



**HAL**  
open science

# Modélisation prédictive de populations d'insectes vecteurs structurées par stade dans des environnements fluctuants

Wallid Kandouci

► **To cite this version:**

Wallid Kandouci. Modélisation prédictive de populations d'insectes vecteurs structurées par stade dans des environnements fluctuants. Statistiques [math.ST]. 2021. hal-03930980

**HAL Id: hal-03930980**

**<https://hal.inrae.fr/hal-03930980>**

Submitted on 9 Jan 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

UNIVERSITÉ DE MONTPELLIER  
FACULTÉ DES SCIENCES  
MASTER 2 : BIOSTATISTIQUE  
RAPPORT DE STAGE :

---

**Modélisation prédictive de populations  
d'insectes vecteurs structurées par stade  
dans des environnements fluctuants**

---

*Auteur :*  
Walid KANDOUCI

*Encadrants :*  
Nicolas SAUVION (INRAE)  
David PLEYDELL (INRAE)

Jury :  
Gilles DIDIER  
Benoite DE SAPORTA  
Elodie BRUNEL



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>6</b>
1.1	Présentation de la structure d'accueil . . . . .	6
1.2	Contexte . . . . .	8
1.3	Objectif . . . . .	12
<b>2</b>	<b>Les données biologiques</b>	<b>13</b>
2.1	Données expérimentales pour estimer la vitesse de développement des différents stades . . . . .	13
2.2	Données météorologiques . . . . .	14
2.2.1	Températures : valeurs manquantes . . . . .	15
<b>3</b>	<b>Modèles classiques de développement intra-stade</b>	<b>16</b>
3.1	Modèles de vitesse de développement en fonction de la température . . . . .	16
3.2	Applications des modèles sur nos données . . . . .	17
3.3	Discussion . . . . .	20
<b>4</b>	<b>Un modèle de dynamique des populations adapté aux environnements fluctuants</b>	<b>21</b>
4.1	Introduction . . . . .	21
4.2	Modèles matriciels de population . . . . .	22
4.2.1	Matrice de Leslie . . . . .	22
4.2.2	Matrice de Lefkovitch . . . . .	23
4.3	Modèles de projections intégrales . . . . .	24
4.3.1	Modèles de projections intégrales classiques (MPI) . . . . .	24
4.3.2	Modèle de projection intégrale d'une matrice de Lefkovitch (PIML) pour les insectes . . . . .	25
4.3.3	Exemple d'implémentation de modèle de projection sur un seul stade . . . . .	27
4.4	Implémentation du modèle PIML sur les données . . . . .	28
4.4.1	Description de l'hétérogénéité des vitesses de développement . . . . .	29
4.4.2	Lien entre les températures fluctuantes et le modèle PIML . . . . .	29
4.4.3	Estimation des paramètres dans les modèles de population à l'aide d'une approche Bayésienne . . . . .	30
4.5	Résultats . . . . .	33
4.6	Discussion . . . . .	39
<b>5</b>	<b>Développement de l'application Psyl'risk</b>	<b>41</b>
5.1	Le réseau actuel d'épidémiologie-surveillance des psylles . . . . .	41
5.2	Une nouvelle interface web, Psyl'Risk . . . . .	41
5.3	Conclusion . . . . .	43
<b>6</b>	<b>Conclusion générale</b>	<b>44</b>
<b>A</b>	<b>Code R - Modèles de Brière, Kontodimas et Taylor</b>	<b>47</b>
<b>B</b>	<b>Données manquantes - 2005</b>	<b>48</b>

<b>C</b>	<b>Code R - Modèle PILM</b>	<b>49</b>
<b>D</b>	<b>Packages Nimble, NimbleTempDev et NimbleAPT</b>	<b>54</b>
<b>E</b>	<b>Fonction stBriere</b>	<b>57</b>
<b>F</b>	<b>Résultats des autres modèles</b>	<b>58</b>

## Table des figures

1	Dynamique des populations de <i>Cacopsylla pruni</i> dans la région de Montpellier en 2018. Les courbes sont obtenues partir de collectes hebdomadaires de psylles dans des massifs de prunelliers par la technique dite de frappage (source <a href="http://cpruni.blogspot.com/">http://cpruni.blogspot.com/</a> ). . . . .	9
2	Photos des différents stades de développement <i>Cacopsylla pruni</i> (photos present par Nicolas Sauvion). . . . .	10
3	Dynamique des populations de <i>Cacopsylla pruni</i> : <b>(A)</b> courbes d’observations hebdomadaires d’observations de la présence des psylles dans un massif de prunelliers <b>(B)</b> Périodes hypothétiques de développement des différents stades <b>(C)</b> Différent scénarios d’émergence des adultes de la nouvelle génération (imagos) .	11
4	Températures enregistrées durant la période du 16/03/2005 au 13/07/2005 (gauche) et zoom sur la semaine du 28/03/2005 au /03/04/2005 permettant de voir les variations de températures enregistrées avec un pas de temps de 3 heures (droite) .	14
5	Valeurs manquantes des températures ajoutées aux données météo France 2005 (Montpellier) . . . . .	15
6	Relation entre la température et la vitesse de développement chez les insectes. <b>(A)</b> Courbe de performance thermique théorique avec le minimum thermique critique ( $T_{min}$ ), le maximum ( $T_{max}$ ), l’optimal ( $T_{opt}$ ) et la température de base ( $T_{base}$ ). <b>(B)</b> Quelques modèles inférant la relation entre la température et la vitesse de développement, d’après (Rebaudo et al. 2018). . . . .	16
7	Courbes des vitesses de développement des différents stades des psylles selon les trois modèles : Brière (rouge), Kontodimas (bleu) et Taylor (vert) sur les données de l’expérimentation 2005 . . . . .	18
8	Un exemple graphique d’un cycle de vie : $W_S$ représente la probabilité de rester dans ce même stade $s$ , $B_S$ la probabilité de passer au prochain stade, $F_S$ la contribution à la prochaine génération (fécondité). . . . .	23
9	Implémentation d’un modèle de projection intégrale de base (MPI) pour la dynamique intra-stade (d’après : (Castano 2017)). <b>(a)</b> Fonction de densité $f$ utilisé pour définir les éléments des matrices <b>W</b> et <b>B</b> (Équations :). <b>(b)</b> Projection du développement dans une cohorte représenté à $t = 0$ , $t = 4$ et $t = 12$ . <b>(c)</b> Probabilités de survie et de compléter un stade en fonction du temps. . . . .	27

10	Représentation graphique d'un cycle de vie simplifié de l'espèce <i>Cacopsylla pruni</i> qui fournit une base pour le développement d'un PIML : A chaque pas de temps, les matrices <b>W</b> représentent la proportion d'individus restant dans une classe de stade donnée, les matrices <b>B</b> représentent la proportion d'individus passant à la classe de stade suivante, et <i>F</i> représente la fécondité. . . . .	28
11	Synthèse des estimations obtenus par MCMC pour les paramètres de SDmodel <sub>1</sub> pour les six stades de développement. Ce tableau montre la moyenne (Mean), l'écart types (SD) et des quantiles pour tous les paramètres. L'erreur standard (SE) a été estimée de deux manières, une prenant en compte l'auto-corrélation (Time.series) et l'autre non (Naive). . . . .	33
12	Vitesse de développement moyenne ( $\mu$ ) en fonction des températures pour les différents stades de développement. En noire la moyenne obtenue à partir de 10000 échantillons MCMC et en rouge l'intervalle de crédibilité à 95%. Ces résultats sont obtenus avec sdModel <sub>1</sub> . . . . .	34
13	Écart-type de la vitesse de développement $\sigma$ en fonction des températures pour les différents stades de développement. En noire la moyenne obtenue à partir de 10000 échantillons MCMC et en rouge l'intervalle de crédibilité à 95%. Ces résultats sont obtenus avec sdModel <sub>1</sub> . . . . .	36
14	Courbes simulées de la proportion de présence des différents stades (de œuf à imago) pour quatre cohortes de psylles au cours du temps ( $\pi_{S,c,t}$ ). Les croix sur les graphiques représentent les proportions observées (œufs, larves, imagos) au cours de l'expérimentation de 2005. L'épaisseur de ces courbes représentent les intervalles de crédibilités à 95%, estimés à partir de 1000 échantillons de MCMC. . . . .	38
15	Nouvelle plateforme "Psyl'risk" . . . . .	42
16	Représentation graphique des données collectés à l'Hérault . . . . .	42
17	Représentation sur la carte des deux espèces A (jaune) et B (rouge) . . . . .	43
18	Vitesse de développement en fonction des températures pour les différents stades de développement. . . . .	58
19	Vitesse de développement en fonction des températures pour les différents stades de développement. . . . .	59
20	Vitesse de développement en fonction des températures pour les différents stades de développement. . . . .	60
21	Vitesse de développement en fonction des températures pour les différents stades de développement. . . . .	61
22	Vitesse de développement en fonction des températures pour les différents stades de développement. . . . .	62
23	Vitesse de développement en fonction des températures pour les différents stades de développement. . . . .	63
24	Vitesse de développement en fonction des températures pour les différents stades de développement. . . . .	64
25	Vitesse de développement en fonction des températures pour les différents stades de développement. . . . .	65
26	Vitesse de développement en fonction des températures pour les différents stades de développement. . . . .	66

27 Vitesse de développement en fonction des températures pour les différents stades de développement. . . . . 67

# Remerciements

Je souhaite remercier dans un premier temps mes encadrants Nicolas SAUVION et David PLEYDELL pour leur disponibilité, le temps passé ensemble et le partage de leurs expertises et de connaissances.

Je remercie Frédéric DEDIEU, Fabienne RIBEYRE, Bernard DUFOUR, Leïla BAGNY-BEILHE, Lorelei GUERY, Gerben Martijn TEN HOOPEN , Jean-Benoit MOREL et l'ensemble de l'équipe PHIM pour leur accueil chaleureux et leur bienveillance.

Enfin je souhaite remercier tous mes enseignants de l'université de Montpellier qui m'ont encadré durant ces dernières années.

Je remercie la plateforme de bioinformatique INRAE MIGALE (MIGALE, INRAE, 2020. Plateforme de bioinformatique Migale, doi : 10.15454/1.5572390655343293E12) pour l'aide et/ou les ressources de calcul et/ou de stockage

# 1 Introduction

## 1.1 Présentation de la structure d'accueil

Mon stage s'est déroulé dans les locaux de l'Unité Mixte de Recherche PHIM, au sein du Campus CIRAD de Baillarguet au nord de Montpellier. J'ai été encadré par deux chercheurs INRAE : Nicolas Sauvion, associé à l'équipe FORISK du pôle PRISM de PHIM ; et David Pleydell associé à une autre unité du Campus, l'unité ASTRE. Enfin, j'ai bénéficié d'une bourse octroyée par l'université Montpellier 2 suite à un appel d'offre lancé par la KIM RIVE.

### a) Le CIRAD

Le Centre de Coopération Internationale en Recherche Agronomique pour le développement (CIRAD, <https://www.cirad.fr/>) est un organisme français de recherche agronomique et de coopération internationale pour le développement durable des régions tropicales et méditerranéennes. Le CIRAD est un établissement public à caractère industriel et commercial créé en 1984. Ses activités relèvent des sciences du vivant, des sciences sociales et des sciences de l'ingénieur appliquées à l'agriculture, à l'alimentation, à l'environnement et à la gestion des territoires.

Les chercheurs du CIRAD s'intéressent à des thématiques telles que la sécurité alimentaire, le changement climatique, la gestion des ressources naturelles, la réduction des inégalités pays du Nord versus pays du Sud, et la lutte contre la pauvreté. Le CIRAD a pour objectif de produire de nouvelles connaissances et de les transmettre à ses partenaires pour accompagner l'innovation et le développement agricole. Le CIRAD met son expertise scientifique et institutionnelle au service des politiques publiques de ces pays et intervient au sein des débats internationaux sur les grands enjeux de l'agriculture. Le CIRAD apporte son soutien à la diplomatie scientifique de la France. Le CIRAD comprend deux centres de recherche à Montpellier (campus Lavalette et campus Baillarguet). J'ai effectué mon stage sur le campus de Baillarguet (Montferrier-sur-Lez).



Cirad Baillarguet

## **b) L'INRAE**

Comme j'ai été encadré par deux chercheurs INRAE, il me semble important de décrire brièvement cet organisme de recherche. L'institut national de recherche pour l'agriculture, l'alimentation et l'environnement (INRAE, <https://www.inrae.fr/>) est un institut de recherche sous la tutelle conjointe du ministère chargé, l'Enseignement Supérieur, la Recherche et l'Innovation (MESRI) et de celui chargé de l'Agriculture et de l'Alimentation (MAA). Les recherches à l'INRAE sont orientées autour de 6 grandes thématiques : l'agroécologie, l'alimentation et la santé globale, la biodiversité, la bioéconomie, le changement climatique et les risques associés, la société et les territoires. L'INRAE est structuré en 14 Départements de Recherche (dont le Département Santé des Plantes) et 18 Centres implantés sur tout le territoire français et aux Antilles-Guyane, dont le Centre de Montpellier.

## **c) L'institut PHIM, le pôle PRISM et l'équipe FORISK**

### **L'UMR PHIM**

L'Unité Mixte de Recherche PHIM (Plant Health Institute of Montpellier) a été récemment formée (officiellement le 1<sup>er</sup> janvier de 2021). Elle regroupe plus d'une centaine d'agents permanents du CIRAD, de l'INRAE, de l'IRD, de Montpellier SupAgro et de l'Université Montpellier 2. Elle accueille de très nombreux étudiants en Master ou thèse. PHIM a pour objectif d'étudier les interactions entre les plantes et leurs environnements qui ont un impact sur l'état sanitaire des plantes, leur croissance et leur productivité. L'unité est structurée en 4 pôles (ITEM, VIROM, PHYTOBIOM et PRISM), regroupant au total 14 équipes (<https://umr-phim.cirad.fr/recherche>).

### **L'Équipe FORISK**

L'équipe FORISK (FORecasting epidemic rISK) est l'une des équipes du pôle PRISM (Plant pathogen and pests : dynamics and RiSk Management). Elle est centrée sur la prédiction du risque épidémique dans un objectif d'épidémiologie. L'équipe a pour missions de déterminer comment les facteurs environnementaux affectent la dynamique spatio-temporelle de maladies, de vecteurs ou de ravageurs des plantes pérennes (cacaoyer, caféier, bananier, Citrus et Prunus), de prédire les risques épidémiques et de contribuer à la veille sanitaire internationale en santé végétale. L'équipe intervient dans différentes régions du monde : Europe, départements ultramarins, Afrique de l'Est et de l'Ouest, Amérique du Sud.

### **Le projet fédérateur KIM RIVE**

Montpellier Université d'Excellence (MUSE) a mis en place des dispositifs thématiques et transversaux, les KIM (pour Key Initiatives MUSE) pour affirmer son identité et favoriser les synergies Recherche, Formation, Économie. Au 1<sup>er</sup> février de cette année a été lancée officiellement la KIM Risks & Vectors (RIVE : <https://muse.edu.umontpellier.fr/key-initiatives-muse/rive/>). Son ambition est de soutenir des recherches sur l'émergence, la transmission vectorielle des agents infectieux et leur contrôle, dans une approche "une seule santé" (One Health) . Ainsi, la KIM Rive encourage fortement les chercheurs en santé humaine, animale et végétale à travailler ensemble notamment par le financement de bourses de Master 2 ou de "petits" projets novateurs. C'est dans ce cadre, que mon équipe d'accueil a pu obtenir une bourse pour mon stage.

## UMR ASTRE

ASTRE est une unité mixte de recherche créée en 2017 et placée sous la tutelle de deux organismes de recherche : le CIRAD et l'INRAE. Elle résulte de la fusion des anciennes unités de recherche AGIRs (Animal et gestion intégrée des risques) et CMAEE (Contrôle des maladies animales, exotiques et émergentes). (<https://umr-astre.cirad.fr/>).

Les travaux conduits par ASTRE ont pour objectif d'améliorer la santé animale pour la sécurité alimentaire et la santé publique au Sud, notamment dans le cadre des changements globaux et des transitions dans les socio-écosystèmes

## 1.2 Contexte

La production de fruits et légumes est un secteur majeur de l'économie française. Ainsi, la France est le 3<sup>ème</sup> producteur de fruits et légumes dans l'Union européenne derrière l'Italie et l'Espagne avec huit millions de tonnes de fruits et légumes produites chaque année, ce qui représente un chiffre d'affaire d'environ 8,1 milliards d'euros (chiffres Eurostat).

Cette production est aussi une production de qualité. En effet, la production française de fruits et légumes bénéficie de nombreux signes officiels de qualité. Pour cela, la filière agriculture fait appel à des méthodes scientifiques innovantes comme par exemple la production fruitière intégrée, qui est un système de production économique de fruits de haute qualité qui priorise des méthodes écologiques plus sûres minimisant le recours à des produits agrochimiques afin de protéger l'environnement et la santé humaine.

### a) L'ESFY : un problème récurrent pour l'arboriculture française

Les abricotiers et les pruniers, deux espèces majeures pour l'arboriculture française, ainsi que plusieurs de leurs porte-greffes sont en situation préoccupante pour leur multiplication et leur production du fait d'une maladie, l'Enroulement Chlorotique de l'Abricotier (ECA ou ESFY pour European stone fruit yellows), causée par une bactérie, '*Candidatus phytoplasma prunorum*' et vecté par un psylle *Cacopsylla pruni* (Sauvion et al. 2020). Le phytoplasme est régulé à l'échelle de l'Union européenne dans le cadre du Règlement (UE) 2016/2031 et du Règlement d'exécution (UE) 2019/829 entrés en vigueur le 14 décembre 2019. De "Organisme de quarantaine" (OQ), ce pathogène est devenu "Organisme réglementé non de quarantaine" (ORNQ).

L'ESFY a fait l'objet d'une expertise de l'EFSA en 2010-12 et d'un projet européen COST de 2010 à 2014 qui a permis notamment d'établir des cartes de distribution des vecteurs et du phytoplasme à l'échelle de l'UE, et d'estimer l'impact de la maladie dans les 17 pays recensés comme concernés (Steffek et al. 2012). Cet impact est variable selon les zones de production et les plantes hôtes. En France, ces dernières années, des taux de contamination (nombre d'arbres malades observés à un instant t) de 10% voire 25% et plus ont été signalés par des arboriculteurs dans les différents bassins de production d'abricots ou de prunes. Ces observations ont justifié plusieurs arrêtés préfectoraux rendant obligatoire la mise en œuvre de mesures de surveillance, de détection et de lutte (dont l'arrachage des arbres contaminés) contre l'ESFY dans toutes les régions particulièrement concernées par cette maladie (Corse, Occitanie, Rhône-Alpes). Ainsi, dans l'Hexagone, l'ESFY continue de poser des problèmes à la filière fruits, malgré le contrôle sanitaire rigoureux des plants fruitiers et les mesures mises en œuvre pour limiter les impacts de la maladie.

Des équipes INRAE (Montpellier, Bordeaux, Valence) travaillent depuis plus de 20 ans sur l'ESFY (Thébaud et al. 2009) en relation étroite avec les professionnels de la filière Fruits et les services de l'État chargés de la surveillance en santé végétale. Dans le cadre de ces collaborations, la question de l'origine des contaminations (voies humaines ? voies naturelles via une dissémination par les insectes vecteurs depuis les Prunus sauvages ?) et des points d'entrées possibles de la maladie (des pépinières aux vergers) a fait l'objet de plusieurs travaux de recherche et notamment une analyse multi-sites dans plusieurs bassins de productions d'abricots (Plaine de Valence, La Crau, Pyrénées-Orientales) (Marie-Jeanne et al. 2020). Aujourd'hui, la lutte contre cette maladie consiste essentiellement à limiter la présence de psylles dans les vergers au début du printemps, au moment de leurs vols de retours d'hivernage (voir ci-après le descriptif de la biologie du psylle). Cette prophylaxie nécessite de bonnes connaissances sur les vecteurs et en particulier sur la dynamique des populations.

### b) Cycle biologique de l'insecte

Le phytoplasme responsable de l'ESFY est disséminé par le psylle *Cacopsylla pruni* (Carraro et al. 1998) inféodé aux plantes du genre Prunus. Les psylles sont associés à un groupe d'insectes dits piqueurs-suceurs, les hémiptères, comprenant notamment les pucerons et les cigales. Ces insectes se nourrissent de la sève phloémienne des arbres. Le psylle est capable de transmettre le phytoplasme au moment de l'insertion de ses stylets dans les vaisseaux du phloème. Ce vecteur, comme l'agent pathogène, est endémique dans tout l'Ouest Palaearctique (Sauvion et al. 2021).

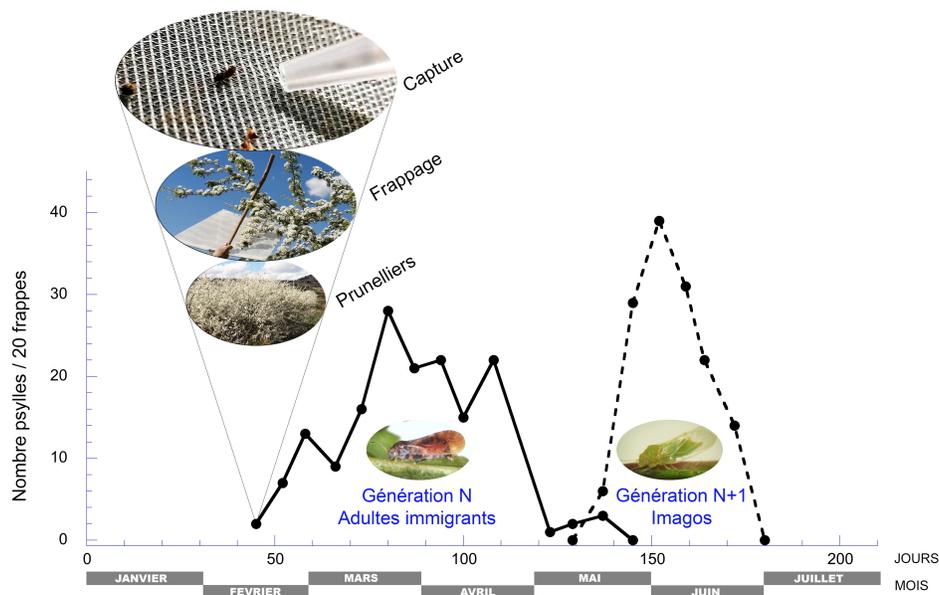


FIGURE 1 – Dynamique des populations de *Cacopsylla pruni* dans la région de Montpellier en 2018. Les courbes sont obtenues partir de collectes hebdomadaires de psylles dans des massifs de prunelliers par la technique dite de frappage (source <http://cpruni.blogspot.com/>).

Le cycle biologique de *C. pruni* conditionne la façon dont les arbres sont contaminés Figure 1. A la fin de l'hiver-début du printemps, les psylles se reproduisent sur des Prunus et en particulier sur les Prunus sauvages (prunelliers et myrobolans) qui sont ses hôtes préférentiels (Labonne et

al. 2004). Sporadiquement cependant, des psylles peuvent se retrouver sur des Prunus cultivés (abricotiers, pruniers, etc) plus ou moins proches des massifs/haies sauvages (Sauvion et al. 2012). C’est à ce moment-là que ces psylles posent problème car c’est le moment où ils peuvent contaminer les vergers (voir comment un peu plus loin).

Les femelles sont capables de s’accoupler plusieurs fois. Il semble qu’elles pondraient ensuite leur œufs très peu de temps après ces accouplements. Ces œufs vont se développer plus ou moins vite en fonction des conditions climatiques. Suivront ensuite cinq stades larvaires (notés L1 à L5) et enfin l’émergence des adultes de la nouvelle génération (nommés imagos), fin mai-début juin, dans la région de Montpellier.

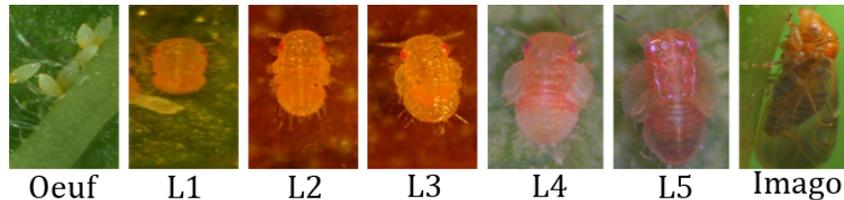


FIGURE 2 – Photos des différents stades de développement *Cacopsylla pruni* (photos présent par Nicolas Sauvion).

### c) Le cycle biologique du pathogène

Le phytoplasme peut être acquis par des larves de stade L5 et surtout les imagos. Une fois ingéré, le phytoplasme est capable de se multiplier dans l’insecte et y persistera jusqu’à sa mort (Thébaud et al. 2009). La quantité de phytoplasme dans le psylle doit atteindre un seuil pour que l’insecte devienne infectieux (c’est-à-dire capable d’inoculer efficacement le pathogène). Ce seuil est atteint en moyenne au bout de 25 jours (Thébaud et al. 2009). D’après les observations de terrain, il semble que les imagos restent très peu de temps sur leur plante-hôte. En effet, dès leurs émergence, ils s’envolent pour une migration plus ou moins longue vers des massifs de conifères où ils trouveront refuge jusqu’au printemps suivant (Marie-Jeanne et al. 2020). Les migrations des psylles entre Prunus et conifères semblent aléatoires et s’étendraient sur plusieurs dizaines de kilomètres, à l’échelle d’un bassin de production (Marie-Jeanne et al. 2020). A la fin de l’hiver-début du printemps, les adultes reviennent sur Prunus pour se reproduire. C’est à ce moment-là que les insectes ayant acquis le phytoplasme l’année précédente sont infectieux et peuvent transmettre le pathogène aux Prunus.

### d) Le cycle biologique du pathogène conditionne la gestion de la maladie

Ces observations montrent l’importance d’appréhender au mieux un facteur clé en épidémiologie, la durée de latence (durée entre le moment où le pathogène est acquis et le moment où il peut être inoculé efficacement). Dans notre cas particulier, cette durée de latence est exceptionnellement longue (8-9 mois) et conditionne fortement les mesures de lutte actuelles contre l’ESFY. En effet, le moment clé d’une contamination potentielle en verger est le moment de présence des psylles immigrants de retour d’hivernage. La lutte actuelle est basée sur une détection précoce des arrivées de psylles par la mise en place d’un réseau d’épidémiosurveillance à l’échelle des bassins de production d’abricots et de prunes dans le sud de la France. Des personnes sont chargées, chaque semaine de janvier à juillet, de quantifier les niveaux de populations de psylles

dans 5 sites différents par la technique dite "de frappe" : une toile tendue de  $80 \times 80\text{cm}$  est placée sous une branche, cette dernière est frappée avec un bâton pour faire tomber les éventuels psylles sur la toile. Les insectes sont ensuite dénombrés et aspirés dans un tube avec un petit aspirateur à bouche. L'opération est répétée sur 20 branches. Les informations de chaque site de collecte sont centralisées chaque semaine dans une base de données par N. Sauvion et diffusées au travers un blog (<https://cpruni.blogspot.com/>), pour enfin être reprises par des BSV (Bulletin Santé du Végétal). Les arboriculteurs disposent ainsi d'informations "rapidement" diffusées pour prendre la décision d'intervenir ou non dans leurs vergers contre le psylle.

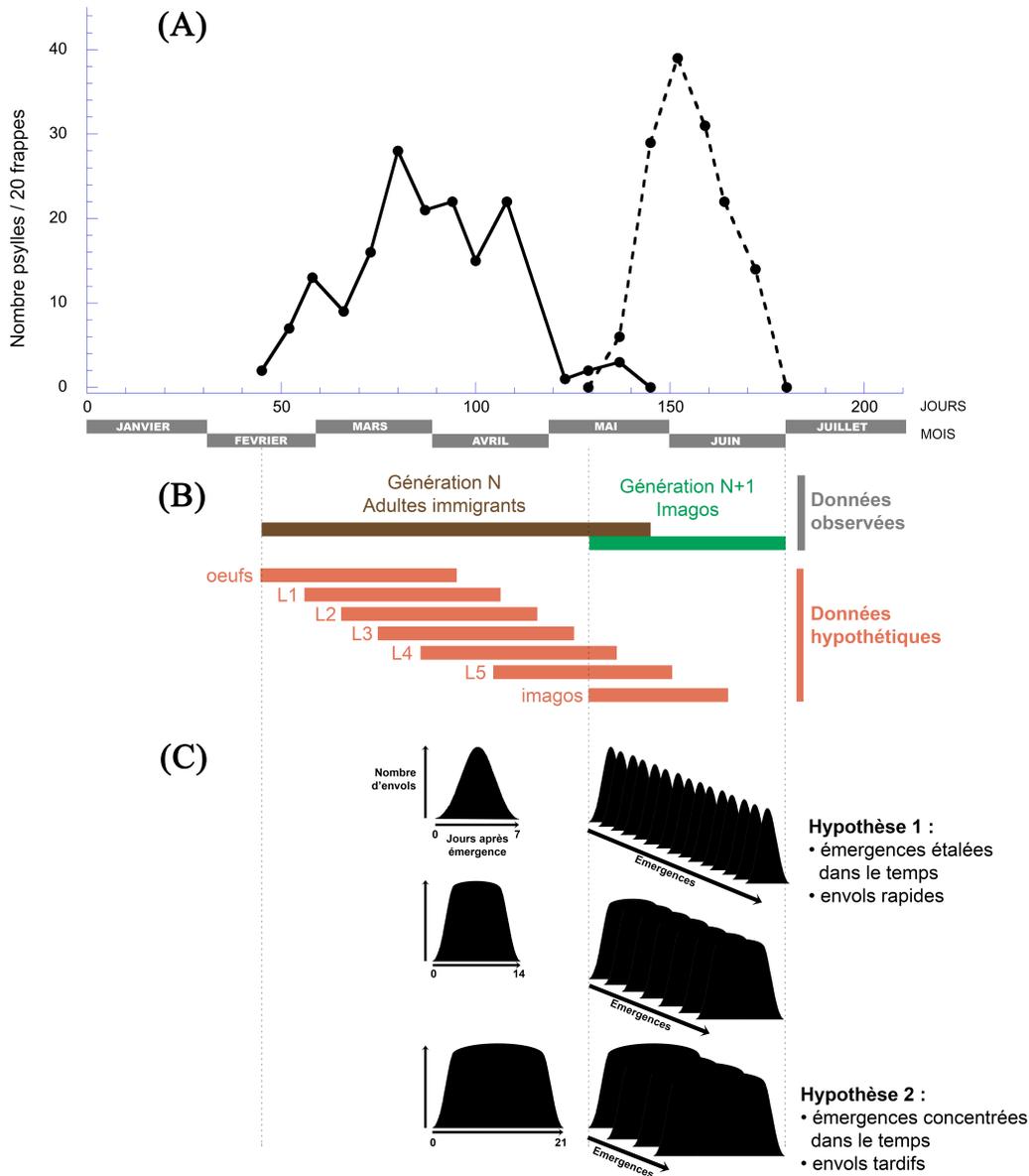


FIGURE 3 – Dynamique des populations de *Cacopsylla pruni* : (A) courbes d'observations hebdomadaires d'observations de la présence des psylles dans un massif de prunelliers (B) Périodes hypothétiques de développement des différents stades (C) Différent scénarios d'émergence des adultes de la nouvelle génération (imagos)

Or, cette stratégie de lutte repose sur une hypothèse très forte : les psylles transmettent le pathogène à leur retour d'hivernage, mais pas avant de s'envoler vers les massifs de conifères au début de l'été. A ce moment-là, des psylles pourraient être porteurs du phytoplasme mais pas encore infectieux, car ils ne resteraient pas assez longtemps ( $> 3$  semaines) sur leur plante-hôte (Prunus) pour multiplier suffisamment le phytoplasme. Cette hypothèse correspond à l'hypothèse 1 de la **Figure 3 (C)** : la courbe observée des imagos correspondrait à des émergences en série d'individus qui resteraient quelques jours sur les Prunus (3 à 4 jours tout au plus). Les émergences s'étaleraient alors sur presque toute la durée observée de présence des imagos. Mais on pourrait imaginer une autre hypothèse (hypothèse 2) : une série d'individus qui émergent tous groupés les premiers jours d'observation d'imagos, et qui restent longtemps (dans l'exemple, jusqu'à 3 semaines) sur leur plant hôte ; une durée suffisante pour que des psylles deviennent infectieux. D'un point de vue pratique, il est quasiment impossible de vérifier formellement ces hypothèses sur le terrain. Comprendre ce pic d'émergence par la modélisation s'avérerait particulièrement informatif.

### 1.3 Objectif

Mon stage visait à modéliser les vitesses de développement des différents stades larvaires en fonction de températures fluctuantes pour prédire l'émergence des adultes de la nouvelle génération en conditions réelles. Pour cela, mon travail a consisté à :

- 1) M'approprier la question biologique et bien comprendre la problématique ;
- 2) Trier et introduire numériquement les données de l'étude faites sur le suivi des différents stades et les introduire sous un format exploitable ;
- 3) Comprendre différents types de modèles utiliser classiquement en dynamique des populations pour modéliser la vitesse de développement en fonction de la température ;
- 4) Implémenter les données sur trois modèles pour appréhender leurs intérêts et leurs limites ;
- 5) Me familiariser avec le modèle choisi pour le stage : le modèle de projections intégrales d'une matrice de Lefkovitch (PIML) ;
- 6) Me familiariser avec la programmation sous *nimble*, le code retenu pour implémenter le modèle PIML sur les données ;
- 7) Réussir à coder un programme sans beug majeur permettant d'obtenir une prédiction cohérente avec les observations de terrain ;
- 8) Développer une application sous *RShiny* pour valoriser tout le travail de recherche mené durant plus de 20 ans au travers une plateforme regroupant toutes les données collectées par le réseau d'épidémiosurveillance.

## 2 Les données biologiques

Dans le cadre de mon stage, j'ai exploité deux types de données biologiques :

- Des données expérimentales de suivi du développement des différents stades de *C. pruni* en conditions semi-naturelles ;
- Des données météorologiques.

### 2.1 Données expérimentales pour estimer la vitesse de développement des différents stades

Afin de collecter des données sur la vitesse de développement des psylles en fonction des températures, Nicolas Sauvion a réalisé une expérience en 2005 en conditions semi-naturelles. Cette expérience a consisté à suivre le développement de cohortes d'insectes, du stade œuf au stade imago, en extérieur, dans des cages d'élevage, à l'abri de la pluie. Le principe de l'expérimentation était le suivant :

Au jour J, au minimum dix femelles ont été mises à pondre sur une feuille de prunier, isolées dans une clip-cage (mini-cage pincée sur une feuille).

A J+1 les femelles ont été supprimées, puis visuellement, les œufs ont été dénombrés. Quelques jours plus tard, et pendant plusieurs jours, les œufs et les larves de stade 1 ont été dénombrés. Cette opération a été répétée pour chaque période de transition supposée (L1 à L2, L2 à L3, etc), jusqu'au stade imago ou jusqu'à ne plus observer de larves. Le [tableau 1](#) donne un exemple de fiche de notation de ces dénombrements. Dans ce cas précis, nous avons observé deux L4, et aucune d'elles ne s'est développée jusqu'au stade L5, peut-être à cause des températures trop faibles au moment de l'expérience.

Plusieurs femelles ont été mises à pondre le même jour, et à des intervalles réguliers de mars 2005 à juin 2005 (plus longue durée d'observation : 70 jours, plus courte durée d'observation : 13 jours) sur 33 plantes pour obtenir des cohortes d'insectes se développant à différentes températures. Ces conditions reflétaient les conditions réelles rencontrées par les psylles en milieu naturel.

Plante 1									
	date	adulte	Oeuf	L1	L2	L3	L4	L5	imago
J	2005-03-16	10	0	0	0	0	0	0	0
J+1	2005-03-29	0	0	0	0	0	0	0	0
J+16	2005-04-01	0	0	0	0	0	0	0	0
J+19	2005-04-04	0	4	23	0	0	0	0	0
J+28	2005-04-13	0	0	10	21	0	0	0	0
J+30	2005-04-15	0	0	0	13	0	0	0	0
J+36	2005-04-21	0	0	0	2	9	0	0	0
J+40	2005-04-25	0	0	0	0	7	0	0	0
J+44	2005-04-29	0	0	0	0	3	2	0	0
J+47	2005-05-02	0	0	0	0	0	0	0	0

TABLE 1 – Exemple de notations du nombre d'individus des différents stades de développement de *C. pruni* sur une plante.

En parallèle, une expérience a été réalisée dans une cellule climatique à température constante (10°C et 17°C) pour essayer d’obtenir plus d’informations aux températures basses.

## 2.2 Données météorologiques

Nous souhaitons estimer la vitesse de développement des différents stades en fonction de la température de manière la plus réaliste possible par rapport aux conditions réelles rencontrées par les psylles. Travailler sur des températures moyennes journalières nous semblait peu satisfaisant car *in natura* d’importants écarts de températures sont observés dans la journée ou d’un jour à l’autre, en particulier au printemps (figure 4). Pour tenir compte de ces écarts, il nous fallait donc pouvoir obtenir des données météorologiques sur un pas de temps aussi petit que possible. Météo France met à disposition gratuitement des données météorologiques enregistrées en France pour 62 stations (<https://www.data.gouv.fr/fr/reuses/code-r-de-recuperation-des-donnees-synop-meteo-france/>). Il est possible d’avoir des données enregistrées avec un pas de temps de 3 heures au minimum. Nous avons fait le choix de travailler avec ces données enregistrées toutes les 3 heures, quitte à tester ultérieurement si un pas de temps supérieurs (4 heures ou 6 heures) est aussi bon pour nos estimations et moins coûteux en temps de calculs.

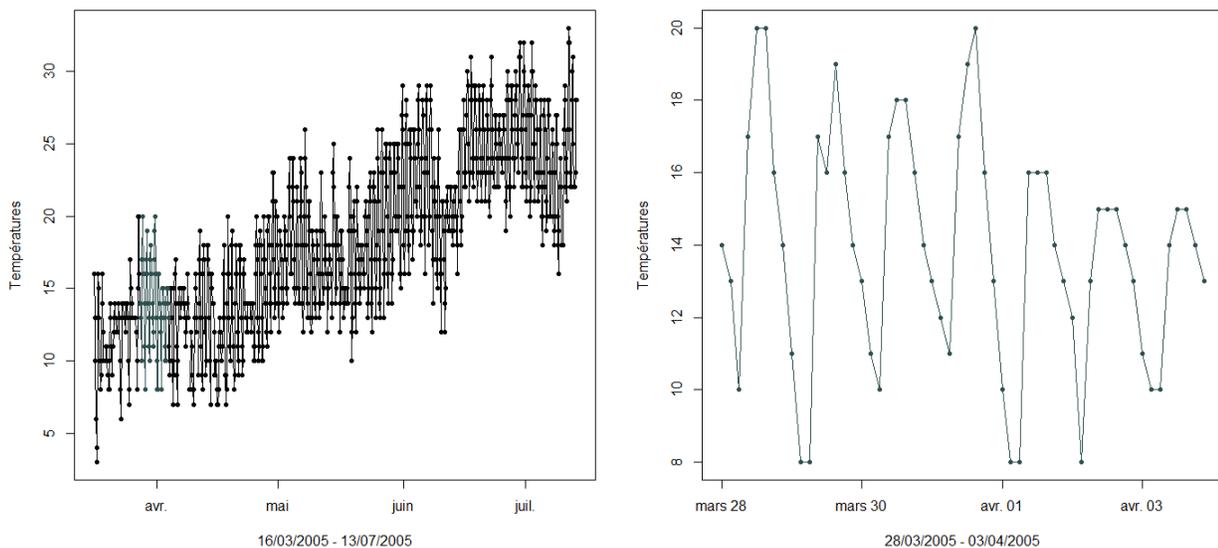


FIGURE 4 – Températures enregistrées durant la période du 16/03/2005 au 13/07/2005 (gauche) et zoom sur la semaine du 28/03/2005 au /03/04/2005 permettant de voir les variations de températures enregistrées avec un pas de temps de 3 heures (droite)

### 2.2.1 Températures : valeurs manquantes

Sur nos données météo de l'année 2005, certaines valeurs de températures étaient manquantes. Plusieurs méthodes basées sur les séries temporelles sont disponibles pour remédier à ce problème, voici les résultats obtenus en utilisant trois méthodes différentes :

modèle	2005-04-08 21 : 00 : 00	2005-04-11 00 : 00 : 00	2005-04-11 03 : 00 : 00
Kalman	10.217565	9.948354	8.607024
ARIMA	12.448108	8.969793	8.127308
Linéaire	10.5	11.0	10.0

TABLE 2 – Comparaison des résultats obtenus pour les trois méthodes utilisés pour remplacés les valeurs manquantes

À l'exception du modèle linéaire qui est assez basique, le filtre de Kalman (finalement choisi) et le modèle ARIMA offrent pour ce cas des valeurs assez similaire et donc le choix d'un modèle ou l'autre n'aura pas un réel impact sur les résultats du modèle. Le code R est disponible en annexe C

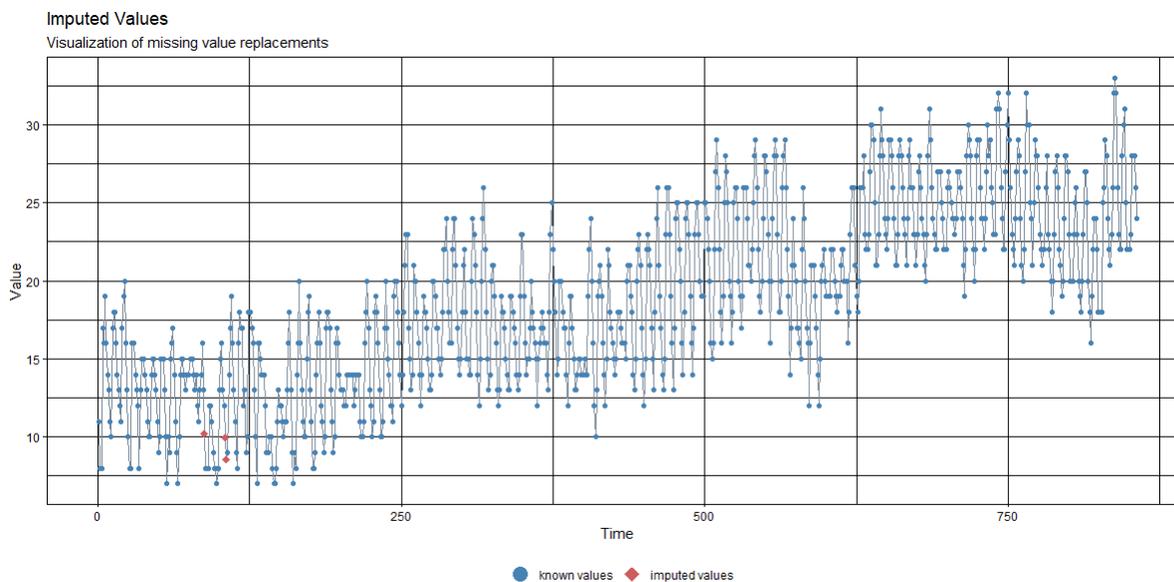


FIGURE 5 – Valeurs manquantes des températures ajoutées aux données météo France 2005 (Montpellier)

### 3 Modèles classiques de développement intra-stade

En écologie, la dynamique des populations est une branche qui s'intéresse au développement dans le temps d'individus au sein d'une population et aux influences de l'environnement sur les effectifs des populations. Dans cette partie, nous allons introduire certaines notions élémentaires de la dynamique des populations. En particulier, nous allons parler de modèles classiquement utilisés pour décrire la vitesse de développement en fonction de la température.

#### 3.1 Modèles de vitesse de développement en fonction de la température

La température est le principal moteur des fonctions clés des insectes, en particulier la survie, la reproduction, le mouvement et le développement. Parmi ces fonctions, le développement a une importance particulière en entomologie appliquée, car il s'agit d'un composant essentielle pour la construction de modèles phénologiques (Rebaudo et al. 2018). Ceux-ci peuvent être utilisés pour prédire les dates d'émergence, la distribution des épidémies et le voltinisme (nombre de génération par an) avec des implications pour la lutte antiparasitaire, l'épidémiologie, la science médico-légale ou encore les élevages de masse d'insectes.

Le développement en fonction de la température chez les insectes est classiquement représenté en utilisant l'inverse de la durée de développement, c'est-à-dire la vitesse de développement (Figure 7).

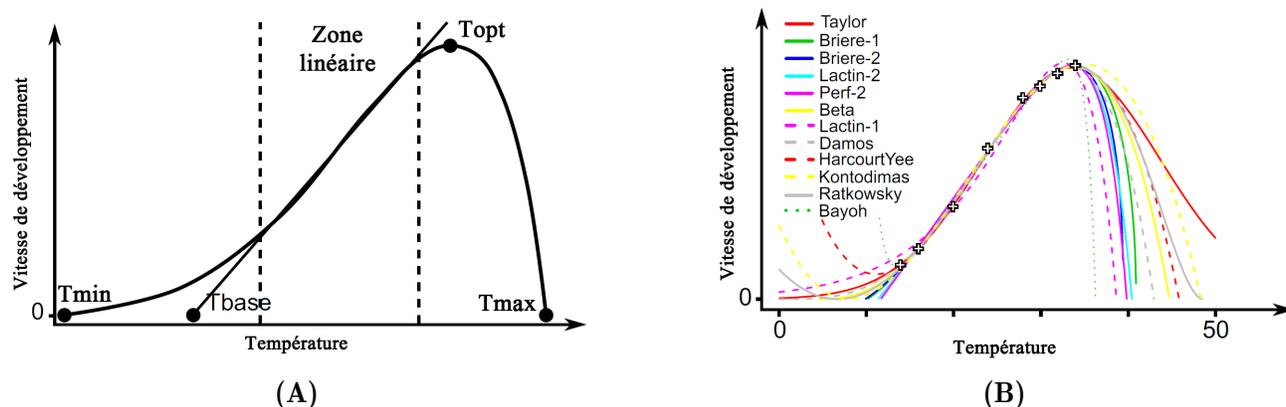


FIGURE 6 – Relation entre la température et la vitesse de développement chez les insectes. (A) Courbe de performance thermique théorique avec le minimum thermique critique ( $T_{min}$ ), le maximum ( $T_{max}$ ), l'optimal ( $T_{opt}$ ) et la température de base ( $T_{base}$ ). (B) Quelques modèles inférant la relation entre la température et la vitesse de développement, d'après (Rebaudo et al. 2018).

En théorie, le développement chez les insectes ne peut commencer qu'à partir d'un minimum thermique critique  $T_{min}$ . Au delà, la vitesse de développement augmente lentement à mesure que la température augmente. Il existe une plage de température où la vitesse de développement est presque linéaire. Si la température augmente encore, la vitesse de développement finit par atteindre une valeur optimale  $T_{opt}$ . Au-delà de cette valeur optimale, la vitesse de développement peut atteindre une valeur critique dite  $T_{max}$ , à partir de laquelle la vitesse de développement devient nulle.

De nombreux modèles ont été développés pour décrire l'effet de la température sur le développement des insectes. Une revue récente en recense les principaux (Rebaudo et al. 2018). Nous avons fait le choix d'en retenir trois parmi eux, essentiellement parce que les paramètres qu'ils estiment ont un sens biologique.

### a) Modèle de Taylor (1981)

Le modèle développé par Fitz Taylor avait pour objectif de modéliser la vitesse de développement de 54 espèces d'insectes, à partir de données sur des régimes de températures annuelles sur un gradient d'altitude et de latitude (Taylor 1981) :

$$R(T) = R_m e^{-\frac{1}{2}\left(\frac{T-T_{opt}}{T_o}\right)^2} \quad (3.1)$$

Où ;  $R(T)$  représente la vitesse de développement en fonction de la température  $T$ ,  $R_m$  la vitesse maximale de développement,  $T_{opt}$  la température optimale et  $T_o$  la vitesse à laquelle la vitesse de développement s'éloigne de  $T_m$ .

### b) Modèle de Brière (1999)

Le modèle de Brière est un modèle de régression non-linéaire qui a pour objectif de décrire la vitesse de développement en fonction de températures constantes de différents types d'insectes. Ce modèle a d'abord été utilisé pour modéliser la vitesse de développement de l'insecte *Lobesia botrana* puis sur différentes autres espèces d'insectes (Brière et al. 1999) :

$$R(T) = \begin{cases} aaT(T - T_{min})(T_{max} - T)^{(1/bb)} & \text{si } T \in [T_{min}, T_{max}] \\ 0 & \text{sinon} \end{cases} \quad (3.2)$$

Où ;  $R(T)$  représente la vitesse de développement en fonction de la température  $T$ ,  $T_{min}$  représente la température minimale,  $T_{max}$  la température maximale et les paramètres  $aa$  et  $bb$  représentent des constantes. À l'extérieur de l'intervalle  $[T_{min}, T_{max}]$ , la vitesse de développement est considérée comme nulle.

### c) Modèle Kontodimas (2004)

Le modèle de Kontodimas est inspiré de différents modèles liant la vitesse de développement à la température (Kontodimas et al. 2004). Dans leurs étude, les auteurs ont étudié la fécondité de deux prédateurs d'insectes, *Nephus includens* et *Nephus bisignatus* :

$$R(T) = \begin{cases} aa(T - T_{min})^2(T_{max} - T) & \text{si } T \in [T_{min}, T_{max}] \\ 0 & \text{sinon} \end{cases} \quad (3.3)$$

Où ;  $R(T)$  représente la vitesse de développement en fonction de la température  $T$ ,  $T_{min}$  représente la température minimale,  $T_{max}$  la température maximale et  $aa$  une constante.

## 3.2 Applications des modèles sur nos données

Nous avons appliqué les trois modèles définis ci-dessus (Brière, Kontodimas et Taylor) sur les données expérimentales de 2005 afin de modéliser l'effet des températures sur la vitesse de développement des différents stades de *C. pruni*. Pour les données obtenues en extérieur, la température correspond à la moyenne des températures journalières moyennes sur la durée de développement du stade considéré. Pour la deuxième expérience à température constante, la température est la température constante elle-même (10°C et 17°C). Les modèles ont été ajustés par la méthode des moindres carrés. Le code est disponible dans l'annexe A.

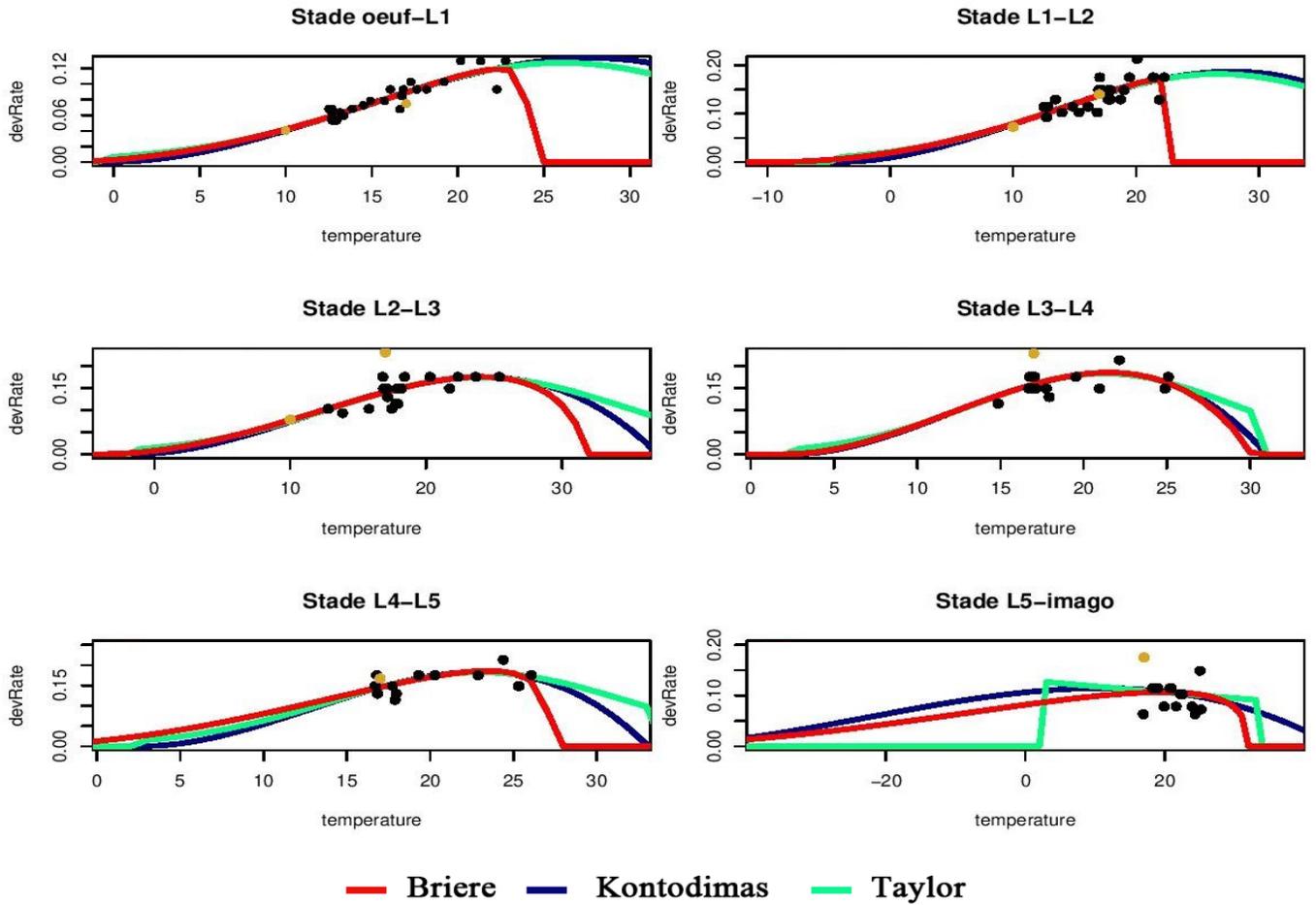


FIGURE 7 – Courbes des vitesses de développement des différents stades des psylles selon les trois modèles : Brière (rouge), Kontodimas (bleu) et Taylor (vert) sur les données de l’expérimentation 2005

La **Figure 7** représente les résultats obtenus avec les trois modèles (Brière, Kontodimas et Taylor) appliqués pour chacun des stades de développement. Les paramètres ont été estimés à partir des données en conditions réelles (points noirs) et à températures constantes (points jaunes (10°C et 17°C)).

Le **tableau 3** suivant contient, pour les différents stades de développement, les valeurs estimées des différents paramètres  $T_{min}$ ,  $T_{max}$  et  $T_{opt}$ .

Pour les modèles de Briere et Kontodimas,  $T_{opt}$  a été estimé de la manière suivante :

$$T_{opt} = \operatorname{argmax}_{t \in [T_{min}, T_{max}]} R(t)$$

Pour le modèle de Taylor, les valeurs de  $T_{min}$  et  $T_{max}$  correspondent respectivement à la première valeur de  $T$  pour laquelle  $R(T)$  est non nulle et à la dernière valeur de  $T$  pour laquelle  $R(T)$  est non nulle.

Stade oeuf-L1			
Modèle	$T_{min}$	$T_{max}$	$T_{opt}$
Briere	-3	24	22
Kontodimas	0	> 40	28
Taylor	0	> 40	26
Stade L1-L2			
Modèle	$T_{min}$	$T_{max}$	$T_{opt}$
Briere	-9	22	22
Kontodimas	-4	> 40	28
Taylor	-4	> 40	27
Stade L2-L3			
Modèle	$T_{min}$	$T_{max}$	$T_{opt}$
Briere	-4	31	24
Kontodimas	-2	37	24
Taylor	-2	37	24
Stade L3-L4			
Modèle	$T_{min}$	$T_{max}$	$T_{opt}$
Briere	3	30	22
Kontodimas	3	30	21
Taylor	3	30	21
Stade L4-L5			
Modèle	$T_{min}$	$T_{max}$	$T_{opt}$
Briere	-6	27	23
Kontodimas	3	33	23
Taylor	2	33	23
Stade L5-imago			
Modèle	$T_{min}$	$T_{max}$	$T_{opt}$
Briere	< -40	31	20
Kontodimas	< -40	> 40	11
Taylor	3	33	3

TABLE 3 – Valeurs de  $T_{min}$ ,  $T_{max}$  estimés et  $T_{opt}$  pour les trois modèles Brière, Kontodimas et Taylor

### 3.3 Discussion

- 1) En analysant nos résultats, on remarque tout d'abord que les trois modèles ne donnent pas les mêmes résultats. Autrement dit, ils ne s'ajustent pas de la même manière aux données. C'est un problème car pour le paramètre  $T_{max}$  en particulier, les valeurs estimées sont très différentes d'un modèle à l'autre (exemple : stade L1-L2 : 22, > 40 et > 40 pour les modèles de Brière, Kontodimas et Taylor).
- 2) Dans la **Figure 7**, on remarque que les 3 modèles sont très sensibles au manque de données pour les températures basses et hautes. Telle que l'expérience 2005 a été conçue, il n'a pas été possible de placer des larves L5 dans des conditions de températures basses voire hautes. Afin d'obtenir de meilleurs résultats, il faudrait enrichir les données en suivant la vitesse de développement des derniers stades sous des températures constantes entre 0°C et 15°C et entre 25°C et 35°C. Ces mesures pourraient être réalisées en cellule climatique, mais elles seraient très longues à réaliser.
- 3) Un inconvénient majeur des trois modèles que nous avons testé, comme pour tous les autres modèles de la littérature ([Rebaudo et al. 2018](#)), est que ces modèles sont basées sur des températures constantes. Ils ne tiennent donc pas compte des fluctuations des températures observable en conditions naturelles. En ce sens, ils ne nous paraissent pas assez réalistes.
- 4) Autre défaut majeurs de ces modèles est qu'ils permettent de modéliser ce qui se passent au sein de chaque stade, mais pas l'ensemble du processus de développement de l'œuf à l'adulte.

## 4 Un modèle de dynamique des populations adapté aux environnements fluctuants

### 4.1 Introduction

Dans la partie précédente, nous avons évoqué quelques modèles permettant de modéliser la vitesse de développement en fonction de la température et nous avons discuté de leurs limites. Notre objectif n'était pas seulement de modéliser indépendamment les vitesses de développement de chacun des stades, mais de simuler l'intégralité du processus de développement de l'insecte du stade œuf au stade imago. Ceci supposait donc de pouvoir aussi tenir compte des transitions entre les différents stades.

Notre objectif était donc triple pour être le plus réaliste possible :

- tenir compte de la variabilité naturelle des températures auxquelles les insectes sont soumis chaque jour et pendant tout leur développement ;
- tenir compte de la variabilité intra et inter stades des durées de développement ;
- tenir compte des variations des délais de passages entre les différents stades.

**Le modèle de projection intégrale d'une matrice de Lefkovitch (PIML)** est apparu comme une solution intéressante pour notre étude. David Pleydell, l'un de mes co-encadrements, avait encadré une thèse sur ce sujet ([Castano 2017](#)). L'idée était de transposer le modèle développé alors au cas d'un moucheron *Culicoïde* à notre cas d'étude sur des températures fluctuantes. En effet, dans la thèse, les paramètres du modèle n'ont été implémentés qu'avec des températures constantes.

Le modèle PIML présente l'intérêt de faciliter la paramétrisation avec des covariables variant dans le temps. Il fournit des probabilités de transition valides pour caractériser les distributions de stades de développements non stables et réduit les erreurs dans les analyses des délais de transitions entre stades.

Mais avant de définir le modèle **PIML** que nous avons développé ([section 4.3.2](#)), et de voir comment nous l'avons appliqué sur nos données, je vais commencer par définir deux types de modèles essentiels pour comprendre le modèle **PIML** :

- les modèles matriciels de populations ([section 4.2](#))
- les modèles de projections intégrales classiques ([section 4.3.1](#))

Ensuite, je décris aussi l'approche bayésienne utilisée pour estimer les paramètres du modèle PIML ([section 4.4.3](#)).

## 4.2 Modèles matriciels de population

Les modèles matriciels de population sont des outils très populaires en écologie pour décrire les cycles de vie et la dynamique des populations structurées (Tonnamg et al. 2017). Dans les modèles matriciels, les individus sont regroupés selon une gamme discrète d'états, et les probabilités de transition définissent la dynamique de passage d'un stade à l'autre.

Les états classiquement utilisés en écologie pour caractériser une population sont l'âge et/ou des stades de développement discrets tels que les stades de développement des insectes (œuf, larves et adulte dans notre cas d'étude). Des modèles plus ou moins complexes peuvent aussi inclure des informations sur le sexe, la génétique, l'aspect physique, la physiologie, l'état de santé, etc. Le modèle de Leslie (Leslie 1945) représente l'un des modèles les plus utilisés en dynamique des populations et le modèle de Lefkovitch (Lefkovitch 1965) représente une généralisation du modèle de Leslie.

### 4.2.1 Matrice de Leslie

Le modèle de Leslie (ou matrice de Leslie) (Leslie 1945) a pour but de décrire la dynamique d'une population animale composée de femelles reproductrices réparties dans  $k+1$  classes formées selon les âges.

A l'instant  $t$ , la connaissance des effectifs  $x_i(t)$  dans la classe numéro  $i$  (pour tout  $i \in [0, k]$ ) permet de prévoir ceux à l'instant  $t+1$  grâce au système suivant :

$$X_{t+1} = LX_t \quad (4.1)$$

où,  $X_t$  désigne le vecteur colonne à  $K+1$  lignes dont les composantes sont les  $x_i(t)$  pour tout  $t$ , tandis que  $L$  représente la matrice de Leslie qui est définie comme suit :

$$L = \begin{pmatrix} f_0 & f_1 & f_2 & \dots & f_{K-1} & f_K \\ s_0 & 0 & 0 & \dots & 0 & 0 \\ 0 & s_1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & s_{K-1} & 0 \end{pmatrix} \quad (4.2)$$

Sachant que les éléments de la première ligne représentent les fécondités :  $f_i$  correspond au nombre moyen de femelles mises au monde entre deux instants successifs  $t$  et  $t+1$  par chaque femelle de la classe  $i$ .

Les éléments diagonaux de la sous-matrice située sous la première ligne représentent les probabilités de survie. Ainsi,  $s_i$  représente la probabilité qu'une femelle appartenant à la classe numéro  $i$  au moment  $t$  soit vivante dans la classe numéro  $i+1$  à l'instant  $t+1$ .

### 4.2.2 Matrice de Lefkovitch

Alors que le modèle de Leslie structure une population par classes d'âge, le modèle de Lefkovitch (Lefkovitch 1965), qui est considéré comme une généralisation du modèle de Leslie, introduit une classification par stades de développement. Comme pour le modèle de Leslie, on a la formule :

$$N_t = M_t N_{t-1} \quad (4.3)$$

où,  $N_t$  représente un vecteur des densités pour une série de  $k$  stades de développement à l'instant  $t$  et  $M_t$  la matrice de Lefkovitch.

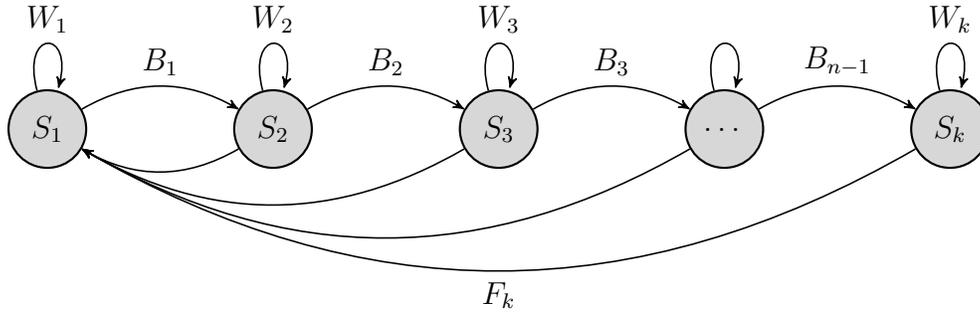


FIGURE 8 – Un exemple graphique d'un cycle de vie :  $W_S$  représente la probabilité de rester dans ce même stade  $s$ ,  $B_S$  la probabilité de passer au prochain stade,  $F_S$  la contribution à la prochaine génération (fécondité).

Au cycle de développement schématisé sur la figure 9, correspond alors la matrice suivante :

$$M = \begin{pmatrix} W_1 & F_2 & F_3 & \cdots & F_k \\ B_1 & W_2 & 0 & \cdots & 0 \\ 0 & B_2 & W_3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & B_{n-1} & W_k \end{pmatrix} \quad (4.4)$$

où :

- $M$  représente la matrice de Lefkovitch ;
- pour n'importe quel stade  $s \in \{1, \dots, k\}$ ,  $W_S$  représente la probabilité de rester dans ce même stade  $s$  (intra-stade) ;
- $B_S$  la probabilité de passer au prochain stade (inter-stade) ;
- $F_S$  la contribution à la prochaine génération (fécondité).

Un défaut majeur de ce type de modèle matriciel est qu'il donne une mauvaise approximation de la variation des durées de développement au sein de chacun des stades considérés. En effet, le modèle matriciel de Lefkovitch impose que les temps de séjour suivent une distribution géométrique. La moyenne de séjour peut être ajustée aux données, mais la variance sera biaisée par la

contrainte qui est la distribution géométrique. Ainsi, les dynamiques transitoires sont fortement affectées par de mauvaises approximations de la variation des temps de séjour (Castano 2017). Une alternative aux modèles matriciels est l'utilisation des modèles de projections intégrales (MPI).

### 4.3 Modèles de projections intégrales

Pour tenir compte de l'hétérogénéité de la durée de développement de chacun des stades, l'idée développée par (Castano 2017) a été de remplacer les probabilités d'une matrice de Lefkovich classique par des modèles dits de projections intégrales (MPI) spécifiques à chaque stade, ce qui permet ainsi de suivre la dynamique interne de chaque stade.

#### 4.3.1 Modèles de projections intégrales classiques (MPI)

Le principe des modèles de projections intégrales classiques est de considérer pour chaque individu une variable  $x$  continue nommée caractéristique (par exemple : le poids). Le vecteur de population est remplacé par une fonction  $n(x, t)$ , où  $n(x, t)dx$  est le nombre d'individus à un instant  $t$  dont la caractéristique est comprise entre  $[x, x + dx]$  (par exemple entre 82kg et 85kg). On définit alors une fonction de noyau de développement comme suit :

$$K(y, x) = P(y, x) + F(y, x) \tag{4.5}$$

où,  $P$  représente la proportion d'individus qui passe de la caractéristique  $x$  à la caractéristique  $y$ , et  $F$  représente le nombre total de descendants ayant la caractéristique  $y$  produit par chaque femelle qui présente la caractéristique  $x$ .

La dynamique de population est alors définie par la formule suivante :

$$n(y, t + 1) = \int_L^U K(y, x)n(x, t)dx \tag{4.6}$$

où  $[L, U]$  est l'intervalle de toutes les caractéristiques possibles.

Définies pour des caractéristiques continues, les discrétisations de la fonction de noyau permettent de traiter les modèles de projections intégrales comme des modèles matriciels à haute résolution.

Néanmoins, une différence entre les modèles matriciels classiques et les modèles de projections intégrales classiques est que les modèles matriciels utilisent généralement des observations pour paramétrer directement les probabilités de transition (ce qui est leur principale faiblesse lors de l'augmentation de la dimension de la matrice pour tenir compte des sous-stades de développement) alors que les modèles de projection intégrales classiques déduisent les probabilités de transition sur la base d'une régression décalée dans le temps de  $x$  pour aboutir à une estimation du noyau  $K$  (Easterling et al. 2000).

Ainsi, un avantage majeur des modèles de projections intégrales est le traitement statistique naturel de la variation et de l'incertitude.

### 4.3.2 Modèle de projection intégrale d'une matrice de Lefkovitch (PIML) pour les insectes

Un inconvénient des modèles de projections intégrales classiques est qu'ils sont généralement utilisés pour caractériser la dynamique des variables facilement mesurables, tel que le poids ou la taille par exemple. Pour les insectes, le principal facteur qui joue n'est pas seulement la température mais plutôt la quantité de chaleur (ou accumulation thermique) nécessaire pour achever un stade de développement. Cette quantité de chaleur ne peut pas être mesurée directement, ce qui rend difficile l'application des modèles MPI dans ce cas. Pour répondre à ce défi, Castano (Castano 2017) a proposé une méthode qui tire parti des modèles de projections intégrales pour suivre le développement intra stade en fonction de l'accumulation thermique. Dans cette thèse, Castano a estimé les paramètres de son modèle sur des températures constantes. En théorie, il est possible d'utiliser la même approche sur des températures fluctuantes. C'est précisément ce que j'ai cherché à faire au cours de mon stage.

Avant de détailler ma démarche, je vais faire une introduction générale du modèle PIML pour le développement des insectes, puis dans la section 4.4, je présenterai comment j'ai estimé les paramètres de modèle avec des températures fluctuantes.

On définit un modèle de projections intégrales pour le développement intra-stade comme suit :

$$n(\delta', t) = \int_0^1 K_t(\delta', \delta) n(\delta, t-1) d\delta \quad (4.7)$$

où,  $n(\delta, t)$  est la densité d'individus ayant un degré de développement  $\delta$  en  $t$  et le noyau  $K_t(\delta', \delta)$  quantifie la proportion d'individus avec un développement  $\delta$  qui survivent et se développent jusqu'à  $\delta'$  en un pas de temps  $t$ .

Le degré de développement  $\delta$  est une représentation de la proportion de développement des individus, avec  $\delta = 0$  représente le degré de développement au début d'une expérience et  $\delta = 1$  l'accomplissement du processus de développement. Ce modèle peut être appliqué à un ou plusieurs stades de développement ou tout un cycle de vie.

Pour les insectes, le noyau de projection doit prendre en compte trois événements :

$$K(y, x) = W(y, x) + B(y, x) + F(y, x) \quad (4.8)$$

où,  $W$  et  $B$  caractérisent le développement intra-stade et le développement inter-stade et  $F$  la fécondité.

Les modèles de projections intégrales sont décrits avec des variables caractéristiques continus. En pratique, il est donc nécessaire d'effectuer discrétisation de la fonction de noyau par une matrice de Lefkovitch que Castano (Castano 2017) a défini comme suit :

$$\begin{bmatrix} \mathbf{n}_1 \\ \mathbf{n}_2 \\ \mathbf{n}_3 \\ \vdots \\ \mathbf{n}_k \end{bmatrix}_t = \begin{bmatrix} \mathbf{W}_1 & \mathbf{F}_2 & \mathbf{F}_3 & \cdots & \mathbf{F}_k \\ \mathbf{B}_1 & \mathbf{W}_2 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_2 & \mathbf{W}_3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{B}_{k-1} & \mathbf{W}_k \end{bmatrix}_t \begin{bmatrix} \mathbf{n}_1 \\ \mathbf{n}_2 \\ \mathbf{n}_3 \\ \vdots \\ \mathbf{n}_k \end{bmatrix}_{t-1} \quad (4.9)$$

où, la matrice représente la nouvelle matrice de Lefkovitch que l'on nomme  $M_t$ . La différence entre cette matrice et celle définie précédemment (4.4) est que l'on considère  $\mathbf{W}_S$  et  $\mathbf{B}_S$  comme étant des sous matrices qui caractérisent le développement intra-stade et le développement inter-

stade respectivement et  $\mathbf{0}$  sont des matrices nulles qui correspondent avec les dimensions des matrices  $\mathbf{W}_S$  et  $\mathbf{B}_S$ .

On rappelle que :

$$\mathbf{N}_t = \mathbf{M}_t \mathbf{N}_{t-1} \quad (4.10)$$

- $\mathbf{N}_t = [\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_k]$  représente une série de vecteurs représentent le développement des individus dans chaque stade;
- $r_S$  le nombre de sous-stades de développement à l'intérieur d'un stade  $S$ , c'est la résolution de la matrice qui discrétise  $\delta$ ;
- les sous-matrices  $\mathbf{W}_S$  et  $\mathbf{B}_S$  sont définies par des modèles de projections intégrale (MPI) à chaque stade;
- le degré de développement  $\delta$  est remplacé par une série de sous-stades discrets  $r_S$  pour chaque étape  $S$ . Les individus commencent chaque stade dans le premier sous-stade discret. Ensuite, ils quittent le stade après avoir terminé les  $r_S$  étapes de développement. Il s'agit ici d'hypothèses du modèle : nous supposons que le développement ne fait que progresser.
- ces probabilités de transition fournissent, pour un stade  $S$ , des éléments pour la matrice triangulaire inférieure  $r_S \times r_S$  suivante :

$$\mathbf{W}_S = \nu_S \begin{bmatrix} p_{0,S} & 0 & 0 & \cdots & \vdots \\ p_{1,S} & p_{0,S} & 0 & \cdots & \vdots \\ p_{2,S} & p_{1,S} & p_{0,S} & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 \\ p_{r-1,S} & p_{r-2,S} & \cdots & \cdots & p_{0,S} \end{bmatrix} \quad (4.11)$$

où les probabilités  $\{p_{0,S}, \dots, p_{r,S}\}$  dépendent de paramètres de développement spécifiques à chaque stade de développement et  $\nu_S$  représente la probabilité de survie qui est spécifique à chaque stade.

- La matrice  $\mathbf{B}_S$  fournit la proportion d'individus passant à l'étape suivante, où le développement est initialisé dans le premier sous-stade. On définit  $\mathbf{B}_S^1$  comme étant la première ligne de la matrice  $\mathbf{B}_S$  avec l'élément  $j$  de  $\mathbf{B}_S^1$  égale à  $\sum_{l=r+1-j}^r p_{l,s}$  et tous les autres éléments de la matrice  $\mathbf{B}_S$  sont nuls. Alors, la matrice  $\mathbf{B}_S$  est définie comme suit :

$$\mathbf{B}_S = \nu_S \begin{bmatrix} p_{r,S} & p_{r,S} + p_{r-1,S} & \cdots & \sum_{l=1}^r p_{l,s} \\ 0 & 0 & \cdots & \vdots \\ 0 & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & 0 \end{bmatrix} \quad (4.12)$$

- Pour obtenir les probabilités  $p_{0,S} \cdots p_{r-1,S}$  d'une façon parcimonieuse, [Castano \(2017\)](#) a proposé d'utiliser des fonctions de distribution cumulée  $F(\Delta|\mu, \sigma)$  et des fonctions de densité  $f(\Delta|\mu, \sigma)$ . Plus précisément, Castano a définie la probabilité  $p_t$  de compléter  $l$  accroissements discrets dans un pas de temps de la façon suivante :

$$p_l = F\left(\frac{l+1}{r+1} \middle| \mu, \sigma\right) - F\left(\frac{l}{r+1} \middle| \mu, \sigma\right) \quad (4.13)$$

où,  $F(\Delta|\mu, \sigma)$  est la fonction de distribution cumulée associée à  $f(\Delta|\mu, \sigma)$  avec  $\Delta$  le taux de développement intra-stade.

$\mu$  et  $\sigma$  représentent respectivement la moyenne et l'écart-type de la distribution  $f(\Delta|\mu, \sigma)$   $f(\Delta|\mu, \sigma)$  est la fonction de densité associé à  $F(\Delta|\mu, \sigma)$ .

### 4.3.3 Exemple d'implémentation de modèle de projection sur un seul stade

Nous allons voir ici un exemple d'implémentation d'un modèle de projection intégrale de base (MPI) tiré de la thèse de [Castano \(2017\)](#) pour la dynamique intra-stade avec une distribution de loi bêta pour définir la variation de développement.

Dans cette exemple, l'approximation matricielle du modèle de projections intégrales pour un seule stade est définie comme suit :

$$\begin{bmatrix} \mathbf{n} \\ c \end{bmatrix}_t = \begin{bmatrix} \mathbf{W} & 0 \\ \mathbf{B}^1 & 1 \end{bmatrix}_t \begin{bmatrix} \mathbf{n} \\ c \end{bmatrix}_{t-1} \quad (4.14)$$

où, pour chaque  $t$ ,  $\mathbf{n}$  donne la distribution de la densité d'une population dans  $r$  sous-stades et  $c$  est la densité cumulée d'individus ayant terminé le stade.  $\mathbf{N}_0$  est initialisé à  $[1, 0, 0, \dots, 0]$ .

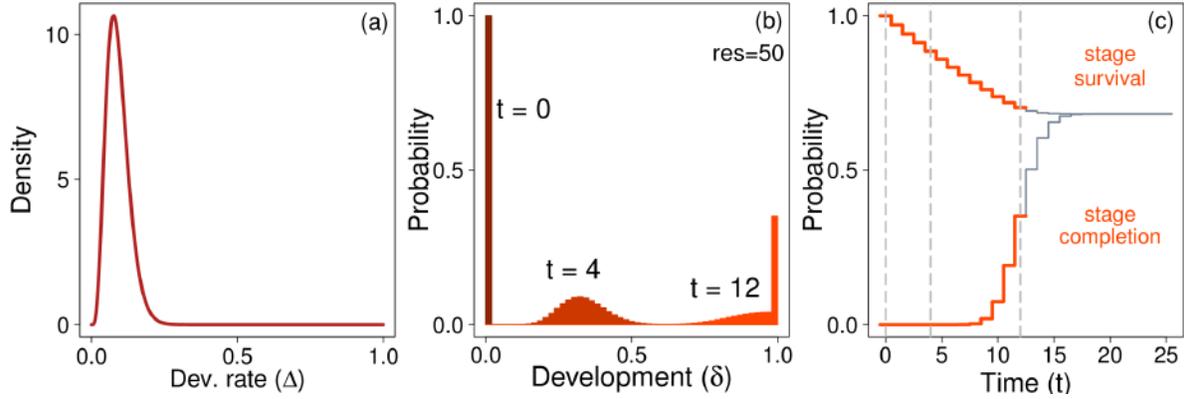


FIGURE 9 – Implémentation d'un modèle de projection intégrale de base (MPI) pour la dynamique intra-stade (d'après : [\(Castano 2017\)](#)).

- (a) Fonction de densité  $f$  utilisé pour définir les éléments des matrices  $\mathbf{W}$  et  $\mathbf{B}$  (Équations :).
- (b) Projection du développement dans une cohorte représenté à  $t = 0$ ,  $t = 4$  et  $t = 12$ .
- (c) Probabilités de survie et de compléter un stade en fonction du temps.

Dans cet exemple, le noyau MPI  $K$  est défini comme le produit d'une probabilité de survie spécifique au stade  $\nu = 0.97$  et d'une fonction de densité de probabilité pour les incréments de développement. La fonction de densité est  $beta(5, 50)$ . Sur la figure :

- (a) : représente la fonction  $f$  qui définit les matrices de développements intra-stades et inter-stades  $\mathbf{W}$  et  $\mathbf{B}$  (4.10) (4.11) ;
- (b) : représente l'évolution (à  $t = 0$ ,  $t = 4$  et  $t = 12$ ) de la distribution du développement des individus au sein du stade ;

- **(c)** : représente le cumule des probabilités de survie et le cumule des probabilités de compléter le stade en fonction du temps au sein du stade. Les lignes pointillées correspondent aux distributions de développement à  $t = 0$ ,  $t = 4$  et  $t = 12$ .

#### 4.4 Implémentation du modèle PIML sur les données

Dans cette partie, je vais présenter la méthodologie derrière l'implémentation du modèle PIML sur les données mises à ma disposition. L'implémentation du modèle a pour objectifs de modéliser l'évolution intra-stade des psylles ainsi que les transitions inter-stades. Le cycle de vie des psylles est défini schématiquement comme suit :

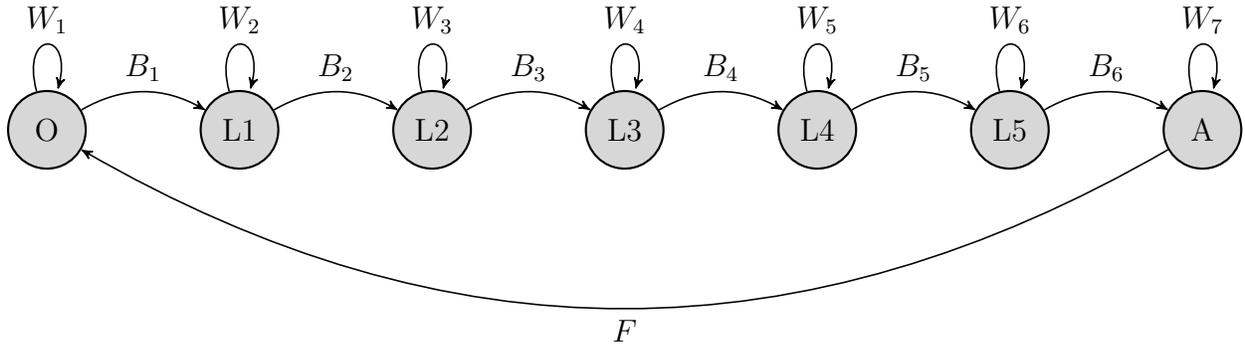


FIGURE 10 – Représentation graphique d'un cycle de vie simplifié de l'espèce *Cacopsylla pruni* qui fournit une base pour le développement d'un PIML : A chaque pas de temps, les matrices **W** représentent la proportion d'individus restant dans une classe de stade donnée, les matrices **B** représentent la proportion d'individus passant à la classe de stade suivante, et **F** représente la fécondité.

Dans le cas de notre étude, on a exclu le facteur fécondité et donc on considère les matrices **F** comme nulles. La matrice  $M_t$  devient :

$$\begin{bmatrix} \mathbf{n}_1 \\ \mathbf{n}_2 \\ \mathbf{n}_3 \\ \mathbf{n}_4 \\ \mathbf{n}_5 \\ \mathbf{n}_6 \\ n_7 \end{bmatrix}_t = \begin{bmatrix} \mathbf{W}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{B}_1 & \mathbf{W}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_2 & \mathbf{W}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}_3 & \mathbf{W}_4 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_4 & \mathbf{W}_5 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_5 & \mathbf{W}_6 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_6^1 & 1 \end{bmatrix}_t \begin{bmatrix} \mathbf{n}_1 \\ \mathbf{n}_2 \\ \mathbf{n}_3 \\ \mathbf{n}_4 \\ \mathbf{n}_5 \\ \mathbf{n}_6 \\ n_7 \end{bmatrix}_{t-1} \quad (4.15)$$

avec **0** sont des matrices nulles avec des dimensions qui correspondent à celles des matrices  $\mathbf{W}_S$  et  $\mathbf{N}_0$  est initialisé à  $[1, 0, 0, \dots, 0]$ .

#### 4.4.1 Description de l'hétérogénéité des vitesses de développement

Le choix d'une fonction  $f(\Delta)$ , pour décrire l'hétérogénéité des vitesses de développement (représentée par les incréments) est flexible. Une option aurait été de choisir une distribution beta (Castano 2017). Nous avons choisi l'option d'utiliser des distributions log-normales. En effet :

- Les distributions log-normales ont deux paramètres (la moyenne et l'écart-type de  $\log(\Delta)$  décrits ici par  $\mu_l$  et  $\sigma_l$ ), donc offrent un compromis entre flexibilité et parcimonie ;
- Elles garantissent que la vitesse de développement est positive ;
- Les distributions log-normales sont des choix populaires dans la littérature (Goodsman et al. 2018), (Briere et al. 1999) .

Nous avons travaillé avec une reparamétrisation de la distribution log-normale. Nous avons choisi pour notre paramétrisation la moyenne et l'écart-type de  $\Delta$ . Nous avons fait cette reparamétrisation car ces paramètres sont plus facilement interprétables d'un point de vue biologique.

Ainsi, la variation individuelle des vitesses de développement est modélisée par la fonction de densité de probabilité :

$$f(\Delta|\mu, \sigma) = \frac{1}{\Delta\sigma_l\sqrt{2\pi}} \exp\left(-\frac{(\ln \Delta - \mu_l)^2}{2\sigma_l^2}\right) \quad (4.16)$$

où :

$$\mu_l = \log(\mu) - \frac{1}{2}\log\left(\frac{\sigma^2 + \mu^2}{\mu^2}\right) \quad , \quad \sigma_l = \sqrt{\log\left(\frac{\sigma^2 + \mu^2}{\mu^2}\right)} \quad (4.17)$$

et  $\mu$  et  $\sigma$  sont respectivement la moyenne et l'écart-type de la distribution, et  $\mu_l$  et  $\sigma_l$  sont les paramètres canoniques à l'échelle logarithme de la distribution (le sous-scripte  $l$  représente l'échelle logarithmique).

#### 4.4.2 Lien entre les températures fluctuantes et le modèle PIML

Dans un premier temps, le modèle de Brière a été choisi pour définir la moyenne des vitesses de développement des différents stades mais un problème est survenu. En effet, les paramètres  $aa$  et  $bb$  dans le modèle de Brière sont fortement corrélés, ce qui conduit à un échantillonnage MCMC (méthode de Monte-Carlo par chaînes de Markov) peu efficace.

Pour rendre  $aa$  et  $bb$  indépendants, une solution serait de diviser la formule de Brière évaluée à  $T$  par la même formule évaluée à  $T_{opt}$  mais cette équation est impossible à résoudre algébriquement. La solution a été d'effectuer une transformation linéaire des températures selon la formule :

$$T' = \frac{T - T_{min}}{T_{max} - T_{min}} \quad (4.18)$$

Ce paramétrage conduit à la nouvelle fonction que nous appellerons **fonction de Brière standardisée**. Cette fonction est disponible sur le package *nimbleTempDev* sous le noms de la fonction *stBriere* :

$$R(T) = \alpha \frac{T'^2(1 - T')^{\frac{1}{B}}}{\kappa^2(1 - \kappa)^{\frac{1}{B}}} \quad (4.19)$$

où :

- $\alpha$  représente l'amplitude de la courbe ;
- $\kappa$  le paramètre représentant la forme de la courbe ;
- $T_{opt}$  est situé à  $T_{min} + (T_{max} - T_{min})\kappa$
- $B$  équivaut au paramètre  $bb$  du modèle de Brière classique (3.2) avec  $B = \frac{\kappa}{2(1-\kappa)}$ .

Le nouveau modèle de Brière standardisé est utilisé afin de modéliser la vitesse de développement moyenne de chaque stade :

$$\mu|T = R(T; T_{min}, T_{max}, \alpha, \kappa) \quad (4.20)$$

Nous avons ensuite défini différentes fonctions alternatives pour modéliser l'écart-type de  $f(\Delta)$  en fonction des températures. Dans la littérature, il existe très peu d'exemples de ce genre. En effet, l'objectif majeur de modélisation est souvent la moyenne de  $f(\Delta)$ , généralement, tous les effets de la température sur l'écart-type sont négligés pour des raisons de simplicité, alors cette partie de mon travail été exploratoire et nouvelle :

SDmodel <sub>1</sub>	$\sigma T = R(T) \times \exp(\beta_0)$
SDmodel <sub>2</sub>	$\sigma T = \mathbf{1}_{R(T)>0} \times \exp(\beta_0)$
SDmodel <sub>3</sub>	$\sigma T = R(T) \times \exp(\beta_0 + \beta_1 T)$
SDmodel <sub>4</sub>	$\sigma T = \mathbf{1}_{R(T)>0} \times \exp(\beta_0 + \beta_1 T)$
SDmodel <sub>5</sub>	$\sigma T = R(T) \times \exp(\beta_0 + \beta_1 T + \beta_2 T^2)$
SDmodel <sub>6</sub>	$\sigma T = \mathbf{1}_{R(T)>0} \times \exp(\beta_0 + \beta_1 T + \beta_2 T^2)$

TABLE 4 – Liste des modèles utilisés pour caractériser l'écart-type.

où,  $R(T)$  représente la vitesse de développement en fonction de la température définie avec la nouvelle fonction de Brière standardisée (4.18).

Nous avons défini ces modèles de cette façon pour les raisons suivantes. Nous avons imposé que  $\sigma|T = 0$  quand  $R(T) = 0$  parce que c'est une contrainte logique sachant que  $\Delta$  doit être non-négative. Les modèles que nous avons définis sont simples et positives pour tout  $T$  pour lequel  $R(T) > 0$

#### 4.4.3 Estimation des paramètres dans les modèles de population à l'aide d'une approche Bayésienne

Nous avons estimé les paramètres  $T_{min}, T_{max}, \alpha, \kappa, \beta_0, \beta_1, \beta_2$  par une approche Bayésienne :

- Les paramètres  $\beta_0, \beta_1$  et  $\beta_2$  sont des paramètres des différents modèles alternatifs pour modéliser l'écart-type. Pour ces paramètres nous avons choisi les priors suivants :

$$\beta_0 \sim \text{Normale}(0, 10^{11})$$

$$\beta_1 \sim \text{Double exp}(\beta)$$

$$\beta_2 \sim \text{Double exp}(\beta)$$

où  $\beta$  suit une loi gamma :  $\beta \sim \text{gamma}(\frac{1}{\text{nombre de stades}=6}, 2)$  pour apporter un rétrécissement (shirnkage) pour  $\beta_1$  et  $\beta_2$  (Bhattacharya et al. 2015), et le prior sur  $\beta_0$  est très peu informative.

- Pour les paramètres de températures minimale et maximale,  $T_{min}$  et  $T_{max}$ , nous avons attribué des lois uniformes avec des bornes très distantes de valeurs typiquement retrouvées par les entomologistes :

$$\begin{aligned} T_{min} &\sim Uniforme[-40, 20] \\ T_{max} &\sim Uniforme[20, 80] \end{aligned}$$

Nous avons choisi ces lois parce qu'ils sont très peu informatives.

- Pour le paramètres  $\alpha$  et  $\kappa$  du modèle de Brière reparamétrisé nous avons choisi les loi bêta non-informatives suivantes :

$$\begin{aligned} \alpha &\sim beta[1, 1] \\ \kappa &\sim beta[1, 1] \end{aligned}$$

Pour la vraisemblance, nous avons fait le choix d'une loi multinomiale, (qui est une généralisation d'une loi binomiale), pour chacun des différents stades que nous souhaitions modéliser. Sa fonction de masse est définie comme tel :

$$f(y|\mathbf{T}_{min}, \mathbf{T}_{max}, \alpha, \kappa, \beta_0, \beta_1, \beta_2) = \prod_{c \in \text{cohortes}} \prod_{\omega \in \text{obs}(c)} \frac{\sum_{S=1}^7 y_{S(c,\omega)}!}{y_{1(c,\omega)}! \cdots y_{7(c,\omega)}!} \pi_{1,c,\omega}^{y_{1(c,\omega)}} \cdots \pi_{7,c,\omega}^{y_{7(c,\omega)}} \quad (4.21)$$

où :

- cohortes : représente toutes les cohortes observées lors de l'étude faite en 2005 (33 cohortes ou plantes) ;
- obs(c) : représente toutes les observations dans chaque cohorte ;
- $\pi_{S,c,\omega} = \sum_{i=1}^r n_{i,S,c,\omega}$  représente la probabilité conditionnelle pour la cohorte  $c$  d'appartenir au stade  $S$  à l'observation  $\omega$ .
- $n_{i,S,c,\omega}$  représente la probabilité conditionnelle pour la cohorte  $c$  d'appartenir au sous-stade  $i$  du stade  $S$  à l'observation  $\omega$ .

$\mathbf{T}_{min}, \mathbf{T}_{max}, \alpha, \kappa, \beta_0, \beta_1, \beta_2$  sont des vecteurs qui prennent six valeurs (le nombre de stades pour lesquels on modélise le développement intra-stade).

Sachant toutes les informations données précédemment dans ce chapitre, la densité postérieur pour chacun des modèles (SDmodel<sub>5</sub> et SDmodel<sub>6</sub>), comprenant tous les paramètres, est calculée dans la forme non-normalisée suivante :

$$\pi(\mathbf{T}_{min}, \mathbf{T}_{max}, \alpha, \kappa, \beta_0, \beta_1, \beta_2 | y) \propto L(y|\mathbf{T}_{min}, \mathbf{T}_{max}, \alpha, \kappa, \beta_0, \beta_1, \beta_2) \pi(\mathbf{T}_{min}, \mathbf{T}_{max}, \alpha, \kappa, \beta_0, \beta_1, \beta_2)$$

où :

- $\pi(\mathbf{T}_{min}, \mathbf{T}_{max}, \alpha, \kappa, \beta_0, \beta_1, \beta_2)$  représente le produit de nos priors ;
- $L(y|\mathbf{T}_{min}, \mathbf{T}_{max}, \alpha, \kappa, \beta_0, \beta_1, \beta_2)$  notre vraisemblance multinomiale ;
- $\pi(\mathbf{T}_{min}, \mathbf{T}_{max}, \alpha, \kappa, \beta_0, \beta_1, \beta_2 | y)$  nos posteriors.

Pour les modèles SDmodel<sub>1</sub>, SDmodel<sub>2</sub>, SDmodel<sub>3</sub> et SDmodel<sub>4</sub> des posteriors équivalents ont été obtenus avec moins de paramètres  $\beta$  pour modéliser  $\sigma|T$ .

## Échantillonnage MCMC

Afin d'estimer les paramètres de ces modèles, nous avons utilisé des algorithmes de Monte-Carlo par Chaines de Markov (MCMC). Sachant que les posteriors sont non-normalisés, la méthode de Metropolis-Hastings est adaptée. Nous avons utilisé le package *nimbleAPT* [D](#) pour construire un seul échantillonneur en block. Nous avons utilisé un simple indice de convergence pour établir la fin des phases de "burn in" – cet indice utilise un test de t pour indiquer quand la moyenne des vraisemblances à posteriori de deux chaînes consécutives n'augmente plus. Nous effectuons ensuite 100000 itérations avec un "thinning" de 10 (garder un échantillon sur 10). Sachant que ces MCMC ont besoin de plusieurs semaines de calculs, nous avons effectué ces calculs sur le cluster de **Migale** de l'INRAE <http://migale.jouy.inra.fr/>. Ensuite, nous avons effectué des tests de convergences avec le package *coda*.

## 4.5 Résultats

Bien que nous avons réussi à estimer les paramètres des six modèles, nous allons ici détailler les résultats du premier modèle  $SDmodel_1$  et nous renvoyons à l'annexe F pour les autres modèles.

### Tableau récapitulatif des valeurs estimées

Pour les 1000 échantillons obtenus par MCMC, nous avons obtenu les estimations des paramètres suivantes :

	statistics.Mean	statistics.SD	statistics.Naive.SE	statistics.Time.series.SE	quantiles.2.5	quantiles.25	quantiles.50	quantiles.75	quantiles.97.5
Tmax.1.	67.79	8.81	0.09	1.27	50.83	60.93	69.74	75.6	79.56
Tmax.2.	60.07	12.85	0.13	1.55	34.11	50.08	61.43	70.73	79.36
Tmax.3.	26.56	0.36	0	0.04	26.04	26.29	26.54	26.75	27.62
Tmax.4.	56.7	13.11	0.13	1.72	33.86	45.5	56.49	67.99	78.48
Tmax.5.	40.08	13.85	0.14	2.85	26.54	28.34	34.7	49.45	73.38
Tmax.6.	53.54	16.38	0.16	2.96	24.36	40.27	54.94	67.64	78.68
Tmin.1.	-17.22	3.88	0.04	0.46	-28.63	-19.2	-16.45	-14.42	-11.81
Tmin.2.	-11.82	1.36	0.01	0.15	-14.42	-12.73	-11.87	-10.86	-9.2
Tmin.3.	-17.19	3.22	0.03	0.36	-24.42	-19.18	-17.15	-15.01	-11.12
Tmin.4.	-21.29	9.89	0.1	1.4	-38.33	-29.17	-21.02	-14.09	-1.44
Tmin.5.	6.34	8.16	0.08	2.18	-22.11	4.33	8.6	11.86	14.1
Tmin.6.	-2.82	15.58	0.16	2.14	-34.99	-14.05	-1.43	11.02	18.89
beta0.1.	-0.17	0.03	0	0	-0.23	-0.19	-0.17	-0.15	-0.12
beta0.2.	-0.92	0.1	0	0.01	-1.14	-0.98	-0.92	-0.86	-0.71
beta0.3.	-1.04	0.14	0	0.02	-1.34	-1.13	-1.03	-0.95	-0.76
beta0.4.	-1.65	0.48	0	0.07	-2.71	-1.94	-1.57	-1.3	-0.89
beta0.5.	-7536.38	4560.26	45.6	888.72	-16919.39	-11118.8	-6775.85	-3653.51	-571.18
beta0.6.	15.09	3.12	0.03	0.44	8.45	13.09	15.5	17.56	19.9
α1	0.03	0	0	0	0.03	0.03	0.03	0.03	0.03
α2	0.24	0.08	0	0.01	0.1	0.17	0.23	0.3	0.39
α3	0.07	0	0	0	0.06	0.07	0.07	0.07	0.07
α4	0.04	0	0	0	0.04	0.04	0.04	0.05	0.05
α5	0.06	0.01	0	0	0.04	0.05	0.06	0.07	0.08
α6	0.53	0.29	0	0.03	0.03	0.29	0.55	0.78	0.97
κ1	0.57	0.08	0	0.01	0.45	0.51	0.56	0.61	0.78
κ2	0.98	0.01	0	0	0.95	0.97	0.99	0.99	1
κ3	0.99	0.01	0	0	0.97	0.99	0.99	1	1
κ4	0.62	0.12	0	0.01	0.39	0.52	0.61	0.7	0.85
κ5	0.51	0.26	0	0.06	0.11	0.27	0.52	0.74	0.94
κ6	0.67	0.24	0	0.03	0.11	0.49	0.71	0.88	0.98

FIGURE 11 – Synthèse des estimations obtenus par MCMC pour les paramètres de  $SDmodel_1$  pour les six stades de développement. Ce tableau montre la moyenne (Mean), l'écart types (SD) et des quantiles pour tous les paramètres. L'erreur standard (SE) a été estimée de deux manières, une prenant en compte l'auto-corrélation (Time.series) et l'autre non (Naive).

Ce tableau fait ressortir cinq points essentiels :

- Les températures estimés  $T_{min}$  des différents stades de développement sont toutes largement en dessous de ( $0^{\circ}C$ ) sauf pour le cinquième stade.
- Les températures  $T_{max}$  des différents stades de développement sont très hautes pour tous les stades à l'exception du troisième (stade L2 - L3).
- On remarque que les valeurs de  $\beta_0$  pour les stades cinq et six sont très différentes de celles des autres stades. Nous remarquons aussi que la valeur (Time.Serie.SE) est très élevée.
- Pour le paramètre  $\kappa$ , le niveau d'incertitude est très hétérogène (les valeurs des quantiles 1.25 et 2.5 sont très variables)
- Pour le paramètre  $\alpha$ , les valeurs  $\alpha_2$  et  $\alpha_6$  sont très supérieurs aux valeurs  $\alpha$  des autres stades.

;

## Courbes de vitesses de développements

La figure suivante représente les estimations des vitesses de développement moyennes des différents stades en fonction des températures :

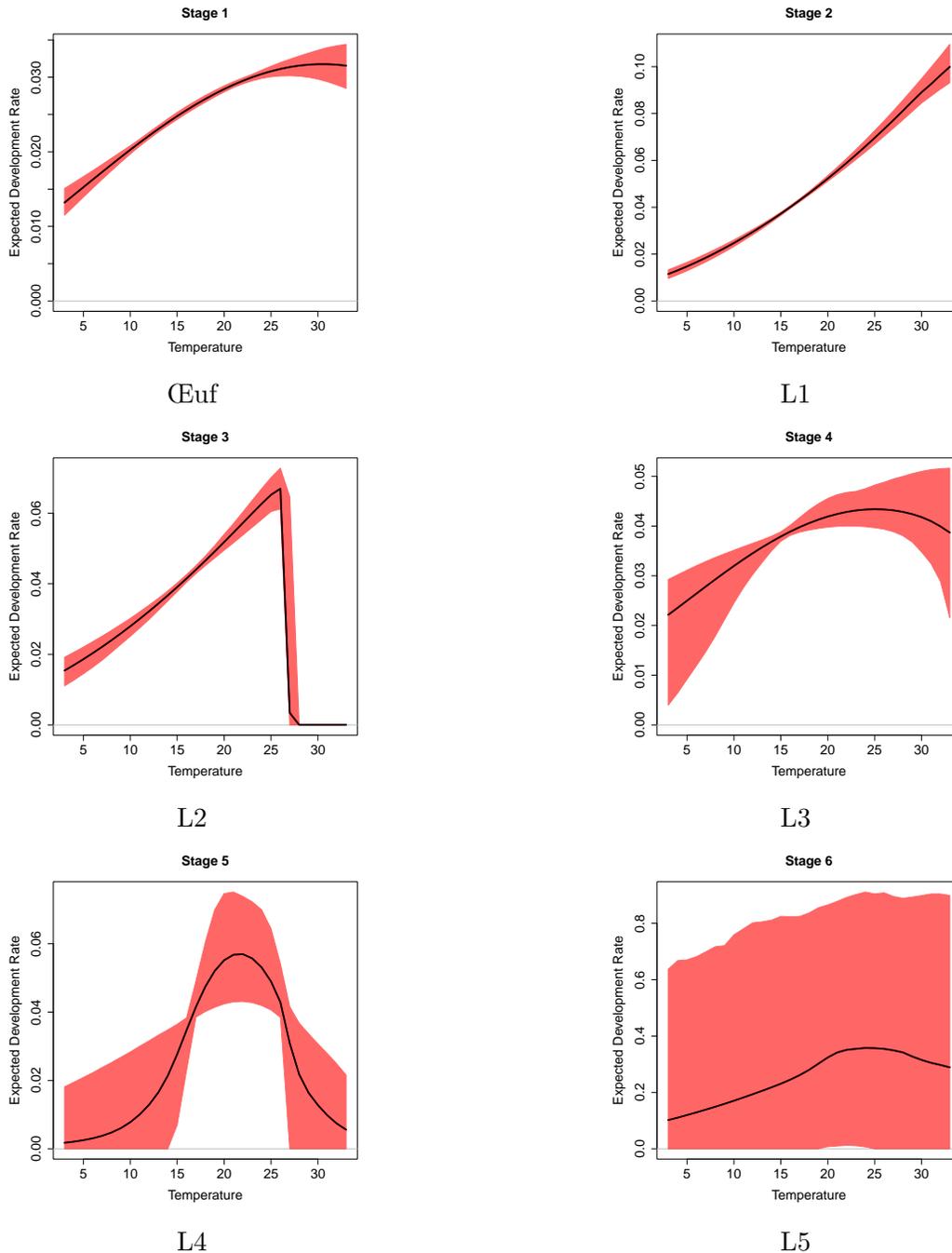


FIGURE 12 – Vitesse de développement moyenne ( $\mu$ ) en fonction des températures pour les différents stades de développement. En noire la moyenne obtenue à partir de 10000 échantillons MCMC et en rouge l'intervalle de crédibilité à 95%. Ces résultats sont obtenus avec  $\text{sdModel}_1$ .

La **Figure 12** représente la moyenne de la vitesse de développement des différents stades larvaires en fonctions des températures avec un intervalle de confiance à 95%. La courbe en noire représente la moyenne (10000 échantillons) de  $f(\Delta)$ , qui est le modèle de Brière standardisé, dont on rappelle la formule :

$$R(T) = \alpha \frac{T'^2(1 - T')^{\frac{1}{B}}}{\kappa^2(1 - \kappa)^{\frac{1}{B}}} \quad (4.22)$$

où,  $\alpha$  représente l'amplitude de la courbe,  $\kappa$  la forme de la courbe,  $T_{min}$  et  $T_{max}$  les températures minimale et maximale de développement.

On remarque dans la **Figure 12** que :

- Pour les deux premiers stades œuf et L1 : On observe que la température maximale est supérieur à 30°C et une température minimale inférieur à 0°C avec un intervalle de crédibilité étroit entre 0°C et 30°C.
- Pour le stade L2 : On remarque que la température maximale ici est à 27°C et que la température minimale est inférieure à 0°C et que l'intervalle de crédibilité s'élargit légèrement comparé aux deux premiers stades de développement. On remarque que la valeur de  $T_{opt}$  est très proche de la valeur de  $T_{max}$ .
- Pour le stade L3 : l'intervalle de crédibilité est large, les températures  $T_{min}$  et  $T_{max}$  sont plus petites que 0°C et plus grande que 30°C respectivement.
- Pour le stade L4 : on remarque que l'intervalle de crédibilité est large ce qui traduit une forte incertitude pour les valeurs de  $T_{min}$  et  $T_{max}$ .
- Pour le stade L5 : l'intervalle de crédibilité est très large et donc l'estimation des paramètres  $T_{min}$  et  $T_{max}$  manque de précision. C'est le stade avec le plus d'incertitude au niveau de l'amplitude  $\alpha$ .

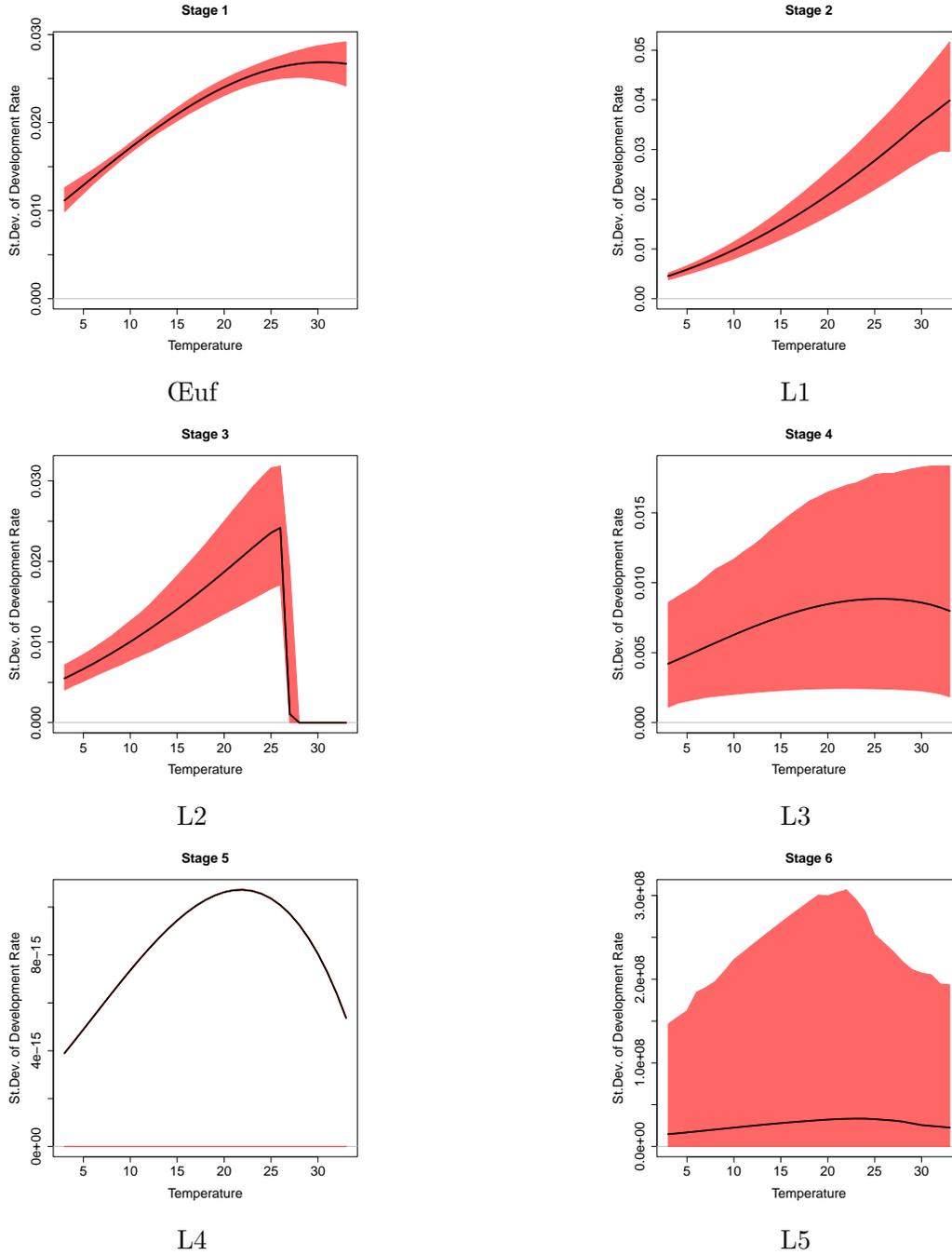


FIGURE 13 – Écart-type de la vitesse de développement  $\sigma$  en fonction des températures pour les différents stades de développement. En noire la moyenne obtenue à partir de 10000 échantillons MCMC et en rouge l'intervalle de crédibilité à 95%. Ces résultats sont obtenus avec `sdModel1`.

La **Figure 13** représente l'écart-type de la vitesse de développement ( $\sigma$ ) des différents stades larvaires en fonctions des températures avec des intervalles de confiance à 95%. La courbe en noire représente l'écart-type de  $f(\Delta)$  dont on rappelle la formule :

$$\sigma|T = R(T) \times \exp(\beta_0) \quad (4.23)$$

où  $R(T)$  est le modèle de Brière standardisé.

On remarque sur la **Figure 13** que :

- Les quatre premières courbes ressemblent beaucoup à celles de la **Figure 12** mais avec des intervalles de crédibilités (en rouge) plus larges.
- Pour le stade L4 : la moyenne des écarts-types est très proche de 0 . De plus, le décalage entre l'intervalle de crédibilité et la moyenne des écarts-types  $\sigma$  (en noir) indique que celle-ci est très influencée par un petit nombre d'échantillons avec des valeurs extrêmes.
- Pour le stade L5 : l'intervalle de crédibilité est extrêmement large.

## Courbes de proportion de stades présent dans les cohortes

A partir des valeurs de paramètres échantillonnés par MCMC, nous avons utilisé le modèle PIML pour simuler le développement (du stade œuf au stade imago) au sein de chaque cohorte en fonction des températures fluctuantes. On rappelle que :

$$\pi_{S,c,t} = \sum_{i=1}^r n_{i,S,c,t} \quad (4.24)$$

cette formule représente la probabilité conditionnelle pour la cohorte  $c$  d'appartenir au stade  $S$  au temps  $t$ . Les évolutions au cours du temps de  $\pi_{S,c}$  sont représentées pour quatre cohortes dans la figure suivante :

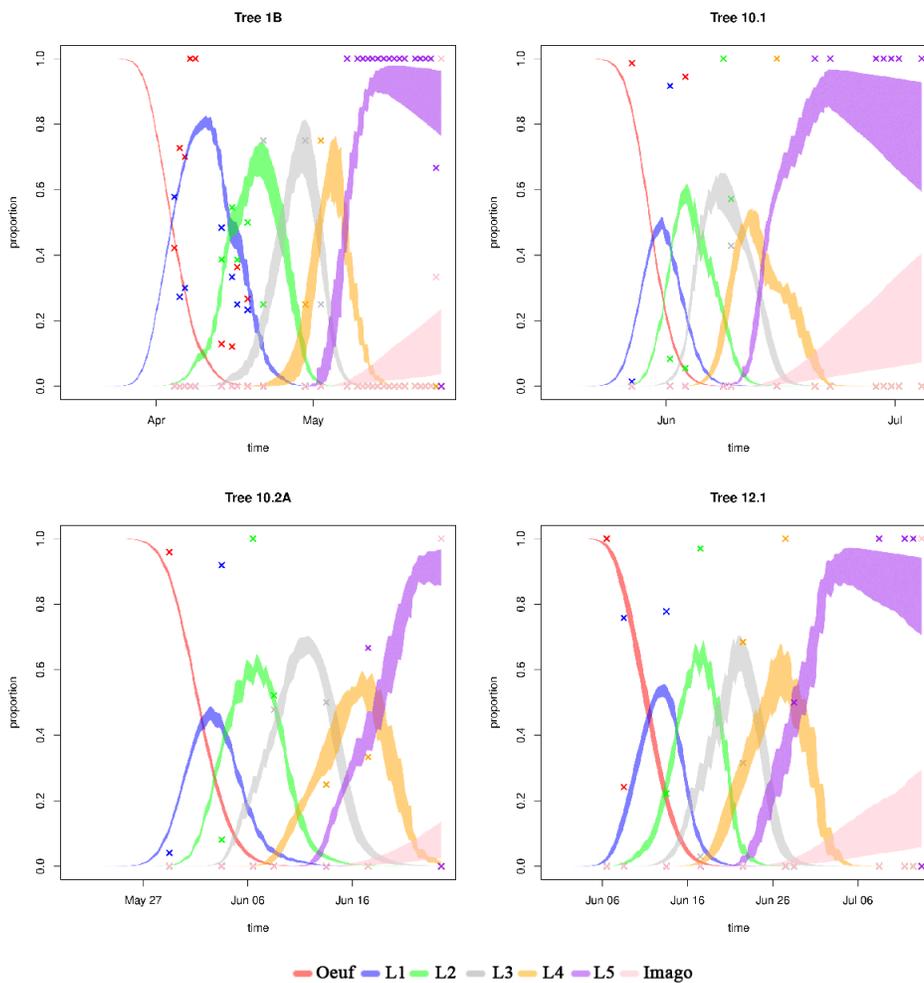


FIGURE 14 – Courbes simulées de la proportion de présence des différents stades (de œuf à imago) pour quatre cohortes de psylles au cours du temps ( $\pi_{S,c,t}$ ). Les croix sur les graphiques représentent les proportions observées (œufs, larves, imagos) au cours de l'expérimentation de 2005. L'épaisseur de ces courbes représentent les intervalles de crédibilités à 95%, estimés à partir de 1000 échantillons de MCMC.

Rappelons que pour chaque cohorte, pour un jour donné d’observation, nous avons comme données le nombre d’individus de chacun des stades. À partir de là, nous en avons déduit la proportion d’individus de chacun des stades. Par exemple, sur la [Figure 14](#) en haut à gauche, le 18 mai, nous avons observé 82% (croix rouge) d’œufs et 18% de larves L1 (croix bleu). On remarque ici que :

- Les intervalles de confiance sont de plus en plus larges au cours du temps.
- Le chevauchement important entre différents stades, par exemple, pour le stade L2 (en vert) chevauche sur les stades œuf (rouge), L1 (bleu), L3 (gris) et L4 (orange).
- Pour chaque stade, on observe un étalement assez important de la période de présence des psylles dans ce stade. Par exemple, la figure en bas à gauche, les L2 (vert) sont observables durant une durée d’environ 15 jours. Tandis que sur la figure en haut à gauche, ce même stade L2 est étalé sur presque 1 mois.
- En comparant les quatre cohortes, on constate que la date à laquelle les femelles sont mises à pondre influence la durée de développement des stades : les durées de développement sont plus longues pour les cohortes où les femelles ont été mises à pondre tôt dans l’année.
- Les données simulées semblent très cohérentes avec les données observées. En particulier, les simulations prédisent de manière cohérente l’apparition des larves L5.
- Certaines des courbes (par exemple L2 en vert) ne sont pas lisses, ce qui est inattendu.

## 4.6 Discussion

Dans cette étude, nous avons exploré l’estimation du modèle **PIML** à partir des données psylles récoltées en conditions semi-naturelles. Le défi était de modéliser la vitesse de développement en fonction des températures fluctuantes alors que jusque-là les modèles utilisés en dynamique des populations étaient basés uniquement sur des données récoltées à des températures constantes. On a montré dans la section 3 les limites de ces modèles.

Le développement du modèle **PIML** nous a confronté à plusieurs obstacles méthodologiques. Afin de contourner ces obstacles, nous avons dû faire trois innovations majeures : reparamétrisation du modèle de Brière, exploration de plusieurs modèles températures dépendant pour  $\sigma$ , représentation creuse de  $M_t$ . Un autre point important est que pour améliorer l’efficacité de l’algorithme MCMC nous avons dû utiliser les algorithmes APT.

D’un point de vue biologique, les projections du modèle **PIML** sont satisfaisantes dans le sens où elles prédisent bien l’apparition des larves du stade L5. Pour autant, il reste des choses à améliorer : les projections de la [Figure 14](#) sont moins lisses que attendu, pour la [Figure 13](#) la différence entre des écarts-types ( $\sigma$ ) des stades L4 et L5 est trop extrême, la température  $T_{max}$  du stade L3 est trop basse d’un point de vue biologique, et les températures  $T_{min}$  de plusieurs stades sont beaucoup trop basses pour être réalistes.

Ce problème indique que nos priors ont été certainement insuffisamment informatifs, ce qui a laissé beaucoup trop de liberté dans l’estimation des paramètres. Une deuxième raison potentielle pour expliquer ces problèmes de paramétrisation viendrait l’équation (4.13). Dans cette équation, il est possible que le  $p_r$  (dernier stade) sous-estimé occasionnellement à cause d’un troncage de  $\Delta = 1$  ce qui est un problème quand  $f(\Delta|\mu, \sigma)$  suit une loi log-normale (comme dans notre cas). Une solution serait d’ajuster cette fonction (4.13) pour éviter ce troncage.

Comme perspectives, ce modèle devra être amélioré avec moins de liberté dans les priors. Ensuite, il faudra utiliser la validation-croisée pour tester le pouvoir prédictif de ce modèle afin de générer des prédictions réalistes sur d'autres sites de collecte de données de psylles.

## 5 Développement de l'application Psyl'risk

### 5.1 Le réseau actuel d'épidémio-surveillance des psylles

Depuis une dizaine d'années, un réseau d'épidémio-surveillance a été mis en place en France pour suivre la présence des psylles *C. pruni* dans différentes régions de production d'abricots et de prunes. Ce réseau consiste à quantifier chaque semaine, de janvier à mai, les psylles présents dans des massifs ou haies de prunelliers de 5 localités : Montauban (Tarn-et-Garonne), Torreilles (Pyrénées-Orientales), Grabels (Hérault), Cavaillon (Vaucluse) , Étoile-sur-Rhône (Drôme). L'échantillonnage est réalisé en frappant 20 branches de prunelliers. Les psylles sont récoltés sur une toile (80x80 cm) et aspirés avec un petit aspirateur à bouche. Les informations obtenues pour chaque site de collecte sont rassemblées dans une base de donnée commune, et diffusées via un blog (<https://cpruni.blogspot.com/>). Ce blog permet d'avoir accès aux données brutes et à des courbes « de vol » réactualisées chaque semaine.

### 5.2 Une nouvelle interface web, Psyl'Risk

Nous souhaitons améliorer la diffusion de ces informations et donner accès en open source à toutes les données, y compris les données de toutes les années antérieures (jusqu'à 20 ans de données pour certains sites). L'objectif principal était que les personnes ayant accès à ces données puissent faire facilement des comparaisons entre sites, entre années pour un même site, au travers un affichage interactif. Nous souhaitons également que les différentes personnes qui interviennent dans les collectes de psylles puissent elles-mêmes rajouter leurs informations dans la base de données commune, pour accélérer le processus de diffusion de l'information. Dans ce but, j'ai développé une application web à l'aide du package R, shiny, que j'ai nommée Psyl'risk. Le package shiny est un package qui permet de créer des applications web interactives sous R, mais aussi en écrivant des scripts sous HTML, CSS et markdown.

La nouvelle application Psyl'risk permettra aux utilisateurs autorisés de rentrer leurs propres données dans la base commune. L'application générera alors automatiquement les courbes de captures associées. Cet outil a vocation de faciliter la prise de décision des arboriculteurs ou des techniciens de terrain pour mieux gérer la lutte contre les psylles, c'est-à-dire de leur permettre de décider rapidement du moment opportun pour utiliser des traitements contre les psylles.



FIGURE 15 – Nouvelle plateforme "Psyl'risk"

En suivant le lien suivant [https://walidkandouci.shinyapps.io/PsylleShinyApp/?\\_ga=2.204860432.1240152510.1625572804-584626372.1618915623](https://walidkandouci.shinyapps.io/PsylleShinyApp/?_ga=2.204860432.1240152510.1625572804-584626372.1618915623) l'utilisateur aura accès à une page d'accueil décrivant l'application. A gauche apparaîtront différents onglets :

- **Visualisation** : donne accès à une première fenêtre donnant la liste des différents sites pour lesquels des données sont disponibles. Cliquer sur les onglets du haut de cette page donne accès aux courbes pour chaque site. Par défaut, toutes les courbes pour toutes les années disponibles dans la base sont affichées (Fig. 16), mais il est possible de sélectionner l'année ou les années de son choix, en utilisant la bannière de droite. L'onglet "Données terrain" permet de visualiser les données brutes pour chaque site.

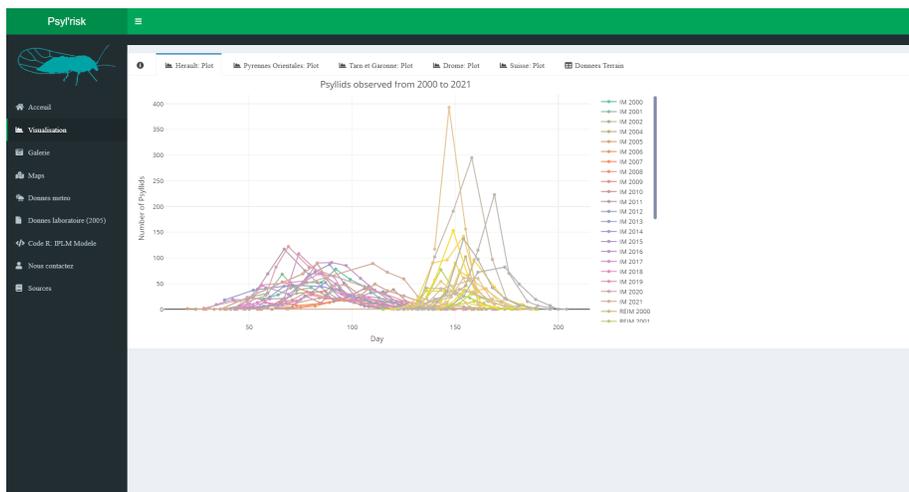


FIGURE 16 – Représentation graphique des données collectés à l'Hérault

- **Galerie** : galerie de photos de *C. pruni* ou collectes.
- **Maps** : donne accès à deux types de carte : **(1)** Carte "bassins" permettant de localiser les points de collecte de psylles du réseau d'épidémiologie géographique; **(2)** carte "Espèces A et B" permettant de visualiser les occurrences des espèces A et B du complexe *C. pruni*, d'après Sauvion et al. 2021 (<https://bdj.pensoft.net/article/68860/>)

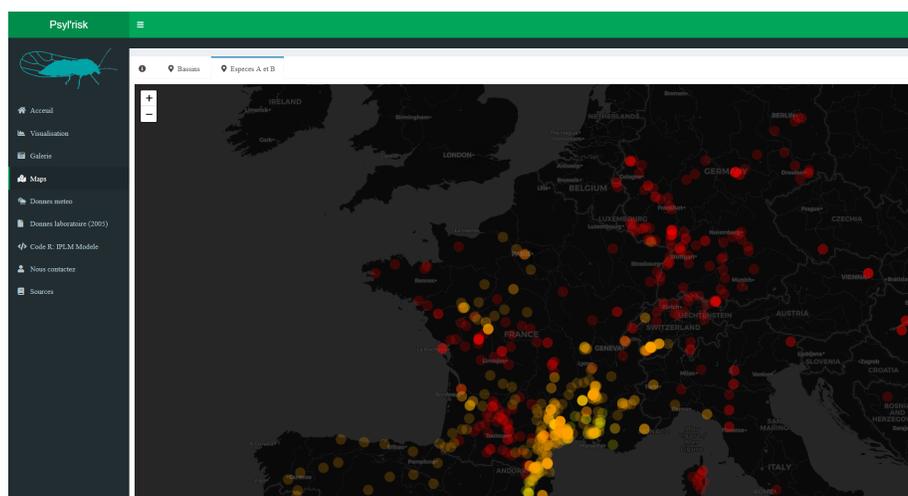


FIGURE 17 – Représentation sur la carte des deux espèces A (jaune) et B (rouge)

- **Données météo** : donne accès aux données météorologiques utilisées au cours du master 2 de Walid Kandouci (mars-août 2021), et téléchargées gratuitement depuis le site de Météo France par le lien : <https://www.data.gouv.fr/fr/reuses/code-r-de-recuperation-des-donnees-synop-meteo-france/>
- **Données du laboratoire (2005)** : pour visualiser les données brutes obtenues au cours de l'expérimentation de 2005 visant à estimer la vitesse de développement des psylles en fonction de la température (pour plus de détails voir résultats du stage de M2 de Walid Kandouci).
- **Code R** : renvoie à une page affichant les liens url pour accéder aux codes sources des programmes développés au cours du master 2 de Walid Kandouci.
- **Contact** : donne les adresses mails pour nous contacter
- **Sources** : liste quelques références d'intérêt ciblées sur le pathosystème *C. pruni/phytoplasme/Prunus*.

### 5.3 Conclusion

L'objectif de l'application **Psy'risk** est de permettre non seulement aux différents chercheurs d'avoir un accès rapide et facile aux données, ainsi que de pouvoir suivre la dynamique des vols de psylles sur les différents sites et de pouvoir les comparer avec les données obtenues lors des années précédentes.

## 6 Conclusion générale

Nous avons présenté dans ce rapport le modèle de projection intégrale pour objectif de l'appliquer pour décrire et modéliser la vitesse de développement en fonction des températures fluctuantes ainsi que de prédire la proportion de présence des différents stade au fil du temps pour l'année 2005 pour l'espèce *Cacopssyla pruni*.

Dans le futur, après certaines améliorations, l'implémentation du modèle PIML pourra être utilisée sur les différents sites de collectes de données. Ainsi il sera possible d'utiliser cette approche pour prédire l'apparition des nouvelles générations de psylles, pour lutter contre le phytoplasme (bactérie) responsable de l'Enroulement Chlorotique de l'Abricotier (ECA ou ESFY pour European stone fruit yellows).

L'application **Psyll'risk** permettra quant à elle de partager les données et visualiser la présence des psylles sur les différents sites.

## Références

- [1] Bhattacharya A, Pati D, Pillai NS & Dunson DB (2015) *Dirichlet–Laplace priors for optimal shrinkage*. *Journal of the American Statistical Association* 110 : 1479-1490. doi :<https://doi.org/10.1080/01621459.2014.960967>.
- [2] Brière J-F, Pracros P, Le Roux A-Y & Pierre J-S (1999) *A novel rate model of temperature-dependent development for arthropods*. *Environmental Entomology* 28 : 22-29. doi :<https://doi.org/10.1093/ee/28.1.22>.
- [3] Carraro L, Osler R, Loi N, Ermacora P & Refatti E (1998) *Transmission of European stone fruit yellows phytoplasma by Cacopsylla pruni*. *Journal of Plant Pathology* 80 : 233-239. doi :<https://www.jstor.org/stable/41997929>.
- [4] Castano MS (2017) *Heterogeneity in hidden developmental processes : inference and analysis for stage-structured populations in fluctuating environments*, PhD : Philosophy. *Physiology and biology of organisms, population and interactions*. University of Antilles, Guadeloupe, France, 191 p.
- [5] Easterling MR, Ellner SP & Dixon PM (2000) *Size-specific sensitivity : applying a new structured population model*. *Ecology* 81 : 694-708. doi :[https://doi.org/10.1890/0012-9658\(2000\)081\[0694 :SSSAAN\]2.0.CO;2](https://doi.org/10.1890/0012-9658(2000)081[0694 :SSSAAN]2.0.CO;2).
- [6] Goodsmann DW, Aukema BH, McDowell NG, Middleton RS & Xu C (2018) *Incorporating variability in simulations of seasonally forced phenology using integral projection models*. *Ecology and evolution* 8 : 162-175. doi :<https://doi.org/10.1002/ece3.3590>.
- [7] Kontodimas DC, Eliopoulos PA, Stathas GJ & Economou LP (2004) *Comparative temperature-dependent development of Nephus includens (Kirsch) and Nephus bisignatus (Boheman)(Coleoptera : Coccinellidae) preying on Planococcus citri (Risso)(Homoptera : Pseudococcidae) : evaluation of a linear and various nonlinear models using specific criteria*. *Environmental Entomology* 33 : 1-11. doi :<https://doi.org/10.1603/0046-225X-33.1.1.C>
- [8] Labonne G & Lichou J (2004) *Data on the life cycle of Cacopsylla pruni, psyllidae vector of European stone fruit yellows (ESFY) phytoplasma, in France*. *Acta Horticulturae* 657 : 465-470.
- [9] Laçki M.K, Miasojedow B (2015). *State-dependent swap strategies and automatic reduction of number of temperatures in adaptive parallel tempering algorithm*. *Statistics and Computing* volume 26, pages951–964
- [10] Lefkovich L.P (1965) *The Study of Population Growth in Organisms Grouped by Stages*. *Biometrics* Vol. 21, No.1 (Mar.,1965), pp. 1-18. doi :<https://doi.org/10.2307/2528348>
- [11] Leslie P.H (1945) *On The Use Of Matrices In Certain Population Mathematics*. *BBiometrika*, Volume 33, Issue 3, November 1945, Pages 183–212. doi :<https://doi.org/10.1093/biomet/33.3.183>
- [12] Marie-Jeanne V, Bonnot F, Thébaud G, Peccoud J, Labonne G & Sauvion N (2020) *Multi-scale spatial genetic structure of the vector-borne pathogen ‘Candidatus Phytoplasma prunorum’ in orchards and in wild habitats*. *Scientific reports* 10 : 5002. doi :[10.1038/s41598-020-61908-0](https://doi.org/10.1038/s41598-020-61908-0).

- [13] Miasojedow B, Moulines E, Vihola M (2015). *An Adaptive Parallel Tempering Algorithm*. *Journal of Computational and Graphical Statistics Volume 22, 2013 - Issue 3*
- [14] Rebaudo F & Rabhi V-B (2018) *Modeling temperature-dependent development rate and phenology in insects : review of major developments, challenges, and future directions*. *Entomologia Experimentalis et Applicata* 166 : 607-617. doi :<https://doi.org/10.1111/eea.12693>.
- [15] Régnière J, Powell J, Bentz B & Nealis V (2012) *Effects of temperature on development, survival and reproduction of insects : experimental design, data analysis and modeling*. *Journal of Insect Physiology* 58 : 634-647. doi :<https://doi.org/10.1016/j.jinsphys.2012.01.010>.
- [16] Sauvion N, Labeyrie B & Brun L (2020) *Enroulement chlorotique de l'abricotier : situation en France*. *Phytoma La Défense des Végétaux (France)* 735 : 9-10.
- [17] Sauvion N, Peccoud J, Meynard CN & Ouvrard D (2021) *Occurrence data for the two *Cacopsylla pruni* cryptic species (Hemiptera : Psylloidea)*. *Biodiversity Data Journal* 9 : e68860. doi :<https://bdj.pensoft.net/article/68860/>.
- [18] Sauvion N, Thébaud G, Marie-Jeanne V, Peyre J, Brun L & Labonne G (2012) *Enroulement chlorotique de l'abricotier (ECA) - les points-clés pour pour gérer cette maladie à phytoplasme en vergers de fruits à noyau*. *Phytoma* 654 : 28-32.
- [19] Smith AF & Roberts GO (1993) *Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods*. *Journal of the Royal Statistical Society : Series B (Methodological)* 55 : 3-23. doi :<https://doi.org/10.1111/j.2517-6161.1993.tb01466.x>.
- [20] Steffek R, Swen Follak S, Sauvion N, Labonne G & MacLeod A (2012) *Distribution of 'Candidatus Phytoplasma prunorum' and its vector *Cacopsylla pruni* in European fruit growing areas : a literature survey*. *EPPO Bulletin* 42 : 191-202. doi :[10.1111/epp.2567](https://doi.org/10.1111/epp.2567).
- [21] Taylor F (1981) *Ecology and evolution of physiological time in insects*. *The American Naturalist* 117 : 1-23. doi :<https://doi.org/10.1086/283683>.
- [22] Thébaud G, Yvon M, Alary R, Sauvion N & Labonne G (2009) *Efficient transmission of 'Candidatus Phytoplasma prunorum' is delayed by eight months due to a long latency in its host-alternating vector*. *Phytopathology* 99 : 265-273. doi :[10.1094/PHYTO-99-3-0265](https://doi.org/10.1094/PHYTO-99-3-0265).
- [23] Tonnang HE, Hervé BD, Biber-Freudenberger L, Salifu D, Subramanian S, Ngowi VB, Guimapi RY, Anani B, Kakmeni FM & Affognon H (2017) *Advances in crop insect modelling methods—Towards a whole system approach*. *Ecological Modelling* 354 : 88-103. doi :<https://doi.org/10.1016/j.ecolmodel.2017.03.015>.

## A Code R - Modèles de Brière, Kontodimas et Taylor

Ces fonctions sont disponibles sur le package *nimbleTempDev*

```
# Briere model
briere <- nimbleFunction(
  run = function(t = double(1), Tmin = double(0), Tmax = double(0),
    aa = double(0), bb = double(0)){
    returnType(double(1))
    result = nimNumeric(length = length(t))
    for (i in 1:length(t)) {
      if (t[i]<=Tmin | t[i]>=Tmax) {
        result[i] <- 0
      }else{
        result[i] <- aa * t[i] * (t[i]-Tmin) * ((Tmax-t[i])^(1/bb))
      }
    }
    return(result)
  }
)
```

```
# Kontodimas model
kontodimas <- nimbleFunction(
  run = function(t = double(1), Tmin = double(0), Tmax = double(0),
    aa = double(0)){
    returnType(double(1))
    result = nimNumeric(length = length(t))
    for (i in 1:length(t)) {
      if (t[i]<=Tmin | t[i]>=Tmax) {
        result[i] <- 0
      }else{
        result[i] <- aa * (t[i] - Tmin)^2 * (Tmax - t[i])
      }
    }
    return(result)
  }
)
```

```
# Taylor model
taylor <- nimbleFunction(
  run = function(t = double(1), Rm = double(0), Tm = double(0),
    To = double(0)){
    returnType(double(1))
    result = nimNumeric(length = length(t))
    for (i in 1:length(t)) {
      result[i] <- Rm * exp(-1/2 * ((t[i] - Tm)/To)^2)
    }
    return(result)
  }
)
```

## B Données manquantes - 2005

```
# DataMeteo2005
DataMeteo2005$temperature <- as.numeric(DataMeteo2005$temperature)
DataMeteo2005$humidity <- as.numeric(DataMeteo2005$humidity)
DataMeteo2005$date <- as.Date(DataMeteo2005$date)
# Option 1: linear type with the function "na.interp"
missingMeteo <- na.interp(DataMeteo2005$temperature)

plot(missingMeteo[1:110],
     type = "l",
     main = "",
     xlab = "jours",
     ylab = "temperatures",
     lwd=3,
     col="red")
lines(DataMeteo2005$temperature[1:110],
      type = "l",
      lwd=3)

length(missingMeteo) == length(DataMeteo2005$temperature)
which(is.na(DataMeteo2005$temperature))
missingMeteo[which(is.na(DataMeteo2005$temperature))]
# Option 2: package "imputeTS"
ggplot_na_distribution(DataMeteo2005$temperature)
imp <- na_kalman(DataMeteo2005$temperature)
ggplot_na_imputations(DataMeteo2005$temperature, imp)

imp[which(is.na(DataMeteo2005$temperature))]

# Option 3: auto.arima()
y <- DataMeteo2005$temperature
fit <- auto.arima(DataMeteo2005$temperature)
kr <- KalmanRun(DataMeteo2005$temperature, fit$model)
id.na <- which(is.na(DataMeteo2005$temperature))

for (i in id.na){
y[i] <- fit$model$Z * kr$states[i,]
}

which(is.na(y))
y[id.na]

#####
## PLOT ##
#####
TS <- ts(y, start = c(1,1), frequency = 7)
autoplot(TS)
```

## C Code R - Modèle PILM

Ce code est disponible sur le lien suivant : <https://github.com/WalidKandouci/CacopsyllaPruni/blob/main/src/modelDefinition.R>

```
#####  
## This script provides the definition of the multi-stageIPM model ##  
#####  
  
#####  
## Packages ##  
#####  
library(here)          ## for locating a suitable base directory  
library(tibble)       ## for tibbles  
library(dplyr)        ## for piping %>%  
library(nimble)       ## for working with BUGS type models in R  
library(nimbleAPT)    ## for adaptive parallel tempering  
library(nimbleTempDev) ## for temperature dependant IPM  
library(lubridate)    ## for dates and times  
library(coda)         ## for mcmc diagnostics  
  
baseDir = here()  
setwd(baseDir)  
load("data/data4nimble.Rdata")  
  
if (!exists("setConstantsElsewhere")) { ## This flag permits other scripts (which  
  source this script) to set the following constants  
  SDmodel = 1 # 2, 3, 4, 5  
}  
  
#####  
## Missing values ##  
#####  
if (is.element("package:imputeTS", search())) {  
  library(imputeTS)      ## for na_kalman  
  ## Impute missing values  
  imp <- na_kalman(meteo$temperature) # use the Kalman filter to input our  
  missing values  
  meteo$temperature <- round(imp) # NA free meteo dataset  
} else {  
  ## A temporary hack because imputeTS won't install on migale  
  meteo$temperature[is.na(meteo$temperature)] = c(10, 10, 9)  
}  
  
#####  
## BUGS code for nimble model ##  
#####  
psyllidCode <- nimbleCode ({  
  #####  
  ## Development kernel at each temperature & for each stage ##  
  #####  
  for (stage in 1:nStagesDev) { # iStag = index for {, L1, L2, L3, L4, L5, imago}  
  ## Priors  
    Tmin[stage] ~ dunif(-40, 20) # dnorm(0, sd=20)  
    Tmax[stage] ~ dunif( 20, 80) # dnorm(40, sd=20)
```

```

logit(amplitudeMean[stage]) ~ dLogitBeta(1,1)
logit(shapeMean[stage]) ~ dLogitBeta(1,1)
beta0[stage] ~ dnorm(0, tau=tauBeta0)
if (SDmodel > 2) {
  beta1[stage] ~ ddexp(location=0, scale=scaleBeta1)
}
if (SDmodel > 4) {
  beta2[stage] ~ ddexp(location=0, scale=scaleBeta2)
}
#####
## Mean development as a function of temperature ##
#####
paras[stage, 1:lTempVec, 1] <- stBriere(T=tempVec[1:lTempVec], Tmin=Tmin[stage], Tmax=Tmax[stage], shape=shapeMean[stage], amplitude=amplitudeMean[stage])
for (iTemp in 1:lTempVec) { # iTemp = index for temperature
## Within stage development at temperature tempVec[iTemp]
  devKernel[stage, iTemp, 1:(res+1)] <- getKernel(paras=paras[stage, iTemp, 1:3], res=res, devFunction = 1)
## Survival
  paras[stage, iTemp, 3] <- 1
## Standard deviation in development as a function of temperature
  if (SDmodel == 1) {
    paras[stage, iTemp, 2] <- paras[stage, iTemp, 1] * exp(beta0[stage])
  } else if (SDmodel == 2) {
    paras[stage, iTemp, 2] <- (paras[stage, iTemp, 1] > 0) * exp(beta0[stage])
  } else if (SDmodel == 3) {
    paras[stage, iTemp, 2] <- paras[stage, iTemp, 1] * exp(beta0[stage] + beta1[stage]*tempVec[iTemp])
  } else if (SDmodel == 4) {
    paras[stage, iTemp, 2] <- (paras[stage, iTemp, 1] > 0) * exp(beta0[stage] + beta1[stage]*tempVec[iTemp])
  } else if (SDmodel == 5) {
    paras[stage, iTemp, 2] <- paras[stage, iTemp, 1] * exp(beta0[stage] + beta1[stage]*tempVec[iTemp] + beta2[stage]*tempVec[iTemp]^2)
  } else if (SDmodel == 6) {
    paras[stage, iTemp, 2] <- (paras[stage, iTemp, 1] > 0) * exp(beta0[stage] + beta1[stage]*tempVec[iTemp] + beta2[stage]*tempVec[iTemp]^2)
  }
}
}
## Hyper-parameters for standard deviation of development models
if (SDmodel > 2) {
## Add a dLogGamma distribution to nimbleTempDev or nimbleAPT
  log(scaleBeta1) ~ dLogGamma(shape=1/nStagesDev, scale=2)
}
if (SDmodel > 4) {
  log(scaleBeta2) ~ dLogGamma(shape=1/nStagesDev, scale=2)
}
#####
## Loop over trees ##
#####
for (tree in 1:nTrees) { # Adding multiple trees means running the IPM seperately for each tree (due to different start dates)
  # IPM projections
  states[tree, 1, 1] <- 1
}

```

```

for (substage in 2:(nStagesDev*res+1)){
  states[tree, 1, substage] <- 0
}
for (time in 1:nSteps[tree]) { # time = index for time-step
  states[tree, time+1, 1:(nStagesDev*res+1)] <- sparseTWstep(states[tree,
time, 1:(nStagesDev*res+1)], devKernel[1:nStagesDev, iMeteoTemp[
iMeteoForObsMat[tree,1] + time - 1], 1:(res+1)])
}
# Likelihood
for (obs in 1:nObs[tree]) {
  for (stage in 1:nStagesDev){
    pStage[tree, obs, stage] <- sum(states[tree, iMeteoForObsMat[tree,obs] -
iMeteoForObsMat[tree,1] + 1, ((stage-1)*res+1):(stage*res)])
  }
  pStage[tree, obs, nStagesTot] <- states[tree, iMeteoForObsMat[tree,obs] -
iMeteoForObsMat[tree,1] + 1, (nStagesDev*res+1)]
  psyllids[tree, obs, 1:nStagesTot] ~ dmultinom(prob = pStage[tree, obs, 1:
nStagesTot], size = psyllidsTotal[tree, obs])
}
}
})

#####
## Constants ##
#####
nObs      = vector("numeric", length = length(psyllids))
nSteps    = vector("numeric", length = length(psyllids))
iMeteoForObs = vector("list", length = length(psyllids))
for (tree in 1:length(iMeteoForObs)) {
  iMeteoForObs[[tree]] = sapply(psyllids[[tree]]$date, function(x) which(abs(
meteo$date-x) == min(abs(meteo$date-x)))) %>% unlist()
  nObs[tree]          = length(iMeteoForObs[[tree]])
  nSteps[tree]        = length(min(iMeteoForObs[[tree]]):max(iMeteoForObs[[tree
]]))
}
iMeteoForObsMat = matrix(NA, nrow = nTrees, ncol = max(nObs))
for (tree in 1:nTrees){
  iMeteoForObsMat[tree, 1:nObs[tree]] = iMeteoForObs[[tree]]
}
stagesDev = c("egg", "L1", "L2", "L3", "L4", "L5")
stagesTot = c("egg", "L1", "L2", "L3", "L4", "L5", "imago")

(res      <- 25)                ## Resolution of within-stage development
(nTrees   <- length(psyllids))
(nStagesDev <- length(stagesDev)) ## Number of developing stages (without imago)
(nStagesTot <- length(stagesTot)) ## Total number of stages (includes imago)
(tempMin   <- min(meteo$temperature, na.rm = TRUE))
(tempMax   <- max(meteo$temperature, na.rm = TRUE))
(tempVec   <- tempMin:tempMax)
(lTempVec  <- length(tempVec))
(lMeteo    <- nrow(meteo))

Const     = list(
  SDmodel  = SDmodel,
  res      = res,

```

```

nTrees          = nTrees ,
nStagesDev      = nStagesDev ,
nStagesTot      = nStagesTot ,
lTempVec        = lTempVec ,
lMeteo          = lMeteo ,
meteoTemp       = meteo$temperature ,
iMeteoTemp      = sapply(meteo$temperature , function(x) which(x == tempVec)),
iMeteoForObsMat = iMeteoForObsMat ,
nSteps          = nSteps ,
tauBeta0        = 1E-11
)

#####
## List available APT output ##
#####
(apOutputFiles = sort(dir(here("APT/"), pattern="*Temps4.txt")))
(apOutputFile = aptOutputFiles[SDmodel])
if (length(aptOutputFiles)!=6)
  stop("Wrong number of APT output files detected.")

#####
## Initial (prior to MCMC) parameter values ##
#####
fileStem        = sub(".txt", "", aptOutputFile)
previous         = read.table(here("APT", aptOutputFile), header=TRUE)
previous         = previous[-round(1:nrow(previous)/2),] # Remove 1/2 the samples
                as burn-in
##
previous = previous %>% tail(1)

Inits = list(
## Last values retrned from a previous run
  Tmin = previous %>% select(grep("Tmin", colnames(previous))) %>% as.numeric(),
  Tmax = previous %>% select(grep("Tmax", colnames(previous))) %>% as.numeric(),
## For SDmodel == 1
  logit_amplitudeMean = previous %>% select(grep("amplitudeMean", colnames(
    previous))) %>% as.numeric(),
  logit_shapeMean      = previous %>% select(grep("shapeMean", colnames(
    previous))) %>% as.numeric(),
## For SDmodel == 2
  beta0      = previous %>% select(grep("beta0", colnames(previous))) %>% as.
    numeric(),
  beta1      = previous %>% select(grep("beta1", colnames(previous))) %>% as.
    numeric(),
  beta2      = previous %>% select(grep("beta2", colnames(previous))) %>% as.
    numeric(),
  log_scaleBeta1 = log((1/nStagesDev) / (1/2)), # Mean = shape / rate (gamma
    distribution)
  log_scaleBeta2 = log((1/nStagesDev) / (1/2)) # Mean = shape / rate (gamma
    distribution)
)

if (is.element(SDmodel, c(2,4,6)))
  Inits$beta0 = rep(-1, nStagesDev) # -2 for models 2 & 6

```

```

#####
## Model data ##
#####
psyllidsArray = array(NA, dim = c(nTrees, max(nObs), nStagesTot))
psyllidsTotal = matrix(NA, nrow=nTrees, ncol=max(nObs))
for (tree in 1:nTrees) {
  psyllidsArray[tree, 1:nObs[tree], 1:nStagesTot] <- psyllids[[tree]] %>% select(
    all_of(stagesTot)) %>% as.matrix()
  psyllidsTotal[tree, 1:nObs[tree]] <- rowSums(psyllidsArray[tree,
    1:nObs[tree], 1:nStagesTot])
}

Data = list(psyllids      = psyllidsArray,
            psyllidsTotal = psyllidsTotal,
            tempVec       = tempVec)

#####
## Build R version of nimble model ##
#####
rPsyllid = nimbleModel(psyllidCode, const=Const, init=Inits, data=Data, calculate
  = FALSE)

#####
## Node lists ##
#####
dataNodes      = rPsyllid$getNodeNames(dataOnly      = TRUE)           #
  The data
stochNodes      = rPsyllid$getNodeNames(stochOnly      = TRUE, includeData = FALSE) #
  The parameters
detNodes        = rPsyllid$getNodeNames(determOnly     = TRUE)           #
  Deterministic nodes
monitorNodes    = rPsyllid$getParents("paras", immediateOnly = TRUE)
## Filter out sumLogProb
## dataNodes = dataNodes[which(dataNodes!="sumLogProb")]
## stochNodes = stochNodes[which(stochNodes!="sumLogProb")]
## detNodes   = detNodes[which(detNodes!="sumLogProb")]

stochNodesUnique = unique(sub("\\[.*", "", stochNodes))
detNodesUnique   = unique(sub("\\[.*", "", detNodes))

#####
## See scripts 'cpuTimeExperiment.R' and 'fitModel.R' for using this model ##
#####

```

## Inférence bayésienne

À mesure que les modèles deviennent plus complexes, de nouveaux paramètres sont inclus et augmentent la sensibilité des sorties aux valeurs des paramètres.

Par conséquent, une évaluation précise des estimations des paramètres et des incertitudes associées est cruciale pour le pouvoir prédictif définissant l'utilité des modèles. L'approche bayésienne est un paradigme naturel qui permet d'évaluer les incertitudes dans les estimations et les prévisions.

L'inférence bayésienne est une approche d'analyse statistique dans laquelle toutes les formes d'incertitude sont exprimées en termes de distributions de probabilité. On entend par information a priori sur le paramètre  $\theta$  toute information disponible sur  $\theta$  en dehors de celle apportée par les observations. L'information a priori sur  $\theta$  est entachée d'incertitude.

Il est naturel de modéliser cette information a priori au travers d'une loi de probabilité, appelée loi a priori. Sa densité est notée  $\pi(\theta)$ . On appelle loi des observations, la loi conditionnelle de  $\mathbf{x}$  sachant  $\theta$ . Sa densité est notée  $f(\mathbf{x} | \theta)$ , que la variable aléatoire  $\mathbf{x}$  soit discrète ou continue.

La loi a posteriori est la loi conditionnelle de  $\theta$  sachant  $\mathbf{x}$ . Sa densité est notée  $\pi(\theta | \mathbf{x})$  et elle est définie comme suit :

$$\pi(\theta | \mathbf{x}) = \frac{f(\mathbf{x} | \theta)\pi(\theta)}{\int_{\Theta} f(\mathbf{x} | \theta)\pi(\theta)d\theta} \quad (\text{C.1})$$

En pratique, il est souvent difficile de générer directement un tel ensemble de variables aléatoires, mais il est souvent possible de construire une chaîne de Markov qui a la cible postérieure comme distribution stationnaire.

En d'autres termes, la chaîne de Markov régit une marche aléatoire qui explore la distribution cible et fournit une série d'échantillons qui, s'ils sont suffisamment grands, peuvent partager de nombreuses propriétés statistiques de l'ensemble souhaité de nombres aléatoires générés indépendamment. Ces méthodes sont connues sous le nom de Markov Chain Monte Carlo (MCMC). Un grand nombre de méthodes MCMC ont évolué à partir de deux techniques clés : l'échantillonneur Gibbs et l'échantillonneur Metropolis-Hastings.<sup>[15]</sup>

## D Packages Nimble, NimbleTempDev et NimbleAPT

Afin de réaliser notre étude, certaines fonctions utilisées lors de la partie implémentation du modèle sur les données sont des fonctions disponibles dans les packages *nimble*, *NimbleTempDev* et *NimbleAPT*. Les deux packages *NimbleTempDev* et *NimbleAPT* sont encore en développement, l'un des objectifs de mon stage était de rendre les fonctions du package *NimbleTempDev* applicable sur différents types de données car ce package n'était implémenté auparavant que sur les données de la thèse faites en 2017 sur un type de données totalement différents des données utilisés lors de mon stage.

### Le package Nimble

NIMBLE propose un nouveau système flexible pour construire des modèles de type BUGS en R qui compile automatiquement le code en C++.

Les principaux avantages de NIMBLE sont qu'il permet de générer des fonctions et des distributions définies par l'utilisateur dans un langage simple de type R, et que la compilation C++ améliore considérablement les temps de calcul.

Des échantillonneurs définis par l'utilisateur peuvent également être générés de la même manière, offrant une flexibilité qui, en même temps, reste relativement accessible pour les écologistes non-

informatiques.

Un autre avantage clé de NIMBLE est qu'il fournit une suite de fonctions pour simuler les modèles BUGS et permet une séparation de l'algorithme et du modèle qui, par rapport à l'écriture d'échantillonneurs MCMC personnalisés pour un modèle, simplifie grandement les scripts d'analyse.

## NimbleTempDev

<https://forgemia.inra.fr/umr-astre/nimbletempdev/>

Une liste de fonctions pour modéliser les taux de développement des différents stades de développement d'un insecte (moyenne et variance) en fonction de la température. Ces fonctions ont été dans un premier temps adaptées à l'origine pour la thèse de Maria Soledad Castano [https://agritrop.cirad.fr/586544/1/CASTANO\\_manuscript\\_avril%202017.pdf](https://agritrop.cirad.fr/586544/1/CASTANO_manuscript_avril%202017.pdf) (encadré par David Pleydell) puis adapté par David Pleydell et moi-même sur mes données.

Ce package comprend les différents modèles décrivant la vitesse de développement en fonction des températures, ainsi que la fonction *setMultiM* qui permet de construire une matrice qui fournit une implémentation discrétisée d'un modèle de projection intégrale (PIM) pour suivre le développement (et la survie) au sein d'une étape à travers plusieurs étapes d'un modèle structuré par étapes.

Cette fonction renvoie une matrice de Lefkovitch haute résolution, qui suit le développement au sein de l'étape, et que nous appelons une matrice de Lefkovitch à projection intégrale (PIML). Cette matrice décrit le développement à travers une ou plusieurs stades.

La fonction *setM* permet de construire la sous-matrice correspondant à chaque étape.

J'ai aidé à développer durant mon stage des fonctions pour exploiter la nature creuse de  $M_t$ . Spécifiquement nous avons développé ensemble les fonctions *getkernel* qui génère une représentation creuse de la matrice  $M_t$  et *sparseTWstep* qui effectue la projection avec l'équation

4.4

## NimbleAPT

<https://forgemia.inra.fr/umr-astre/nimbleAPT>

Une liste de fonctions pour échantillonner les distributions postérieures de modèles bayésiens (écrites dans l'implémentation de Nimble du langage BUGS) via un algorithme dit "adaptive parallel tempering" (APT) basé sur les (Miasojedow 2013) et (Leçki 2015)

La principale motivation de ce package est que, lorsque les distributions a posteriori cibles sont multimodales, les algorithmes MCMC standard fonctionnent souvent mal. Dans ces situations, les algorithmes standard proposent souvent peu ou pas de sauts entre les modes, et peuvent donc fournir de très mauvaises approximations pour cibler les distributions postérieures. Le package nimbleAPT permet aux utilisateurs d'effectuer une trempe parallèle adaptative (APT) comme solution potentielle pour l'échantillonnage de distributions postérieures multimodales.

La trempe parallèle est une technique MCMC dans laquelle la probabilité postérieure (d'un modèle bayésien) est « chauffée », « ou tempérée » à divers degrés. À mesure que la température augmente, les probabilités deviennent plus plates et cela peut augmenter la probabilité et la fréquence des sauts entre les modes.

En pratique, une "échelle de température" (un ensemble classé de températures) est établie, MCMC est effectué dans chaque échelon de l'échelle de température, et une étape MCMC spéciale entre les échelons est introduite afin que les ensembles de paramètres puissent monter et descendre l'échelle de la température.

L'inférence bayésienne est faite sur les échantillons postérieurs de l'échelon non chauffé uniquement. Des algorithmes de trempe parallèle adaptative règlent automatiquement les températures de l'échelle de température de sorte que les taux d'acceptation cibles pour les sauts proposés entre les échelons soient atteints.

Dans notre étude, nous avons testé des échelles de température avec 4,8,12,16 et 20 températures différentes. Ces tests ont indiqué que l'APT pourrait améliorer l'efficacité des MCMC par rapport aux échantillonneurs par défaut proposés par *nimble*. Cependant, il y avait peu d'avantage à utiliser plus de 4 températures, c'est pour cette raison que pour notre analyse finale, nous avons utilisé 4 températures.

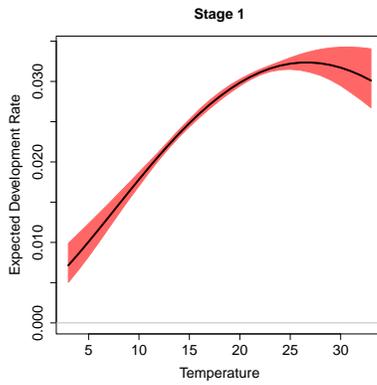
## E Fonction `stBriere`

La fonction `stBriere`, disponible dans le package `nimbleTempDev` (<https://forgemia.inra.fr/umr-astre/nimbletempdev/-/blob/master/nimbleTempDev/R/temperatureResponseCurves.R>) est notre nouvelle version du modèle de Brière standardisé :

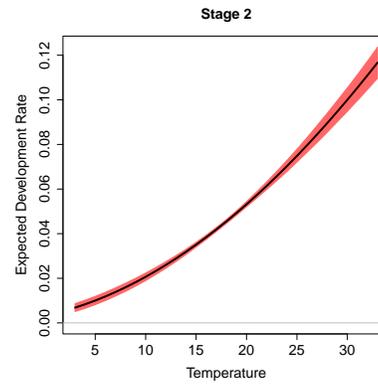
```
stBriere = nimbleFunction(  
  run = function(T      = double(1) ,  
                 Tmin   = double(0) ,  
                 Tmax   = double(0) ,  
                 shape  = double(0) ,  
                 amplitude = double(0)) {  
    returnType(double(1))  
    n           = length(T)  
    result      = nimNumeric(length = n)  
    T01         = nimNumeric(length = n)  
    T01[1:n]   = (T[1:n]-Tmin) / (Tmax-Tmin) # Transform temperatures to [0,1]  
    inverseB   = (2*(1-shape))/shape  
    B          = 1/inverseB  
    mode       = 2*B/(1+2*B)  
    for (i in 1:n) {  
      if (T01[i]<=0 | T01[i]>=1) {  
        result[i] <- 0  
      } else {  
        result[i] <- amplitude * (T01[i]^2 * (1-T01[i])^inverseB) / (mode^2 * (1-  
mode)^inverseB)  
      }  
    }  
    result[result < 0] <- 0  
    return(result[1:n])  
  }  
)
```

# F Résultats des autres modèles

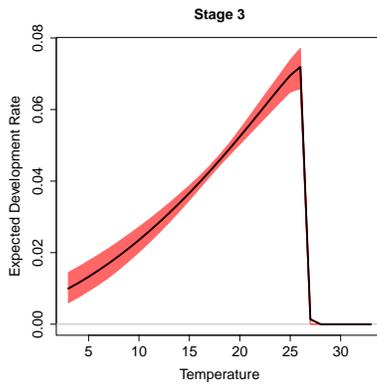
## Modèle 2 : Moyenne



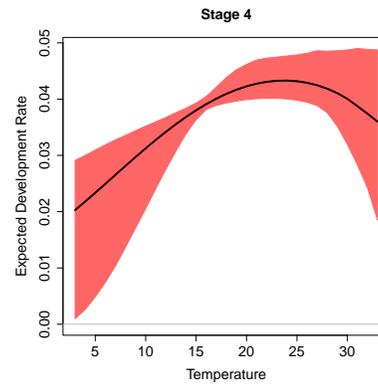
(a) Stade oeuf-L1



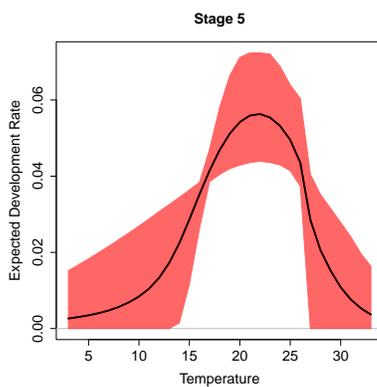
(d) Stade L1-L2



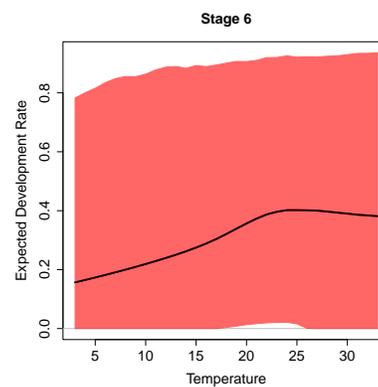
(b) Stade L2-L3



Stade L3-L4



(c) Stade L4-L5



Stade L5-imago

FIGURE 18 – Vitesse de développement en fonction des températures pour les différents stades de développement.

## Modèle 2 : SD

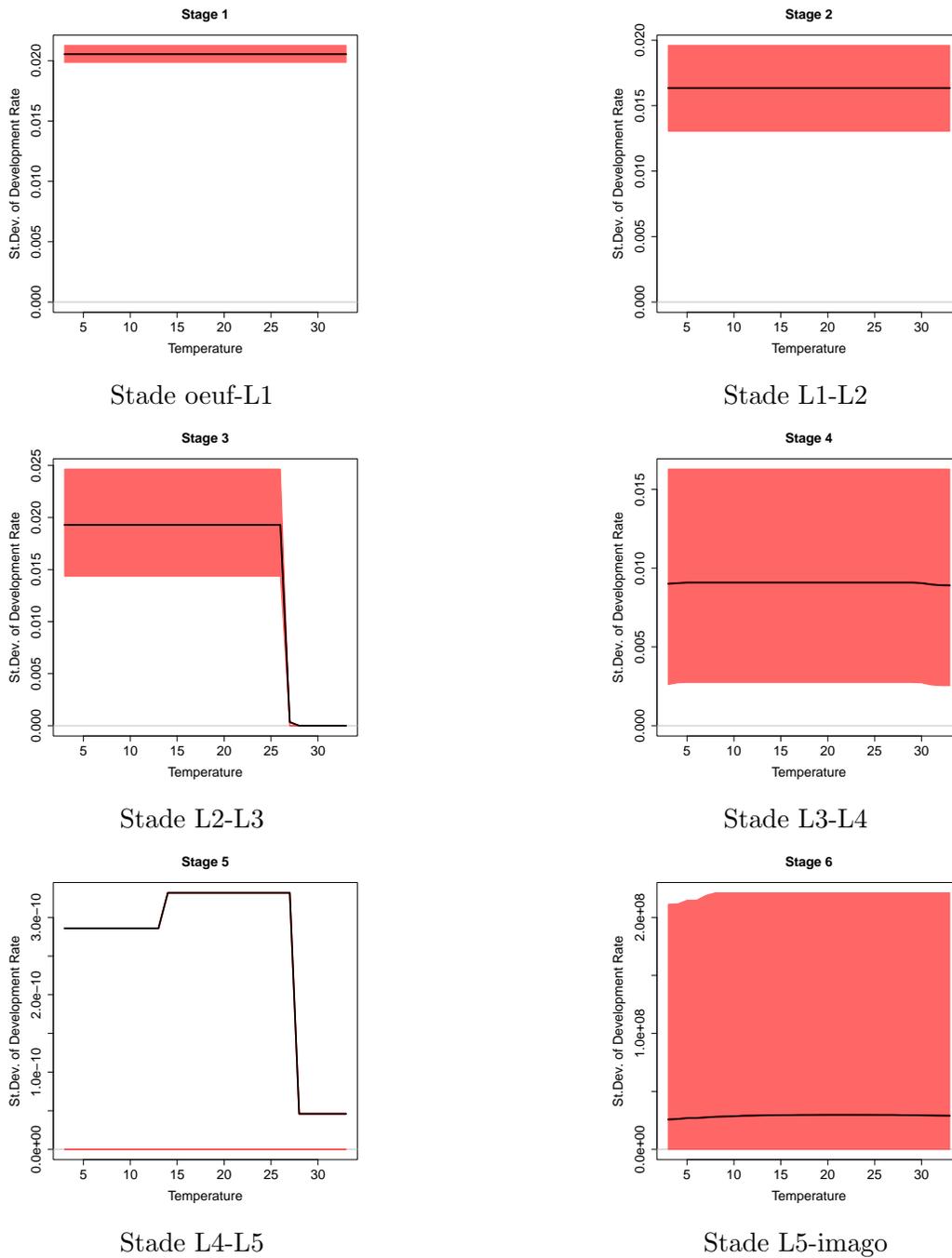


FIGURE 19 – Vitesse de développement en fonction des températures pour les différents stades de développement.

### Modèle 3 : Moyenne

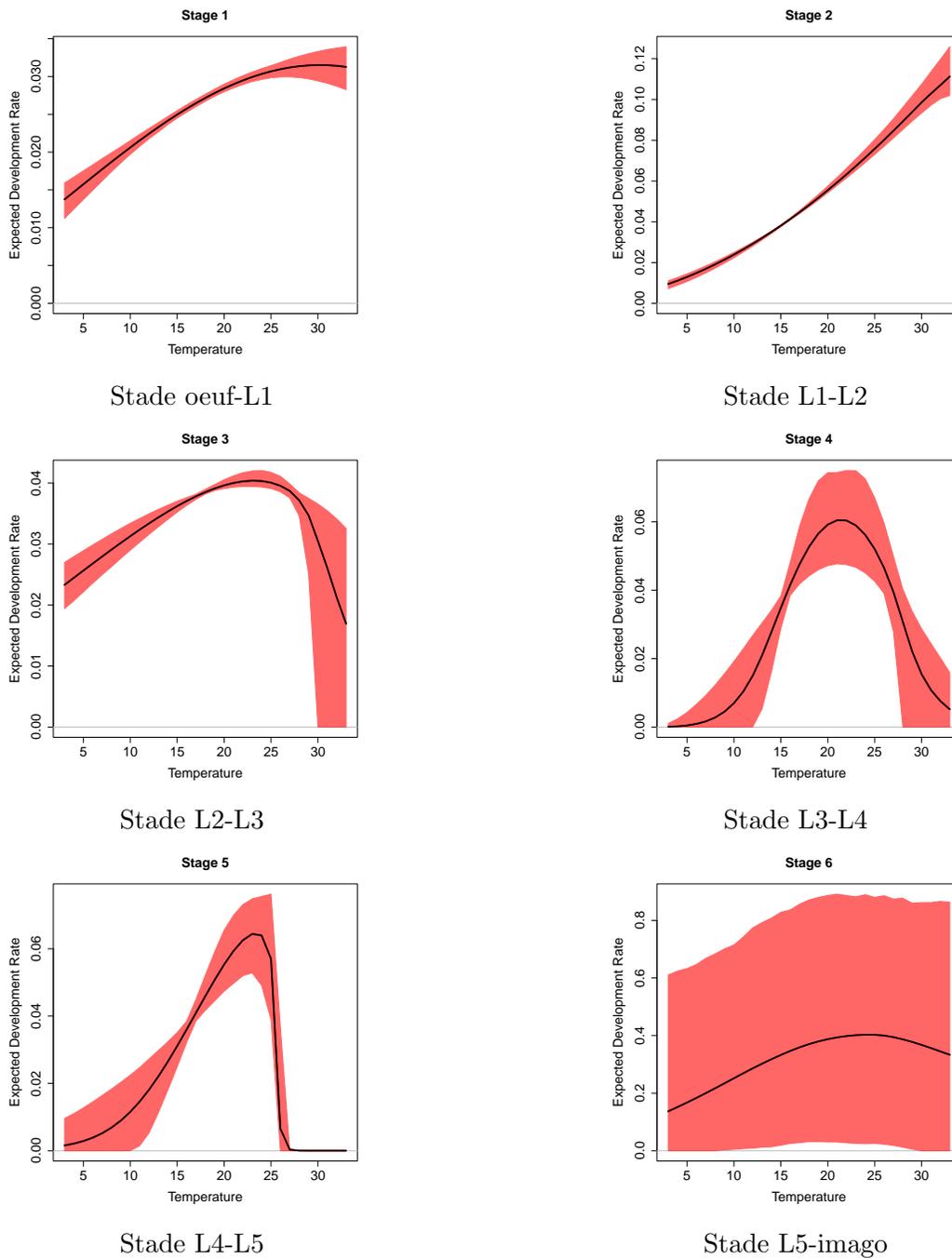


FIGURE 20 – Vitesse de développement en fonction des températures pour les différents stades de développement.

### Modèle 3 : SD

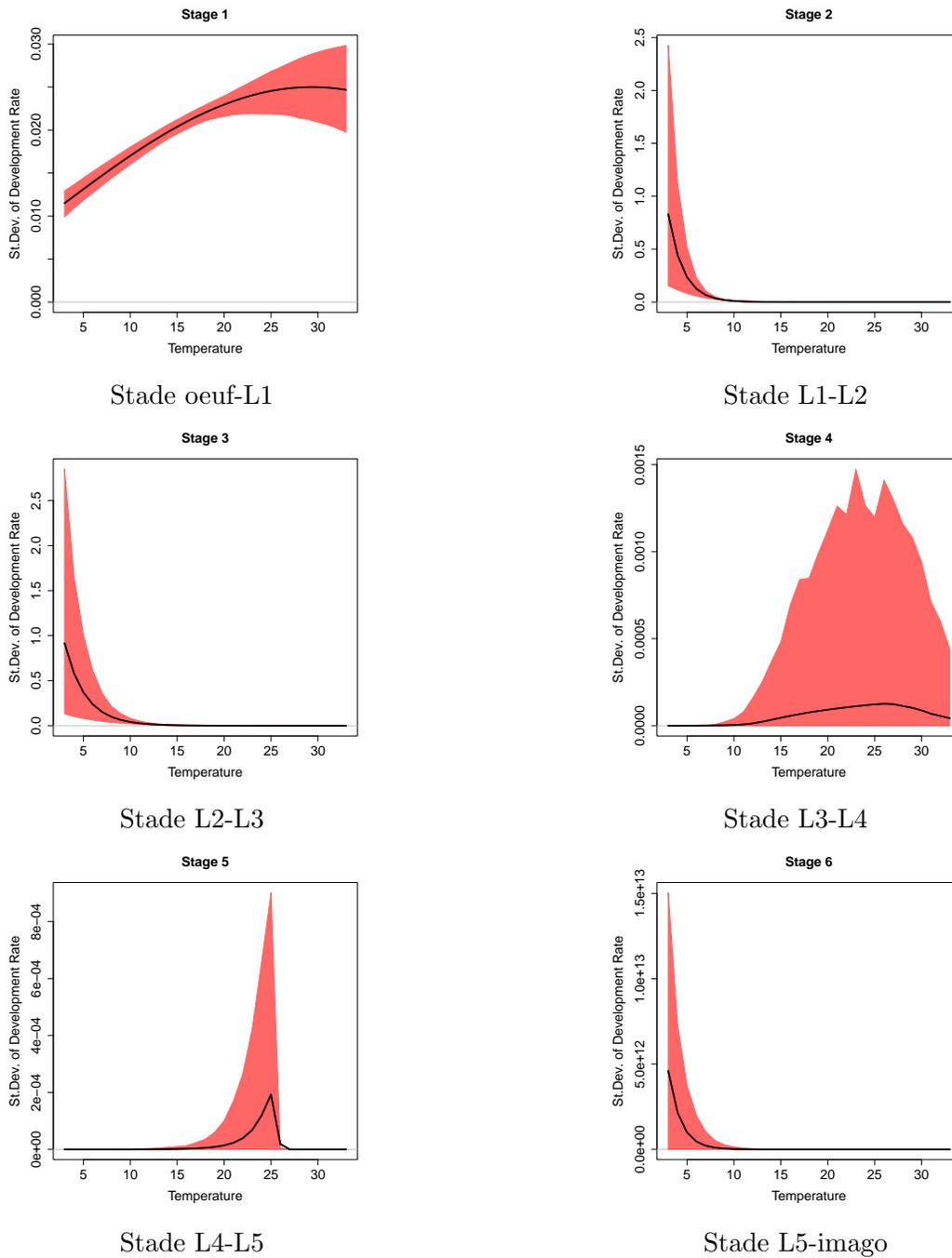


FIGURE 21 – Vitesse de développement en fonction des températures pour les différents stades de développement.

## Modèle 4 : Moyenne

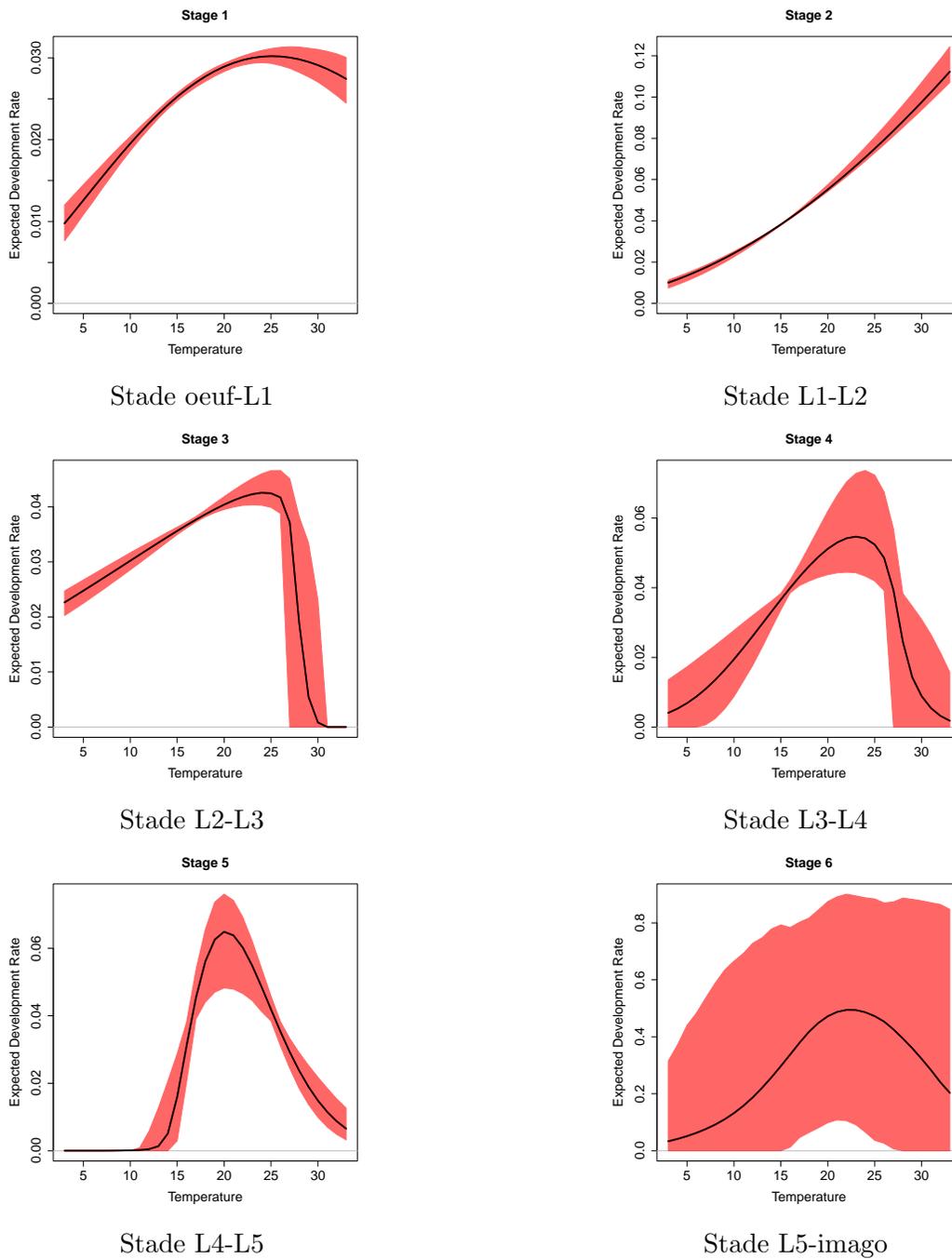


FIGURE 22 – Vitesse de développement en fonction des températures pour les différents stades de développement.

## Modèle 4 : SD

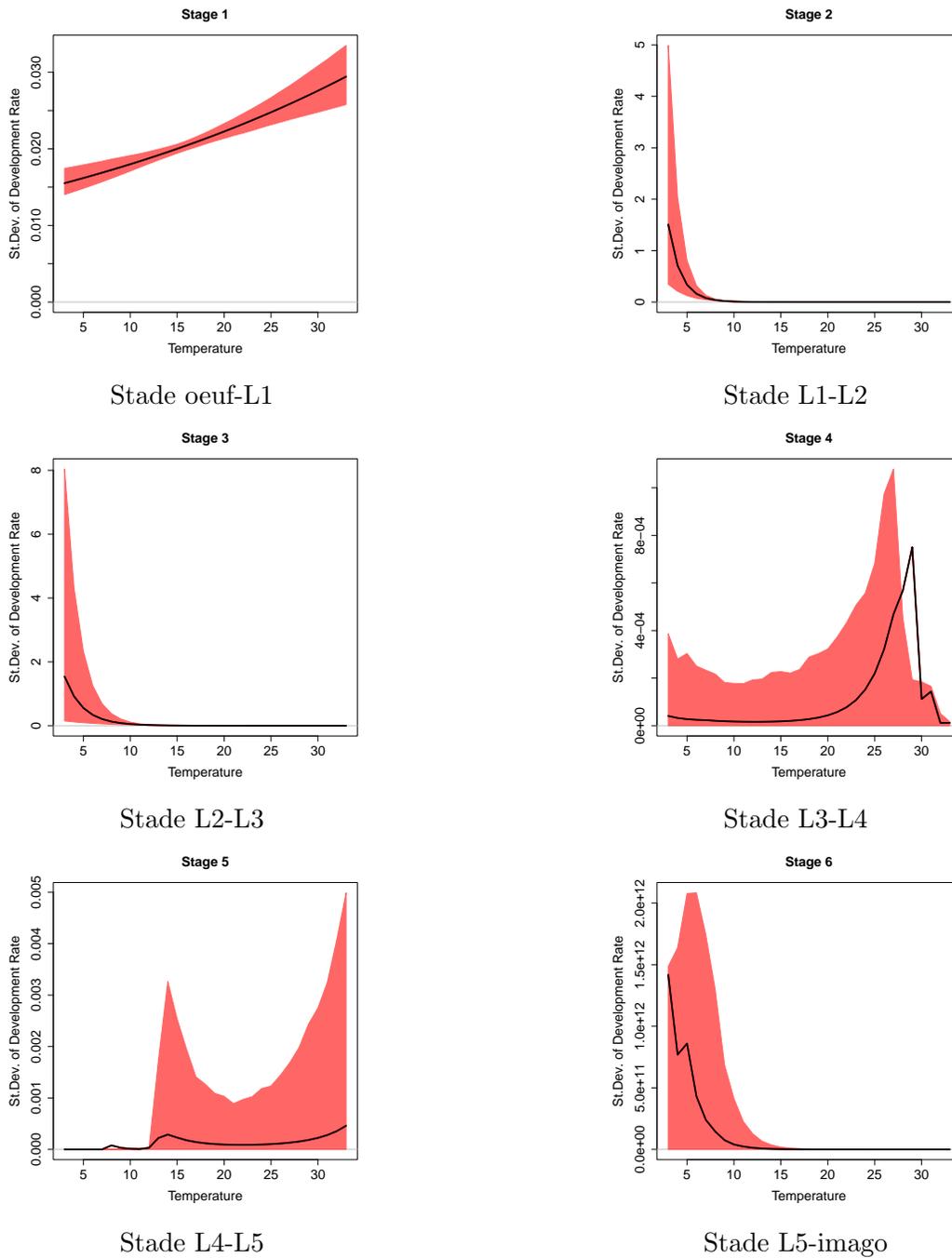


FIGURE 23 – Vitesse de développement en fonction des températures pour les différents stades de développement.

## Modèle 5 : Moyenne

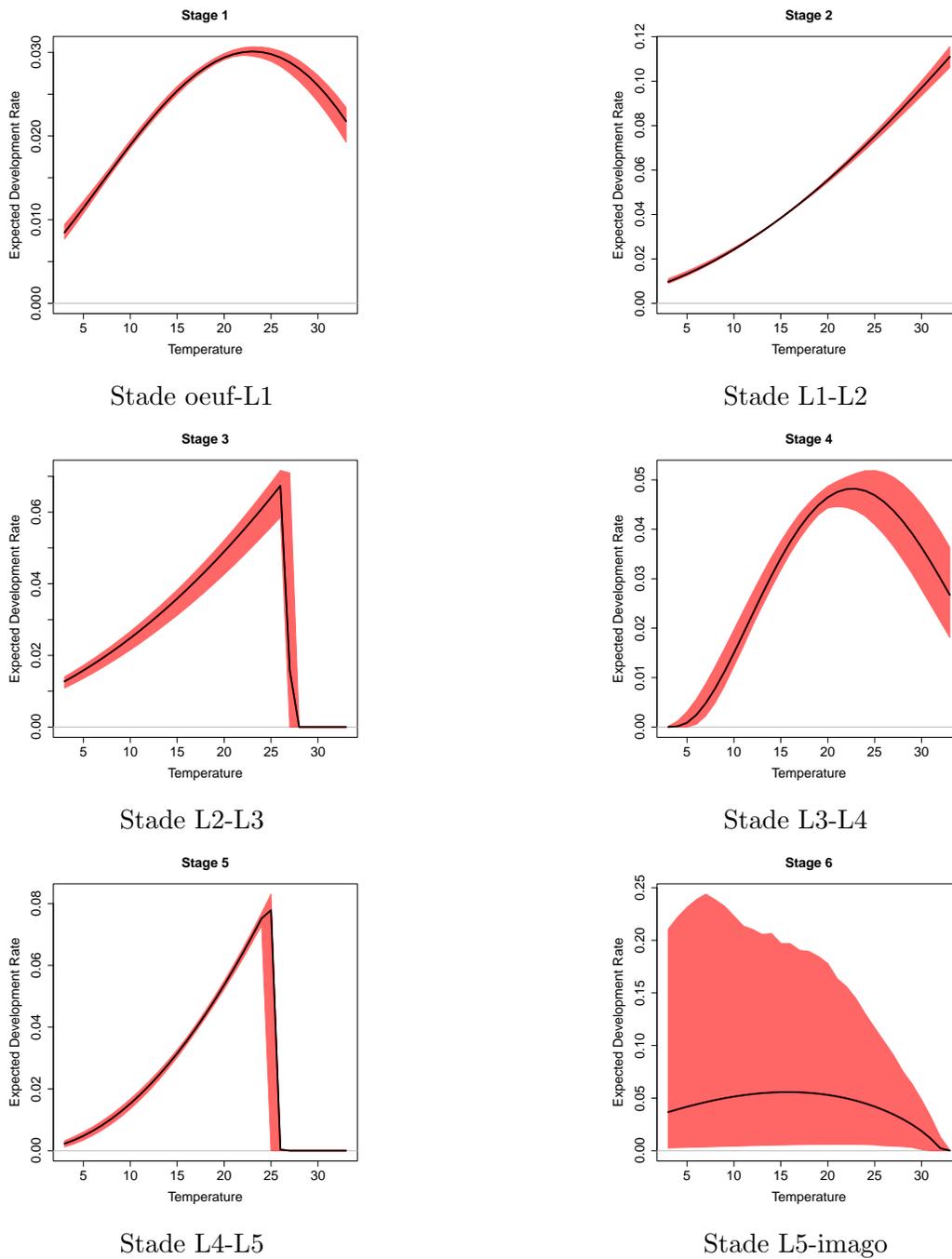


FIGURE 24 – Vitesse de développement en fonction des températures pour les différents stades de développement.

## Modèle 5 : SD

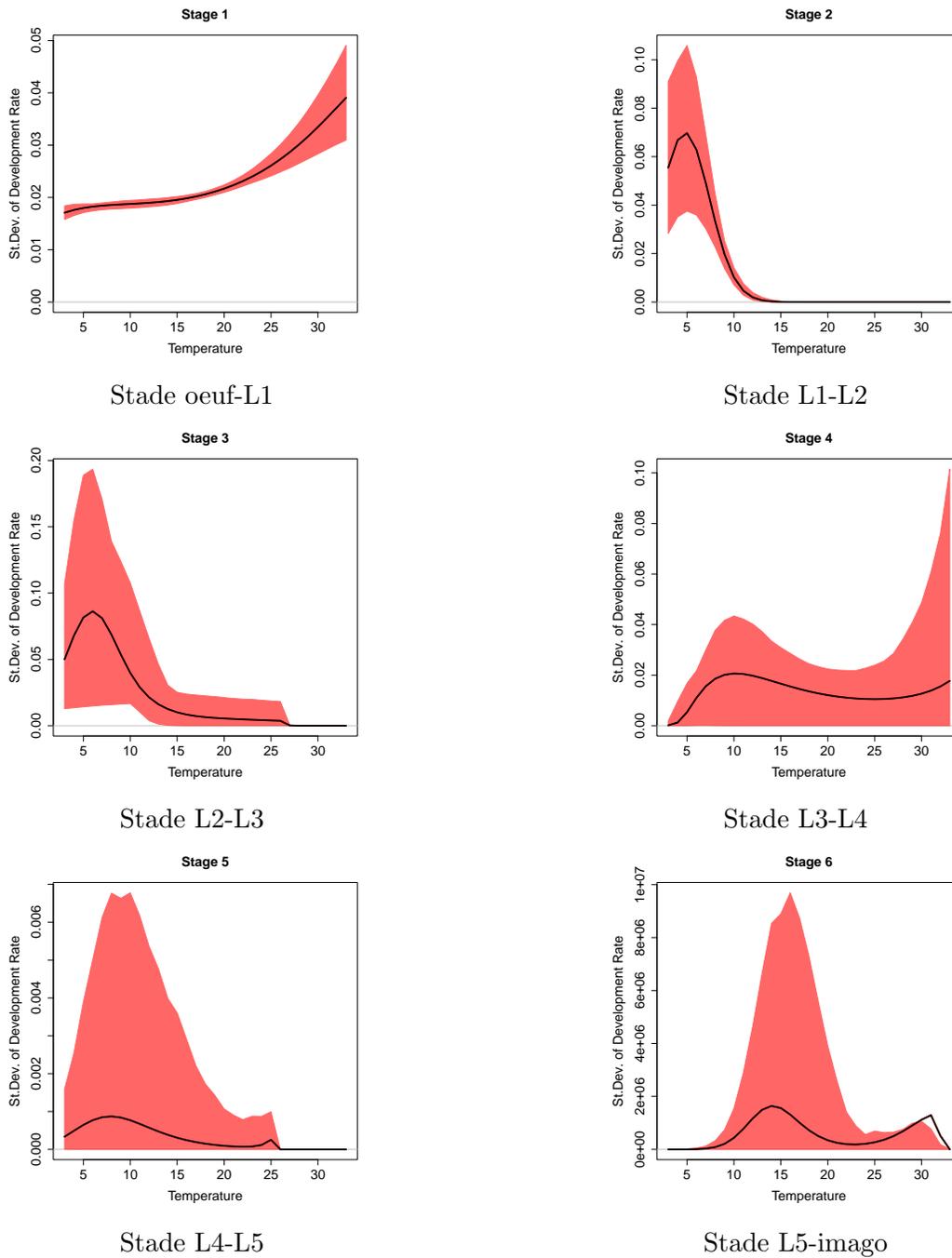


FIGURE 25 – Vitesse de développement en fonction des températures pour les différents stades de développement.

## Modèle 6 : Moyenne

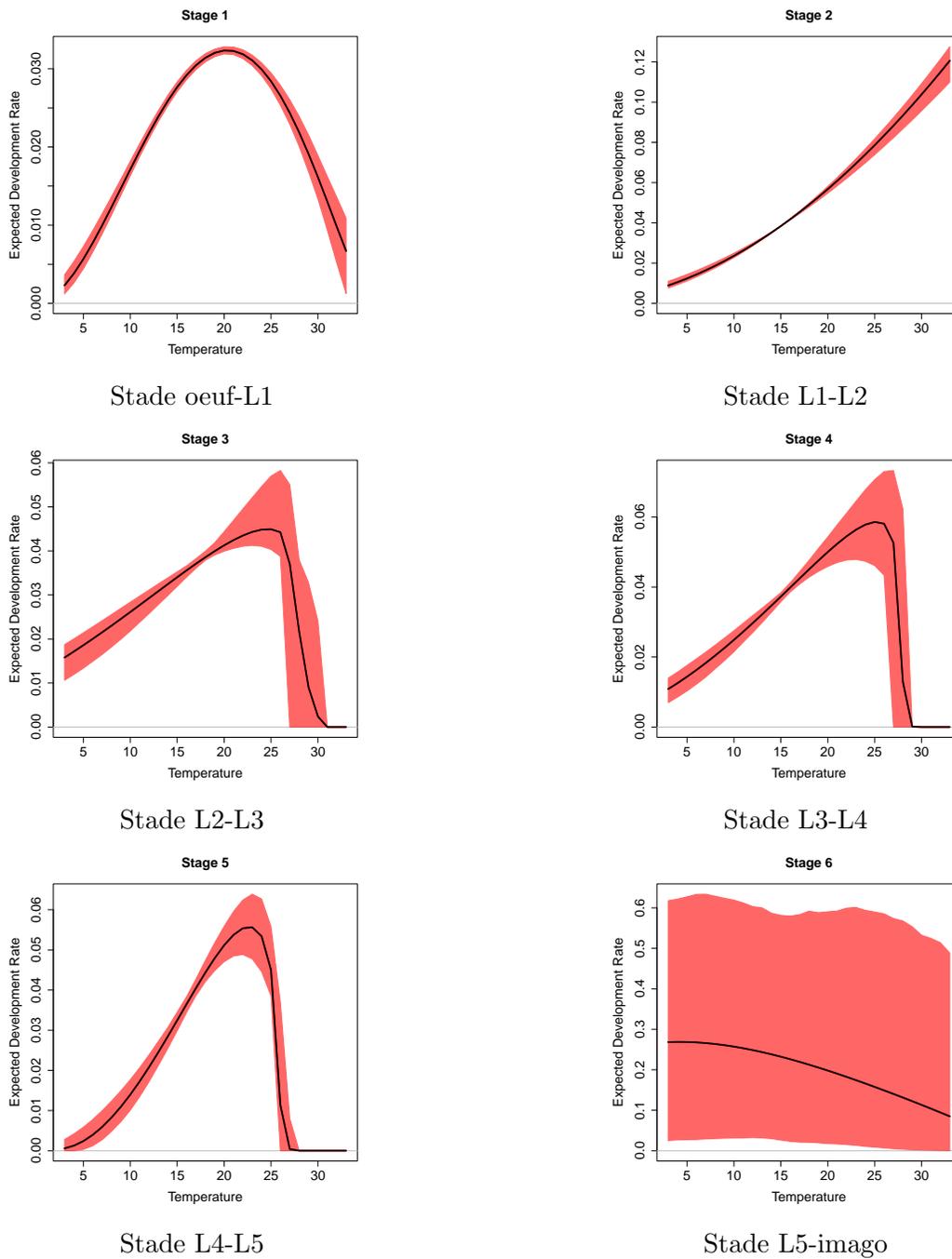


FIGURE 26 – Vitesse de développement en fonction des températures pour les différents stades de développement.

## Modèle 6 : SD

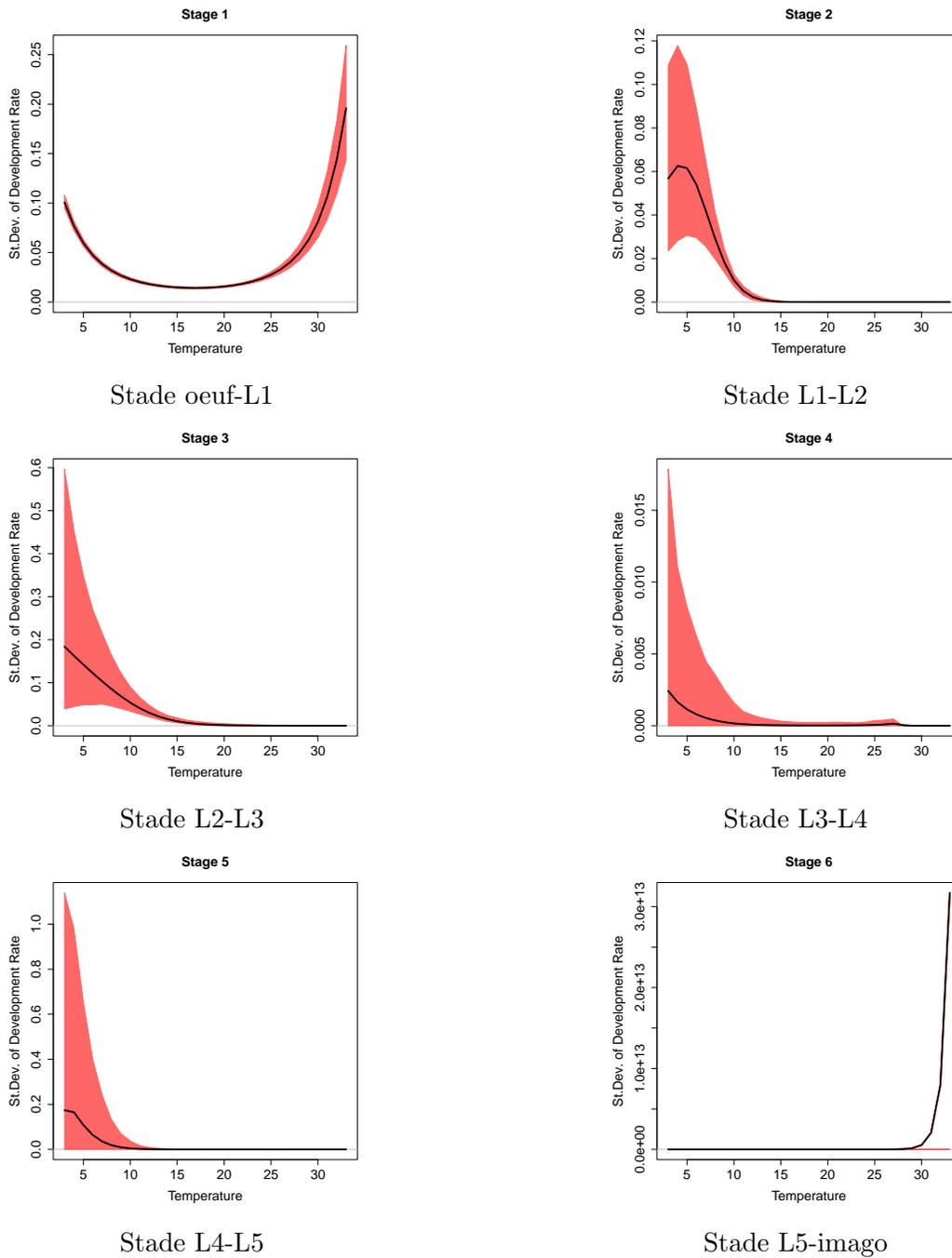


FIGURE 27 – Vitesse de développement en fonction des températures pour les différents stades de développement.