

Mémoire de stage

Présenté par

Janloup Taïno THALY

Master 1 Biodiversité, écologie, évolution

Parcours Biodiversité végétale et gestion des écosystèmes tropicaux (BIOGET)

Sujet :

Intégration du modèle neutre de la biodiversité à un package R de modélisation multi-agent

Tuteur : Dr Éric MARCON

Soutenu publiquement le 10 Juillet 2024

À Montpellier

Devant le jury suivant :

Dr Dominique LAMONICA

Rapporteure

Dr Éric MARCON

Examineur

MdC Méline AUBRY-KIENTZ

Examinatrice

RESUME

Comprendre les écosystèmes est crucial pour la santé écologique et les objectifs de gestion. Les modèles actuels peuvent être insuffisants, ce qui rend les simulations particulièrement utiles. Ce projet intègre la théorie neutre de la biodiversité de Hubbell—qui met l'accent sur l'équivalence des espèces, les processus stochastiques et l'équilibre entre communautés locales et méta-communautés—dans le package R SimComm en utilisant la programmation orientée objet en R6. Cette intégration permet de créer des communautés avec des abondances et des distributions spatiales variables, et de les simuler selon des règles écologiques spécifiques. Le package permet également des simulations en chaîne et génère des graphiques descriptifs à différents niveaux, tout en conservant la modularité des paramètres écologiques. Bien que SimComm soit moins performant que NetLogo en termes de calcul, sa flexibilité en fait un outil puissant pour la recherche écologique. Cette intégration représente une avancée potentielle dans la modélisation théorique en écologie, reliant approches théoriques et computationnelles pour informer la conservation de la biodiversité et la gestion des ressources naturelles.

ABSTRACT

Understanding ecosystems is crucial for ecological health and management objectives. Current models may fall short, making simulations a valuable avenue. This project integrates Hubbell's Neutral Theory of Biodiversity—emphasizing species equivalence, stochastic processes, and the balance between local and meta-communities—into the R package SimComm using object-oriented programming in R6. This integration allows for the creation of communities with variable abundance and spatial patterns, and their simulation according to ecological rules. The package also enables chained simulations and produces descriptive graphs across different levels, maintaining modularity in ecological parameters. While SimComm is modular and data-rich but less computationally efficient than NetLogo, its flexibility makes it a powerful tool for ecological research. This integration represents a possible advancement in theoretical ecology modeling, bridging theoretical and computational approaches to inform biodiversity conservation and resource management strategies.

Keywords : neutral theory, modelling, simulations, agent-based models, spatial aggregation, R.

TABLE DES MATIÈRES

RESUME	2
ABSTRACT	2
TABLE DES MATIÈRES	3
1. INTRODUCTION.....	5
2. MATERIEL ET METHODES.....	8
2.1. Architecture de SimComm	8
2.1.1. Création des communautés	8
2.1.2. Simulation du modèle neutre.....	9
2.1.3. Multi-simulations.....	12
2.2. NetLogo	12
2.3. Comparaisons entre SimComm et NetLogo	13
2.3.1. Comparaison quantitative	13
2.3.2. Comparaison qualitative.....	14
3. RESULTATS	15
3.1. Applications du package SimComm	15
3.2. Comparaisons avec NetLogo	17
3.2.1. Comparaisons quantitatives	17
3.2.2. Comparaisons qualitatives.....	19
4. DISCUSSION.....	21
5. CONCLUSION	24
6. REFERENCES.....	27
7. ANNEXES	31

1. INTRODUCTION

Les écosystèmes, unités fondamentales de la biosphère, constituent un réseau complexe d'interactions entre les organismes et leurs milieux, respectivement appelés biocénose et biotope (Odum, 1969). Ils varient en taille, allant des petites mares aux vastes forêts tropicales (Trudgill, 2007), et en complexité biotique ou abiotique, ce qui rend chaque écosystème unique tant dans l'espace que dans le temps (Deevey, 1969).

Comprendre les interactions en jeu permet d'évaluer les paramètres de santé écologique, tels que leur capacité à fournir des services écosystémiques vitaux, y compris la pollinisation, la régulation du climat et le cycle des nutriments (Costanza *et al.*, 1997), mais aussi leur résilience face à diverses perturbations comme les changements climatiques, les invasions biologiques ou les activités anthropiques (Folke *et al.*, 2004).

Cette mesure de la santé écologique a une importance cruciale dans le domaine de la conservation de la biodiversité (Groom *et al.*, 2006), ainsi que dans la gestion des ressources naturelles, particulièrement dans le contexte de stress climatique croissant et des besoins agro-alimentaires accrus (Tilman *et al.*, 2001).

Face à la diversité des écosystèmes, l'établissement de modèles compréhensifs pour chacun d'entre eux est systématique. Par la méthode scientifique, les premiers modèles à voir le jour sont des modèles observationnels, basés sur la mesure directe de la biodiversité, leur échantillonnage et le calcul d'indices tels que la biodiversité alpha, beta et les indices de diversité de Shannon ou Simpson (Whittaker, 1972). Ces modèles sont cependant limités par l'effort d'échantillonnage, les biais potentiels dans le temps, l'espace ou expérimentaux, sans oublier les facteurs confondants (Gotelli et Colwell, 2001).

À partir de là, des modèles analytiques et théoriques, fondés sur des bases de données observationnelles établies, ont été développés pour décrire et prédire la dynamique des communautés écologiques (Hanski et Hanski, 1999). Un des modèles les plus communément admis est celui de la théorie de la biogéographie insulaire de MacArthur et Wilson (1967), qui propose que le nombre d'espèces sur une île est déterminé par un équilibre dynamique entre l'immigration et l'extinction.

De cette théorie, Hubbell a proposé en 1979, puis en 2001, un modèle neutre de la biodiversité et de la biogéographie. Ce modèle propose une extension de la théorie de MacArthur en introduisant la notion de neutralité écologique (Hubbell, 2001). Contrairement à la théorie des niches, qui stipule que chaque espèce occupe une niche unique (Hutchinson, 1957), la neutralité écologique part de l'hypothèse que les espèces d'une même communauté sont écologiquement équivalentes. Elles ont donc les mêmes probabilités de naissance, de mort et la même fitness.

La théorie d'Hubbell propose une compréhension théorique des communautés dans un cadre prédéfini pour assurer cette neutralité : toutes les espèces de la communauté étudiée appartiennent au même niveau trophique, la communauté étudiée intègre un jeu à somme nulle, les espèces étant écologiquement équivalentes, les individus le sont aussi et partagent donc les mêmes probabilités de mortalité, de mutation ainsi que de migration depuis une communauté externe (Hubbell, 2001). Ce modèle est particulièrement utile pour comprendre la distribution de l'abondance des espèces dans les écosystèmes complexes.

Les conséquences clés du modèle neutre, où toutes les espèces au sein d'une communauté ont les mêmes chances de survie et de reproduction, dans un jeu à somme nulle, où les ressources sont limitées et où la population totale est constante, se traduit en une direction en une durée indéfinie vers la mono-dominance spécifique sans introduction de nouvelles espèces. (Hubbell, 2001). Le modèle intègre donc des processus stochastiques de naissance, de mort et de migration, soulignant l'importance des événements aléatoires dans la structuration des communautés écologiques (Alonso *et al.*, 2006).

Cependant, la stochasticité induite par ce modèle n'est pas fondamentalement intégrée dans les formules mathématiques, et l'inclusion des notions de variance et de marge d'erreur induit une interprétation approximative des dynamiques des communautés (Alonso *et al.*, 2006). En outre, les modèles bidimensionnels d'abondance et de temps ne prennent pas en compte les notions de dispersion ni de répartition spatiale inhérente aux biomes des écosystèmes (Chave, 2004).

De ce fait, l'utilisation de modèles computationnels peut répondre aux questions liées à la stochasticité (Grimm *et al.*, 2006), mais aussi offrir une représentation graphique de la structure spatiale des communautés (Grimm *et al.*, 2006). Des outils tels que MATLAB, Python, R et NetLogo sont particulièrement adaptés pour ces besoins (Wilensky, 1999).

Les modèles multi-agents, en particulier, permettent de simuler les interactions individuelles entre les agents (organismes) et leur environnement. Cette approche offre une grande flexibilité et une

précision accrue dans la modélisation des processus écologiques complexes (Bonabeau, 2002). Par exemple, NetLogo est un environnement de programmation permettant de créer des modèles basés sur des agents pour explorer la dynamique des systèmes naturels et sociaux (Tisue et Wilensky, 1999). Ces modèles peuvent intégrer des comportements individuels, des interactions entre individus et des processus environnementaux, offrant ainsi une compréhension plus complète et détaillée des dynamiques écologiques (Railsback et Grimm, 2012).

En combinant le modèle neutre de la biodiversité avec des outils de modélisation multi-agent, il devient possible de simuler des écosystèmes de manière plus réaliste et d'explorer des scénarios variés. Cela permet d'évaluer l'impact des perturbations environnementales, des politiques de conservation et des changements dans l'utilisation des terres sur la biodiversité et la santé des écosystèmes (Grimm et al., 2006). L'intégration de ces modèles dans des frameworks computationnels avancés peut ainsi contribuer à des stratégies de gestion plus efficaces et durables (Tang & Bennett, 2010).

Dans le cadre de mon étude, j'ai donc l'opportunité de pouvoir apporter une réponse à ce qui semble manquer dans l'environnement des packages de R et introduire une application de programmation orienté objet dans un modèle d'écologie théorique.

Le but de mon travail, au sein de la structure d'AgroParisTech Montpellier et sous la direction de Dr. Éric Marcon, sera donc de reprendre un package préexistant et permettant une visualisation simple et maîtrisée des communautés écologiques dans le temps et dans l'espace, qui constitue le package SimComm (Marcon et Rousteau, 2024), afin d'intégrer les éléments définissant le modèle neutre, tel que les règles écologiques mais aussi la différenciation entre communauté locale et méta-communauté. J'ai en parallèle l'objectif de constituer un package utilisable pour un utilisateur informé, notamment par l'introduction d'une modularité poussée mais aussi de permettre un processus de simulation à la chaîne pour en tirer des tendances que l'on peut exploiter visuellement ou statistiquement.

2. MATERIEL ET METHODES

2.1. Architecture de SimComm

2.1.1. Création des communautés

Le package SimComm initialement utilisait les fonctions contenues dans le fichier *Patterns.R* afin de générer des communautés via le package *entropart*. La pratique posait 2 problèmes techniques. Il n'était pas possible de modifier ou afficher les valeurs de la communauté générée après la création de celle-ci. La communauté ne comportait pas les variables des paramètres écologiques inhéremment associées à celle-ci.

De ce fait, le choix se porte sur la création d'une nouvelle classe nommée « *community_param* » dans le fichier *Communities.R*. (Figure 1) La classe comporte nativement les attributs publics de taille de la communauté ainsi que de la richesse spécifique, respectivement nx/ny , et S . Elle comporte aussi les arguments relatifs à la création d'une communauté par la fonction *rCommunity* du package *entropart*.

Lors de l'instanciation de l'objet avec *community_param\$new()*, la méthode *set_values()*, est appelée afin de remplir les valeurs par défaut de tout les attributs de construction de communauté. Après cette instanciation, l'utilisateur peut utiliser *set_values()* pour intégrer des nouveaux paramètres.

La génération de la communauté est identifiée dans la fonction *draw_matrix()*, sans arguments, qui récupère les attributs posés par *set_values()* afin de générer puis stocker une communauté dans l'attribut *the_matrix*. Auquel cas l'utilisateur ne souhaite pas faire appel à une génération entropique ou souhaite disposer les individus sans hasard, les attributs *distribute* et *style*, acceptent respectivement les options « *entropart*, *uniform*, *random* » et « *random*, *uniform*, *checkerboard* ». L'option « *random* » de *distribute* fait un simple appel à *rnorm()* avec S en argument. L'option « *uniform* » construit la communauté sous condition que la taille sélectionnée de la communauté soit divisible par le nombre d'espèces. Pour *style*, il s'agit ici de positionnement des valeurs dans une matrix. « *uniform* », appelle la fonction *matrix()* en prenant compte de nx/ny et « *checkerboard* » ignore la distribution de l'abondance des espèces afin de forcer une alternance du positionnement de chaque individu de chaque espèce.

De là, deux nouvelles classes sont construites sur la base de « *community_param* ». Les différences entre celles-ci correspondent aux différences entre la communauté locale et la méta-communauté.

Ainsi, au-delà de différentes valeurs par défaut lors de l'instanciation sans argument de l'objet, un nouvel attribut associé aux paramètres écologiques sont ajoutés. Le choix de la distinction de ces objets est doublement justifié par la praticabilité d'intégration future de nouveaux modèles avec des différentes règles régissant les communautés.

De ce fait, la classe « *local_pc* » ajoute un taux de mortalité, *death_rate*, par défaut à 0,1 et « *meta_pc* » ajoute un taux d'immigration et un taux de spéciation, respectivement à 0.005 et 0.00001.

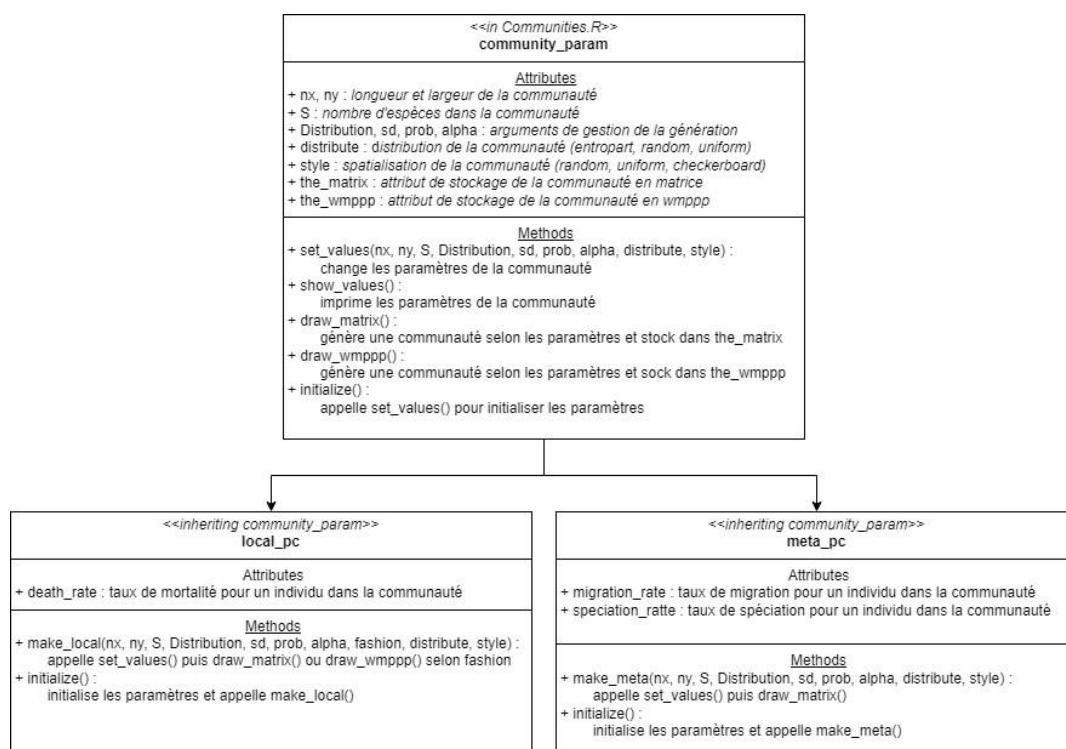


Figure 1 : Diagramme classes de création des communautés

2.1.2. Simulation du modèle neutre

Le cœur de la simulation du modèle neutre est présente dans la classe « *cm_hubbell* », dans le fichier *CM_Models.R* qui est hérite de « *community_matrixmodel* » qui hérite elle-même de « *community_model* ». Ces dernières sont écrites dans le fichier *Classes.R* et constitue la structure fondationnelle de la simulation. (Figure 2).

La classe « *community_model* » contient les fonctions *run()* et *evolve()* qui, respectivement permettent à l'objet d'appliquer la fonction *evolve()* et inscrit les règles selon lesquelles les *pattern* peuvent être modifiés entre les étapes. Dans cette classe, *evolve()* n'est que déclarée et les détails se doivent

d'être inscrites dans les classes inférieures. À l'instar de *evolve()*, la fonction *disturb()* est créé, déclarée et intégrée en parallèle dans cette classe.

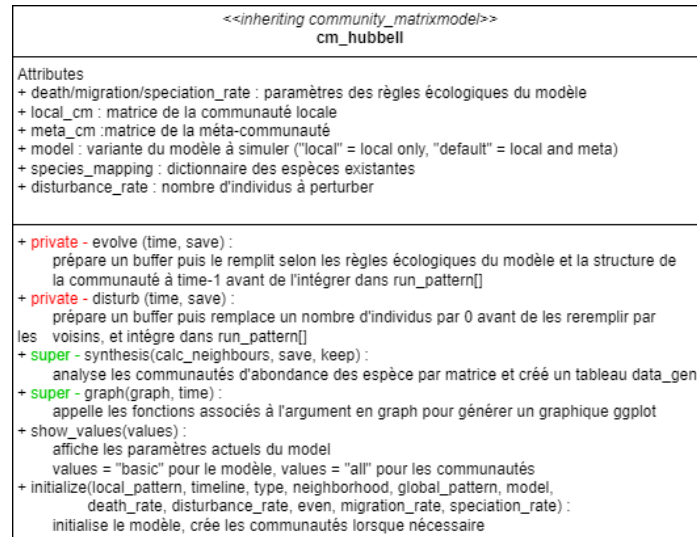


Figure 2 : Diagramme de la classe du modèle de Hubbell

La classe « *community_matrixmodel* » apporte les spécificités liées aux matrices à la simulation. La fonction *neighbours()* définit les règles de voisinages de von Neumann, Moore 1 et 2, ainsi que le voisinage globale. Cette fonction renvoie un vecteur des valeurs comprises dans le voisinage des coordonnées x et y de la cellule spécifiée.

Afin de pouvoir interpréter les communautés simulées, 2 fonctions sont créées dans cette classe, *synthesis()* et *graph()*. Comme son nom l'indique, *synthesis* crée un tableau qui, pour chaque étape de la simulation, enregistre le nom de l'espèce, l'abondance de chacune d'elles et leur classement au sein de l'étape temporelle. Une option *calc_neighbours* est intégrée afin de mesurer un index d'agrégation spatiale, calculé sur la moyenne de voisins de même espèces donnée pour un individu de chaque espèce, puis divisée par la taille du voisinage. Cet index est borné entre 0 et 1. Des options booléennes existent pour permettre l'allocation à une variable (*save*) ou l'archivage dans un fichier local (*keep*).

La fonction *graph()* cependant fait appel à un dictionnaire de graphiques pré-écrits dans fichier *Graphs.R*. Ce fichier comprend actuellement 13 fonctions générant chacune leur propre graphique avec la transformation éventuelle des données. Une liste comportant le dictionnaire des graphiques est disponible à la fin du fichier et comportent les variables qui peuvent être entrés comme argument dans la fonction *graph()*. Choisir « *all* » déclenche la génération de tous les graphiques définis. (Figure 3)

```
map_of_graphs <- list(
  "timed_abundance" = timed_abundance,
  "timed_relative_abundance" = timed_relative_abundance,
  "overall_abundance_distribution" = overall_abundance_distribution,
  "overall_abundance_rank" = overall_abundance_rank,
  "relative_abundance_rank" = relative_abundance_rank,
  "neighbors_rank" = neighbors_rank,
  "neighbors_time" = neighbors_time,
  "neighbors_abundance" = neighbors_abundance,
  "neighbors_abundance_time" = neighbors_abundance_time,
  "plot_neighbors_over_time" = plot_neighbors_over_time,
  "plot_abundance_and_neighbors" = plot_abundance_and_neighbors,
  "plot_neighbors_abundance" = plot_neighbors_abundance,
  "cumul_new_extinct_species" = cumul_new_extinct_species
)
```

Figure 3 Dictionnaire des graphiques dans le fichier Graphs.R de SimComm

Finalement, la classe essentielle du modèle neutre, « *cm_hubbell* » prend en argument les paramètres fondamentaux des classes parentes : un *local_pattern* de communauté, sa *timeline*, le type de distinction entre les individus (ie : *Species*) et le *neighborhood*. En raison de la particularité des communautés dans *local_cm* et *meta_cm*, la classe peut accepter toutes les valeurs des paramètres écologiques *death/speciation/migration_rate*, mais aussi le *global_pattern* de la méta communauté. Afin d'inclure les dynamiques de perturbations dans le modèle, *event* et *disturbance_rate* prennent en argument, respectivement, des vecteurs d'entiers et un entier pour spécifier le nombre d'individus à perturber aux étapes choisies.

L'ensemble des arguments sont initialisés dans l'objet déclaré de « *cm_hubbell* » puis exploité pour la création des communautés lorsque demandé et la gestion des méthodes privées *evolve()* et *disturb()*. La fonction *evolve()* parcourt les éléments de la matrice contenue dans *pattern* (qui provient de la variable *the_matrix* des communautés) et applique un dé, *runif(1,0,1)* afin de vérifier les conditions d'enclenchement des dynamiques écologiques. Ainsi, une condition sur *migration_rate* est vérifiée avant *mortality_rate*. Lorsque la migration est sélectionnée, l'individu est remplacé par une valeur aléatoire de la table des espèces de la méta-communauté. Dans le cas inverse, la valeur origine sera celle du vecteur retourné par la fonction *neighbors()*.

La spéciation existant exclusivement dans la méta-communauté dans le modèle de Hubbell, la stratégie ici est de lancer le dé autant de fois qu'il y a d'espèces et de créer une nouvelle entrée dans le tableau des espèces, *species_mapping*, lorsque la fonction *runif()* donne un nombre inférieur au taux de spéciation.

La fonction *disturb()*, cependant, est écrite de telle sorte à respecter la règle du jeu à somme nulle du modèle. Ainsi, lorsqu'appelé, cette fonction transformera un nombre défini d'individus dans la communauté en 0, avant d'ensuite sélectionner via *neighbours()* un voisin qui recolonisera l'espèce laisse vide, avant la fin de l'étape dans la simulation.

2.1.3. Multi-simulations

Une méthode de simulation de masse est intégrée dans le package SimComm sous la forme d'une classe nommée « *model_simulation* ».

Cette classe prend en argument, lors de son instantiation, les arguments liés au nombre de simulation à faire tourner, *count*, ainsi qu'à leur durée, *timeline*. Il est possible de sélectionner le voisinage ainsi que de donner des communautés (*local_pattern* et *global_pattern*) afin d'uniformiser la situation initiale de la communauté.

Après instantiation, une fonction *simulate()* intègre le cœur de la classe et enregistre dans une liste de taille *count*, le résultat de la fonction *synthesis()* appliquée à chaque simulation. Il convient d'utiliser cette synthèse afin de minoriser la charge en mémoire du programme. Stocker des millions de matrices pèse rapidement plusieurs méga-octets. L'attribut, *data_synth*, comporte l'ensemble de toutes les *synthesis()* avec le numéro associé à chaque simulation. Les données générées peuvent ensuite être transformées avec la fonction *new_data()* et stocké dans l'attribut *data_graph*. Enfin, la fonction *graph()* du modèle appelle une fonction *ggplot()* pour l'affichage graphique des données.

2.2. NetLogo

Northwestern University, l'organisation en charge de la maintenance de NetLogo par le biais de son créateur, Uri Wilensky, propose une page composée de modèles créés par la communauté d'utilisateurs. Accessible à cette page : <https://ccl.northwestern.edu/netlogo/models/community/>, le Dr Ruwan Punchi-Manage a mis à disposition un modèle exhaustif des modèles neutres appliqués à la biodiversité et la biogéographie. (Punchi-Manage, 2023).

Après avoir exporté ce modèle, une suppression des éléments non-pertinents au cadre du stage présent est faite, ne conservant que les 2 sliders des règles écologiques (immigration et spéciation) ainsi que les variables de création de la communauté (*J* et *w* pour le nombre d'individus et le nombre d'espèces respectivement). Les graphiques pertinents à l'étude de l'abondance des espèces sont les seuls conservés.

Afin de permettre un archivage des données, un cluster d'objets permettant l'affichage du temps écoulé, la limite du nombre d'étapes à simuler dans le modèle et l'emplacement du stockage de l'archive sont créés (Fig3).

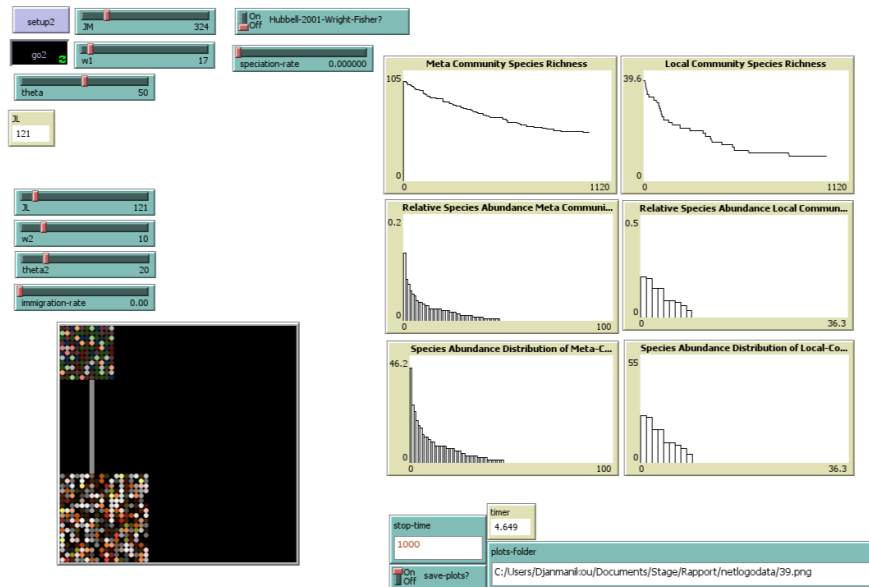


Figure 3 : Snapshot de NetLogo sur un modèle de la théorie neutre après 1000 steps

2.3. Comparaisons entre SimComm et NetLogo

2.3.1. Comparaison quantitative

La vitesse d'exécution du modèle est testée sur le package SimComm et sur NetLogo.

Afin de minoriser la charge sur le CPU de la machine de travail, un Thinkpad E485 de Janvier 2019, comportant un processeur AMD Ryzen 3 2200U avec Radeon Vega Mobil Gx de 2.50 GHz et 8.00go de mémoire RAM installée sur Window 10 x64, l'ensemble des paramètres écologiques sont minimisées voire éliminées afin d'obtenir des données sur la vitesse maximale possible avec le moins d'interférence.

Les règles communes à chaque simulation du modèle sont donc :

- Taux d'immigration = 0
- Taux de spéciation = 0
- Durée de la simulation (nombre d'étapes) = 1000

D'autre part, lors de la création de la communauté locale, 3 paramètres sont étudiés :

- 10 espèces dans une communauté de 100 individus
- 20 espèces dans une communauté de 400 individus
- 50 espèces dans une communauté de 2500 individus.

Et finalement, le modèle de simulation sur SimComm sera testé avec un voisinage global correspondant au fonctionnement de NetLogo ainsi qu'à la base théorique du modèle de Hubbell, ainsi qu'avec un voisinage correspondant à von Neumann, où les 4 cellules adjacentes à la cellule focale sont considérées pour les événements de recolonisation après mortalité.

Chacune des simulations sera répétée dans chaque paramètres de communauté 10 fois afin d'obtenir de minimiser la variance due à l'utilisation du CPU. La machine de travail n'est pas utilisée pendant le déroulement de chacune des simulations.

2.3.2. Comparaison qualitative

Au-delà des capacités techniques de vitesse, il convient d'offrir une appréciation comparative des caractéristiques qualitatives des 2 outils. Un utilisateur étant la cible d'un tel package, les besoins de celui-ci peuvent être multiple. Pour une appréciation spatiale de la dynamique des communautés, une bonne fluidité visuelle est recommandée. Dans le cas où l'utilisateur souhaite explorer les effets des différentes variables des communautés et des règles écologiques, il conviendra donc de juger de la modularité des modèles, c'est-à-dire la capacité à combiner diverses hypothèses de paramètres. Après la génération de la communauté et la simulation du modèle, il est souvent utile de pouvoir conserver les données produites pour des futures manipulations ou une conservation de la communauté de départ, le stockage est donc un élément à apprécier. Enfin, l'accessibilité du programme est primordiale afin de permettre au plus grand nombre de manipuler les simulations dans un cadre prédéfini, avec le minimum d'adaptation ou de courbe d'apprentissage. Une synthèse est faite sous forme de tableau afin d'évaluer les qualités de chacun des logiciels dans leur valeur relative l'un de l'autre.

3. RESULTATS

Le package peut être installé sur RStudio par le biais des lignes de commandes suivantes :

```
> library("remotes")
> remotes::install_github("TainoT/SimComm")
```

3.1. Applications du package SimComm

Le package permet la création diversifiée et personnalisée de la communauté initiale du modèle. La fonction `local_pc$new()` génère une communauté locale avec les valeurs par défaut, où *distribute* = « *entropart* » et *style* = « *random* », pour une distribution a priori réaliste de la distribution des communautés. L'appel de cette fonction avec l'argument « *uniform* » pour les deux options *distribute* et *style* crée une communauté ordonnée et triée. Enfin, une option *style* = « *checkerboard* » existe afin de démontrer les possibilités de la structure initiale de la communauté. (Figure 4)

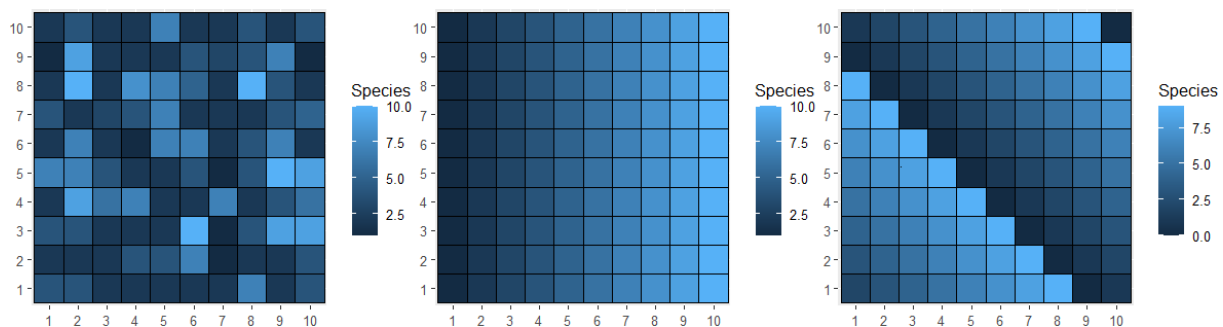


Figure 4 : 3 types de distribution des communautés, « entropart/random », « uniform/uniform », « checkerboard »

En utilisant la fonction `model_name <- cm_hubbell$new()`, un modèle comportant une communauté aléatoirement générée selon les règles par défaut, ainsi qu'une méta-communauté, sont construites. Il est possible d'intégrer une communauté créée précédemment en les assignant aux options « *local_pattern* » et « *global_pattern* ». Le choix du modèle de simulation, c'est-à-dire la décision d'inclure ou non la dynamique entre une méta-communauté et une communauté locale (ie : migration) peut être précisé par l'option « *model* ». Une option pour déclencher des perturbations à des étapes sélectionnées est accessible par le biais des options « *disturbance rate* », qui prend en argument le nombre d'individus à perturber, et « *event* » qui précise les étapes où enclencher la perturbation. Il est fondamental de choisir une *timeline* de la forme `0:1234` afin de permettre le bon fonctionnement de la fonction `model_name$run()`. Pareillement, il convient de mettre en argument à cette dernière fonction « *save = TRUE* » afin de permettre, avec

`model_name$synthesis()`, la synthèse des données de structuration de la communauté dans un tableau qui sera exploitée dans la fonction `model_name$graph()`.

De ce fait, la Figure 5 présente la distribution des espèces dans le temps au cours de la simulation du modèle selon les configurations suivantes : aucun argument, `model = « local »`, `disturbance_rate = (self$nx*2)/2`, (ie : la moitié des individus de la communauté sont perturbés)

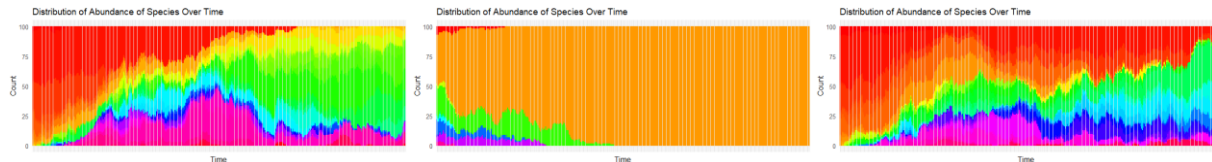


Figure 5 : 3 types de distribution des espèces dans le temps selon le modèle et la perturbation. Le 1^{er} modèle inclut l'immigration, le 2nd est régit uniquement par la dérive, le 3^{ème} inclut l'immigration et des perturbations régulières

De multiples options existent pour la synthèse graphique des communautés, pour inclure, entre autres, les distributions relatives de l'abondance des espèces, la dynamique d'extinction des espèces, la corrélation entre l'agrégation spatiale des espèces et leur abondance. (Figure 6)

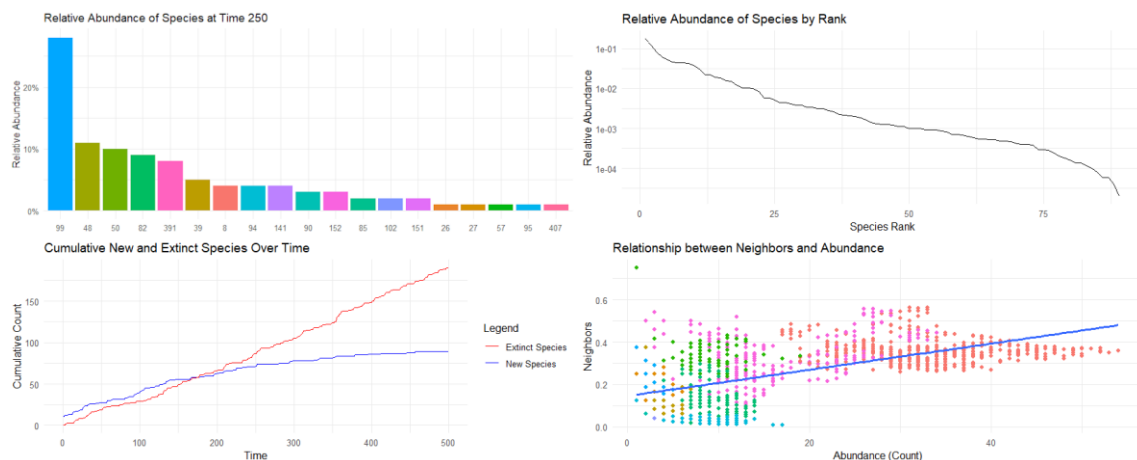


Figure 6 4 exemples de graphiques existants dans le package SimComm. De haut en bas et de gauche vers la droite :
Abondance relative des espèces à temps 250, Abondance relative des espèces sur une échelle log,
Nombre cumulatif de nouvelles espèces et d'extinctions, Relation entre le taux d'agrégation spatiale et l'abondance

Finalement, par la fonction `simulation_name <- model_simulation$new()` qui peut intégrer le choix du modèle ou les communautés initiales en argument, il est possible de générer avec `simulation_name$simulate()`, un nombre choisi de simulations individuelles du modèle sur une timeline donnée. Après, quoi l'utilisateur peut lancer les fonctions `simulation_name$new_data()` et `simulation_name$graph()` pour obtenir, dans la version actuelle du package, des courbes associées à l'abondance de l'espèce la plus abondante ainsi que la moyenne et la régression linéaire proposée.

La figure 7 présente le résultat de cette exécution avec en option *model* = « local » ou « default ».

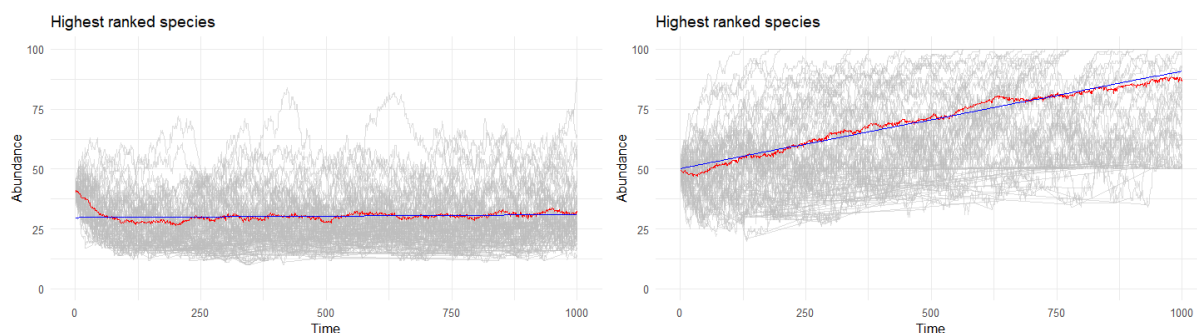


Figure 7 : 2 types de résultats graphique de l'abondance de l'espèce la mieux classée par simulation et la moyenne résultante, selon le type de modèle choisi (sans ou avec dérive)

3.2. Comparaisons avec NetLogo

3.2.1. Comparaisons quantitatives

La figure 3 représente un barplot avec en abscisse la variable du nombre d'espèces et de la largeur de la communauté. Il est donc à noter que le nombre d'individus dans la communauté est le carré de la valeur associée en abscisse. En ordonnée, sur une échelle log, est le temps de travail computationnel d'une simulation en fonction des différents paramètres de voisinage ou de l'outil utilisé.

De ce fait, il y a une grande discrédance observée dans la vitesse d'exécution d'une simulation du modèle neutre de Hubbell. Pour les mêmes paramètres écologiques minimaux, c'est-à-dire à *immigration* = 0 et *spéciation* = 0, avec $S = 10$ espèces dans une communauté de 100 individus, le modèle, tourné pendant 1000 étapes, prendra un temps relativement court dans chacune des situations. Sur NetLogo, ce temps avoisine les 3,11 secondes alors que sur R, indépendamment du voisinage, cette valeur s'élève entre 5,11 et 5,47 secondes. Bien qu'à échelle humaine, ces valeurs peuvent être marginales, il reste question d'une augmentation de la durée d'entre 66,24 et 75,80% entre NetLogo et SimComm.

Cette tendance est amplifiée lorsque la simulation introduit $S = 20$ espèces dans une communauté de 400 individus. L'augmentation du travail computationnel sur NetLogo est de 2 secondes, soit 64,63 % par rapport à $S = 10$. Sur R, cependant, lorsque le voisinage est global, comme il l'est sur NetLogo, la durée totale du calcul avoisine 56,8 secondes par simulation de modèle, soit 224,22% de plus par rapport à NetLogo dans les mêmes paramètres. Quand le choix du voisinage est au

minimum, c'est-à-dire le voisinage de von Neumann, cette durée n'augmente que de 240% à 17 seconds environ.

Enfin, lorsqu'une simulation plus poussée est choisie, avec 50 différentes espèces et 2500 individus, tous les modèles accusent le coup à leur échelle. Le modèle sur NetLogo prendra environ 19,3 secondes à s'exécuter sur 1000 étapes. Il s'agit ici d'une augmentation de 276,95% par rapport à la simulation paramétrée à $S = 20$. Sur SimComm, ce chiffre est largement plus imposant dans la perspective d'un voisinage global où l'utilisateur doit patienter 5120 secondes pour voir la complétion du modèle, soit un peu moins d'une heure et demie. L'augmentation par rapport à $S = 20$ est alors de près de 10 000%. Pour référence, lorsqu'un voisinage von Neumann est sélectionné, la simulation dure 117 secondes, ce qui est 2,28% de la durée en voisinage global. Ces résultats démontrent clairement que NetLogo offre une performance computationnelle nettement supérieur à R invariablement du paramètre de voisinage ou de taille de communauté initiale.

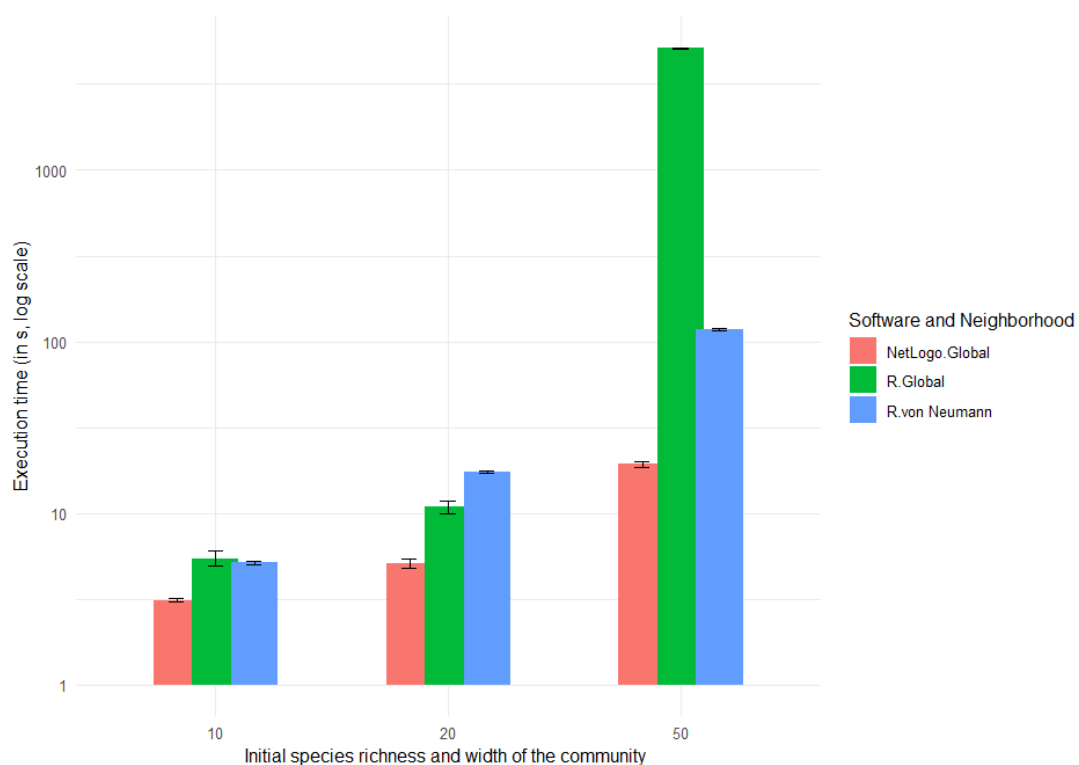


Figure 8 Histogramme du temps d'exécution des simulations du modèle Hubbell, selon la taille initiale de la communauté, puis selon le type de logiciel et le voisinage appliqué en paramètre

3.2.2. Comparaisons qualitatives

Visuels : NetLogo est précisément conçu pour l'aspect graphiques des simulations multi-agent. De ce fait, il est possible de créer une multitude de style d'agent avec un jeu de couleur largement varié. Pour un rythme des tics normal, il y a peu de la latence entre les étapes et l'utilisateur novice peut apprécier l'évolution des communautés. D'autre part, les graphiques associés à la simulation tournent en même temps, laissant à l'utilisateur une contextualisation en temps réel.

Sur R, la génération graphique est faite via l'onglet « Plot » de RStudio, ce qui exclut donc l'aspect de lecture parallèle des données avec les graphiques associés. Cela dit, la génération graphique de *plot()* reste largement appréciable et il n'y a pas de « brouillon visuel ». L'utilisateur peut pareillement régler le temps entre les étapes.

Modularité : Sur NetLogo, les paramètres écologiques sont disponibles sous formes de sliders, modifiables depuis l'interface, pour permettre une plus fine calibration. Les paramètres des communautés tels que la taille de la communauté ou le nombre d'espèces sont enregistrés avant le lancement du modèle par pression du bouton d'initialisation « Setup ». Les règles écologiques sont aussi des slides qui adaptent en temps réel les dynamiques de la communauté. Il est donc possible de faire intervenir à la volée une migration impromptue ou une plus grande probabilité de spéciation.

Par le package SimComm, les paramètres de la communauté et les règles écologiques sont autant disponibles, cependant l'aspect « temps réel » n'existe pas et tout calcul de l'environnement est fait avant la génération de la simulation. D'un autre côté, la génération de la communauté peut être plus fine. Le package accepte 3 options de distribution d'abondance des espèces ainsi que 3 options de répartition spatiale des espèces à l'état initial. Il est aussi possible d'ajouter des presets choisis par l'utilisateur. Finalement, la notion de voisinage est entièrement intégrée dans le package, permettant une règle écologique de dispersion, en accord avec la notion de neutralité.

Stockage : Les capacités de stockage des résultats sont limitées sur NetLogo. Il est possible de conserver un snapshot de l'interface sur un trigger ou de collecter les coordonnées des objets graphiques générés. Cependant, sur R, tout élément peut être sauvegarder pour future utilisation que ça soit de l'analyse des données ou l'archivage des résultats scientifiques. Toutes les tables, matrices, objets générés par la simulation, à toutes les étapes, ainsi que les paramètres, visuels et graphiques peuvent être conservée par simple allocation de variable sous R, avec la commande <-

.

Accessibilité : NetLogo comme R demande à minima l'installation d'un programme associé pour l'exécution complète du modèle sur les machines informatiques. Selon les modalités de l'organisation responsable de NetLogo, il est par ailleurs possible d'intégrer le modèle dans une interface web, éliminant le besoin d'installation de logiciel et facilitant encore plus la simulation du modèle pour un utilisateur novice dans le cas où celui-ci peut être lancé par la pression de 2 boutons d'initialisation puis de lancement (« Setup2 » puis « Go2 »).

Le package SimComm, écrit en R6, demande au contraire une période d'apprentissage plus longue pour un langage relativement plus bas niveau. Pour obtenir les mêmes résultats, il convient à l'utilisateur de comprendre un minimum de notions de programmation pour intégrer les différents paramètres des communautés à différents modèles de simulation.

Outil	R	NetLogo
Visuel	+	+
Modularité	++	+
Stockage	++	-
Accessibilité	-	++

Tableau 1 : Synthèse de l'appréciation qualitative de R et NetLogo

4. DISCUSSION

Le package SimComm modifié intègre la base fondamentale du modèle de Hubbell dans une matrice. La communauté initiale et locale est régie par les événements stochastiques de la dérive démographique et l'immigration depuis une méta-communauté, empêchant la dynamique de tendre vers une mono-dominance d'une espèce dans la communauté, mais aussi la notion de voisinage.

En tant que package, SimComm conserve une grande modularité et liberté de stockage des données, permettant une exploitation future des données brutes comme transformées par d'autres packages de mesure de la biodiversité tels que *vegan* (Oksanen *et al.*, 2012). Bien qu'il soit possible de simplement altérer les règles de la fonction *evolve()*, le cadre de travail et l'objectif éventuel de distribution à un membre de la communauté scientifique requièrent une refonte de l'architecture du package.

Il est intéressant de pouvoir inclure les avantages de la méthode computationnelle vis-à-vis de celle analytique. L'ajout de la dimension de la spatialisation lors de la création de la communauté apporte un élément plus réaliste à la structure de la communauté écologique. La fragmentation des habitats à l'échelle du paysage, malgré la dégradation des écosystèmes à l'échelle des parcelles, est positivement liée à la biodiversité (Riva et Fahrig, 2022). Bien que la théorie neutre constitue une hypothèse nulle, elle n'est pas conçue pour exclure l'implémentation de nouvelles règles d'environnement tant que celles-ci respectent la loi de la neutralité, c'est-à-dire que les règles doivent s'appliquer uniformément à tous les agents ou individus du modèle (Hubbell, 2001).

Les démonstrations présentées dans les résultats peuvent indiquer avec justesse la direction d'influence des paramètres sur le modèle et apporter des réponses théoriques à des questions réalistes. Par exemple, une disposition spatiale uniforme des végétaux est courante dans l'industrie agro-alimentaire (Tilman *et al.*, 2001). La simulation du modèle peut indiquer la vitesse de dispersion des semences dans le temps comparé à d'autres types de spatialisation des plantes (Nathan et Muller-Landau, 2000). Une autre application du modèle peut exister à une échelle plus petite dans le domaine de la microbiologie (Prosser *et al.*, 2007) ou encore à une échelle humaine, sur la propagation d'éléments de culture qui peuvent migrer, muer ou disparaître (Cliff *et al.*, 1992). Les études montrent que les variables basées sur le voisinage jouent un rôle crucial dans la structuration spatiale des forêts, influençant la dynamique des espèces et la diversité (Aguirre

Calderon *et al.*, 2003). De ce fait, la création d'un index de voisinage calculé sur la moyenne d'individu de même espèce à l'individu focal ainsi que sur le type de voisinage, tel que Moore ou von Neumann. (Breckling *et al.*, 2011) permet à l'utilisateur de pouvoir évaluer une éventuelle relation entre l'abondance, la spatialisation initiale ou les paramètres écologique et cette mesure de l'agrégation spatiale des espèces. Par exemple, l'extinction groupée des espèces a des effets plus forts et plus durables sur la dynamique des méta communautés, expliquant l'importance de la structure spatiale des extinctions dans la récupération des communautés (Vercken, 2021).

Par ailleurs, l'intégration des simulations de masse ouvre encore plus les applications scientifiques. Les chaînes de Markov et le Machine Learning étant des domaines de plus en plus utilisés dans la communauté scientifique depuis les dernières décennies, dans le sillage des prouesses technologiques, ces méthodes permettent d'éliminer un problème souvent récurrent dans la méthode scientifique observationnelle qui est le biais lié au faible nombre de données, tout en ajoutant de la robustesse aux formules mathématiques élaborées à partir de ces données (Hastie *et al.*, 2009). Par exemple, l'application des chaînes de Markov Monte Carlo en écologie communautaire permet de modéliser et d'analyser des systèmes écologiques complexes avec une plus grande précision (Cobb et Chen, 2003). Plus simplement, faire une moyenne de 5 sets de données de dynamiques des populations peut entraîner une grande variance et fausser rapidement les interprétations si le 6ème set de données est entièrement contraire aux 5 premiers. La simulation de masse évite donc les interprétations fragiles (Tisue et Wilensky, 1999).

Les packages à portée écologique sur R sont souvent très spécialisés dans l'analyse des données récoltées, mais moins abondants dans la création de données théoriques et leur simulation. C'est dans ce but que NetLogo a été développé en 1999 (Tisue et Wilensky, 1999) et cela se reflète dans les résultats de vitesse d'exécution des simulations. Bien que la simulation de la communauté locale de NetLogo inclue celle de la méta-communauté par design originel, il convient de noter que le calcul du temps écoulé ne commence qu'au début de la simulation et non à l'appel de la création de la communauté. Cela était particulièrement frappant lors des tests $S = 50$, où la génération de la communauté locale était plus longue que la simulation de celle-ci pendant 1000 étapes. Cette génération de table d'individus étant beaucoup plus brève sur R, étant son domaine de prédilection, il conviendra de poursuivre les tests réalisés en poussant au plus loin les limites de tailles des communautés.

Il est possible de poser l'hypothèse que, dès que le package SimComm est optimisé pour la gestion du voisinage global, le temps de génération et de simulation serait plus avantageux pour

SimComm que pour NetLogo. Cet élément est d'autant plus pertinent dans le cas où nous analysons des métacommunautés qui peuvent comprendre plusieurs millions d'individus (Hubbell, 2006). Dans le cas des grands nombres, l'optimisation du voisinage global est probablement une priorité numéro un. L'implémentation actuelle est telle qu'à chaque cellule de la matrice, un appel de la matrice précédente entière est fait afin de réaliser le sampling via la méthode privée *neighbours()*. Une limitation à un appel par pas de simulation serait certainement en faveur de SimComm par rapport à NetLogo. Mais reste l'objet d'études supplémentaires

Au-delà de cette optimisation, il reste à convenir que l'application scientifique de NetLogo et de SimComm semble remplir des niches différentes. Alors que SimComm est très modulable, il reste peu accessible au grand public. À l'inverse, NetLogo est plus limité en fonction, mais remplit très bien une fonction pédagogique de démonstration du modèle. Cela n'exclut pas une démonstration via SimComm, mais elle s'inscrirait dans une configuration plus large d'apprentissage à la gestion des multi-agents ou à l'implémentation dans d'autres packages de mesures des simulations.

En outre, il est nécessaire de rappeler le cadre relativement limitant de la prédiction des dynamiques de communautés par la théorie neutre. Comme Hubbell le signale, il reste aberrant pour tout écologiste de ne considérer que la neutralité en omettant la notion des niches. Attribuer des attributs différents à chacun des individus n'est cependant pas une innovation, nous pouvons précisément distinguer les uns des autres par l'espèce qui leur est associée. Il pourrait cependant être utile d'exploiter les méthodes de voisinages pour simuler les effets allélopathiques des plantes. Par exemple, (Aguirre Calderon *et al.*, 2003) ont démontré l'importance des interactions de voisinage dans la structuration des forêts, suggérant que des mécanismes similaires pourraient être utilisés pour modéliser les effets allélopathiques dans SimComm.

L'autre caractère limitant de la théorie neutre est le strict cadre d'un niveau trophique unique. Sachant que tout écosystème non fermé et contrôlé comporte des interactions biotiques entre différentes espèces sans qu'elles soient en compétition pour les mêmes ressources (Tilman *et al.*, 2001), une proposition d'extension du modèle neutre par la méthode computationnelle pourrait inclure plusieurs communautés locales en parallèle, dont les paramètres d'interactions tels que la prédation seraient contrôlés par l'environnement, à l'instar de l'objet *cm_hubbell()*. La neutralité serait alors conservée à l'intérieur des communautés respectives tout en permettant des interactions inter-trophiques.

Enfin, le troisième élément limitant de la théorie est le jeu à somme nulle, excluant la proposition d'espaces vides, qu'ils soient occupés par une perturbation telle qu'un feu propagatif ou par une frontière naturelle comme une rivière, limitant alors les possibilités de dispersion pour chacun des agents uniformément. Intégrer des éléments de fragmentation du paysage, temporelle ou non, comme proposé par Riva et Fahrig (2022), pourrait permettre de concilier l'abstraction de la théorie d'écologie théorique avec les applications réelles des modèles écologiques.

5. CONCLUSION

L'objectif de mon stage était d'intégrer la théorie neutre de la biodiversité et de la biogéographie du Hubbell afin d'apporter une entrevue dans la simulation et compréhension des dynamiques écologiques dans le langage R. En combinant les concepts fondamentaux de cette théorie avec les capacités de la modélisation multi-agent, j'ai pu faire écrire un outil exploitable et modulable qui permet d'explorer l'évolution des abondances des espèces aux seins des écosystèmes.

Le package SimComm offre une plateforme puissante pour la simulation de communautés, avec une grande liberté pour la génération initiale de celles-ci, en prenant compte des processus stochastiques de mortalité, migration et de spéciation, tout en ajoutant une 3^{ème} dimension avec la structuration spatiale et les notions de voisinages, associés aux processus de dispersions des espèces.

Le cadre du package reste cependant limité vis-à-vis de NetLogo lorsqu'il s'agit de faire la démonstration simple et pédagogique des mécanismes de la théorie neutre, mais permettra, avec un travail nécessaire d'optimisation et éventuellement une plus grande capacité computationnelle des machines, de subvenir aux besoins scientifiques vis-à-vis de l'étude des communautés écologiques et l'application des hypothèses nulles proposées par la théorie. Les résultats obtenus restent significatifs et pertinents pour les objectifs de modélisations.

Une progression du package peut être envisagée par l'intégration de la théorie des niches, des autres interactions biotiques telle que la prédation, mais aussi l'inclusion des barrières limitantes à la dispersion ou l'interaction entre plusieurs communautés locales par niveau trophique. Après quoi, il conviendra d'explorer l'intégration des données réelles et des structures environnementales observées afin de permettre une application à l'échelle humaine dans les objectifs de conservation ou de gestion des écosystèmes qu'apportent les modèles dans leur définition large. Une implémentation du package dans un environnement pre-existant au sein des

autres packages de la communauté scientifique et sur CRAN est également une option à explorer, notamment en jonction avec d'autres langages de programmations qui profiterais des forces de R tout en compensant ses faiblesses.

En conclusion, ce travail m'a permis d'étendre les capacités de modélisation du package mais aussi apporter une perspective nouvelle sur l'élaboration des modèles computationnels complexe et leurs applications en écologie et également une plus fine compréhension théorique des interactions écologiques et des dynamiques des communautés dans le temps et l'espace.

6. REFERENCES

- Aguirre Calderon O.A., Hui G., Gadow K., et Pérez J. 2003. An analysis of spatial forest structure using neighborhood-based variables. *Forest Ecology and Management*, 183, p. 137-145. DOI: 10.1016/S0378-1127(03)00102-6
- Alonso D., Etienne R.S., et McKane A.J. 2006. The merits of neutral theory. *Trends in Ecology & Evolution*, 21(8), p. 451-457. DOI: 10.1016/j.tree.2006.03.019
- Bonabeau E. 2002. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*, 99(suppl_3), p. 7280-7287. DOI: 10.1073/pnas.082080899
- Breckling B., Pe'er G., et Matsinos Y. 2011. Cellular Automata in Ecological Modelling. Dans : *Modelling Complex Ecological Dynamics: An Introduction into Ecological Modelling for Students, Teachers & Scientists*, p. 105-117. DOI: 10.1007/978-3-642-05029-9_8
- Chave J. 2004. Neutral theory and community ecology. *Ecology Letters*, 7(3), p. 241-253. DOI: 10.1111/j.1461-0248.2003.00566.x
- Cliff A., Pred A., et Hägerstrand T. 1992. Classics in human geography revisited: Hägerstrand, T. 1967: Innovation diffusion as a spatial process. Chicago: University of Chicago Press. Translation and postscript by Allan Pred. *Progress in Human Geography*, 16(4), p. 541-544. DOI: 10.1177/030913259201600403
- Cobb G.W. et Chen Y.-P. 2003. An Application of Markov Chain Monte Carlo to Community Ecology. *The American Mathematical Monthly*, 110(4), p. 265-288. DOI: 10.1080/00029890.2003.11919964
- Costanza R., d'Arge R., de Groot R., Farber S., Grasso M., Hannon B., Limburg K., Naeem S., O'Neill R.V., Paruelo J., Raskin R.G., Sutton P., et van den Belt M. 1997. The value of the world's ecosystem services and natural capital. *Nature*, 387(6630), p. 253-260. DOI: 10.1038/387253a0
- Deevey E.S. 1969. MARGALEF, R. 1968. Perspectives in ecological theory. Univ. Chicago Press, Chicago, Ill. 111 p. \$4.50. *Limnology and Oceanography*, 14(2), p. 313-315. DOI: 10.4319/lo.1969.14.2.0313
- Folke C., Carpenter S., Walker B., Scheffer M., Elmqvist T., Gunderson L., et Holling C. s. 2004. Regime Shifts, Resilience and Biodiversity in Ecosystem Management. *Annual Review of Ecology, Evolution, and Systematics*, 35, p. 557-581. DOI: 10.1146/annurev.ecolsys.35.021103.105711
- Gotelli N.J. et Colwell R.K. 2001. Quantifying biodiversity: procedures and pitfalls in the measurement and comparison of species richness. *Ecology Letters*, 4(4), p. 379-391. DOI: 10.1046/j.1461-0248.2001.00230.x
- Grimm V., Berger U., Bastiansen F., Eliassen S., Ginot V., Giske J., Goss-Custard J., Grand T., Heinz S., Huse G., Huth A., Jepsen J., Jørgensen C., Mooij W., Müller B., Pe'er G., Piou C., Railsback S., Robbins A., et Deangelis D. 2006. A Standard Protocol for Describing Individual-Based and Agent Based Models. *Ecological Modelling*, 198, p. 115-126. DOI: 10.1016/j.ecolmodel.2006.04.023

- Groom M.J., Meffe G.K., et Carroll C.R. 2006. *Principles of conservation biology*. 3rd ed. Sunderland, Mass. : Sinauer Associates, 779 p.
- Hanski I. et Hanski I. 1999. *Metapopulation Ecology*. Oxford, New York : Oxford University Press, 324 p. (Oxford Series in Ecology and Evolution).
- Hastie T., Tibshirani R., et Friedman J. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition (Springer Series in Statistics)*.
- Hubbell S.P. 2006. Neutral Theory and the Evolution of Ecological Equivalence. *Ecology*, 87(6), p. 1387-1398. DOI : 10.1890/0012-9658(2006)87[1387:NTATEO]2.0.CO;2
- Hubbell S.P. 2001. *The Unified Neutral Theory of Biodiversity and Biogeography (MPB-32)*. Princeton University Press Disponible sur : <https://www.jstor.org/stable/j.ctt7rj8w> (Consulté le 3 avril 2024).
- Marcon E. et Rousteau A. 2024. *SimComm: A Simulator of Communities*. R package version 0.1-3, <https://github.com/EricMarcon/SimComm>, <https://ericmarcon.github.io/SimComm>. Disponible sur : <https://github.com/EricMarcon/SimComm> (Consulté le 4 juillet 2024).
- Nathan R. et Muller-Landau H.C. 2000. Spatial patterns of seed dispersal, their determinants and consequences for recruitment. *Trends in Ecology & Evolution*, 15(7), p. 278-285. DOI : 10.1016/S0169-5347(00)01874-7
- Odum E.P. 1969. The Strategy of Ecosystem Development. *Science*, 164(3877), p. 262-270. DOI : 10.1126/science.164.3877.262
- Oksanen J., Blanchet F.G., Kindt R., Legendre P., Minchin P., O'Hara R., Simpson G., Solymos P., Stevenes M., et Wagner H. 2012. *Vegan: Community Ecology Package*. R package version 2.0-2.
- Prosser J.I., Bohannan B.J.M., Curtis T.P., Ellis R.J., Firestone M.K., Freckleton R.P., Green J.L., Green L.E., Killham K., Lennon J.J., Osborn A.M., Solan M., van der Gast C.J., et Young J.P.W. 2007. The role of ecological theory in microbial ecology. *Nature Reviews Microbiology*, 5(5), p. 384-392. DOI : 10.1038/nrmicro1643
- Punchi-Manage R. 2023. *NetLogo Unified Neutral Non-Neutral Model*. <http://netlogo/models/Unified-Neutral-Non-Neutral-Model>.
- Railsback S. et Grimm V. 2012. *Agent-Based and Individual-Based Modeling: A Practical Introduction*.
- Riva F. et Fahrig L. 2022. Landscape-scale habitat fragmentation is positively related to biodiversity, despite patch-scale ecosystem decay. *Ecology Letters*, 26. DOI : 10.1111/ELE.14145
- Tilman D., Fargione J., Wolff B., D'Antonio C., Dobson A., Howarth R., Schindler D., Schlesinger W.H., Simberloff D., et Swackhamer D. 2001. Forecasting agriculturally driven global environmental change. *Science (New York, N.Y.)*, 292(5515), p. 281-284. DOI : 10.1126/science.1057544
- Tisue S. et Wilensky U. 1999. NetLogo: A Simple Environment for Modeling Complexity.
- Trudgill S. 2007. Tansley, A.G. 1935: The use and abuse of vegetational concepts and terms. *Ecology* 16, 284 307. *Progress in Physical Geography - PROG PHYS GEOG*, 31, p. 517-522. DOI : 10.1177/0309133307083297
- Vercken E. 2021. Unity makes strength: clustered extinctions have stronger, longer-lasting effects on metacommunities dynamics. *Peer Community In Ecology*, DOI : 10.24072/pci.ecology.100084

Whittaker R.H. 1972. Evolution and Measurement of Species Diversity. *Taxon*, 21(2/3), p. 213-251. DOI : 10.2307/1218190

Wilensky U. 1999. *NetLogo Home Page*. Disponible sur : <https://ccl.northwestern.edu/netlogo/> (Consulté le 3 avril 2024).

7. ANNEXES

Annexe 1 : Données brutes relatives à la vitesse d'exécution du modèle en fonction de la communauté initiale, du logiciel utilisé et du voisinage appliqué.

```
> execution_speed_data
```

time	S	software	neighborhood
1	3.027	10	NetLogo Global
2	3.030	10	NetLogo Global
3	3.065	10	NetLogo Global
4	3.068	10	NetLogo Global
5	3.087	10	NetLogo Global
6	3.126	10	NetLogo Global
7	3.127	10	NetLogo Global
8	3.128	10	NetLogo Global
9	3.152	10	NetLogo Global
10	3.261	10	NetLogo Global
11	4.807	20	NetLogo Global
12	4.885	20	NetLogo Global
13	4.892	20	NetLogo Global
14	4.930	20	NetLogo Global
15	5.000	20	NetLogo Global
16	5.018	20	NetLogo Global
17	5.129	20	NetLogo Global
18	5.206	20	NetLogo Global
19	5.432	20	NetLogo Global
20	5.889	20	NetLogo Global
21	18.504	50	NetLogo Global
22	18.522	50	NetLogo Global
23	18.771	50	NetLogo Global
24	18.826	50	NetLogo Global
25	18.857	50	NetLogo Global
26	19.308	50	NetLogo Global
27	19.837	50	NetLogo Global

28	19.905	50	NetLogo	Global
29	20.082	50	NetLogo	Global
30	20.258	50	NetLogo	Global
31	4.800	10	R	Global
32	5.100	10	R	Global
33	5.120	10	R	Global
34	5.130	10	R	Global
35	5.240	10	R	Global
36	5.280	10	R	Global
37	5.560	10	R	Global
38	5.630	10	R	Global
39	6.310	10	R	Global
40	6.530	10	R	Global
41	62.680	20	R	Global
42	58.360	20	R	Global
43	58.050	20	R	Global
44	58.690	20	R	Global
45	56.860	20	R	Global
46	56.550	20	R	Global
47	62.170	20	R	Global
48	59.860	20	R	Global
49	59.130	20	R	Global
50	55.890	20	R	Global
51	5076.727	50	R	Global
52	5090.321	50	R	Global
53	5097.726	50	R	Global
54	5102.575	50	R	Global
55	5105.908	50	R	Global
56	5107.485	50	R	Global
57	5136.889	50	R	Global
58	5153.109	50	R	Global
59	5160.229	50	R	Global
60	5167.489	50	R	Global

61	5.000	10	R	von Neumann
62	5.050	10	R	von Neumann
63	5.100	10	R	von Neumann
64	5.100	10	R	von Neumann
65	5.180	10	R	von Neumann
66	5.190	10	R	von Neumann
67	5.200	10	R	von Neumann
68	5.210	10	R	von Neumann
69	5.210	10	R	von Neumann
70	5.410	10	R	von Neumann
71	17.220	20	R	von Neumann
72	16.440	20	R	von Neumann
73	16.420	20	R	von Neumann
74	16.740	20	R	von Neumann
75	16.490	20	R	von Neumann
76	16.230	20	R	von Neumann
77	17.230	20	R	von Neumann
78	16.640	20	R	von Neumann
79	16.300	20	R	von Neumann
80	16.440	20	R	von Neumann
81	114.950	50	R	von Neumann
82	115.250	50	R	von Neumann
83	115.670	50	R	von Neumann
84	116.370	50	R	von Neumann
85	116.380	50	R	von Neumann
86	116.670	50	R	von Neumann
87	118.810	50	R	von Neumann
88	119.350	50	R	von Neumann
89	119.370	50	R	von Neumann
90	119.680	50	R	von Neumann
>				