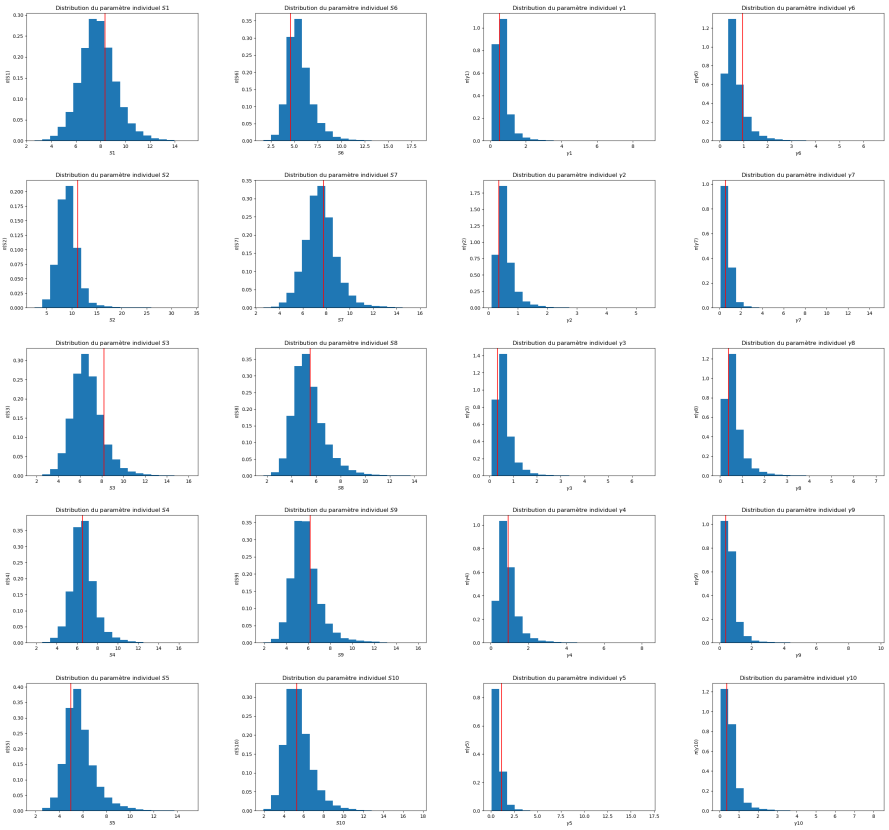


Figure 5.16 – En bleu, distributions empiriques, pour le modèle de Schneider sans compétition, des paramètres individuels de la chaîne de Markov obtenue avec l'algorithme MHWG initialisé sur une réalisation de la loi *a priori*. En rouge, la valeur utilisée pour simuler les données. À gauche, les S_i, à droite, les γ_i.

On déduit de la figure 5.14 le support d'une loi stationnaire pour cette chaîne de Markov. On élimine les premières valeurs (20%) de la chaîne pour pouvoir construire un histogramme. On obtient ainsi la Distributions empiriques d'une loi stationnaire de la chaîne de Markov construite (figure 5.15 pour les hyper-paramètres et figure 5.16 pour les paramètres individuels).

Sur les figures 5.15 et 5.16, les distributions *a posteriori* semblent toutes avoir leur mode proche de la vraie valeur. Une interprétation possible de cette remarque est que la valeur la plus probable *a posteriori* est proche de la vraie valeur ayant servi à simuler les données. Ce résultat est rassurant puisque, lorsque le nombre d'observations tend vers l'infini, la distribution *a posteriori* tend vers un Dirac en la vraie valeur.

5.1.2 ... à un modèle de Schneider avec compétition

Le modèle utilisé ici est celui explicité dans la section 4.2. Pour appliquer l'algorithme de Metropolis-Hastings Within Gibbs (algorithme 4), il nous suffit d'isoler les densités des lois conditionnelles à constante multiplicative près.

Rappelons la notation $f_{i,j}(S,\gamma,X,\sigma_x^2,\sigma_S)$ pour la solution de (10) pour l'individu i au temps t_j, avec les paramètres individuels S, γ et X et les paramètres de compétition σ_x^2 et σ_S.

Pour les paramètres individuels d'un individu $1 \leq i \leq N$, on obtient :

$$\begin{aligned}
p(S_i|...) &\propto p(s^o|S,\gamma,X,\sigma_x^2,\sigma_S,\sigma^2) \times p(S_i|a_S,b_S) \\
&\propto \prod_{i'=1}^{n}\prod_{j=1}^{M} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{\left(s^o_{i',j} - f_{i',j}(S,\gamma,X,\sigma_x^2,\sigma_S)\right)^2}{2\sigma^2}\right] \\
&\quad \times \frac{1}{B(a_S,b_S)} \frac{(S_i - s_m)^{a_S-1}(S_M - S_i)^{b_S-1}}{(S_M - s_m)^{a_S+b_S-2}} \mathbb{1}_{s_m \leq S_i \leq S_M} \\
&\propto (S_i - s_m)^{a_S-1} (S_M - S_i)^{b_S-1} \\
&\quad \exp\left[-\frac{\sum_{i'=1}^{n}\sum_{j=1}^{M}\left(s^o_{i',j} - f_{i',j}\left(S,\gamma,X,\sigma_x^2,\sigma_S\right)\right)^2}{2\sigma^2}\right] \mathbb{1}_{s_m \leq S_i \leq S_M}
\end{aligned}$$

$$\begin{aligned}
p(\gamma_i|...) &\propto p(s^o|S,\gamma,X,\sigma_x^2,\sigma_S,\sigma^2) \times p(\gamma_i|a_\gamma,b_\gamma) \\
&\propto \prod_{i'=1}^{n}\prod_{j=1}^{M} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{\left(s^o_{i',j} - f_{i',j}(S,\gamma,X,\sigma_x^2,\sigma_S)\right)^2}{2\sigma^2}\right] \\
&\quad \times \frac{1}{B(a_\gamma,b_\gamma)} \frac{(\gamma_i)^{a_\gamma-1}(\gamma_M - \gamma_i)^{b_\gamma-1}}{\gamma_M^{a_\gamma+b_\gamma-2}} \mathbb{1}_{0 \leq \gamma_i \leq \gamma_M} \\
&\propto (\gamma_i)^{a_\gamma-1}(\gamma_M - \gamma_i)^{b_\gamma-1} \exp\left[-\frac{\sum_{i'=1}^{n}\sum_{j=1}^{M}\left(s^o_{i',j} - f_{i',j}(S,\gamma,X,\sigma_x^2,\sigma_S)\right)^2}{2\sigma^2}\right] \mathbb{1}_{0 \leq \gamma_i \leq \gamma_M}
\end{aligned}$$

Une étape acceptation/rejet de type Metropolis-Hastings nécessite le calcul du rapport des densités. On rappelle que $S_M = s_m e^{R_M}$. Pour tout $1 \leq i \leq N$, en notant y le candidat proposé et f_x la solution du schéma avec le paramètre x, on a, lorsque $s_m \leq y \leq S_M$,

$$r_{S_i}(S_i,y) = \left(\frac{y - s_m}{S_i - s_m}\right)^{a_S-1} \left(\frac{S_M - y}{S_M - S_i}\right)^{b_S-1} \exp\left[-\frac{\sum (s^o_{i,j} - f_y)^2 - (s^o_{i,j} - f_{S_i})^2}{2\sigma^2}\right].$$

Afin d'optimiser au mieux les calculs et d'éviter les zéros machines, on travaillera avec le logarithme du rapport :

$$\begin{aligned}
&\log\left(r_{S_i}(S_i,y)\right) = (a_S - 1)\left[\log(y - s_m) - \log(S_i - s_m)\right] + \\
&\quad (b_S - 1)\left[\log(S_M - y) - \log(S_M - S_i)\right] - \frac{\sum(2s^o_{i,j} - f_y - f_{S_i})(f_{S_i} - f_y)}{2\sigma^2}.
\end{aligned}$$

De même, lorsque $0 \leq y \leq \gamma_M$,

$$\begin{aligned}
&\log\left(r_{\gamma_i}(\gamma_i,y)\right) = (a_\gamma - 1)\left[\log y - \log\gamma_i\right] + \\
&\quad (b_\gamma - 1)\left[\log(\gamma_M - y) - \log(\gamma_M - \gamma_i)\right] - \frac{\sum(2s^o_{i,j} - f_y - f_{\gamma_i})(f_{\gamma_i} - f_y)}{2\sigma^2}.
\end{aligned}$$

D'autre part, pour les paramètres de compétition,

$$
\begin{aligned}
p(\sigma_x^2|\cdots) &\propto p(s^o|\gamma, S, \sigma_x^2, \sigma_S, \sigma^2, X) \times p(\sigma_x^2|\alpha_x, \beta_x) \\
&\propto \prod_{i'=1}^{n}\prod_{j=1}^{M} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{\left(s^o_{i',j} - f_{i',j}(\gamma, S, \sigma_x^2, \sigma_S, X)\right)^2}{2\sigma^2}\right] \\
&\quad \times \frac{\beta_x^{\alpha_x}}{\Gamma(\alpha_x)} \left(\frac{1}{\sigma_x^2}\right)^{\alpha_x+1} \exp\left[-\frac{\beta_x}{\sigma_x^2}\right] \mathbb{1}_{\sigma_x^2>0} \\
&\propto \left(\frac{1}{\sigma_x^2}\right)^{\alpha_x+1} \exp\left[-\frac{\sum_{i'=1}^{n}\sum_{j=1}^{M}\left(s^o_{i',j} - f_{i',j}(\gamma, S, \sigma_x^2, \sigma_S^2, X)\right)^2}{2\sigma^2} - \frac{\beta_x}{\sigma_x^2}\right] \mathbb{1}_{\sigma_x^2>0}
\end{aligned}
$$

Alors, lorsque $y > 0$, le rapport de densité est

$$
r_{\sigma_x^2}(\sigma_x^2, y) = \left(\frac{\sigma_x^2}{y}\right)^{\alpha_x+1} \exp\left[-\frac{\sum_{i,j}(s^o_{i,j} - f_y)^2 - (s^o_{i,j} - f_{\sigma_x^2})^2}{2\sigma^2} - \frac{\beta_x}{y} + \frac{\beta_x}{\sigma_x^2}\right].
$$

Donc, si $y > 0$,

$$
\begin{aligned}
\log\left(r_{\sigma_x^2}(\sigma_x^2, y)\right) &= (\alpha_x + 1)\left(\log(\sigma_x^2) - \log(y)\right) + \\
&\qquad \beta_x\left(\frac{1}{\sigma_x^2} - \frac{1}{y}\right) - \frac{\sum_{i,j}(2s^o_{i,j} - f_y - f_{\sigma_x^2})(f_{\sigma_x^2} - f_y)}{2\sigma^2}.
\end{aligned}
$$

De même, si $y > 0$,

$$
\begin{aligned}
\log\left(r_{\sigma_S}(\sigma_S, y)\right) &= (\alpha_S + 1)\left(\log(\sigma_S) - \log(y)\right) + \\
&\qquad \beta_S\left(\frac{1}{\sigma_S} - \frac{1}{y}\right) - \frac{\sum_{i,j}(2s^o_{i,j} - f_y - f_{\sigma_S})(f_{\sigma_S} - f_y)}{2\sigma^2}.
\end{aligned}
$$

Pour les paramètres de population,

$$
\begin{aligned}
p(a_S|\cdots) &\propto \prod_{i=1}^{N} p(S_i|a_S, b_S) \times p(a_S) \\
&\propto \prod_{i=1}^{N} \frac{1}{B(a_S, b_S)} \frac{(S_i - s_m)^{a_S-1}(S_M - S_i)^{b_S-1}}{(S_M - s_m)^{a_S+b_S-2}} \mathbb{1}_{s_m \le S_i \le S_M} \\
&\quad \times \frac{1}{\sup_{a_S} - \inf_{a_S}} \mathbb{1}_{\inf_{a_S} \le a_S \le \sup_{a_S}} \\
&\propto \frac{1}{B(a_S, b_S)^N} \left(\frac{\prod_{i=1}^{N}(S_i - s_m)}{(S_M - s_m)^N}\right)^{a_S} \mathbb{1}_{\inf_{a_S} \le a_S \le \sup_{a_S}}
\end{aligned}
$$

Donc, si $\inf_{a_S} \leq y \leq \sup_{a_S}$,

$$r_{a_S}(a_S, y) = \left(\frac{B(a_S, b_S)}{B(y, b_S)}\right)^N \left(\frac{\prod_{i=1}^{N}(S_i - s_m)}{(S_M - s_m)^N}\right)^{y - a_S}.$$

Et ainsi,

$$\log\left(r_{a_S}(a_S, y)\right) = N\left(\log(B(a_S, b_S)) - \log(B(y, b_S))\right) + (y - a_S)\left(\sum_{i=1}^{N} \log\left((S_i - s_m)\right) - N\log(S_M - s_m)\right).$$

Un calcul similaire donne les logarithmes de r_{a_γ}, r_{b_S} et r_{b_γ}.

Enfin, lorsque les plantes non observées sont de positions inconnues,

$$\begin{aligned} p(x_i|\cdots) &\propto p(s^o|\gamma, S, \sigma_x^2, \sigma_S, \sigma^2, X) \times p(x_i) \\ &\propto \prod_{i'=1}^{n}\prod_{j=1}^{M} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{\left(s^o_{i',j} - f_{i',j}(\gamma, S, \sigma_x^2, \sigma_S, X)\right)^2}{2\sigma^2}\right] \times \mathbb{1}_{[0,L]^2}(x_i) \\ &\propto \exp\left[-\frac{\sum_{i'=1}^{n}\sum_{j=1}^{M}\left(s^o_{i',j} - f_{i',j}(\gamma, S, \sigma_x^2, \sigma_S, X)\right)^2}{2\sigma^2}\right] \mathbb{1}_{[0,L]}(x_i). \end{aligned}$$

Donc, si $0 \leq y \leq 1$, le rapport de densité vaut

$$\log(r_{x_i}(x_i, y)) = -\frac{\sum_{i,j}(2s^o_{i,j} - f_y - f_{x_i})(f_{x_i} - f_y)}{2\sigma^2}.$$

On obtient, avec les mêmes calculs, le logarithme de $r_{y_i}(y_i, y)$.

Résultats

Un algorithme de Metropolis-Hastings Within Gibbs a été appliqué pour un modèle de population à $N = 20$ individus dont seulement la moitié ($n = 10$) est observée à $M = 10$ temps différents. La figure A.31 (en annexe A) représente des données simulées dont on dispose. La figure A.32 représente les croissances des plantes non observées.

Une simplification a été effectuée :

— $\inf_{a_S} = \inf_{a_\gamma} = \inf_a$,
— $\sup_{a_S} = \sup_{a_\gamma} = \sup_a$,
— $\inf_{b_S} = \inf_{b_\gamma} = \inf_b$,
— $\sup_{b_S} = \sup_{b_\gamma} = \sup_b$.

Le tableau 5.3 regroupe les valeurs utilisées pour simuler et inférer le modèle.

Lois *a priori*							
α_x	β_x	α_S	β_S	$\inf_a$	$\sup_a$	$\inf_b$	$\sup_b$
2.0	2.0	10.0	1.1	1.0	4.0	1.0	7.0

Constantes du modèle et paramètres de simulation					
$\inf(S_i)$	$\sup(S_i)$	$\sup(\gamma_i)$	Taille initiale	Côté du champs	Variance d'observation
s_m	S_M	γ_M	s_0	L	σ
5e-2	1.0	1.0	0.2	1.0	0.01

Paramètres d'inférence	
Variance d'exploration	Nombre d'itérations
σ_i	K
0.5	100 000

Table 5.3 – Valeurs des paramètres utilisés pour simuler et inférer le modèle de Schneider avec compétition à l'aide d'un MHWG.

5.1.3 Cas des positions inconnues pour les individus non observés.

Le premier modèle auquel l'algorithme a été appliqué est celui qui considère que l'on n'a aucune information sur les individus non observés. C'est le cas par exemple dans les cultures agricoles en champs.

L'algorithme nécessite environ 20h de calculs.

La figure 5.17 montre l'évolution des états pris, sur les coordonnées des hyperparamètres, par la chaîne de Markov qui est construite au cours de l'algorithme. La figure 5.18 représente la Distributions empiriques d'une loi stationnaire de cette chaîne.

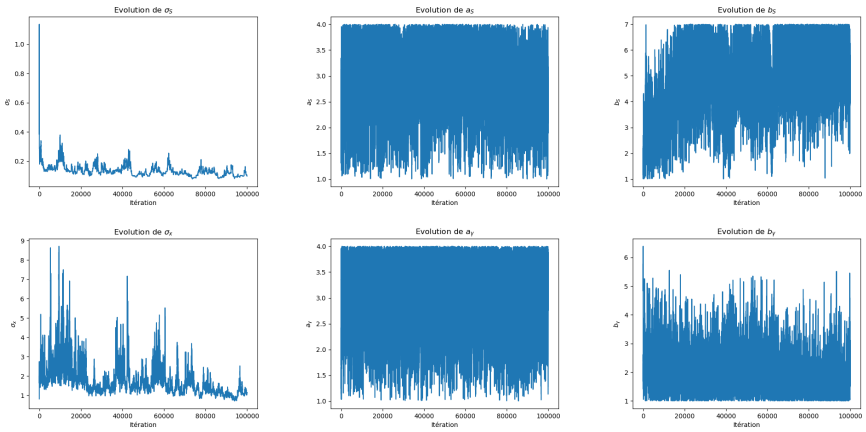


Figure 5.17 – Évolution de l'état des hyperparamètres de la chaîne de Markov, pour le modèle de Schneider avec compétition, au cours des itérations de l'algorithme de MHWG initialisé sur une réalisation de la loi *a priori*.

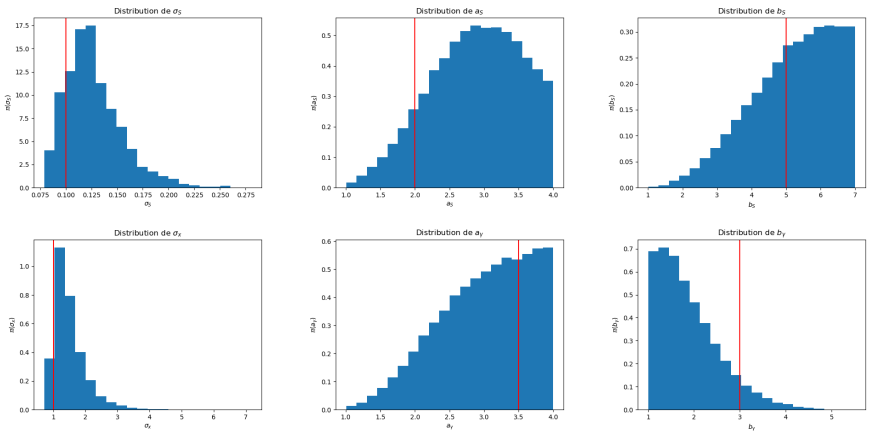


Figure 5.18 – En bleu, distributions empiriques, pour le modèle de Schneider avec compétition, des hyperparamètres de la chaîne de Markov obtenue avec l'algorithme MHWG initialisé sur une réalisation de la loi *a priori*. En rouge, la valeur utilisée pour simuler les données.

On regarde aussi les distributions *a posteriori* empiriques des paramètres individuels S_i (figure 5.19). On observe deux comportements distincts : les distributions des S_i pour les individus observés, *i.e.* $1 \leq i \leq n$, sont proches d'un Dirac en les vraies valeurs tandis que pour les individus non observés, *i.e.* $n < i \leq N$, les distributions sont à support très large et sont toutes similaires.

Cette différence peut s'expliquer par le fait que les individus non observés sont interchangeables. Ainsi, le modèle n'est pas identifiable et les paramètres individuels des individus non observés, $(S_i, \gamma_i, X_i)_{n<i\leq N}$, ont la même loi *a posteriori*. Ces variables aléatoires sont dites **échangeables** : pour toute permutation σ de $\{n+1, \cdots, N\}$, la variable aléatoire permutée $\left(S_{\sigma(i)}, \gamma_{\sigma(i)}, X_{\sigma(i)}\right)_{n<i\leq N}$ a même loi que la variable originale $(S_i, \gamma_i, X_i)_{n<i\leq N}$. Ainsi, la distribution *a posteriori* est la même pour tous les (S_i, γ_i, X_i) tels que $n < i \leq N$.

Une solution possible à ce problème d'identifiabilité est de traiter le cas suivant : on suppose que l'on dispose des positions de toutes les plantes, celles qui sont observées et celles qui ne le sont pas.

5.1.4 Cas des positions connues pour les individus non observés.

On suppose ici que les positions de tous les individus sont connues. C'est le cas par exemple pour des expériences contrôlées.

La figure 5.20 représente la Distributions empiriques d'une loi stationnaire de la chaîne de Markov obtenue par un algorithme MHWG en fixant les positions des individus non observés à leur vraie valeur, celle ayant servi à simuler les données. On observe des différences avec le cas précédent (figure 5.18), notamment sur la distribution des paramètres a_S et b_S.

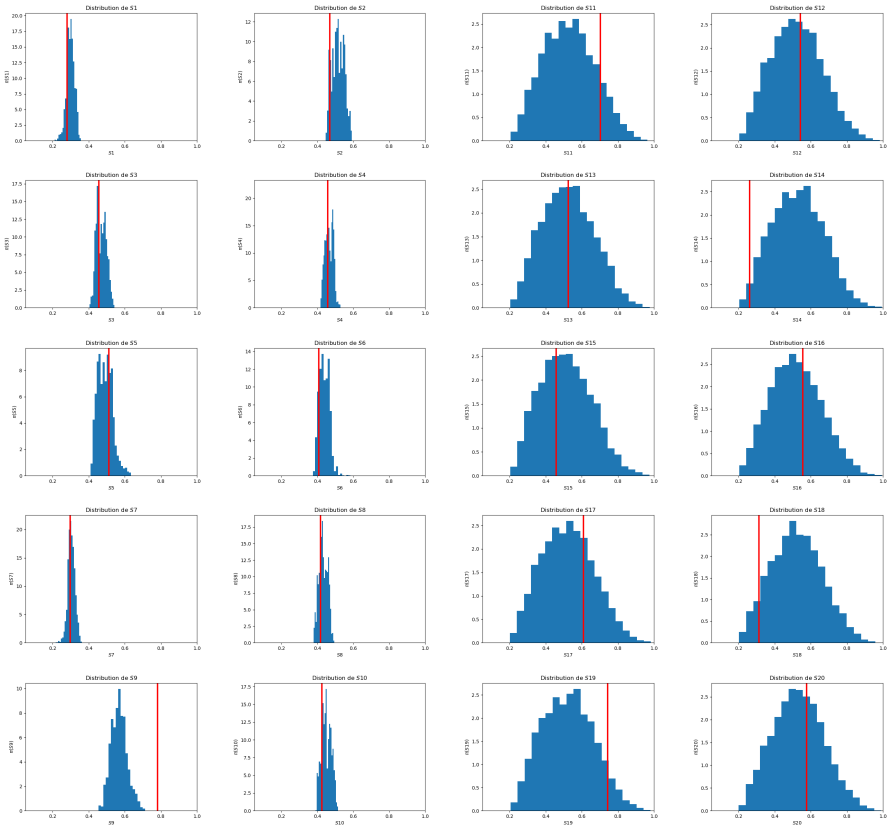


Figure 5.19 – En bleu, distributions empiriques, pour le modèle de Schneider avec compétition, des paramètres individuels S_i de la chaîne de Markov obtenue avec l'algorithme MHWG initialisé sur une réalisation de la loi *a priori*. En rouge, la valeur utilisée pour simuler les données. À gauche, les individus observés, à droite, les individus non observés.

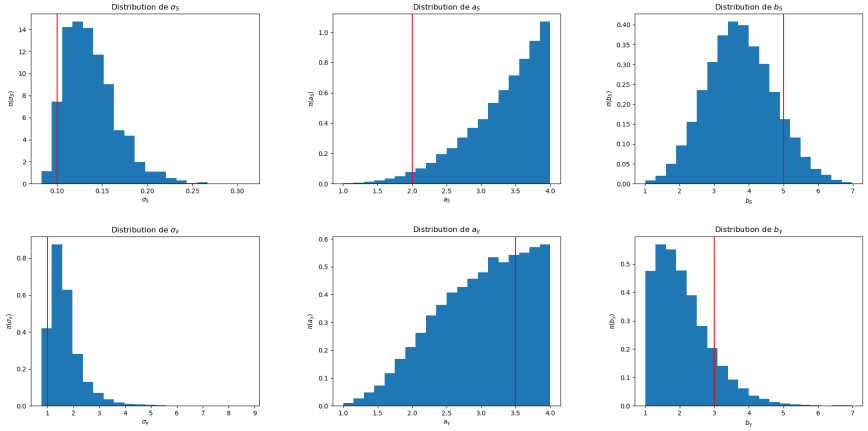


Figure 5.20 – En bleu, distributions empiriques, pour le modèle de Schneider avec compétition et à positions connues de tous les individus, des hyperparamètres de la chaîne de Markov obtenue avec l'algorithme MHWG initialisé sur une réalisation de la loi *a priori*. En rouge, la valeur utilisée pour simuler les données.

La figure 5.21 représente les distributions empirique des paramètres individuels S_i. Les S_i des individus non observés semblent avoir des distributions similaire, comme dans le cas précédent de la figure 5.19.

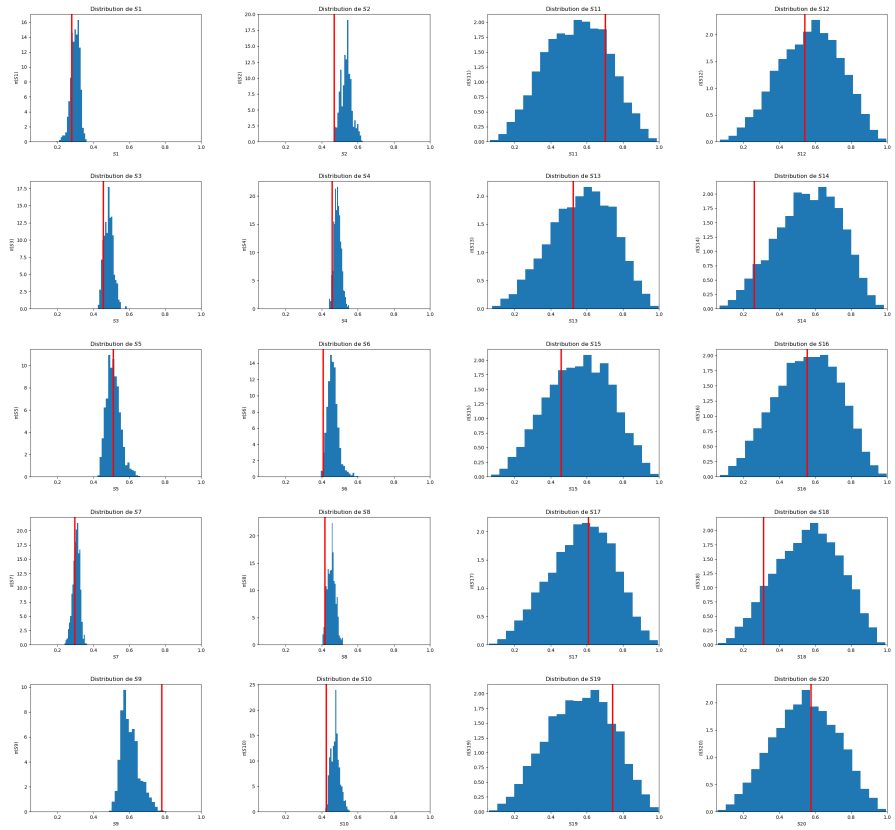


Figure 5.21 – En bleu, distributions empiriques, pour le modèle de Schneider avec compétition et à positions connues de tous les individus, des paramètres individuels S_i de la chaîne de Markov obtenue avec l'algorithme MHWG initialisé sur une réalisation de la loi *a priori*. En rouge, la valeur utilisée pour simuler les données. À gauche, les individus observés, à droite, les individus non observés.

Ce comportement général peut s'expliquer par un effet de population. Lorsque le nombre de plantes tend vers l'infini, les tailles des plantes aux différents instants t se comportent comme des variables aléatoires indépendantes distribuées selon une même loi $\mu[t]$, appelée loi limite de champs moyen [Della Noce et al., 2019].

5.2 Application d'un Monte-Carlo Hamiltonien à un modèle de Schneider avec compétition

Dans cette section, on va utiliser le modèle de la section 4.2 auquel on va effectuer une inférence bayésienne à l'aide d'un Monte-Carlo Hamiltonien (algorithme 6). La dimension de l'espace d'état dans ce modèle est 66, ce qui justifie l'application d'un tel algorithme, adapté à la grande dimension.

On écrit la densité complète sous la forme $\pi \propto \exp(-U)\mathbb{1}_A$ où A désigne le support de la loi jointe. On considère ici tous les paramètres, c'est à dire $S, \gamma, X, \sigma_x^2, \sigma_S, a_\gamma, b_\gamma, a_S$, et b_S. La variance d'observation σ^2 est une constante connue. La densité complète du modèle est donnée par :

$$p(s^o, S, \gamma, X, \sigma_x^2, \sigma_S, a_\gamma, b_\gamma, a_S, b_S) \propto \exp\left(-U(S, \gamma, X, \sigma_x^2, \sigma_S, a_S, b_S, a_\gamma, b_\gamma)\right) \mathbb{1}_A(S, \gamma, X, \sigma_x^2, \sigma_S, a_\gamma, b_\gamma, a_S, b_S)$$

avec

$$U(S, \gamma, X, \sigma_x^2, \sigma_S, a_S, b_S, a_\gamma, b_\gamma) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{M}\left(s^o_{i,j} - f_{i,j}(S, \gamma, X, \sigma_x^2, \sigma_S)\right)^2}{2\sigma^2}$$
$$+N(a_S+b_S-2)\log(S_M-s_m)-(a_S-1)\sum_{i=1}^{N}\log(S_i-s_m)-(b_S-1)\sum_{i=1}^{N}\log(S_M-S_i)$$
$$+N(a_\gamma+b_\gamma-2)\log(\gamma_M)-(a_\gamma-1)\sum_{i=1}^{N}\log(\gamma_i)-(b_\gamma-1)\sum_{i=1}^{N}\log(\gamma_M-\gamma_i)$$
$$+N\left[\log(B(a_S,b_S))+\log(B(a_\gamma,b_\gamma))\right]+\frac{\beta_x}{\sigma_x^2}+(\alpha_x+1)\log(\sigma_x^2)+\frac{\beta_S}{\sigma_S}+(\alpha_S+1)\log(\sigma_S),$$

et

$$A =]s_m, S_M[^N \times]0, \gamma_M[^N \times [0, L]^{N-n} \times \mathbb{R}_+^* \times \mathbb{R}_+^* \times [inf_a, sup_a] \times [inf_b, sup_b] \times [inf_a, sup_a] \times [inf_b, sup_b].$$

Résultats

Le tableau 5.4 regroupe les valeurs de paramètres utilisées pour inférer le modèle de Schneider avec compétition à l'aide d'un Monte-Carlo Hamiltonien, initialisé sur une réalisation des lois *a priori*. L'algorithme étant lent, le nombre d'itérations est fixé à $K = 10000$.

Paramètres d'inférence							
Échelles (matrice de masse)							
s_S	s_γ	s_x	s_y	s_{σ_x}	s_{σ_S}	$s_{a_S} = s_{a_\gamma}$	$s_{b_S} = s_{b_\gamma}$
$S_M - s_m$	γ_M	L	L	$\frac{\beta_x}{\alpha_x + 1}$	$\frac{\beta_S}{\alpha_S + 1}$	$\sup_a - \inf_a$	$\sup_b - \inf_b$

Paramètres du saute-moutons		Nombre d'itérations
L_{HMC}	ε	K
30	0.01	10 000

Table 5.4 – Valeurs utilisées pour inférer le modèle de Schneider avec compétition à l'aide d'un HMC.

Sur la figure 5.22, le tracé de l'évolution de l'état des hyperparamètres de la chaîne de Markov nous montre que l'algorithme n'a pas exploré tout l'espace. Plusieurs raisons peuvent expliquer ce résultat. Dans un premier temps, l'algorithme est très lent. Le nombre d'itérations ne permet donc pas d'obtenir assez de réalisations. Ensuite, l'algorithme HMC est difficile à ajuster.

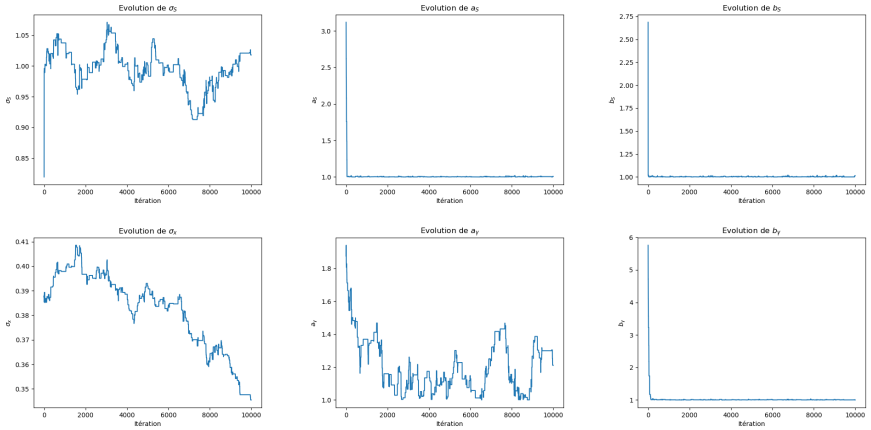


Figure 5.22 – Évolution de l'état des hyperparamètres de la chaîne de Markov, pour le modèle de Schneider avec compétition, au cours des itérations de l'algorithme HMC initialisé sur une réalisation de la loi *a priori*.

La figure 5.23 montre les distributions des paramètres individuels. La même tendance que précédemment peut être observée : les paramètres des individus observés sont estimés sans difficulté tandis que les paramètres des individus non observés ont des distributions empiriques plus larges. L'échangeabilité des variables aléatoires associées aux individus non observés explique ce comportement (voir section 5.1.3).

Afin d'explorer au mieux l'espace des hyperparamètres, on fait le choix d'appliquer ensuite au modèle de Schneider avec compétition, un algorithme combinant Metropolis-Hastings Within Gibbs et le HMC.

5.3 Application d'un Monte-Carlo Hamiltonien dans Gibbs à un modèle de Schneider avec compétition

Dons cette section, on applique l'algorithme hybride Gibbs Within HMC (algorithme 8) au modèle de Schneider présenté dans la section 4.2. Lorsqu'on réduit le HMC aux paramètres individuels, la loi à considérer dans l'étape du HMC est la loi conditionnelle des paramètres individuels sachant les hyperparamètres : $p(S, \gamma, X | a_S, b_S, a_\gamma, b_\gamma, \sigma_S, \sigma_x^2) \propto \exp\left[U(S, \gamma, X)\right] \mathbb{1}_A$.

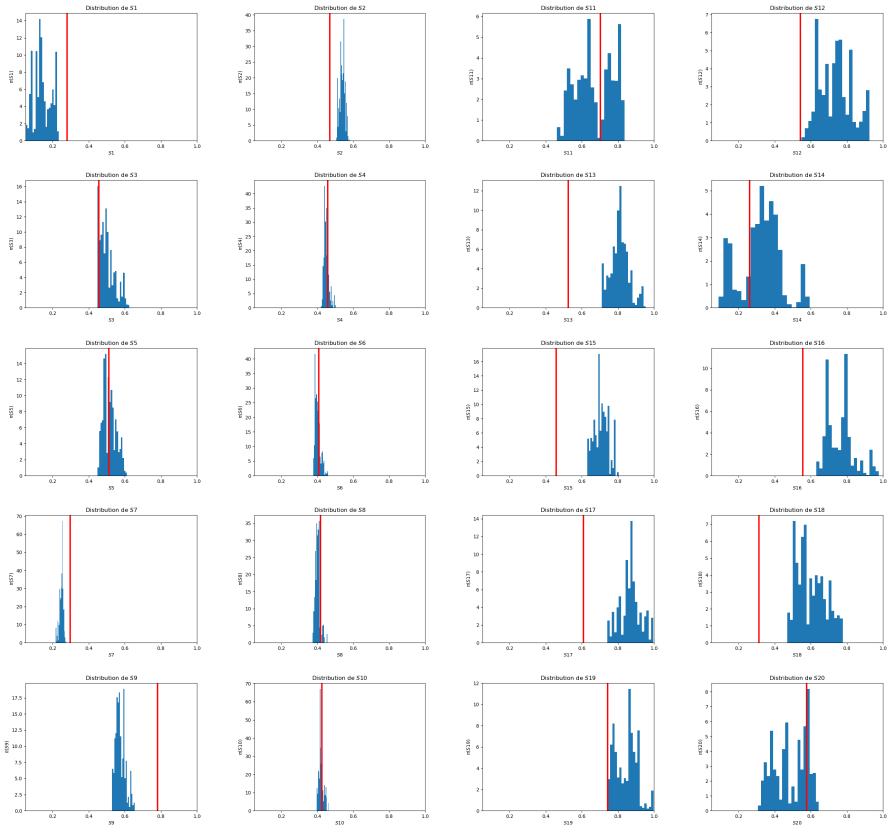


Figure 5.23 – En bleu, distributions empiriques, pour le modèle de Schneider avec compétition et à positions connues de tous les individus, des paramètres individuels S_i de la chaîne de Markov obtenue avec l'algorithme HMC initialisé sur une réalisation de la loi *a priori*. En rouge, la valeur utilisée pour simuler les données. À gauche, les individus observés, à droite, les individus non observés.

L'expression de l'énergie potentielle est donnée par :

$$U(S,\gamma,X) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{M}\left(s_{i,j}^{o} - f_{i,j}(S,\gamma,X,\sigma_x^2,\sigma_S)\right)^2}{2\sigma^2} - (a_S-1)\sum_{i=1}^{N}\log(S_i - s_m)$$
$$-(b_S-1)\sum_{i=1}^{N}\log(S_M - S_i) - (a_\gamma-1)\sum_{i=1}^{N}\log(\gamma_i) - (b_\gamma-1)\sum_{i=1}^{N}\log(\gamma_M-\gamma_i).$$

Le support de la loi conditionnelle est

$$A =]s_m, S_M[^N \times]0,\gamma_M[^N \times [0,L]^{N-n}.$$

Comme les hyperparamètres sont inférés par une étape de Metropolis-Hastings Within Gibbs (algorithme 5), on reprend les calculs des logarithmes des rapports $r_{\sigma_x^2}$, r_{σ_S}, r_{a_S}, r_{a_γ}, r_{b_S} et r_{b_γ} de la section 5.1.2.

Résultats

Le tableau 5.5 regroupe les valeurs des paramètres choisis pour inférer le modèle de Schneider à l'aide d'un algorithme de Gibbs combiné à un HMC. L'algorithme a été initialisé sur une réalisation des lois *a priori*.

Paramètres d'inférence							
Échelles (matrice de masse)				Paramètres du saute-moutons		Variance d'exploration	Nombre d'itérations
s_S	s_γ	s_x	s_y	L_{HMC}	ε	σ_i	K
$S_M - s_m$	γ_M	L	L	15	0.001	0.5	5 000

Table 5.5 – Valeurs des paramètres utilisés pour inférer le modèle de Schneider avec compétition à l'aide d'un Gibbs Within HMC.

La figure 5.24 montre l'évolution des états pris par les hyperparamètres. On observe que les états varient correctement : on peut donc en tracer la Distributions empiriques (figure 5.25).

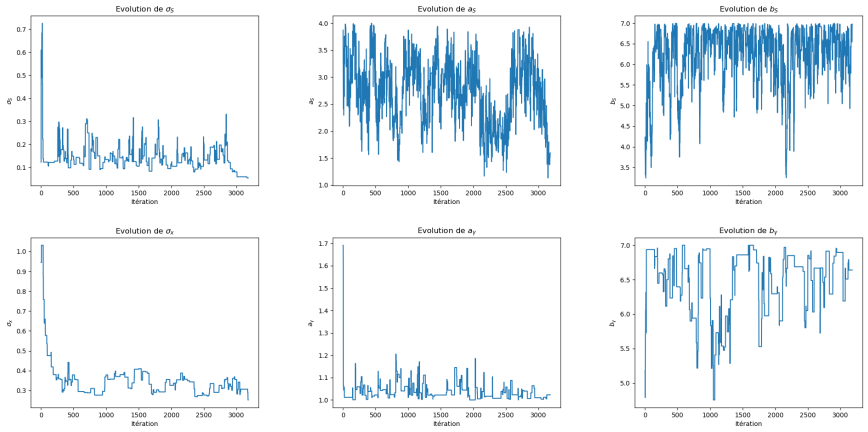


Figure 5.24 – Évolution de l'état des hyperparamètres de la chaîne de Markov, pour un modèle de Schneider avec compétition, au cours des itérations de l'algorithme Gibbs Within HMC initialisé sur une réalisation de la loi *a priori*.

D'autre part, les distributions empiriques des paramètres individuels sont représentées pour les individus 1 et 2 (observés) et 11 et 12 (non observés) sur la figure 5.26. L'algorithme a beaucoup de mal à retrouver les paramètres individuels. Plus spécifiquement, le taux de croissance γ_i, qui se trouve en facteur avec le taux de compétition dans l'expression (10), semble mal estimé. De plus, il semble proche de 0, ce qui signifie que la plante n'évolue pas. L'algorithme est peut être coincé dans une région avec une densité *a posteriori* forte qui n'est pas le mode.

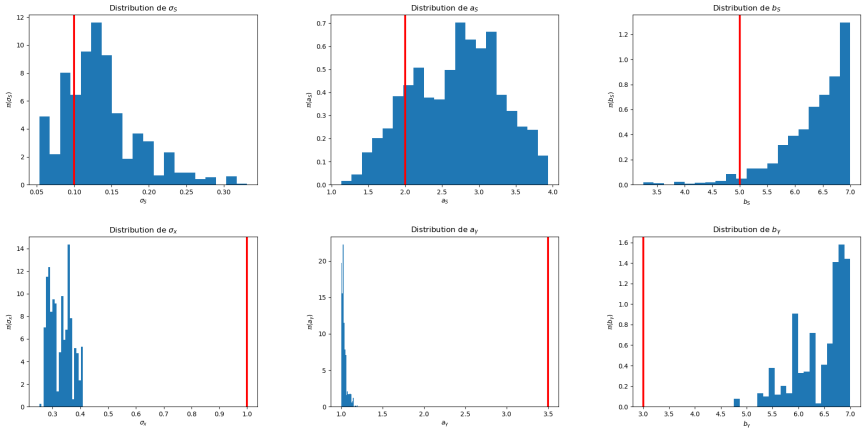


Figure 5.25 – En bleu, distributions empiriques, pour le modèle de Schneider avec compétition, des hyperparamètres de la chaîne de Markov obtenue avec l'algorithme Gibbs Within HMC initialisé sur une réalisation de la loi *a priori*. En rouge, la valeur utilisée pour simuler les données.

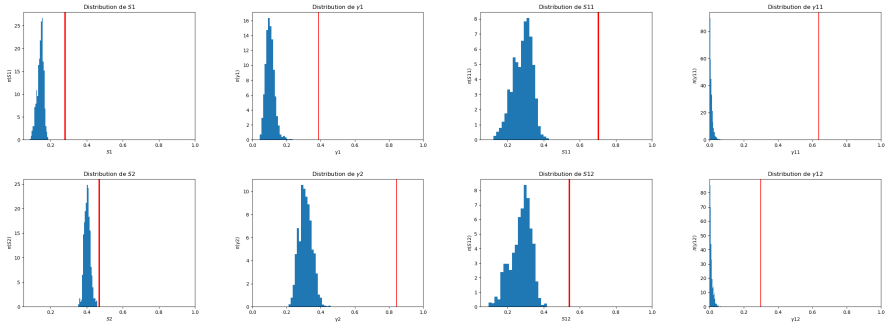


Figure 5.26 – En bleu, distributions empiriques, pour un modèle de Schneider avec compétition, des paramètres individuels (S_i, γ_i), pour les individus 1 et 2 (observés) et 11 et 12 (non observés), obtenues avec l'algorithme Gibbs Within HMC initialisé sur une réalisation de la loi *a priori*. En rouge, la valeur utilisée pour simuler les données. À gauche, les individus 1 et 2, à droite, les individus 11 et 12.

Cet algorithme a de grandes difficultés à converger. De plus, le temps de calcul pour chaque itération est trop important pour que l'on puisse effectuer suffisamment d'itérations. Les calculs des gradients en grande dimension semblent poser problème.

L'algorithme Gibbs Within HMC, comme pour le HMC classique, est difficile à ajuster. Il est possible que la longueur de trajectoire choisie ($\tau = \varepsilon \times L_{HMC}$) ne soit pas adaptée au modèle et aux données.

6 Application sur des données expérimentales.

6.1 Protocole expérimental.

Les données à notre disposition proviennent d'expérimentations sur du Colza, réalisées en 2012-2013 à la station expérimentale de l'INRA à Grignon (78), sur la variété Pollen [Baey et al., 2018]. Les plantes ont été semées dans des godets puis replantées dans des bacs en mi-septembre. Les plantes sont réparties dans dix bacs de taille identique. Nos données ne concernent que deux bacs, les bacs 3 et 4, qui ont la particularité d'être à forte densité. Ces bacs contiennent 42 plantes qui sont espacées de façon régulière (figure B.34 en annexe B).

Deux types de mesures ont été effectués :

— Des relevés du nombre de feuilles ont été faits de manière hebdomadaire, disponibles en annexe B.1 ;

— Des mesures de masses des feuilles ont été réalisées à la sortie de l'hiver, après 200 jours d'expérience. Les profils de biomasse relevée en fonction du rang de la feuilles sont représentés en annexe B.2.

Les prélèvements ont été effectués avant que la plante n'atteigne le stade de montaison. La biomasse est donc répartie uniquement dans les feuilles, ce qui correspond à notre modèle exposé dans la section 4.3.6. Dans la suite, les deux bacs sont considérés comme deux réalisations indépendantes de ce même modèle.

6.2 Estimation des paramètres individuels.

Les données sur le nombre de feuilles permettent l'estimation des paramètres d'organogenèse basée sur le modèle présenté dans la section 4.3.6. Les paramètres individuels ont été estimés sur 83 plantes (l'une des 84 plantes étant très certainement morte au cours de l'expérience) avec un modèle mixte.

Résultats

L'estimation des paramètres d'organogenèse est réalisée avec le logiciel Monolix [Lixoft SAS, 2018].

Les calculs de critères BIC ont permis de déterminer les effets que l'on peut considérer comme fixes : τ^R et τ^{init}. Les paramètres individuels sont donc les deux phyllochrones $\omega^{(1)}$ et $\omega^{(2)}$. Un exemple d'ajustement individuel réalisé avec Monolix est disponible sur la figure 6.27.

Les résultats obtenus pour les paramètres de population sont résumés dans le tableau 6.6.

τ^{init}	τ^R	$\omega^{(1)}_{pop}$	σ_1	$\omega^{(2)}_{pop}$	σ_2	σ	$Corr_{\left(\omega^{(1)}_{pop},\omega^{(2)}_{pop}\right)}$
240.26	703.10	35.97	0.10	14.72	0.21	0.63	0.62

Table 6.6 – Paramètres de population estimés avec le logiciel Monolix dans le modèle d'organogenèse GreenLab.

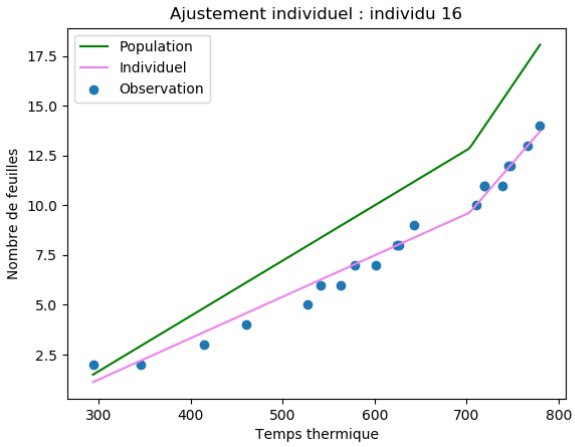


Figure 6.27 – Ajustement individuel avec Monolix pour l'individu 16 sur le modèle d'organogenèse GreenLab. En bleu, les observations, en rose la courbe d'ajustement individuel, en vert, la courbe de population.

6.3 Estimation bayésienne sur le modèle de population.

Les résultats présentés dans cette section ont été obtenus à l'aide d'un algorithme de MHWG (algorithme 4), initialisé sur l'estimateur des moindres carrés. Le modèle utilisé est celui de la section 4.3.6. Pour cette étude bayésienne, le paramètre b est supposé constant, fixé à la valeur de l'estimateur des moindres carrés.

Des calculs préalables sont nécessaires pour la mise en oeuvre de l'algorithme. Les calculs sont similaires à ceux effectués précédemment pour les modèles de Schneider (Section 5). Pour un paramètre x, la notation $q_{i,j}^{mod}(x)$ désigne la quantité de biomasse au temps final sur la feuille de rang j de l'individu i calculée avec le modèle où tous les paramètres sont fixés sauf x. Le candidat pour l'étape d'acceptation/rejet, x^*, est à distinguer du paramètre courant, x. N_{obs} désigne le nombre d'observations total.

Pour le paramètre μ, la densité conditionnelle est

$$\begin{aligned} p(\mu|\cdots) &\propto p(q^o|\mu, a, \sigma_x^2, \sigma_S)\ p(\mu|m_\mu, \sigma_\mu) \\ &\propto \frac{1}{\sqrt{2\pi\sigma^2}^{N_{obs}}} \exp\left[-\frac{\sum_{i,j}\left(q_{i,j}^o - q_{i,j}^{mod}(\mu)\right)^2}{2\sigma^2}\right] \\ &\quad \times \frac{1}{\mu\sqrt{2\pi\sigma_\mu^2}} \exp\left[-\frac{\log(\mu) - m_\mu}{2\sigma_\mu^2}\right] \mathbb{1}_{\mu>0} \end{aligned}$$

D'où, lorsque $\mu^* > 0$,

$$r_\mu(\mu, \mu^*) = \frac{\mu}{\mu^*} \exp\left[\frac{\sum_{i,j}\left(q^o_{i,j} - q^{mod}_{i,j}(\mu)\right)^2 - \left(q^o_{i,j} - q^{mod}_{i,j}(\mu^*)\right)^2}{2\sigma^2}\right] \times \exp\left[\frac{(\log(\mu) - m_\mu)^2 - (\log(\mu^*) - m_\mu)^2}{2\sigma_\mu^2}\right]$$

Ainsi,

$$\log(r_\mu(\mu, \mu^*)) = \log(\mu) - \log(\mu^*) + \frac{\sum_{i,j}\left(q^o_{i,j} - q^{mod}_{i,j}(\mu)\right)^2 - \left(q^o_{i,j} - q^{mod}_{i,j}(\mu^*)\right)^2}{2\sigma^2} + \frac{\log(\mu)^2 - \log(\mu^*)^2 + 2m_\mu(\log(\mu) - \log(\mu^*))}{2\sigma_\mu}$$

De même, pour le paramètre de la fonction de puits a, lorsque $a* > 1$,

$$\log(r_a(a, a^*)) = \log(a-1) - \log(a^*-1) + \frac{\sum_{i,j}\left(q^o_{i,j} - q^{mod}_{i,j}(a)\right)^2 - \left(q^o_{i,j} - q^{mod}_{i,j}(a^*)\right)^2}{2\sigma^2} + \frac{\log(a-1)^2 - \log(a^*-1)^2 + 2m_a(\log(a-1) - \log(a^*-1))}{2\sigma_a}$$

D'autre part, pour les paramètres de compétition, lorsque ${\sigma_x^2}^* > 0$,

$$\log\left(r_{\sigma_x^2}(\sigma_x^2, {\sigma_x^2}^*)\right) = (\alpha_x + 1)(\log(\sigma_x^2) - \log({\sigma_x^2}^*)) + \beta_x\left(\frac{1}{\sigma_x^2} - \frac{1}{{\sigma_x^2}^*}\right) + \frac{\sum_{i,j}\left(q^o_{i,j} - q^{mod}_{i,j}\left(\sigma_x^2\right)\right)^2 - \left(q^o_{i,j} - q^{mod}_{i,j}\left({\sigma_x^2}^*\right)\right)^2}{2\sigma^2}$$

et, lorsque $\sigma_S^* > 0$,

$$\log(r_{\sigma_S}(\sigma_S, \sigma_S^*)) = (\alpha_S + 1)(\log(\sigma_S) - \log(\sigma_S^*)) + \beta_S\left(\frac{1}{\sigma_S} - \frac{1}{\sigma_S^*}\right) + \frac{\sum_{i,j}\left(q^o_{i,j} - q^{mod}_{i,j}(\sigma_S)\right)^2 - \left(q^o_{i,j} - q^{mod}_{i,j}(\sigma_S^*)\right)^2}{2\sigma^2}$$

La loi *a priori* sur σ^2 est conjuguée. Avec des calculs similaires à (17),

$$p(\sigma^2|\cdots) = \mathcal{IG}\left(\frac{N_{obs}}{2} + e, \frac{1}{2}\sum_{i,j}\left(q^o_{i,j} - q^{mod}_{i,j}\right)^2 + f\right).$$

Résultats

Le tableau 6.7 regroupe les valeurs des paramètres des lois *a priori* et des paramètres d'inférence utilisés pour appliquer un algorithme de MHWG. Les variances d'exploration sont différentes selon les paramètres : pour les paramètres μ et a, on choisi une variance d'exploration $\sigma_{\mu,a} = 0.01$ tandis que pour les paramètres de compétition, on choisi $\sigma_{\sigma_S,\sigma_x} = 0.1$. Comme le *prior* sur σ^2 est conjugué, l'étape d'acceptation/rejet de Metropolis-Hastings n'est pas nécessaire, d'où l'absence de variance d'exploration pour ce paramètre.

Lois *a priori*									
m_μ	σ_μ	m_a	σ_a	e	f	α_S	β_S	α_x	β_x
1.0	1.0	0	1.0	2.0	1.0	2.0	1.0	2.0	1.0

Paramètres d'inférence		
Variances d'exploration		Nombre d'itérations
$\sigma_{\mu,a}$	$\sigma_{\sigma_S,\sigma_x}$	K
0.01	0.1	100 000

Table 6.7 – Valeurs des paramètres utilisés pour inférer le modèle adapté de GreenLab avec compétition à l'aide d'un Metropolis-Hastings Within Gibbs.

L'algorithme a été initialisé sur une approximation numérique de l'estimateur des moindres carrés, θ_{LS}. La figure 6.28 présente l'évolution des états de la chaîne de Markov au cours des itérations de l'algorithme. On observe que l'algorithme a bien convergé. On peut donc établir la distribution empirique de la distribution stationnaire, représentée sur la figure 6.29.

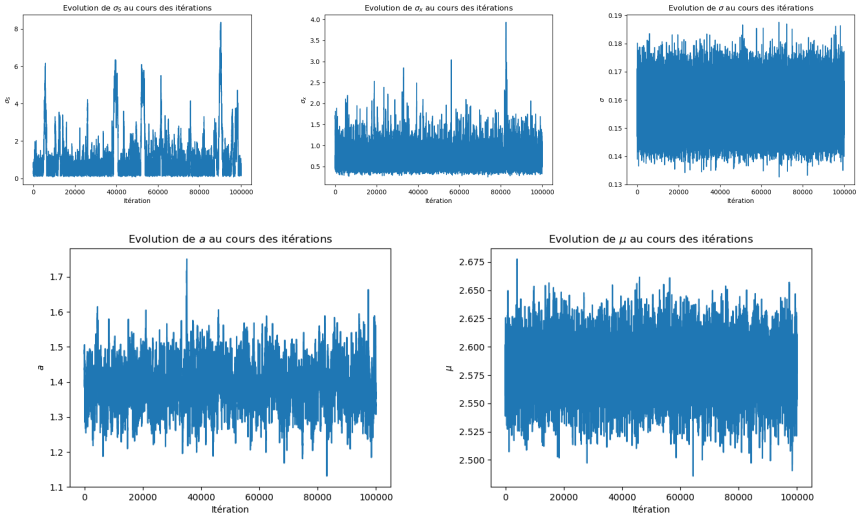


Figure 6.28 – Évolution de l'état des paramètres de la chaîne de Markov, pour le modèle adapté de GreenLab avec compétition, au cours des itérations de l'algorithme de Metropolis-Hastings Within Gibbs initialisé sur l'estimateur des moindres carrés θ_{LS}.

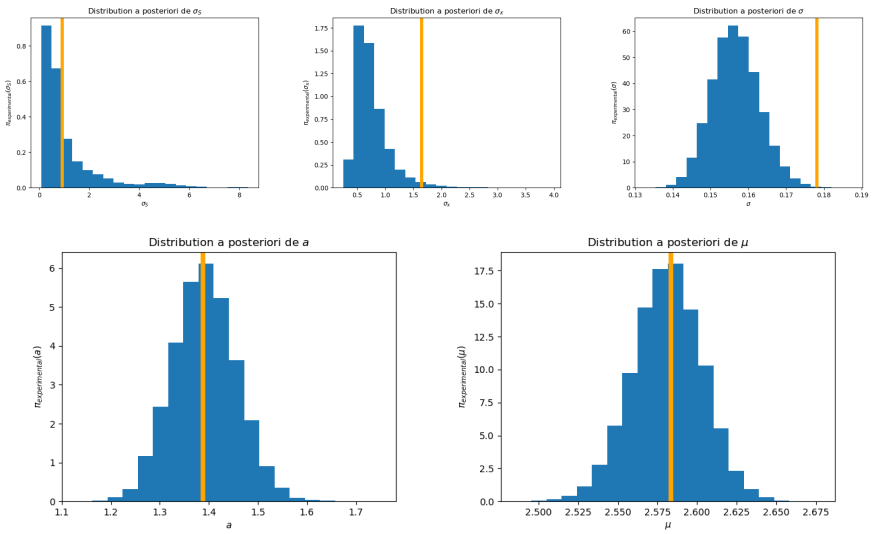


Figure 6.29 – En bleu, distributions empiriques, pour le modèle adapté de GreenLab avec compétition, des paramètres de la chaîne de Markov obtenue avec l'algorithme Metropolis-Hastings Within Gibbs initialisé sur l'estimateur des moindres carrés θ_{LS}. En orange, θ_{LS}.

Le résultat obtenu sur la figure 6.29 est satisfaisant. Cependant, une variabilité individuelle sur le paramètre a de la fonction de puits pourrait améliorer considérablement le modèle. Les temps de calculs importants nécessaires pour inférer ce modèle incluant de la variabilité individuelle sur l'un des paramètres empêchent d'observer la convergence de l'algorithme.

7 Conclusion et perspectives.

Lors de ce stage, des algorithmes de type Monte-Carlo par Chaîne de Markov ont été appliqués à des modèles de croissance de plantes en population hétérogène. Les modèles étudiés ayant la particularité de considérer que les individus ne poussent pas de manière indépendante mais en interaction, l'inférence en est d'autant plus complexe.

Nous avons débuté avec un modèle de Schneider (2006). Dans un premier temps, le modèle a été considéré pour des individus indépendants puis un modèle avec compétition entre les plantes a été introduit. Un premier algorithme basé sur un Metropolis-Hastings Within Gibbs a été élaboré, en Julia, pour chacun de ces deux modèles. L'échangeabilité des individus sur lesquels on a aucune information a constitué une première observation. L'algorithme de Monte-Carlo Hamiltonien a ensuite été étudié afin de se familiariser avec ses propriétés avant de l'appliquer à un modèle de Schneider avec compétition entre les individus. La difficulté de la grande dimension est apparue très rapidement (20 plantes) et s'est traduite par de longs temps de calculs, notamment dans les algorithmes de type Monte-Carlo Hamiltonien où les calculs de gradients sont fréquents.

Une adaptation d'un modèle GreenLab a ensuite permis l'application d'un algorithme de Metropolis-Hastings Within Gibbs à des données réelles. Les limites de la grande dimension apparaissent clairement avec ce modèle. Appliqué sur 83 plantes, le modèle considéré introduit de la variabilité individuelle uniquement par le biais de constantes. Lorsque l'on essaye d'introduire de la variabilité individuelle sur un paramètre du modèle, on ajoute un nombre de paramètres de l'ordre du nombre de plantes. Or, chaque itération de l'algorithme nécessite de simuler la croissance des 83 plantes simultanément. Les temps de calculs ne permettent plus d'obtenir de résultats satisfaisants en un temps raisonnable.

L'algorithme de Monte-Carlo Hamiltonien, pourtant adapté à la grande dimension, est compliqué à ajuster. Ses performances pourraient être améliorées à l'aide, par exemple, d'un algorithme de calcul du gradient, spécifique au modèle. De plus, des algorithmes tels que le No-U-Turn [Homan and Gelman, 2014] permettent d'éviter l'artefact de la périodicité des trajectoires.

Une idée pour contrer ce problème de grande dimension serait d'utiliser une approximation numérique de type champs moyen. En effet, [Della Noce et al., 2019] montre que lorsque le nombre d'individus tend vers l'infini, les caractéristiques des individus aux différents instants se comportent comme des variables aléatoires identiquement distribuées selon une loi limite, la loi de champs moyen. En trouvant une approximation empirique de cette loi, il serait possible de simuler les individus de manière indépendante tout en leur faisant subir les effets de la compétition induite par le reste de la population.

A Données simulées

A.1 Modèle de Schneider sans compétition

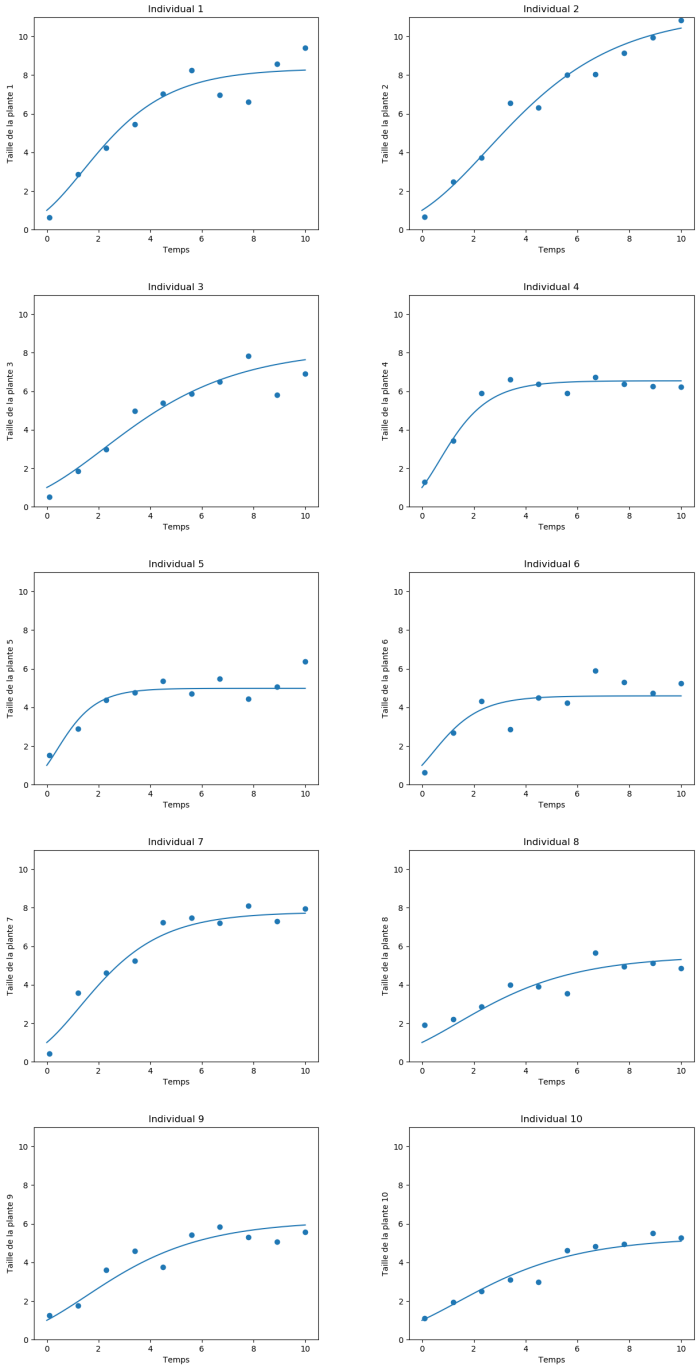


Figure A.30 – Données simulées pour le modèle de Schneider sans interaction. En trait plein, les courbes de croissances simulées, en points, les données bruitées.

A.2 Modèle de Schneider avec compétition

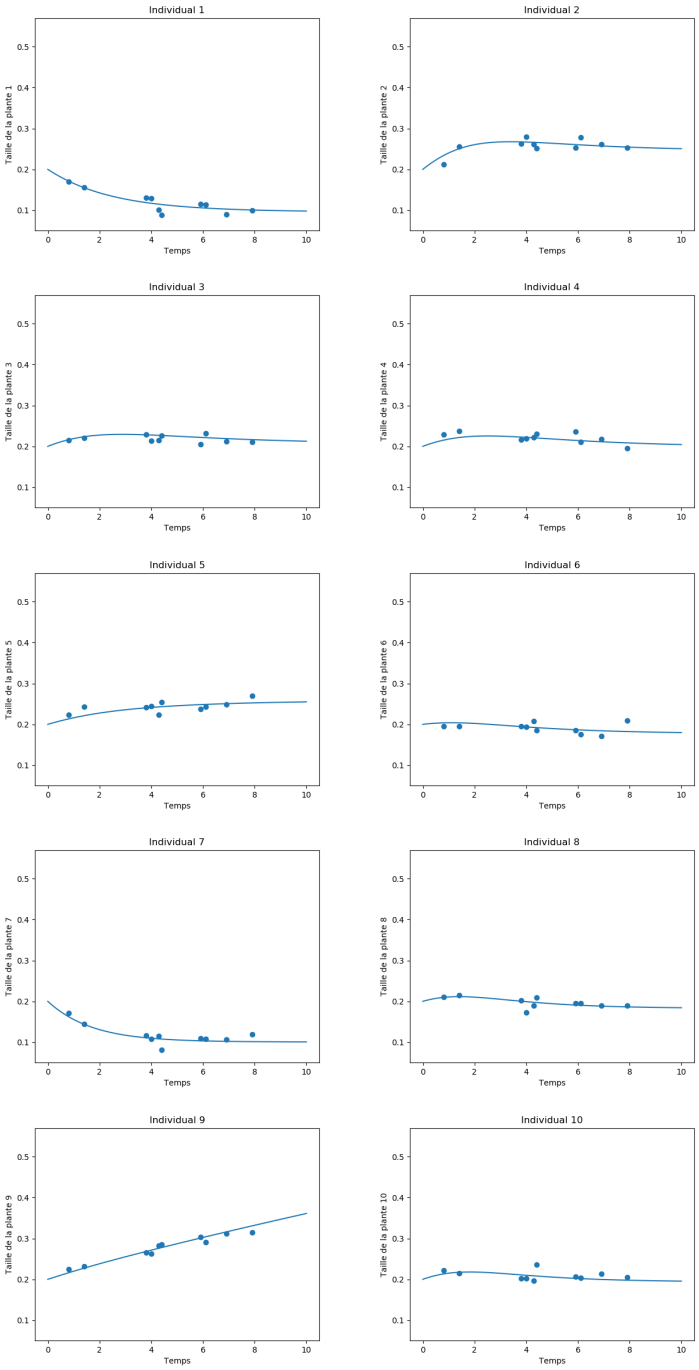


Figure A.31 – Données simulées et observées pour le modèle de Schneider avec compétition. En trait plein, les courbes de croissances simulées, en points, les données bruitées.

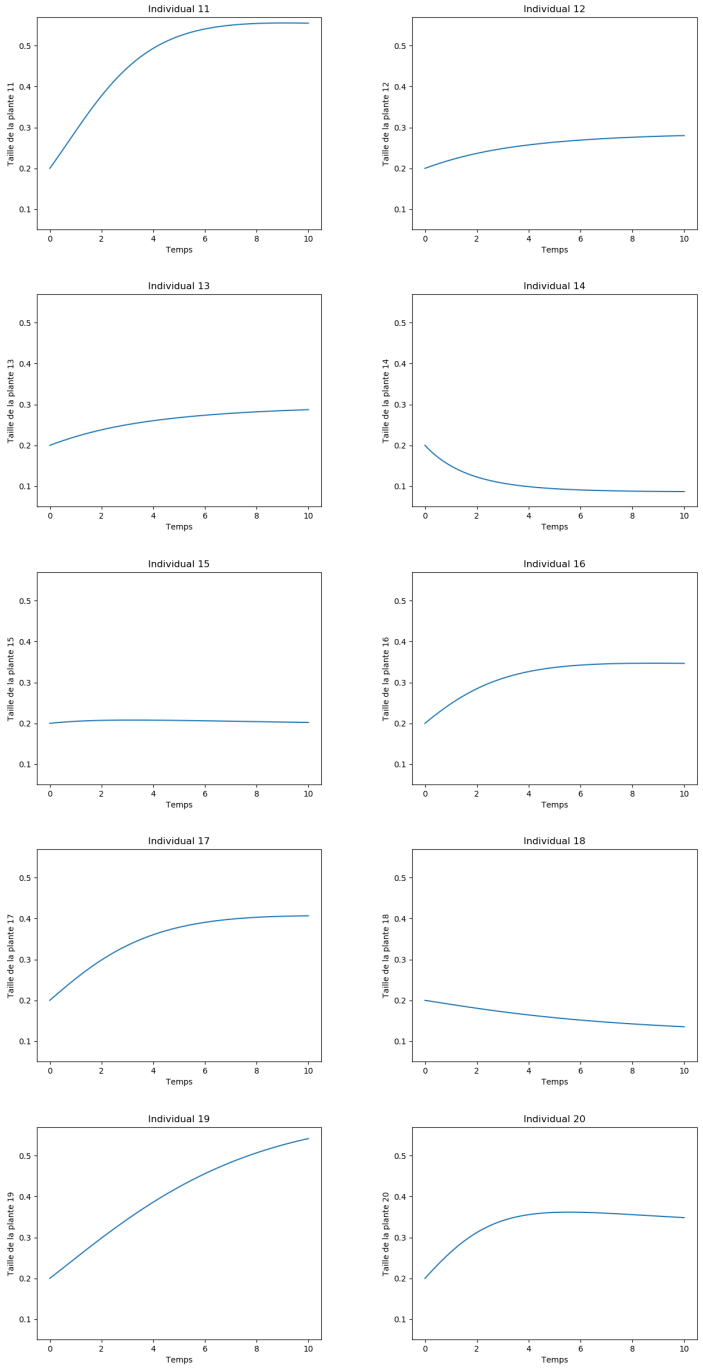


Figure A.32 – Données simulées non observées pour le modèle de Schneider avec compétition.

B Données réelles

B.1 Données sur le nombre de feuilles

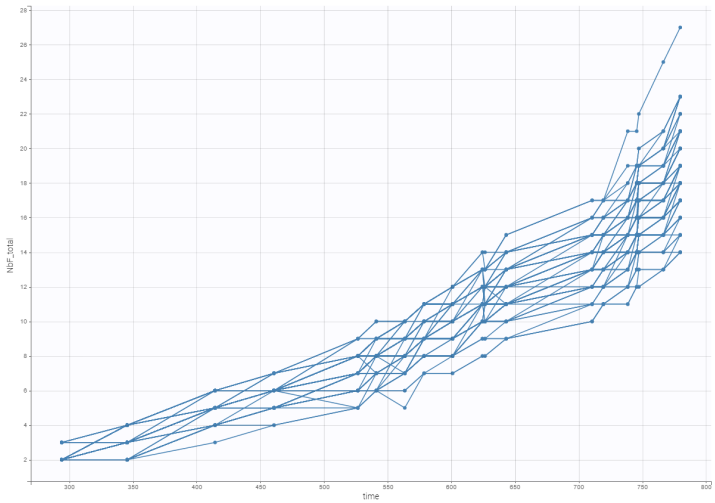


Figure B.33 – Nombre de feuilles en fonction du temps thermique pour 83 plantes de Colza.

B.2 Données sur les profils de biomasse

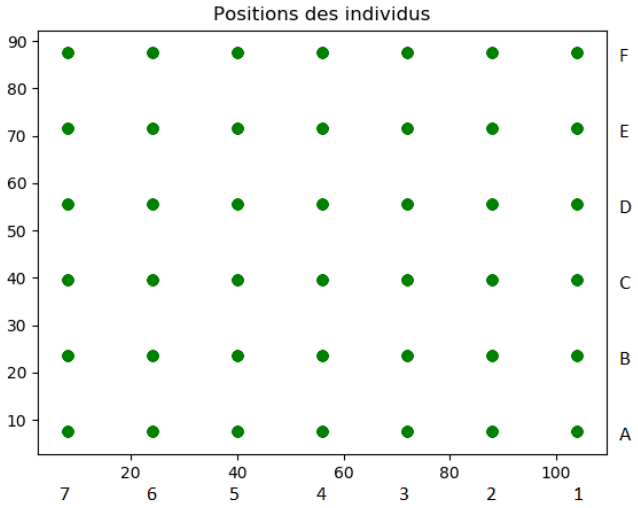


Figure B.34 – Positions des plantes de colza (x_i, y_i) (en cm) dans un bac.

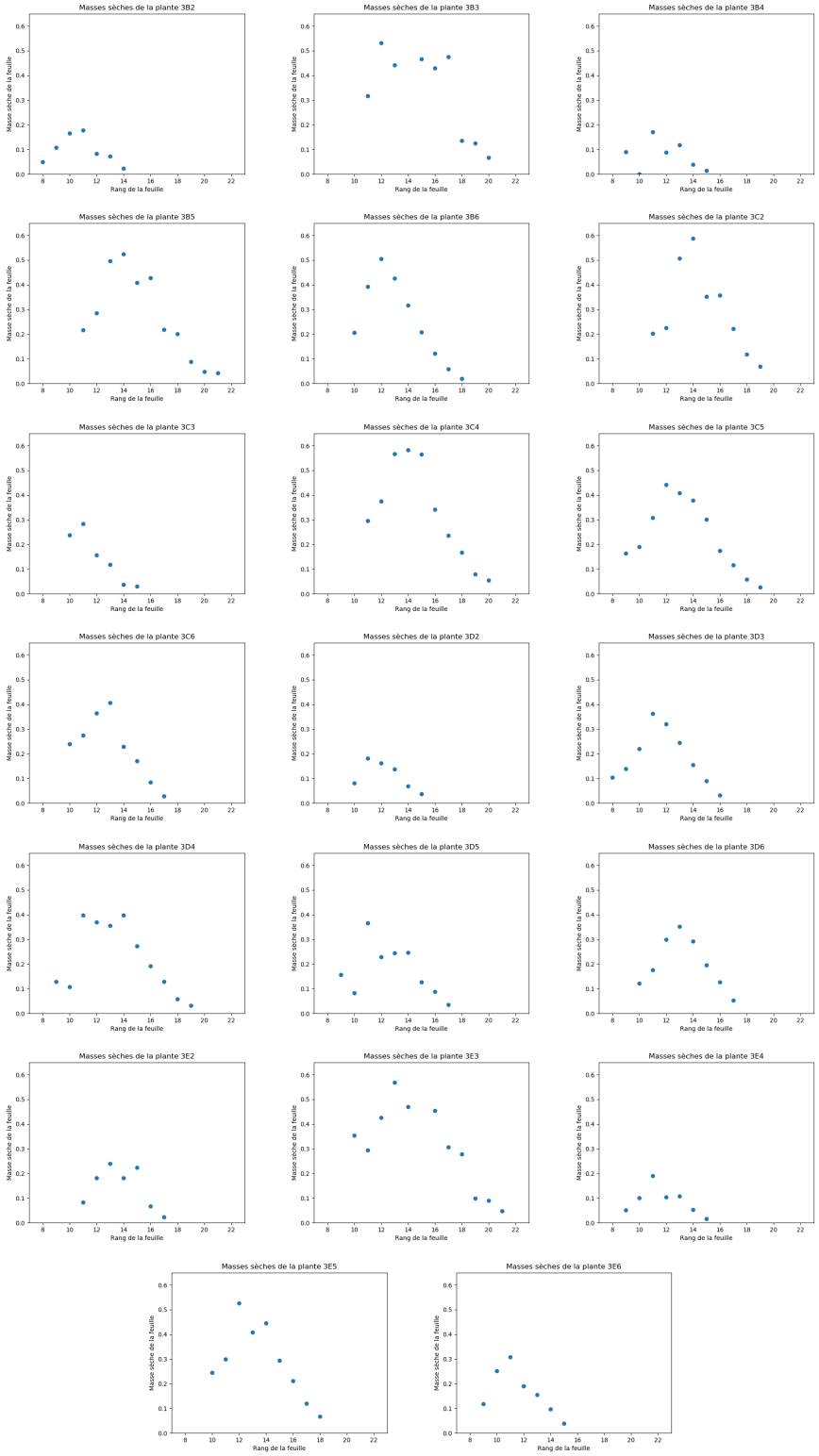


Figure B.35 – Données du bac 3 (20 plantes). Profils de biomasse : masse sèche (en g) en fonction du rang sur le tige.

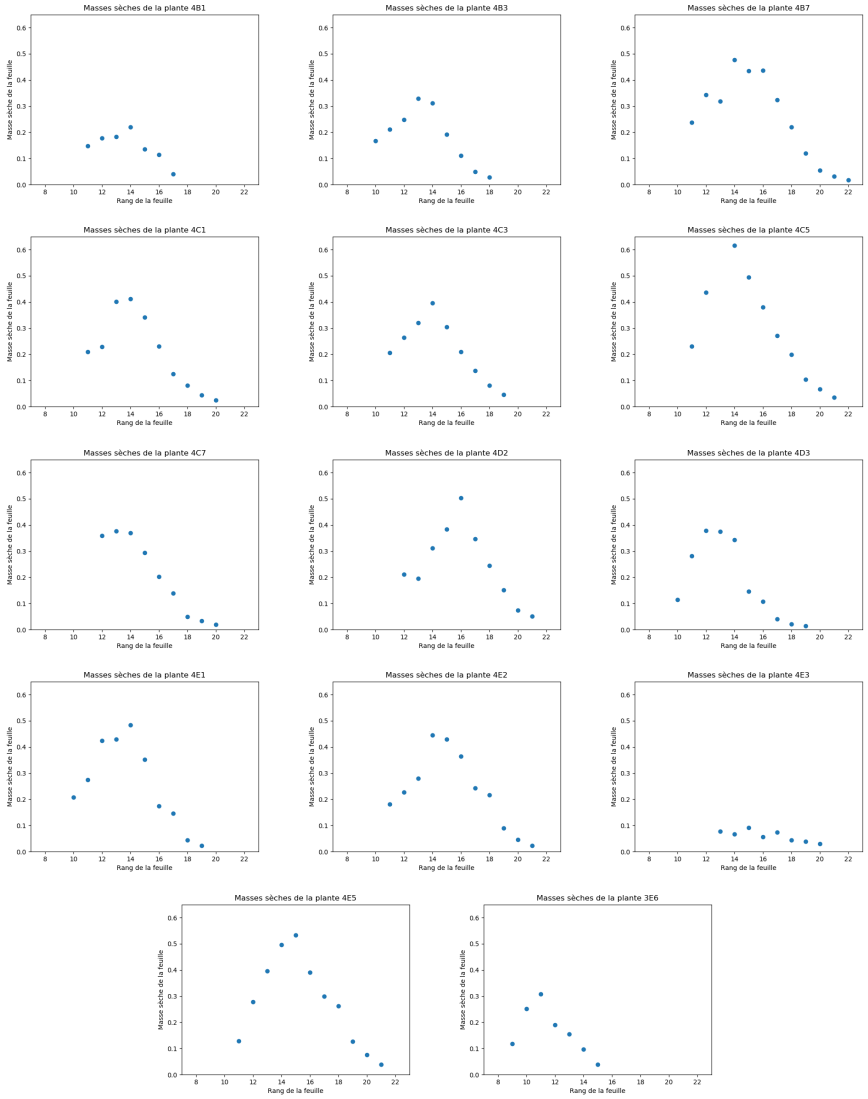


Figure B.36 – Données du bac 4 (14 plantes). Profils de biomasse : masse sèche (en g) en fonction du rang sur le tige.

C Un exemple de code Julia (version 1.1)

```
#####################################
#   CHARGEMENT DES PACKAGES         #
#####################################
using CSV
using Distributions
using DataFrames
using Optim
using JLD

#####################################
#      CHARGEMENT DES DONNEES       #
#####################################
#Ce fichier contient les données de masses sèches
data_ms=CSV.read("data_param.csv"; delim=";")
#Ce fichier contient les paramètres individuels (positions)
individual_parameters=CSV.read("individual_parameters.csv";delim=",")

#Ce fichier contient les variables environnementales
time_data=CSV.read("Env_PAR_Temp_2013.csv";delim=",")

#######################
#      STRUCTURES     #
#######################

# Définition d'un individu :
# Les caractéristiques qui ne vont pas évoluer au cours du temps
struct Fixes
    ID
    bac
    colonne
    ligne
    x::Float64
    y::Float64
    omega::Float64
    ttInit::Float64
    omega2::Float64
    Nbrangmin::Integer
    Nbrangmax::Integer
end
# Les caractéristiques qui vont évoluer au cours du temps :
# Le feuillage
mutable struct Leaf
    thermal_time::Float64
    rank::Float64
    sink::Float64
    mass::Float64
end
const Foliage=Array{Leaf,1}

# Les caractéristiques globales
mutable struct Variables
    thermal_time::Float64
    active_surface::Float64
    biomass::Float64
    biomass_increment::Float64
    demand::Float64
    foliage::Foliage
end
```

```
mutable struct Individual
    theta::Fixes
    X::Variables
end
const Population=Array{Individual,1}

# Les paramètres de population
mutable struct Hyperparameters
    mu::Float64
    a::Float64
    b::Float64
    sigma::Float64
    sigma_x::Float64
    sigma_s::Float64
end
const MC=Array{Hyperparameters,1}

# Les constantes
struct Constantes
    tau_e::Float64
    tau_l::Float64
    Tb::Float64
    kB::Float64
    Spr::Float64
    Temp::Array{Float64,1}
    PAR::Array{Float64,1}
    sm::Float64
    e::Float64
    q_seed::Float64
    tau_rupt::Float64
end

######################
#   MODELISATION     #
######################

#Competition between two individuals
function competition(individual1::Individual,individual2::Individual,
    theta::Hyperparameters,eta::Constantes)
    return tanh(individual2.X.biomass/eta.sm)/(2*(1 +((individual1.
        theta.x-individual2.theta.x)^2 + (individual1.theta.y-
        individual2.theta.y)^2)/theta.sigma_x^2))*(1 + tanh((
        individual2.X.biomass-individual1.X.biomass)/theta.sigma_s))
end

#Competition of a population on an individual
function competition(individual1::Individual,pop::Population,theta::
    Hyperparameters,eta::Constantes)
    return mean([competition(individual1,individual2,theta,eta) for
        individual2 in pop])
end

#Biomass production at time t of individual1 receiving the competition
    of a population of plants pop
function biomass_production!(t::Integer,individual1::Individual,pop::
    Population,theta::Hyperparameters,eta::Constantes)
    c=competition(individual1,pop,theta,eta)
    biomass_increment=eta.PAR[t] * theta.mu * eta.Spr *(1-exp(-eta.kB*
        individual1.X.active_surface/eta.Spr))*(1 .-c)
    individual1.X.biomass += biomass_increment
    individual1.X.biomass_increment=biomass_increment
end
```

```
#Update of the thermal time for leaves
function update_thermal_time!(thermal_increment::Float64,leaf::Leaf)
    leaf.thermal_time += thermal_increment
end

#Update of the thermal time for the individual
function update_thermal_time!(thermal_increment::Float64,individual::
    Individual)
    individual.X.thermal_time += thermal_increment
    for leaf in individual.X.foliage
        update_thermal_time!(thermal_increment,leaf)
    end
end

# Number of leaves at the end of the day
function compute_leaf_number(tt::Float64, theta::Fixes, tau_rupt::
    Float64)
    if tt < tau_rupt
        return Int(floor(1 + tt/theta.omega))
    else
        return Int(floor(1 + tau_rupt/theta.omega + (tt-tau_rupt)/theta
            .omega2))
    end
end

# Add some leaves to an individual
function update_architecture!(thermal_increment::Float64,thermal_time::
    Float64,individual::Individual,eta::Constantes)
    current_leaf_number=length(individual.X.foliage)
    futur_leaf_number=compute_leaf_number(individual.X.thermal_time,
        individual.theta,eta.tau_rupt)
    to_create_leaf_number=futur_leaf_number - current_leaf_number
    if to_create_leaf_number > 0
        for j=1:to_create_leaf_number
            push!(individual.X.foliage,Leaf(thermal_increment,
                current_leaf_number+j,0,0))
        end
    end
end

# Demand of biomass for a leaf
function compute_demand!(leaf::Leaf,a::Float64,b::Float64,tau::Float64)
    leaf.sink=(leaf.thermal_time/tau)^(a-1) * (1 - leaf.thermal_time/
        tau)^(b-1)
end

# Demand for an individual
function compute_demand!(individual::Individual,theta::Hyperparameters,
    eta::Constantes)
    demand=0
    for leaf in individual.X.foliage
        if leaf.thermal_time<eta.tau_e
            compute_demand!(leaf,theta.a,theta.b,eta.tau_e)
            demand+=leaf.sink
        else
            leaf.sink=0
        end
    end
    individual.X.demand=demand
end
```

```
# Biomass allocated to a leaf
function biomass_allocation!(biomass::Float64,demand::Float64,leaf::
    Leaf)
    leaf.mass += biomass * leaf.sink / demand
end

#Biomass allocation for an individual
function biomass_allocation!(individual::Individual)
    for leaf in individual.X.foliage
        biomass_allocation!(individual.X.biomass_increment,individual.X
            .demand,leaf)
    end
end

#Active surface of an individual
function compute_active_surface!(individual::Individual,eta::Constantes
    )
    active_mass=0
    for leaf in individual.X.foliage
        if leaf.thermal_time<eta.tau_l
            active_mass += leaf.mass
        end
    end
    individual.X.active_surface=active_mass/eta.e
end

#Individual update (independant of the population)
function individual_update!(individual::Individual, thermal_time::
    Float64, thermal_increment::Float64,theta::Hyperparameters,eta::
    Constantes)
    update_thermal_time!(thermal_increment,individual)
    update_architecture!(thermal_increment,thermal_time,individual,eta)
    compute_demand!(individual,theta,eta)
    compute_active_surface!(individual,eta)
    biomass_allocation!(individual)
end

#Update of the population state (interaction between individuals)
function update!(t::Integer,thermal_time::Float64,pop::Population,theta
    ::Hyperparameters,eta::Constantes)
    thermal_increment=max(0,eta.Temp[t]-eta.Tb)
    thermal_time += thermal_increment
    N=length(pop)
    pop_old=deepcopy(pop)
    Threads.@threads for i in 1:N
        individual=pop[i]
        if thermal_time>individual.theta.ttInit
            individual_update!(individual,thermal_time,
                thermal_increment,theta,eta)
        end
        biomass_production!(t,individual,pop_old[(1:N).!=i],theta,eta)
    end
    return thermal_time
end

########################################
#   IMPORTATION DES ESTIMATIONS      #
#     EFFECTUEES AVEC MONOLIX        #
########################################
#Ce fichier contient les données sur le nombre de feuilles
nb_feuilles=CSV.read("nbFeuillesCompil_aveccolonneennb.csv",delim=";")
```

```
#Ce fichier contient les paramètres individuels (MONOLIX)
individual_organogenesis_param=CSV.read("Organogenese_avec_rupture/
    IndividualParameters/estimatedIndividualParameters.csv", delim=";")

##############################
#   SIMULATION DU MODELE     #
##############################
# simulate a population according to the data (time and constantes)
function simulate(population::Population,theta::Hyperparameters,eta::
    Constantes)
    pop=deepcopy(population)
    thermal_time=0.0
    t=1
    while t<length(eta.Temp)
        thermal_time=update!(t,thermal_time,pop,theta,eta)
        t +=1
    end
    return pop
end

# Fonction qui prend en entrée l'ID d'une plante et en retourne sa
    position x,y
function IDtoPos(ligne,colonne)
    x=8+(ligne-1)*16
    y=7.5+(colonne-1)*16
    return(x,y)
end
function construct_individual(ID,omega,ttInit,omega2)
    Bac=nb_feuilles.Bac[nb_feuilles.IdPlante.==ID][1]
    Colonne=nb_feuilles.Column1[nb_feuilles.IdPlante.==ID][1]
    Ligne=nb_feuilles.ligne[nb_feuilles.IdPlante.==ID][1]
    x,y=IDtoPos(Ligne,Colonne)
    if ID in data_ms.plante
        n_min=data_ms.Nbrangmin[data_ms.plante .== ID][1]
        n_max=data_ms.Nbrangmax[data_ms.plante .== ID][1]
    else
        n_min=0
        n_max=21
    end
    th=Fixes(ID,Bac,Colonne,Ligne,x,y,omega,ttInit,omega2,n_min,n_max)
    seed=Leaf(0,1,0,eta.q_seed)
    foliage=Foliage()
    push!(foliage,seed)
    X=Variables(0,0,eta.q_seed,eta.q_seed,0,foliage)
    return Individual(th,X)
end
function construct_individual(i)
    ID=individual_organogenesis_param.plante[i]
    omega=individual_organogenesis_param.Phyllo_SAEM[i]
    ttInit=individual_organogenesis_param.ttInit_SAEM[i]
    omega2=individual_organogenesis_param.Phyllo2_SAEM[i]
    return construct_individual(ID,omega,ttInit,omega2)
end

function construct_initial_pop_bac3(eta::Constantes)
    pop=Population()
    for i=1:42
        push!(pop, construct_individual(i))
    end
    return pop
end
```

```
function construct_initial_pop_bac4(eta::Constantes)
    pop=Population()
    for i=43:83
        push!(pop, construct_individual(i))
    end
    return pop
end

tau_e=400 ; tau_l=600 ; Tb=4.5 ; kB=0.7 ; Spr=exp(-3.5) ; sm=1 ; e
    =119.81 ; q_seed=0.008 ; tau_rupt=703.1
eta=Constantes(tau_e,tau_l,Tb,kB,Spr,time_data.Temp,time_data.PAR,sm,e,
    q_seed)
pop3_init=construct_initial_pop_bac3(eta)
pop4_init=construct_initial_pop_bac4(eta)

########################################
#          INITIALISATION :            #
#   MOINDRES CARRES GENERALISES        #
########################################
# Création d'une Dataframe avec uniquement les données utiles
Observations=DataFrame()
Observations.ID=data_ms.plante[(data_ms.phyto_porteur .!== 0)]
Observations.ms=data_ms.ms[(data_ms.phyto_porteur .!== 0)]
Observations.phyto_porteur=data_ms.phyto_porteur[(data_ms.phyto_porteur
     .!== 0)]
# Calculer les moindres carrés
function compute_LS(pop::Population)
    LS=0
    for individual in pop
        ID=individual.theta.ID
        if ID in Observations.ID
            q_obs=Observations.ms[Observations.ID .== ID]
            q_mod=[]
            for leaf in individual.X.foliage
                if leaf.rank in Observations.phyto_porteur[Observations
                    .ID .== ID]
                    push!(q_mod, leaf.mass)
                end
            end
            while length(q_mod)<length(q_obs)
                push!(q_mod,0)
            end
            LS += sum((q_mod .- q_obs).^2)
        end
    end
    return LS
end
function LS(theta::Hyperparameters,eta::Constantes)
    pop3=simulate(pop3_init,theta,eta)
    pop4=simulate(pop4_init,theta,eta)
    return compute_LS(pop3) + compute_LS(pop4)
end
function g(theta)
    mu,a,b,sigma,sigma_s,sigma_x=theta
    theta=Hyperparameters(mu,a,b,sigma,sigma_s,sigma_x)
    return LS(theta,eta)
end
# Initialisation sur l'estimateur des moindres carrés
theta0=[exp(1.15),exp(1.2),exp(1.1),0.1,1,1]
res=optimize(g,theta0)
theta_LS=Optim.minimizer(res)
mu,a,b,sigma,sigma_s,sigma_x=theta_LS
```

```
theta_LS=Hyperparameters(mu,a,b,sigma,sigma_s,sigma_x)
@save "results/LS_real_data/Res_theta_LS.jld" theta_LS

##########################################
#         INFERENCE BAYESIENNE :         #
#    METROPOLIS HASTINGS WITHIN GIBBS    #
##########################################
function Gibbs(theta0::Hyperparameters, v::Array{Float64,1},
    n_iterations::Integer, eta::Constantes)
    theta=deepcopy(theta0)
    current_LS=LS(theta0,eta)
    MarkovChain=MC()
    for k=1:n_iterations
        # mu
        mu_star=rand(Normal(theta.mu, v[1]))
        if mu_star>0
            theta_star=deepcopy(theta)
            theta_star.mu=mu_star
            LS_star=LS(theta_star,eta)
            log_r_mu=log(theta.mu) - log(mu_star) + (current_LS -
                LS_star)/(2 * theta.sigma^2) + (log(theta.mu)^2 - log(
                mu_star)^2 + 2*m_mu*(log(theta.mu) - log(mu_star)))/(2
                * sigma_mu)
            if log(rand())<log_r_mu
                current_LS=LS_star
                theta.mu=mu_star
            end
        end

        # a
        a_star=rand(Normal(theta.a,v[2]))
        if a_star>1
            theta_star=deepcopy(theta)
            theta_star.a=a_star
            LS_star=LS(theta_star,eta)
            log_r_a=log(theta.a-1) - log(a_star-1) + (current_LS -
                LS_star)/(2 * theta.sigma^2) + (log(theta.a-1)^2 - log(
                a_star-1)^2 + 2*m_a*(log(theta.a-1) - log(a_star-1)))
                /(2 * sigma_a^2)
            if log(rand())<log_r_a
                current_LS=LS_star
                theta.a=a_star
            end
        end

        # b=Constante

        # sigma_x
        sigma_x_star=rand(Normal(theta.sigma_x,v[3]))
        if sigma_x_star>0
            theta_star=deepcopy(theta)
            theta_star.sigma_x=sigma_x_star
            LS_star=LS(theta_star,eta)
            log_r_sigma_x=(alpha_x+1)*(log(theta.sigma_x^2)-log(
                sigma_x_star^2)) + beta_x*(1/(theta.sigma_x^2) - 1/
                sigma_x_star^2) + (current_LS - LS_star)/(2 * theta.
                sigma^2)
            if log(rand())<log_r_sigma_x
                current_LS=LS_star
                theta.sigma_x=sigma_x_star
            end
        end
```

```
        # sigma_S
        sigma_s_star=rand(Normal(theta.sigma_s,v[4]))
        if sigma_s_star>0
            theta_star=deepcopy(theta)
            theta_star.sigma_s=sigma_s_star
            LS_star=LS(theta_star,eta)
            log_r_sigma_s=(alpha_s+1)*(log(theta.sigma_s)-log(
                sigma_s_star)) + beta_s*(1/(theta.sigma_s) - 1/
                sigma_s_star) + (current_LS - LS_star)/(2 * theta.sigma
                ^2)
            if log(rand())<log_r_sigma_s
                current_LS=LS_star
                theta.sigma_s=sigma_s_star
            end
        end

        # sigma2
        theta.sigma=sqrt(rand(InverseGamma(n_obs/2 + e, current_LS/2 +
            f)))
        push!(MarkovChain, deepcopy(theta))
    end
    return MarkovChain
end

#####################################
#   LANCEMENT DE L'ALGORITHME   #
#####################################
n_obs=309
# Choix des priors
m_mu=1; sigma_mu=1
m_a=0; sigma_a=1
e=2; f=1
alpha_s=2; beta_s=1
alpha_x=2; beta_x=1
# Paramètres d'inférence
v=[0.01,0.01,0.1,0.1]
K=100000
# Sauvegarde des résultats
new_dir="results/Gibbs_real_data"
mkdir(new_dir)
Res=Gibbs(theta_LS,v,K,eta)
@save  new_dir * "/Res_gibbs.jld" Res v K
```

Références

[Arnold et al., 2013] Arnold, V., Vogtmann, K., and Weinstein, A. (2013). Mathematical Methods of Classical Mechanics. Graduate Texts in Mathematics. Springer New York.

[Baey, 2014] Baey, C. (2014). Modelling inter-individual variability in plant growth models and model selection for prediction. Theses, Ecole Centrale Paris.

[Baey et al., 2018] Baey, C., Mathieu, A., Jullien, A., Trevezas, S., and Cournède, P.-H. (2018). Mixed-Effects Estimation in Dynamic Models of Plant Growth for the Assessment of Inter-individual Variability. Journal of Agricultural, Biological, and Environmental Statistics, 23(2) :208–232.

[Betancourt, 2017] Betancourt, M. (2017). A conceptual introduction to hamiltonian monte carlo. Preprint arXiv :1701.02434.

[Bolker et al., 2009] Bolker, B. M., Brooks, M. E., Clark, C. J., Geange, S. W., Poulsen, J. R., Stevens, M. H. H., and White, J.-S. S. (2009). Generalized linear mixed models : a practical guide for ecology and evolution. Trends in Ecology & Evolution, 24(3) :127 – 135.

[Borg et al., 2018] Borg, J., Kiær, L., Lecarpentier, C., Goldringer, I., Gaufreteau, A., Saint-Jean, S., Barot, S., and Enjalbert, J. (2018). Unfolding the potential of wheat cultivar mixtures : A meta-analysis perspective and identification of knowledge gaps. Field Crops Research, 221 :298 – 313.

[Bédard, 2008] Bédard, M. (2008). Optimal acceptance rates for Metropolis algorithms : Moving beyond 0.234. Stochastic Processes and their Applications, 118(12) :2198–2222.

[Comets et al., 2008] Comets, E., Brendel, K., and Mentré, F. (2008). Computing normalised prediction distribution errors to evaluate nonlinear mixed-effect models : The npde add-on package for R. Computer Methods and Programs in Biomedicine, 90(2) :154–66.

[Cournede et al., 2008] Cournede, P., Mathieu, A., Houllier, F., Barthélémy, D., and De Reffye, P. (2008). Computing competition for light in the GREENLAB model of plant growth : a contribution to the study of the effects of density on resource acquisition and architectural development. Annals of Botany, 101 :1207–1219.

[Della Noce et al., 2019] Della Noce, A., Mathieu, A., and Cournède, P.-H. (2019). Mean field approximation of a heterogeneous population of plants in competition. Working paper or preprint.

[Gelman et al., 1996] Gelman, A., Roberts, G. O., and Gilks, W. R. (1996). Efficient metropolis jumping rules. Bayesian Statistics, pages 599–607.

[Geman and Geman, 1984] Geman and Geman (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. IEEE Transactions On Pattern Analysis And Machine Intelligence.

[Gilks et al., 1995] Gilks, W. R., Best, N. G., and Tan, K. K. C. (1995). Adaptive rejection metropolis sampling within gibbs sampling. Applied Statistics, 44(4) :455.

[Hastings, 1970] Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications. Biometrika, 57(1) :97–109.

[Homan and Gelman, 2014] Homan, M. D. and Gelman, A. (2014). The no-u-turn sampler : Adaptively setting path lengths in hamiltonian monte carlo. J. Mach. Learn. Res., 15(1) :1593–1623.

[Kuhn, 2004] Kuhn, Estelle, L. M. (2004). Coupling a stochastic approximation version of em with an mcmc procedure. ESAIM : Probability and Statistics, 8 :115–131.

[Liu, 2008] Liu, J. (2008). Monte Carlo strategies in scientific computing. Springer Verlag.

[Lixoft SAS, 2018] Lixoft SAS (2018). Monolix version 2018R1. Antony, France : Lixoft SAS. http ://lixoft.com/products/monolix/.

[Metropolis et al., 1953] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. The Journal of Chemical Physics, 21(6) :1087–1092.

[Neal, 2010] Neal, R. M. (2010). MCMC using Hamiltonian dynamics. Handbook of Markov Chain Monte Carlo, 54 :113–162.

[Schneider et al., 2006] Schneider, M. K., Law, R., and Illian, J. B. (2006). Quantification of neighbourhood-dependent plant growth by bayesian hierarchical modelling. Journal of Ecology.

[Tang et al., 2018] Tang, Q., Tewolde, H., LIU, H., Ren, T., Jiang, P., Zhai, L., Lei, B., Lin, T., and Liu, E. (2018). Nitrogen uptake and transfer in broad bean and garlic strip intercropping systems. Journal of Integrative Agriculture, 17(1) :220 – 230.

[Tatarinova and Schumitzky, 2015] Tatarinova, T. and Schumitzky, A. (2015). Nonlinear Mixture Models. IMPERIAL COLLEGE PRESS.

[Viaud, 2018] Viaud, G. (2018). Statistical methods for the genotypic differentiation of plants using growth models. Theses, Université Paris-Saclay.