Modélisation et visualisation des liens entre cinétiques de variables agro-environnementales et qualité des produits dans une approche parcimonieuse et structurée

Girault Gnanguenon Guesse

THÈSE POUR OBTENIR LE GRADE DE DOCTEUR DE L'UNIVERSITÉ DE MONTPELLIER

En Biostatistique

École doctorale Information, Structures et Systèmes (I2S, ED 166)

Unité de recherche Mathématiques, Informatique et STatistique pour l'Environnement et l'Agronomie (MISTEA, UMR 0729)

Modélisation et visualisation des liens entre cinétiques de variables agro-environnementales et qualité des produits dans une approche parcimonieuse et structurée

Présentée par Girault B.S. GNANGUENON GUESSE

Le 22 Octobre 2021

Sous la direction de Nadine HILGERT et Thierry SIMONNEAU

Devant le jury composé de

Élodie BRUNEL-PICCININI, Maître de conférences, Université de Montpellier	Examinatrice
Bénédicte FONTEZ, Maître de conférences, Institut Agro - Montpellier SupAgro	Co-Encadrante de thèse
Romain GLÈLÈ KAKAÏ, Professeur des universités, Université d'Abomey-Calavi	Examinateur
Nadine HILGERT, Directrice de recherche, INRAE - Montpellier	Directrice de thèse
Patrice LOISEL, Chargé de recherche, INRAE - Montpellier	Co-Encadrant de thèse
Tristan MARY-HUARD, Chargé de recherche, INRAE - Le Moulon	Rapporteur
Philippe PIERI, Chargé de recherche, INRAE - Bordeaux	Examinateur
Thierry SIMONNEAU, Directeur de recherche, INRAE - Montpellier	Co-Directeur de thèse
Nancy TERRIER, Chargée de recherche, INRAE - Montpellier	Invitée
Anne-Françoise YAO, Professeur des universités, Université Clermont-Auvergne	Présidente / Rapporteure

UNIVERSITÉ DE MONTPELLIER

Dédicace

À :

➔ mon père Denis GNANGUENON GUESSE pour tous les efforts et sacrifices consentis. Tes attentes ont toujours été une marche qui une fois franchie, permettait d'entrevoir l'infinité des possibilités. J'accepte volontiers le relais.

➔ ma feue mère Marie Houesse BOKO pour ses conseils et son suivi. Celle dont la simple présence a permis beaucoup. Repose en paix.

Remerciements

Cette œuvre, aussi modeste soit-elle n'aurait pu être réalisée sans le concours, l'appui, les conseils et l'attention soutenue de plusieurs personnes. Ainsi, je saisis cette opportunité pour témoigner toute ma reconnaissance à l'équipe d'encadrement constituée de Nadine HILGERT, Thierry SIMONNEAU, Bénédicte FONTEZ et Patrice LOISEL. Travailler à vos côtés, est sans doute l'aventure scientifique et humaine la plus exaltante qu'il m'ait été donné de vivre à ce jour. Moi, j'en sors grandi malgré mes moments de doute. Votre présence et votre intérêt pour ces travaux ont rendu possible, l'atteinte des objectifs fixés.

Mes remerciements vont aussi à Tristan MARY-HUARD et Anne-Françoise YAO qui, malgré leurs occupations, ont pris le temps de rapporter cette thèse. J'exprime aussi toute ma gratitude aux autres membres du jury Élodie BRUNEL-PICCININI, Romain GLÈLÈ KAKAÏ, Philippe PIERI et Nancy TERRIER, notre invité. J'en profite aussi pour renouveler ma reconnaissance à Toussaint LOUGBEGNON et Laurent HOUESSOU qui m'ont initié et nourri mon intérêt pour la recherche scientifique. Cet intérêt a été une boussole qui m'a conduit à l'achèvement de cet œuvre. Sur ce chemin, j'ai croisé Pierre-Yves LOUIS et Yousri SLAOUI à qui je renouvelle aussi ma reconnaissance.

Cette reconnaissance va aussi à Isabelle Sanchez, qui par sa maîtrise avancée du langage R, m'a permis d'apprendre, d'avancer rapidement et d'avoir un regard externe sur la mise en forme du programme informatique que propose cette thèse. Je me saisis donc de l'occasion pour remercier tous les membres de l'UMR Mathématiques, Informatique et STatistique appliquées à l'Environnement et l'Agronomie (MISTEA), en particulier son directeur Pascal NEVEU et la jeune équipe de développement qui y travaille. Échanger avec eux sur nos tâches complètement différentes, m'a permis d'appréhender une autre facette des enjeux liés au big data en agronomie. Je ne saurais finir ce paragraphe sans une attention particulière pour Baptiste OGER, l'autre doctorant qui a partagé mon bureau, que dis-je notre bureau, pendant ces années de thèse.

En dehors de cet univers professionnel, je remercie aussi toutes mes fréquentations et amis qui ont fait de mon séjour à Montpellier, un grand moment de partage. Pêle-mêle un grand merci à Lynda pour son accueil dès les premiers jours ; à Bill, Judicael, Dan, Fabrice, Houefa, Brunel, Pamela, Eva, Thierry, Georges, Mariano, Nawalyath, Lorraine, Bachirou, David, Elysé, Gimmy, Antoine, Lionel, Samuel, Éric, Enock, ... un grand merci aussi.

Je renouvelle aussi toute ma reconnaissance et mon dévouement à ma grande fratrie, Gildas, Bernice, Gladis, Alida, Rodon, Marius, Laurenda, Christelle, Maryse, Tatiana. Aux

amis devenus frères voire plus, Achille, Immaculée, Martial, je dis simplement merci pour tout. Pour toutes mes omissions, je demande pardon tout en vous disant ma reconnaissance. Moi, je suis devenu qui je suis, en partie grâce à vous.

Table des matières

Table des figures

Liste des tableaux

Résumé

L'essor de l'agriculture numérique permet de plus en plus d'observer de manière automatisée et parfois à haute fréquence des dynamiques d'élaboration de la production et de sa qualité en fonction du climat. Les données issues de ces observations dynamiques peuvent être considérées comme des données fonctionnelles. Analyser ce nouveau type de données nécessite d'étendre les outils statistiques usuels au cas fonctionnel ou d'en proposer de nouveaux.

Nous avons proposé dans cette thèse une nouvelle approche (SpiceFP : Sparse and Structured Procedure to Identify Combined Effects of Functional Predictors) permettant d'expliquer les variations d'une variable réponse scalaire par deux ou trois prédicteurs fonctionnels dans un contexte d'influence conjointe de ces derniers. Une attention particulière a été apportée à l'interprétabilité des résultats via l'utilisation de classes d'intervalles combinées définissant une partition du domaine d'observation des facteurs explicatifs. Les développements récents autour des modèles LASSO (Least Absolute Shrinkage and Selection Operator) ont été adaptés pour estimer les régions d'influence dans la partition via une régression pénalisée généralisée. L'approche intègre aussi une double sélection, de modèles (parmi les partitions possibles) et de variables (pour une partition donnée) à partir des critères d'information AIC et BIC. La présentation méthodologique de l'approche, son étude grâce à des simulations ainsi qu'une étude de cas basée sur des données réelles ont été présentés dans le chapitre 2.

Les données réelles utilisées au cours de cette thèse proviennent d'une expérimentation viticole visant à mieux comprendre l'impact du changement climatique sur l'accumulation d'anthocyanes dans les baies. L'analyse de ces données dans le chapitre 3 à l'aide de l'approche SpiceFP que nous avons étendue a permis d'identifier un impact négatif des combinaisons matinales de faible irradiance (inférieure à environ $100\,\mu mol\,m^{-2}\,s^{-1}$ ou $45\,\mu mol\,m^{-2}\,s^{-1}$ selon l'état avancé-retardé des baies) et température élevée (supérieure à environ 25 °C). Une légère différence induite par la température de la nuit a été observée entre ces effets identifiés en matinée.

Dans le chapitre 4 de cette thèse, nous proposons une implémentation de l'approche proposée sous la forme d'un package **R**. Cette implémentation fournit un ensemble de fonctions permettant de construire les intervalles de classes suivant des échelles linéaire ou logarithmique, de transformer les prédicteurs fonctionnels grâces aux classes d'intervalles combinées puis de mettre en oeuvre l'approche en deux ou trois dimensions. D'autres fonctions facilitent la réalisation de post-traitements ou permettent à l'utilisateur de s'intéresser à d'autres modèles que ceux retenus par l'approche comme par exemple une moyenne de différents modèles.

Mots clés : Régressions pénalisées, Interaction, critères d'information, scalar-on-function, coefficients interprétables, microclimat de la vigne.

Abstract

The development of digital agriculture allows to observe at high frequency the dynamics of production according to the climate. Data from these dynamic observations can be considered as functional data. To analyze this new type of data, it is necessary to extend the usual statistical tools to the functional case or develop new ones.

In this thesis, we have proposed a new approach (SpiceFP : Sparse and Structured Procedure to Identify Combined Effects of Functional Predictors) to explain the variations of a scalar response variable by two or three functional predictors in a context of joint influence of these predictors. Particular attention was paid to the interpretability of the results through the use of combined interval classes defining a partition of the observation domain of the explanatory factors. Recent developments around LASSO (Least Absolute Shrinkage and Selection Operator) models have been adapted to estimate the areas of influence in the partition via a generalized penalized regression. The approach also integrates a double selection, of models (among the possible partitions) and of variables (areas inside a given partition) based on AIC and BIC information criteria. The methodological description of the approach, its study through simulations as well as a case study based on real data have been presented in chapter 2 of this thesis.

The real data used in this thesis were obtained from a vineyard experiment aimed at understanding the impact of climate change on anthcyanins accumulation in berries. Analysis of these data in chapter 3 using SpiceFP and one extension identified a negative impact of morning combinations of low irradiance (lower than about $100\,\mu mol\,m^{-2}\,s^{-1}$ or $45\,\mu mol\,m^{-2}\,s^{-1}$ depending on the advanced-delayed state of the berries) and high temperature (higher than about $25\,°C$). A slight difference associated with overnight temperature occurred between these effects identified in the morning.

In chapter 4 of this thesis, we propose an implementation of the proposed approach as an **R** package. This implementation provides a set of functions allowing to build the class intervals according to linear or logarithmic scales, to transform the functional predictors using the joint class intervals and finally to execute the approach in two or three dimensions. Other functions help to perform post-processing or allow the user to explore other models than those selected by the approach, such as an average of different models.

Keywords : Penalized regressions, Interaction, information criteria, scalar-on-function, interpretable coefficients, grapevine microclimate.

Chapitre 1

Introduction générale

1.1 Un défi pour la science des données : l'émergence de nouveaux capteurs et de grande quantité de données pour mieux comprendre les effets du climat sur les cultures

Ces dernières années, des variations importantes du climat ont été observées, obligeant parfois les êtres vivants à s'adapter. Les végétaux, qu'il s'agisse de végétation naturelle ou de cultures nécessitant une intervention humaine ne sont pas épargnés par ces variations [103] qui représentent non seulement un enjeu environnemental, mais surtout alimentaire et économique. C'est par exemple le cas de la filière viticole en France où les acteurs sont obligés de s'adapter aux bouleversements causés par les effets du changement climatique [56, 77, 60, 81]. Ces bouleversements affectent directement le suivi des vignes, les dates de vendanges et indirectement la qualité du raisin récolté. Cet état de chose n'est pas sans conséquences sur les différents produits finaux issus de la filière. Et outre cette filière, ces observations sont généralisables à l'échelle mondiale à d'autres productions agricoles vitales comme le blé [64], le riz [100], le maïs [54], etc.

Afin de garder le contrôle sur la quantité et la qualité de la production, il est nécessaire de mieux comprendre leurs déterminismes physiologiques internes pour les différentes cultures et leurs interactions avec le climat. Pour atteindre cet objectif, les chercheurs peuvent de plus en plus compter sur de récents développements technologiques (capteurs, robots, drones, etc.) visant à augmenter la qualité et le rendement tout en ayant un faible impact environnemental [71, 116]. La diversité de ces technologies permet de suivre la production d'une culture à différents stades mais aussi à différentes échelles allant de l'intégralité d'une parcelle à une plante, ou un élément de la plante (un épi sur un pied de maïs ou encore une baie de raisin sur une vigne par exemple).

Dans cette thèse, nous nous intéressons aux données fournies par différents capteurs : cer-

tains permettant d'effectuer des mesures du micro-climat à haute fréquence et d'autres à une fréquence plus grossière et renseignant sur l'évolution de la quantité/qualité de la production par culture. À partir de ces données, l'objectif est d'expliquer (différent de prédire) ou plus simplement d'identifier les conditions climatiques qui influencent la quantité ou la qualité de la production. La différence que nous soulignons entre explication et prédiction dans ce contexte réside dans l'interprétabilité du résultat. Donc, si un résultat permet d'estimer convenablement la quantité ou la qualité d'une production à partir des données du climat sans fournir clairement des leviers d'influence permettant d'inverser la tendance, il ne nous est pas utile. La filière qui nous servira plus précisément de cas d'application tout au long de nos travaux est la filière viticole et il sera question de mettre à jour des intervalles (plage de valeurs) de variables climatiques, favorables ou défavorables à l'accumulation d'anthocyanes dans les baies, un des métabolites essentiels pour la qualité des vins rouges notamment.

Pour ce premier chapitre, il est présenté dans la suite de la section 1, la description de l'expérimentation d'où proviennent les données de notre cas d'application, la question agronomique associée à cette expérimentation ainsi que sa formulation statistique. La section 2 présente une revue des modèles de régression pour données fonctionnelles disponibles dans la littérature et la section 3 expose brièvement l'approche parcimonieuse et structurée que nous avons développée et implémentée dans cette thèse pour identifier les effets combinés des prédicteurs fonctionnels. Dans la section 4, nous montrons comment elle a été utilisée pour apporter un début de réponse à la question agronomique associée au cas d'application. Le plan du présent document est ensuite détaillé dans la section 5.

1.1.1 Présentation de l'expérimentation

L'expérimentation sur laquelle nous avons pu nous appuyer pour cette thèse a été conduite par deux unités de recherche à savoir le Laboratoire d'Ecophysiologie des Plantes sous Stress Environnementaux (LEPSE) et celui des Sciences pour L'Œnologie (SPO), dans le cadre du projet européen Innovine. Elle s'est déroulée l'année 2014 sur des vignes de Syrah. Le plan des expériences est présenté dans la figure 1.1.1. On y observe trois rangs (blocs) de ceps de vignes numérotées de 1 à 3. Chaque rang contient $(10-2)\times 2 = 16$ ceps numérotés (premiers (1, 11) et derniers (10, 20) ceps de chaque dispositif non pris en compte dans l'étude pour éviter des effets de bordure), représentés par des étoiles rouges et subdivisés en deux groupes à savoir :

— un premier groupe de ceps dont les pieds sont recouverts par un dispositif appelé OpenTop (noté OT) afin de provoquer une élévation de la température (réchauffe les pieds des vignes),

— un deuxième groupe de ceps dont les pieds ne sont pas recouverts et qui servent de dispositif témoin (noté C pour « control »). Les notations utilisées dans le protocole expérimental seront conservés pour ces travaux.

Outre l'organisation par rang et dispositif, une différenciation a été aussi faite suivant l'exposition au soleil des baies de raisin, mesurée via la Densité de Flux de Photons Photosyn-

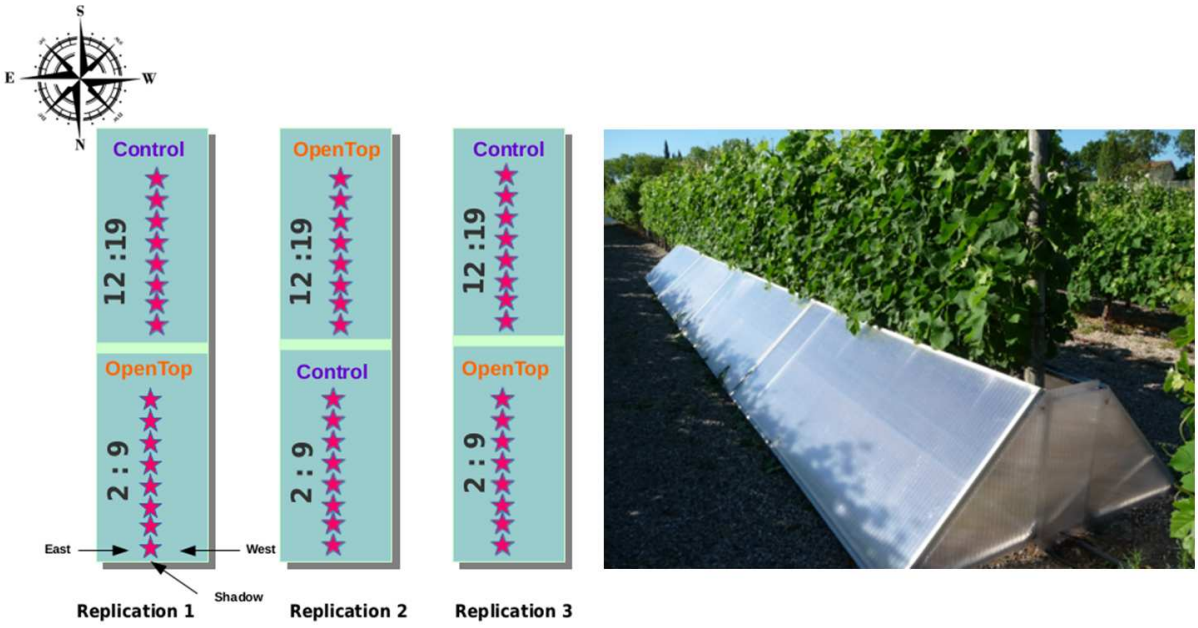


FIGURE 1.1.1 – Schéma du plan d'expérience (à gauche) et image du dispositif OpenTop (à droite)

thétiques (PPFD en $\mu mol/s/m^2$). Sur un même cep, on retrouve ainsi des baies exposées à l'est (E pour « East »), et à l'ouest (W pour « west »). Une organisation particulière du feuillage a permis aussi d'observer des baies à l'ombre (S pour « shadow »). Une telle expérimentation a permis de faire varier le microclimat à l'échelle de la vigne mais aussi à l'échelle de la baie de raisin afin d'observer leurs effets respectifs. Des mesures de température et d'irradiance (PPFD) sont réalisées toutes les douze (12) minutes sur des grappes de raisin pendant toute la période de maturation. Les capteurs de température sont à l'échelle du grain et ceux d'irradiance à l'échelle du demi-rang. La qualité du raisin est quant à elle mesurée à intervalle d'environ une semaine par un équipement automatisé d'estimation de la concentration en anthocyanes des baies de raisin, sans destruction des échantillons. Cet équipement, le multiplex, renseigne sur un indice nommé Indice de Ferari (FI). Il s'agit d'une mesure non-destructive de la teneur en anthocyanes des grappes [1]. D'un point de vue agronomique, la question de recherche revient à expliquer les variations hebdomadaires de l'Indice de Ferari par les observations à haute fréquence du micro-climat.

Avant de nous intéresser au pendant statistique de la question agronomique formulée dans le précédent paragraphe, quelques précisions sont nécessaires. Intéressons-nous à l'individu statistique. Il est représenté dans cette expérimentation par une combinaison de modalités d'observation ordonnées comme suit : l'année (nous ne nous intéresserons qu'à l'année 2014), le numéro du rang (1, 2 ou 3), la présence ou l'absence du dispositif OpenTop (OT ou

C), l'exposition (E, W ou S) et le numéro du cep (un nombre entre 2 à 9 ou 12 à 19). Ainsi, l'individu 2014.2.C.W_4 représente une baie ayant été observée l'année 2014 et située à l'ouest du cep 4 sur le demi-rang témoin du rang 2. Les mesures de température sont effectuées à l'échelle de la baie. L'Indice de Ferari est mesuré pour un groupe de baie (grappe de raisin). L'irradiance est mesurée à l'échelle d'un demi-rang. Un demi-rang contient 8 ceps et représente la moitié d'un rang doté ou non du dispositif OpenTop. Lorsque l'observation est effectuée à l'échelle du demi-rang, le numéro de cep (élément après _) est retiré de la nomenclature. Au total le plan d'expérience compte $n = 144$ individus statistiques ($i = 1, \ldots, n$).

1.1.2 Formulation statistique de la question agronomique

Notons respectivement $\mathcal{A}$ et $\mathcal{B}$ les observations de température et d'irradiance. Fournies à différents instants $t \in T$ (toutes les 12 minutes durant une période d'intérêt), $\mathcal{A}_i(t) \in \mathbb{R}$ et $\mathcal{B}_i(t) \in \mathbb{R}$ représentent respectivement les valeurs prises par les fonctions $\mathcal{A}_i$ et $\mathcal{B}_i$ à l'instant t. Ces différentes données sont des mesures répétées au fil du temps sur les mêmes individus dans l'optique de pouvoir capter une dynamique, une évolution. Ces observations longitudinales ont l'habitude d'être fortement corrélées les unes aux autres. Ce sont des séries temporelles. Mais dans ces travaux, nous nous intéressons à elles comme appartenant à la famille des variables fonctionnelles regroupant courbes, spectres, images, ... [89]. Une variable aléatoire $\mathcal{X}$ est dite fonctionnelle si ses valeurs sont dans un espace de dimension infinie et une observation X de $\mathcal{X}$ est appelée donnée fonctionnelle. En réalité, seulement quelques points discrets du phénomène continu sont observés $\mathcal{X} = \{X(t) : t \in T\}$ [29]. Bien que ce domaine de recherches soit récent en statistiques (il apparaît avec l'avènement des données massives), différents travaux se sont intéressés aux problèmes statistiques les impliquant, qu'il s'agisse de régression [75] , de clustering [49], de classification [7], de statistique multivariée [104], ou non paramétrique [29], etc.

Notre attention sera portée sur les régressions impliquant des variables fonctionnelles. En fonction de l'implication de ces variables dans la régression, en tant que variable réponse ou variable explicative, on peut distinguer trois types de régressions : celles de type scalar-on-function où la variable fonctionnelle est une variable explicative, celles de type function-on-scalar où la variable fonctionnelle est une variable réponse puis celles de type function-on-function où la variable explicative et la variable réponse sont des variables fonctionnelles [58].

Dans le cadre de cette expérimentation, les variables fonctionnelles que sont la température et l'irradiance sont des variables explicatives. Elles sont toutes deux fonction de t et observées aux mêmes instants. La variable réponse provient quant à elle des courbes d'indice de Ferari, qui sont aussi des variables fonctionnelles $(FI_i)_{i=1,\ldots,n} = \{FI_i(d) : d \in d_1, \ldots, d_9\}$. Cet indice est observé quasi-hebdomadairement pendant 8 semaines (s), soit 9 dates d. Si l'objectif avait été d'expliquer toute la courbe des Indices de Ferari par les courbes climatiques, nous aurions eu recours à des régressions de type function-on-function [65]. Deux obstacles mériteraient toutefois d'être soulignés pour une telle modélisation : 1- les courbes d'indice de Ferari sont construites à partir 9 points d'observation (très peu, on risquerait de

mal reconstruire les courbes à expliquer) ; celles du micro-climat le sont avec plus de 6000 points d'observation (beaucoup plus d'informations) 2- en cas de variation (hebdomadaire, ou par quinzaine, etc.) du processus physiologique sous-jacent, un modèle global pour les 8 semaines d'observation pourrait s'avérer non informatif. Pour limiter ce risque, le choix a été fait de rechercher des plages d'influence climatique semaine par semaine. Ainsi, pour une semaine donnée, la variable réponse est la variation de l'Indice de Ferari $\Delta FI_i \in \mathbb{R}, i = 1, \ldots, n$ qui correspond à la différence de l'indice $FI_i(s)$ entre deux dates d_s et d_{s+1} délimitant une période s. d_s et d_{s+1} correspondent aussi respectivement aux limites inférieure ($\underline{T}$) et supérieure ($\bar{T}$) de T. Nous nous intéresserons donc dans ces travaux aux régressions de type scalar-on-function [90]. Nous prendrons aussi en compte à chaque semaine, la valeur de l'indice en début de semaine. D'un point de vue statistique, il sera donc question d'identifier une fonction $\mathcal{F}$ telle que :

$$\Delta FI_i(s) := FI_i(d_{s+1}) - FI_i(d_s); \qquad s = 1, \ldots, 8 \tag{1.1.1}$$

$$= \mathcal{F}_s\left(\mathcal{A}_i(t), \mathcal{B}_i(t), FI_i(d_s)\right); \quad t \in [d_s, d_{s+1}] \tag{1.1.2}$$

Les fonctions F_s seront estimées à s fixé, de façon autonome. L'évolution de l'Indice de Ferari sera donc modélisée semaine par semaine. Aussi, chaque fonction F_s devra être aisément interprétable. Dans le contexte de ces données, cela revient à identifier des combinaisons de plages de température et d'irradiance influençant la dynamique de l'indice de Ferari.

1.2 Modèles de régression pour données fonctionnelles

Les premiers modèles de régression pour données fonctionnelles ont été développés pour prédire une variable réponse scalaire centrée y_i avec une variable explicative fonctionnelle $x_i = \{x_i(t) : t \in T\}$. [30] répartit en 3 catégories les approches pouvant être utilisées pour solutionner ce problème à savoir : les outils usuels de la statistique multivariée, la modélisation spécifique aux données fonctionnelles ou une approche basée sur des techniques d'agrégation de modèles tout en évitant ou contrôlant le sur-ajustement. Dans cette section, nous nous intéressons dans un premier temps à chacune des catégories d'approches précitées puis ensuite aux régressions impliquant plusieurs prédicteurs fonctionnels.

1.2.1 Modèles de régression pénalisée

Utiliser les méthodes usuelles de la statistique multivariée revient généralement à omettre l'idée suivant laquelle les observations $x_i(t)$ sont des observations discrètes de fonctions continues à des instants t et à les considérer comme $\mathcal{T} = card(T)$ variables indépendantes $x_{i,1}, \ldots, x_{i,j}, \ldots, x_{i,\mathcal{T}}$. Ainsi, avec ces variables "*supposées indépendantes*", un modèle de régression linéaire multiple s'écrirait :

$$y_i = x_{i,1}\beta_1 + \ldots + x_{i,j}\beta_j + \ldots + x_{i,\mathcal{T}}\beta_{\mathcal{T}} + \varepsilon_i \; ; i = 1, \ldots, n \tag{1.2.1}$$

avec $\beta_1, \ldots, \beta_{\mathcal{T}}$ les coefficients à estimer et ε_i l'erreur du modèle. Hormis la simplicité d'un tel modèle, les hypothèses H1 et H2 ne sont pas vérifiées :

- **H1** $Rang(x'x) = \mathcal{T}$ **c'est à dire que** $(x'x)^{-1}$ **existe :** dans le cadre de l'équation (1.2.1), les valeurs $x_{i,.}$ correspondent à des mesures répétées sur un même individu statistique i à chaque instant t. Ce qui génère de la colinéarité.
- **H2** $n > \mathcal{T}$ **c'est à dire que le nombre d'individus soit supérieur au nombre de variables :** ces variables fonctionnelles étant observées à grande fréquence, il est habituel que le nombre d'instants ou de pas d'observations dépasse largement le nombre d'individus. Une alternative naïve au non-respect de cette deuxième hypothèse (et uniquement cette deuxième) serait de recourir à une sélection de variables à l'aide d'une régression pas à pas (approche coûteuse en temps de calcul) ascendante [10]. Elle permettra de démarrer avec un modèle vide et de rajouter à l'aide d'un critère à chaque étape le terme le plus significatif tout en gardant à l'esprit que le nombre de variables maximal dans le modèle devra être inférieur à n. Une telle solution permet d'identifier certains instants d'observation t ayant une influence sur la variable réponse. Une des variantes de la régression pas à pas est descendante et nécessite l'estimation du modèle complet (sans intercept) avec $\mathcal{T}$ coefficients, ce qui nous ramène au problème que pose la deuxième hypothèse. Dans ce cas, il est possible d'utiliser des méthodes de régression pénalisée que nous introduisons dans le paragraphe suivant.

Pour gérer le contexte du nombre d'individus inférieur au nombre de variables, il est possible d'estimer les coefficients par le biais d'une régression Ridge [47]. Il s'agit d'une régression linéaire avec une contrainte quadratique sur les coefficients qui peuvent être estimés via l'expression :

$$\beta^{ridge} = \underset{\beta \in \mathbb{R}^{\mathcal{T}}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{\mathcal{T}} x_{i,j}\beta_j \right)^2 + \lambda \sum_{j=1}^{\mathcal{T}} \beta_j^2 \tag{1.2.2}$$

avec λ le paramètre qui contrôle la pénalisation. Le modèle (1.2.1) est un modèle sans intercept (y étant centré). Même si le modèle avait été écrit avec un intercept, ce dernier n'est pas pénalisé. Différentes méthodes peuvent être utilisées pour la sélection du paramètre λ. Nous y reviendrons ultérieurement dans ce chapitre. La pénalisation Ridge est de type norme l_2 et permet d'estimer tous les coefficients mais en cherchant à les réduire (faisant tendre vers 0), son appellation usuelle étant "shrinkage". Outre cette pénalisation, d'autres modèles pénalisés sont présents dans la littérature et le lecteur pourra trouver une revue de ces méthodes au sein des travaux de [31].

L'une des pénalisations les plus utilisées ces dernières années est le Least Absolute Shrinkage and Selection Operator (LASSO) [112]. Son estimateur s'écrit :

$$\beta^{lasso} = \underset{\beta \in \mathbb{R}^{\mathcal{T}}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{\mathcal{T}} x_{i,j}\beta_j \right)^2 + \lambda \sum_{j=1}^{\mathcal{T}} |\beta_j| \tag{1.2.3}$$

Le Lasso est une pénalisation de type norme l_1. Elle permet non seulement de pouvoir résoudre des problèmes de régression pour lesquels $n < \mathcal{T}$ mais fournit aussi une solution

parcimonieuse en annulant certains coefficients β_j. Ce qui permet indirectement de faire de la sélection de variables. Le LASSO présente toutefois quelques limites. Lorsque $\mathcal{T} > n$, il sélectionne au maximum n variables. Une autre de ses limites est qu'en cas de forte colinéarité entre les prédicteurs comme c'est le cas pour les vecteurs composant la variable fonctionnelle x, le LASSO tend à sélectionner une variable au détriment de celles qui lui sont fortement corrélées [45]. Pour lever cette contrainte, différentes adaptations ont été formulées comme :

— **l'Elastic-net [126] :** cette pénalisation associe dans un même modèle une pénalisation de norme l_1 (LASSO) et une pénalisation de norme l_2 (Ridge). La pénalisation de norme l_1 génère un modèle parcimonieux tandis que celle de norme l_2 supprime la limitation du nombre de variables sélectionnées lorsque $\mathcal{T} > n$, encourage un "*effet de groupe*" dans la sélection des variables et stabilise le chemin de régularisation associée à la norme l_1. L'estimateur Elastic-net s'écrit :

$$\beta^{en} = \underset{\beta \in \mathbb{R}^{\mathcal{T}}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{\mathcal{T}} x_{i,j}\beta_j \right)^2 + \lambda_1 \sum_{j=1}^{\mathcal{T}} |\beta_j| + \lambda_2 \sum_{j=1}^{\mathcal{T}} \beta_j^2 \tag{1.2.4}$$

avec $\lambda_1 \geq 0$ et $\lambda_2 \geq 0$ les paramètres de pénalisation. Pour des raisons d'estimation des différents paramètres, il est fréquent de reformuler cette pénalisation sous la forme $J(\beta) = (1-\alpha) \sum_{j=1}^{\mathcal{T}} |\beta_j| + \alpha \sum_{j=1}^{\mathcal{T}} \beta_j^2$ avec $0 \leq \alpha = \frac{\lambda_2}{\lambda_2+\lambda_1} \leq 1$ un hyperparamètre (à définir) contrôlant la contribution de chaque pénalité. Avec une telle reformulation et en fixant l'hyperparamètre α, seul un paramètre de régularisation λ (à estimer) est nécessaire.

— **le Fused Lasso [113] :** cette modification du LASSO prend en compte un voisinage entre les variables explicatives et pénalise en utilisant une norme l_1 les coefficients mais aussi la différence entre deux coefficients consécutifs (ce qui induit une constance locale des coefficients). En effet, pour que ce voisinage puisse être pris en compte, un notion d'ordre (ou de dépendance) est introduite entre les variables explicatives. Cette notion de dépendance permet de gérer la colinéarité associée aux observations temporelles. Son estimateur se formule comme suit :

$$\beta^{fl} = \underset{\beta \in \mathbb{R}^{\mathcal{T}}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{\mathcal{T}} x_{i,j}\beta_j \right)^2 + \lambda_p \sum_{j=1}^{\mathcal{T}} |\beta_j| + \lambda_f \sum_{j=2}^{\mathcal{T}} |\beta_j - \beta_{j-1}| \tag{1.2.5}$$

avec λ_p le paramètre pénalisant la parcimonie des coefficients et λ_f le paramètre pénalisant leur fusion. En ôtant $\lambda_f \sum_{j=2}^{\mathcal{T}} |\beta_j - \beta_{j-1}|$ ou en fixant $\lambda_f = 0$, on retrouve bien l'estimateur du LASSO présenté dans l'équation (1.2.3).

— **le Group Lasso [123] :** l'originalité de cette version du LASSO réside dans : *i)* la répartition des $\mathcal{T}$ variables au sein de K groupes $G = \{g_1, g_2, \ldots, g_K\}$ à fournir et *ii)* la sélection parcimonieuse de groupes et non de variables. Notons $\beta^{(k)}$ le vecteur de

coefficients associé au groupe g_K, l'estimateur du Group LASSO s'écrit :

$$\beta^{gl} = \underset{\beta \in \mathbb{R}^{\mathcal{T}}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{\mathcal{T}} x_{i,j}\beta_j \right)^2 + \lambda_g \sum_{k=1}^{K} \sqrt{w_k} ||\beta^{(k)}||_2 \tag{1.2.6}$$

avec λ_g le paramètre de régularisation, w_k le poids associé au groupe g_k et $||.||_2$ la norme l_2 . Lorsque les groupes sont constitués d'une seule variable, le Group LASSO équivaut au LASSO. [106] ont introduit une variante de ce modèle intitulée le Sparse Group Lasso qui représente une combinaison linéaire associant LASSO et Group LASSO, fournissant ainsi des coefficients parcimonieux à la fois entre et dans les groupes de variables.

La liste des adaptations présentées ici n'est pas exhaustive et plusieurs autres travaux ont introduit diverses pénalisations [82, 119, 127], etc. [114] ont proposé dans leurs travaux une encapsulation de nombre de ces modèles en pénalisant via la norme l_1 le produit entre une matrice de pénalisation et le vecteur des coefficients. La forme de cette matrice définit le modèle pénalisé auquel on s'intéresse.

Pour ces différents modèles, le choix d'un ou de plusieurs paramètres de pénalisation peut être effectué à l'aide des techniques de type cross-validation [32], bootstrap [59] ou encore par l'utilisation d'un critère d'information [33]. Aussi, [27] ont introduit le Least Angle Regression (LARS), un algorithme de sélection de modèle, moins gourmand en temps de calcul que les méthodes traditionnelles de sélection de modèles. Une simple modification de l'algorithme LARS implémente le LASSO et calcule toutes les estimations LASSO possibles pour un problème donné, facilitant ainsi le choix du paramètre de pénalisation λ.

1.2.2 Modèles spécifiques aux variables fonctionnelles

Nous nous intéressons dans cette section à diverses extensions de la théorie du modèle linéaire classique au cadre des données fonctionnelles. Nous prendrons comme cas d'étude, la prédiction d'une variable scalaire y_i par une variable fonctionnelle x_i. Les observations x_i, dépendant de t, sont considérées comme des observations à pas discrets d'une variable continue. Différentes techniques sont utilisées pour reconstruire cet élément continu. La reconstruction peut se faire par le biais d'une interpolation lorsque les observations sont obtenues sans erreur ou par le biais d'un lissage lorsque les observations sont bruitées. Ce deuxième cas peut conduire à l'estimation d'une courbe $\widehat{x_i}$ pour chaque individu statistique (lorsque qu'on dispose suffisamment d'observations à l'échelle d'un individu) ou à l'inférence de structure de population comme des fonctions de moyenne ou de covariance. Cette vision fonctionnelle des données présente aussi l'avantage de pouvoir travailler en fonction des méthodologies, avec des dérivées de courbes [29]. Aussi, les observations $x_i(t)$ peuvent être fournies à pas constant (intervalle régulier) ou non. Elles peuvent être observées avec des données manquantes. Différentes étapes de "pré-traitement des données fonctionnelles" sont alors nécessaires avant de s'intéresser à leur modélisation [50].

Le modèle de régression linéaire fonctionnelle [17, 89] avec réponse scalaire est l'extension naturelle du modèle de régression linéaire multiple présentée dans l'équation (1.2.1) et peut-être formulée comme suit :

$$y_i = \int_T x_i(t)\beta(t)\,dt + \varepsilon_i,\ i = 1, \ldots, n \tag{1.2.7}$$

avec $\beta(t)$ la fonction coefficient et ε_i les résidus i.i.d de moyenne nulle et de variance constante σ^2. Différents types de méthodes d'estimation de la fonction coefficient sont présentes dans la littérature [58, 90, 88, 120] ; etc. On peut citer par exemple les méthodes de réduction de dimension suivantes :

1. **les méthodes utilisant une base de fonctions prédéfinies (de type spline [18] par exemple) et qui peuvent recourir ou non à des pénalisations :** l'idée de ces méthodes est de se doter d'une base orthonormée $(\varphi_k)_{k=1,\ldots,K}$ de taille K dans l'espace des fonctions permettant une reformulation de la fonction coefficient $\beta(t) = \sum_{k=1}^{K} \mathrm{b}_k\varphi_k(t)$ à l'aide d'un vecteur de coefficients de projection $\mathrm{b} \in \mathbb{R}^K$. On peut ainsi écrire :

$$\int_T x(t)\beta(t)\,dt = \sum_{k=1}^{K} \mathrm{b}_k \int_T x(t)\varphi_k(t)dt \tag{1.2.8}$$

$$= \sum_{k=1}^{K} \mathrm{x}_k\mathrm{b}_k, \quad \text{avec } x(t) = \sum_{k=1}^{K} \mathrm{x}_k\varphi_k(t) \tag{1.2.9}$$

   Une telle reformulation permet de réduire le modèle (1.2.7) au modèle (1.2.1) et offre une possibilité d'estimation des b_k. Une fois les $\widehat{\mathrm{b}}_k$ obtenus, une estimation de la fonction coefficient peut être obtenue via l'expression $\widehat{\beta}(t) = \sum_{k=1}^{K} \widehat{\mathrm{b}}_k\varphi_k(t)$. Un des désavantages de ces méthodes est que l'estimateur des coefficients dépend de la forme de la base de fonctions (φ_k), choisie de manière assez subjective en se basant sur la régularité des x_i (celle de β n'étant pas connue). La contrainte liée au choix de K peut être levée en ayant recours à une pénalisation des coefficients b_k. Dans ce cas, un estimateur $\widehat{\mathrm{b}}$ peut s'écrire :

$$\widehat{\mathrm{b}} = \underset{\mathrm{b}\in\mathbb{R}^K}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - \int_T x_i(t) \left(\sum_{k=1}^{K} \mathrm{b}_k\varphi_k(t) \right) dt \right)^2 + \lambda P(\mathrm{b}) \tag{1.2.10}$$

   avec $\lambda > 0$ le paramètre de pénalisation et $P(\mathrm{b})$ une fonction de pénalisation. La forme de cette fonction de pénalisation varie en fonction de la base utilisée et des propositions sont présentes dans littérature qu'il s'agisse des splines [69], des ondelettes [124], etc. En ce qui concerne les bases de fonctions, même si l'utilisateur a le choix entre une diversité de bases existantes, il n'est pas toujours évident que ces bases permettent la prise en compte de particularités associées aux données fonctionnelles. Dans ce cas, l'utilisateur peut s'intéresser à d'autres types de bases de fonctions comme celles discutées dans le point 2.

2. **les méthodes utilisant une base de fonctions construite à partir des données fonctionnelles :** la particularité de ces méthodes réside dans le fait que les bases de fonctions ne sont pas pré-spécifiées mais elles sont estimées à partir des observations x_i, et donc adaptées à la régularité des x_i. Un exemple fréquent est l'utilisation d'une base de fonctions propres, associée à la fonction de covariance estimée par le biais d'une Analyse en Composantes Principales Fonctionnelles [105]. Le lecteur pourra se référer à la section 3.2 du livre de [58] pour une présentation détaillée de la notion de Composantes Principales Fonctionnelles (CPF) puis à la section 4.6 du même livre pour celle de la régression sur CPF. En substance, il est question, partant des observations fonctionnelles $x = \{x_i(t);\, i = 1, \ldots, n;\, t \in T\}$, de la fonction de moyenne ayant pour estimateur $\widehat{\mu}(t) = \sum_{i=1}^{n} x_i(t)$ et de la fonction de covariance ayant pour estimateur $\widehat{c}(t,s) = \frac{1}{n}\sum_{i=1}^{n}(x_i(t) - \widehat{\mu}(t))(x_i(s) - \widehat{\mu}(s))$ de :
   - décomposer les observations fonctionnelles dans une base de Karhunen-Loeve via l'expression :
   $$x(t) = \mu(t) + \sum_{j=1}^{\infty} \xi_j \nu_j(t) \tag{1.2.11}$$
   avec $\nu_j(t)$ les Composantes Principales Fonctionnelles (ce sont les fonctions propres de la fonction de covariance c et donc solutions de l'équation $\int c(t,s)\nu(s)\,ds = \lambda\nu(t)$) et ξ_j des variables aléatoires appelées scores et qui sont données par $\xi_j = \int (x(t) - \mu(t))\,\nu_j(t)\,dt$
   - reformuler le modèle linéaire fonctionnel en utilisant une approximation des observations fonctionnelles à l'aide de p CPF ($j = 1, \ldots, p$) :
   $$y_i = \int_T \beta(t) \left(\widehat{\mu}(t) + \sum_{j=1}^{p} \widehat{\xi}_{i,j}\widehat{\nu}_j(t) \right) dt + \varepsilon_i \tag{1.2.12}$$
   $$= \beta_0 + \sum_{j=1}^{p} \widehat{\xi}_{i,j}\beta_j + \varepsilon_i \tag{1.2.13}$$
   avec $\beta_0 = \int_T \beta(t)\widehat{\mu}(t)\,dt$ et $\beta_j = \int_T \beta(t)\widehat{\nu}_j(t)\,dt$. L'une des importantes étapes de cette méthodes reste le choix de p, le nombre de composantes principales fonctionnelles estimées que l'on garde pour l'ajustement du modèle.

   Une extension de cette approche aux modèle linéaires généralisés est proposée par [92]. Aussi [25] propose une extension de la régression des moindres carrés partiels (une approche itérative et une alternative intéressante aux projections sur la base des composantes principales) pour les données fonctionnelles.

Il est essentiel de préciser que les deux familles de méthodes fonctionnelles (1 et 2) que nous venons de présenter ont deux caractéristiques communes :

— **ces méthodes fonctionnelles sont des modèles linéaires (paramétriques) :** cette précision est importante car elle nous permet d'introduire l'existence de méthodes fonctionnelles non-paramétriques. Exactement comme dans le cadre des méthodes fonctionnelles paramétriques, les méthodes fonctionnelles non-paramétriques sont une extension des méthodes classiques non-paramétriques. Une introduction à ces modèles peut être retrouvée dans le livre de [29]. Dans un contexte où l'on n'a pas toujours suffisamment d'informations sur les liens existant entre les variables, les méthodes fonctionnelles non-paramétriques permettent de faire le moins d'hypothèses possibles sur la forme du lien entre la réponse y et la variable explicative fonctionnelle x, les distributions des variables aléatoires ou encore les paramètres de la régression. Ce qui permet de s'inspirer d'informations provenant des données pour la construction du modèle.

— **ces méthodes fonctionnelles permettent d'ajuster des modèles prédictifs, de type boîte noire qui ne sont pas aisément interprétables :** l'interprétation du lien entre réponse et prédicteurs fonctionnels devient de plus en plus difficile au fur et à mesure que la forme de la fonction de coefficients $\beta(t)$ devient compliquée. Les formes prises par les fonctions de coefficients des deux familles (1 et 2) de méthodes fonctionnelles (réduction de dimension, etc.) ne sont pas aisément interprétables. Afin de lever cette contrainte, [52] introduit l'approche FLiRTI (Functional Linear Regression That's Interpretable) qui utilise des idées de sélection de variables, appliquées à diverses dérivées de $\beta(t)$, pour produire des estimations qui sont interprétables, flexibles et précises. La méthode FLiRTI peut générer des sections de coefficients $\beta(t)$, constantes, linéaires et quadratiques par morceaux. Elle offre aussi l'avantage d'être parcimonieuse, fournissant ainsi des sections exactement nulles lorsque qu'aucune relation apparente n'existe entre réponse et prédicteurs. Dans le même esprit, la méthode BLISS (Bayesian functional Linear regression with Sparse Stepwise functions) introduite par [41] permet aussi de construire pour les régressions de type scalar-on-function, un modèle interprétable permettant de prendre en compte des connaissances a priori et fournissant un indicateur de confiance. Pour obtenir un modèle interprétable dans la méthode BLISS, les solutions sont développées sur un ensemble de fonctions constantes par morceaux :

$$\beta(t) = \sum_{k=1}^{K} \frac{b_k}{|\mathcal{I}_k|} \mathbb{1}_{\mathcal{I}_k}(t) \tag{1.2.14}$$

où les b_k sont des réels, les $\mathcal{I}_k$ sont des intervalles de $[0, 1]$ et K un hyperparamètre à déterminer. À partir de cette combinaison de fonctions constantes par morceaux, (1.2.14), [41] reformule le modèle de régression linéaire fonctionnel en un modèle de régression linéaire multiple où le design dépend d'intervalles $\mathcal{I}_k$ et de fonctions indicatrices associées.

Il est essentiel de retenir à cette étape, que ces méthodes interprétables ont en commun l'utilisation de combinaison de fonctions continues par morceaux associée à l'utilisation d'une sélection des coefficients non nuls qui permet d'identifier le support/domaine où les variables explicatives ont un effet. On parle alors de modélisation interprétable.

Ce qui indirectement induit une facilité d'interprétation des résultats fournis par ces méthodes.

1.2.3 Méthodes nécessitant une complémentarité ou une agrégation de modèles

En dépit de la diversité des méthodes existantes pour analyser les données fonctionnelles, le choix d'une méthode implique habituellement des concessions à faire en termes d'hypothèses, de coût de calcul, d'information à capturer sur les données, etc. [30] ont montré dans leurs travaux que la richesse et la complexité de l'information disponible dans les données fonctionnelles pourrait être mieux captée par le biais d'approches complémentaires (de type "boosting") permettant de combiner les avantages de diverses méthodes. Les auteurs montrent aussi que cette complémentarité des modèles (grâce à la combinaison des résultats de diverses méthodes sur le même jeu de données) existent à la fois au sein des méthodes de statistique multivariée, au sein des méthodes pour données fonctionnelles et en combinant les deux types d'approches. Aussi, certains logiciels ou package d'analyse de données fonctionnelles intègrent des approches de type boosting dans leurs solutions [16]. Outre les régressions de type scalar-on-function, ces méthodes ensemblistes se développent de plus en plus pour les données fonctionnelles [15]

D'un autre côté, on peut noter dans la littérature ces dernières années, un engouement pour l'implication des méthodes de type "bagging" comme les forêts aléatoires [14] dans l'analyse des données fonctionnelles en général [87, 40, 78, 44]. En ce qui concerne les modèles de type scalar-on-function, l'utilisation des forêts aléatoires n'est pas forcément une solution évidente. Mais en fonction de la nature de l'information à tirer de l'analyse, il est possible de penser à des modèles différents de ceux posés dans les équations (1.2.1) et (1.2.7). L'idée principale étant de poser d'autres questions pour obtenir d'autres réponses. Par exemple, au lieu de s'intéresser aux périodes d'influence nécessitant l'estimation des coefficients en fonction du temps $\beta(t)$, les nouveaux objectifs pourraient être d'identifier les seuils de la variable explicative fonctionnelle déclenchant un phénomène donné. Ces nouveaux objectifs nécessitent de construire de nouveaux modèles qui peuvent impliquer diverses transformations des données fonctionnelles. Dans le cadre des courbes de température présentées dans nos données par exemple, ces nouvelles variables construites pour chaque individu à partir de connaissances agronomiques peuvent être : le nombre de jour où la température minimale ou maximale a été inférieure ou supérieure à une valeur donnée, le nombre de jours nécessaire pour atteindre un certaine valeur de la variable réponse, l'écart moyen entre les températures minimales et maximales, les différences de température en le jour et la nuit, etc. Et dans un cadre un peu plus complexe, les mêmes questions peuvent impliquer plusieurs variables fonctionnelles explicatives, avec le besoin de prendre en compte leurs interactions. Nous nous intéressons à ces modèles dans le paragraphe suivant.

1.2.4 Extension de la régression à plusieurs prédicteurs fonctionnels

Lorsque plusieurs prédicteurs fonctionnels sont impliqués dans la régression, il est possible d'utiliser une généralisation du modèle (1.2.7) qui présente aussi des défis pour l'estimation des coefficients ou pour la sélection des variables sans être forcément interprétables. Plusieurs généralisations de ce modèle existent dans la littérature et sont présentées comme des modèles de régression fonctionnelle multiple. Elles impliquent dans le même modèle des variables explicatives fonctionnelles et parfois même des variables explicatives non-fonctionnelles. Ces modèles sont en général additifs [101]. Ils peuvent impliquer aussi des interactions au sens des méthodes classiques de statistique multivariée, c'est à dire que le terme d'interaction est un produit entre les différentes covariables fonctionnelles. Nous nous intéresserons dans cette section à un modèle de régression fonctionnelle multiple additif impliquant p variables fonctionnelles $x_{i1}(t), \ldots, x_{ip}(t)$ observées sur p domaines T_j avec $i = 1, \ldots, n$ et $j = 1, \ldots, p$. Ce modèle [90, 63] se formule comme suit :

$$y_i = \sum_{j=1}^{p} \int_{T_j} x_{ij}(t)\beta_j(t)dt + \varepsilon_i \tag{1.2.15}$$

avec $\beta_j(t)$ les coefficients fonctionnels à estimer et ε_i les résidus. Afin d'estimer les coefficients, [63] propose une approche impliquant une Analyse en Composantes Principales Fonctionnelles. Outre l'utilisation des composantes principales, d'autres méthodes existent pour prendre en main ces difficultés. Estimant que l'utilisation des composantes principales impliquait d'ignorer la structure spatiale existant dans leurs courbes, [70] se sont plutôt servis d'un produit tensoriel de B-splines qui leur a permis d'estimer les coefficients tout en conservant cette structure spatiale. Ce produit tensoriel de B-splines leur a permis d'identifier des espaces issus de combinaisons de domaines des différentes covariables fonctionnelles en lieu et place de coefficients dépendant d'une longueur d'onde. Quant à [86], la structure de dépendance existant parmi des variables fonctionnelles a été modélisée par des graphes. Ces différents cas illustrent bien comment les objectifs à atteindre ou encore la spécificité des variables fonctionnelles observées incitent à adapter ou à proposer de nouvelles méthodes d'estimation. Nous introduisons dans la prochaine section une nouvelle méthode de résolution des problèmes de type scalar-on-functions permettant d'atteindre des objectifs bien définis à partir de données ayant certaines spécificités.

1.3 SpiceFP : une procédure parcimonieuse et structurée pour identifier les effets combinés des prédicteurs fonctionnels

Nous nous intéresserons dans cette thèse à un problème de régression impliquant une réponse scalaire et deux (à trois) prédicteurs fonctionnels (observés sur un même domaine T) sous l'hypothèse selon laquelle les variables fonctionnelles influencent conjointement la réponse. Une autre contrainte que nous nous imposons est une facilité d'interprétation des

résultats. Nous avons développé pour cela la méthode SpiceFP, présentée dans le deuxième chapitre de cette thèse. Il s'agit avant tout d'une approche exploratoire exploitant des outils de la statistique inférentielle. Pour ce faire, elle combine :

— l'interprétabilité associée à une base de fonctions indicatrices (présentée en section 1.3.1 et détaillée dans le chapitre 2) obtenues par le biais de tableaux de contingence (base de fonctions construites à partir des données fonctionnelles),

— l'utilisation de graphe permettant d'intégrer une structure de dépendance entre les fonctions indicatrices composant une même base.

— l'efficacité des modèles de régression linéaire pénalisée (section 1.2.1) et des algorithmes associés dans l'identification et la sélection de variables.

Pour n individus, considérons une réponse scalaire y et deux variables explicatives fonctionnelles $\mathcal{A}(t)$ et $\mathcal{B}(t)$.

1.3.1 Les collections de base de fonctions indicatrices

On se fixe un vecteur $u = (n_{\mathcal{A}}, n_{\mathcal{B}}) \in \mathbb{N}^2$. À partir de ces entiers naturels, calculons les $n_{\mathcal{A}}+1$ et $n_{\mathcal{B}}+1$ bornes de classes nécessaires à la construction de $n_{\mathcal{A}}$ et $n_{\mathcal{B}}$ intervalles de classe. Pour la variable fonctionnelle $\mathcal{A}$, notons les bornes des intervalles $L_{\mathcal{A}}(v),\ v = 1, \ldots, n_{\mathcal{A}}+1$ et les intervalles $I_{\mathcal{A}}(v) = [L_{\mathcal{A}}(v), L_{\mathcal{A}}(v+1)[, v = 1, \ldots, n_{\mathcal{A}}$. Respectivement pour $\mathcal{B}$, on obtient les bornes $L_{\mathcal{B}}(w)$ et les intervalles $I_{\mathcal{B}}(w)$. Pour un individu i, il devient ainsi possible de construire à partir des courbes $\mathcal{A}_i(t)$ et $\mathcal{B}_i(t)$, $t \in T$, un histogramme bivarié ou un tableau de contingence en calculant l'expression :

$$C^u_{i,(v,w)} = \sum_{t \in T} \mathbb{1}_{\mathcal{A}_i(t) \in I^u_{\mathcal{A}}(v),\ \mathcal{B}_i(t) \in I^u_{\mathcal{B}}(w)} = Card\ \{t \in T | \mathcal{A}_i(t) \in I^u_{\mathcal{A}}(v), \mathcal{B}_i(t) \in I^u_{\mathcal{B}}(w)\}, \quad (1.3.1)$$

pour tout $v = 1, \ldots, n_{\mathcal{A}}\ ,\ w = 1, \ldots, n_{\mathcal{B}}$ et chaque $u = (n_{\mathcal{A}}, n_{\mathcal{B}})$. En vectorisant (empilement colonne par colonne) et en transposant chaque tableau de contingence, le résultat, noté $X^u_i = {}^tVect(C^u_i),\ X^u_i \in \mathbb{R}^{n_{\mathcal{A}} n_{\mathcal{B}}}$, devient la $i$ème ligne de la matrice X^u qui sera utilisée pour l'estimation de coefficients. $C^u_{i,(v,w)}$ est une approximation du temps (ou nombre d'instants d'observation du climat) passé par l'individu i (observations $\mathcal{A}_i$ et $\mathcal{B}_i$) simultanément dans les conditions $I^u_{\mathcal{A}}(v) \times I^u_{\mathcal{B}}(w)$. Nous appellerons ces conditions les modalités conjointes qui sont une collection d'intervalles 2D dans lesquelles les observations ont été projetées pour un vecteur de partitionnement u fixé. Ces éléments nous permettent d'introduire la base fonctionnelle $\mathcal{I}^u$ définie par les fonctions indicatrices :

$$\mathcal{I}^u(a,b) = \mathbb{1}_{a \in I^u_{\mathcal{A}}(v),\ b \in I^u_{\mathcal{B}}(w)} \quad (1.3.2)$$

pour $a \in [\underline{\mathcal{A}}, \bar{\mathcal{A}}]$ et $b \in [\underline{\mathcal{B}}, \bar{\mathcal{B}}]$. $\underline{\mathcal{A}}$ et $\underline{\mathcal{B}}$ sont les valeurs minimales des observations fonctionnelles puis $\bar{\mathcal{A}}$ et $\bar{\mathcal{B}}$ leurs valeurs minimales.

1.3.2 Le modèle proposé dans l'approche SpiceFP

Le modèle de SpiceFP est défini, pour chaque partition u et chaque individu i, par :

$$y_i = X_i^u \ \beta^u + \varepsilon_i, \tag{1.3.3}$$

avec β^u les coefficients à estimer sur les intervalles 2D et ε_i des résidus gaussiens i.i.d.. Il s'agit d'un modèle de régression linéaire multiple où les variables explicatives sont les fréquences associées aux classes d'intervalles jointes $\left(I_{\mathcal{A}}^u(v) \times I_{\mathcal{B}}^u(w)\right)_{v,w}$.

1.3.3 L'estimation des coefficients

À la nouvelle matrice X^u construite, on associe une matrice E^u constituée d'arêtes et définissant les liens entre les colonnes de X^u. Ces liens sont définis en fonction de la contiguité entre les intervalles de classes suivant $\mathcal{A}$ ou suivant $\mathcal{B}$. On obtient ainsi un graphe $G^u = (V^u, E^u)$ dont les noeuds sont constituées des modalités conjointes contenues dans la matrice X^u et les arêtes définies par la matrice E. Le Generalized Fused Lasso [122] présenté dans l'équation (1.3.4), est ensuite utilisé dans l'optique d'identifier une ou plusieurs modalités conjointes ayant un effet sur la réponse y. Cette régression est parcimonieuse car se servant de la pénalisation LASSO pour la sélection de variables mais aussi structurée dans l'espace des variables $\mathcal{A}$ et $\mathcal{B}$ car elle pénalise en plus la différence entre deux coefficients liés, les amenant à avoir des valeurs proches.

$$\beta^{gfl} = \underset{\beta \in \mathbb{R}^{n_{\mathcal{A}} n_{\mathcal{B}}}}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{n} \left(y_i - X_i^u \beta\right)^2 + \lambda_p \sum_j |\beta_j| + \lambda_f \sum_{(j,j') \in E^u} |\beta_j - \beta_{j'}| \tag{1.3.4}$$

avec λ_p le paramètre de régularisation de la parcimonie et λ_f celui de la fusion.

1.3.4 L'approche proprement dite

Compte-tenu de la transformation des variables fonctionnelles utilisée, il est essentiel de remarquer que chaque transformation est indexée d'une part et intrinsèquement associée d'autre part au vecteur de partitionnement u utilisé. Changer de vecteur de partitionnement induit une nouvelle estimation de la base de fonctions indicatrices $\mathcal{I}^u$ et par conséquent une modification des modalités conjointes, affectant directement le résultat obtenu. Le choix de u devient donc un verrou essentiel à lever. Étant donné que le modèle utilisé facilite la liaison entre les variables (modalités conjointes), un bon vecteur de partitionnement devrait permettre l'obtention d'intervalles de classe très fins dans le but de les regrouper une fois dans la régression. Même lorsque les variables fonctionnelles sont observées de manière dense, la recherche d'intervalles très fins amènent à des modalités conjointes ayant très peu ou pas d'observations, rendant difficile l'estimation de coefficients pour ces modalités. Afin de lever cette contrainte, nous avons donc opté pour une approche en deux principales étapes :

— une première étape où différents vecteurs de partitionnement sont utilisées pour reconstruire différentes transformation des variables fonctionnelles

— une deuxième étape où un modèle est construit pour chaque transformation des variables fonctionnelles et la meilleure transformation retenue grâce à un critère d'information.

1.3.5 Lien avec un modèle fonctionnel

Afin de construire un pont entre l'approche SpiceFP et un modèle fonctionnel conceptuel dépendant des variables $\mathcal{A}$ et $\mathcal{B}$, le modèle SpiceFP peut être vu comme une approximation du modèle fonctionnel 1.3.5 :

$$y_i = \int_{\underline{\mathcal{A}}}^{\bar{\mathcal{A}}} \int_{\underline{\mathcal{B}}}^{\bar{\mathcal{B}}} g_{\mathcal{A}_i(.),\mathcal{B}_i(.)}(a,b)\beta^u(a,b)\ da\ db + \varepsilon_i, \tag{1.3.5}$$

où $g_{\mathcal{A}_i(.),\mathcal{B}_i(.)}$ représente une quantification du temps passé par un individu i avec $\mathcal{A}_i(.)$ au voisinage de a et $\mathcal{B}_i(.)$ au voisinage de b. Ici, la relation entre la variable à expliquer y_i et les valeurs observées des prédicteurs $\mathcal{A}_i(.)$ et $\mathcal{B}_i(.)$ est supposée stable dans le temps. Par conséquent, certaines classes d'intervalles jointes ont un impact sur y_i et le temps passé dans ces classes est important.

1.3.6 Extensions de l'approche SpiceFP

Deux extensions de l'approche ont été proposées dans le cadre de cette thèse et implémentées dans le package (présentées dans le chapitre 4) :

— une approche SpiceFP en trois dimensions (3D) : cette extension permet de rajouter une nouvelle dimension à celles des variables explicatives ayant une influence conjointe. Cette nouvelle dimension pourrait bien être une troisième variable fonctionnelle ou une variable catégorielle disposant d'au moins deux modalités (un comptage en 2D est ainsi effectué pour chaque modalité de la variable catégorielle).

— une approche SpiceFP itérative d'agrégation de modèles : pour cette extension, un premier modèle est construit à partir de SpiceFP sur les données initiales puis un deuxième l'est en utilisant comme variable réponse dans SpiceFP, les résidus issus du premier modèle, etc.

1.4 Plages d'influence de température et d'irradiance affectant l'accumulation d'anthocyanes dans le raisin

Une fois, l'approche développée, nous l'avons utilisé pour l'analyse du jeu de données présenté dans la section 1.1.1. Cette analyse nous permet dans un premier temps d'identifier des zones d'influence du couple température-irradiance sur l'accumulation des anthocyanes mais aussi dans un second temps d'utiliser l'approche ainsi que ses extensions de manière plus étendue.

SpiceFP permettant de s'intéresser à des phénomènes stables sur toute une plage d'observation (une période de temps dans le cadre de nos données), nous avons dans un premier temps, procédé à différents découpages avant son utilisation. Nous avons commencé par identifier des phases d'accumulation puis décidé de nous intéresser à celle où l'accumulation était la plus importante. Ce qui nous a amené à appesantir nos travaux sur la semaine 3. Les baies de raisin observées, n'étant pas toutes au même niveau d'accumulation en début de cette semaine, il a fallu conditionner nos analyses par rapport à ce niveau d'accumulation en séparant en deux groupes (retardés et avancés). Pour chacun de ces deux groupes, un autre découpage a été réalisé, cette fois-ci sur la journée (supposant que l'accumulation ne se fait pas uniformément à l'échelle d'une même journée) : lever du soleil à 12H, 12H au coucher du soleil et coucher au lever du soleil. Pour toutes les analyses effectuées, nous ne nous sommes intéressés qu'à la période du lever du soleil à 12H. L'analyse des avancés n'a pas été concluante tandis que celle des retardés a permis d'identifier un effet négatif des irradiances faibles ($<$100 $\mu mol.m^{-2}.s^{-1}$) et températures élevées (entre 25°C et 35°C) le matin (entre le lever du soleil et midi) sur l'accumulation des anthocyanes.

Désireux de prendre en compte l'information disponible dans la bibliographie et relative à une influence de la température de la nuit, nous avons catégorisé les nuits, en nuits chaudes et nuits froides grâce à un seuil fourni par l'histogramme des températures du coucher du soleil à minuit. Nous avons ensuite mené une analyse en deux étapes : dans un premier temps, nous avons effectué deux analyses séparées (SpiceFP) à partir des deux jeux de données semaine 3 - retardés - nuits chaudes et semaine 3 - retardés - nuits froides puis dans un second temps, effectué une analyse SpiceFP 3D à deux itérations permettant de prendre en compte outre la température et l'irradiance, la température de la nuit dans une troisième dimension. Les comptages d'observations dans chaque condition ont été ainsi faits dans des tableaux 3D et non 2D. Les nouveaux résultats obtenus renforcent bien les résultats obtenus dans l'analyse du jeu données semaine 3 - retardés et soulignent aussi une diversification de cette influence négative en fonction de la température de la nuit. Dans ce chapitre, des visualisations ont été rajoutées en annexes permettant d'avoir une idée de la stabilité des résultats mais aussi de s'intéresser aux nouvelles variables explicatives ayant permis l'obtention de ces résultats.

1.5 Implémentation de l'approche proposée

L'implémentation de SpiceFP s'est faite sous le logiciel R. La fonction principale est nommée comme l'approche, **spicefp**. Elle permet de l'exécuter intégralement et renvoie comme principale sortie les coefficients estimés, présentés au format 2D ou 3D. Les coefficients 3D sont fournis par l'extension de l'approche à une troisième variable fonctionnelle. L'algorithme de l'approche peut être subdivisé en trois grandes étapes :

1. la création des matrices candidates : c'est à cette étape que les variables explicatives fonctionnelles sont transformées par le biais de fonctions indicatrices en une collection de nouvelles matrices explicatives X^u, auxquelles sont associées les matrices E^u contenant les arêtes. Ces nouvelles matrices peuvent être :

— obtenues suivant le canevas fourni par le package : dans ce cas les matrices X^u sont créées par la fonction **candidates** et les matrices E^u fournies par les fonctions **getD2dSparse** (de **genlasso**) et **getD3dSparse** (de **SpiceFP**) par défaut ou l'argument **penfun** des fonctions **evaluate.candidates** ou **spicefp**. **candidates** fournit pour tous les u, $C^u_{i,(v,w)}$ présenté dans l'équation (1.3.1) tout en laissant à l'utilisateur la possibilité de choisir (logbreaks, linbreaks, etc.) ou de fournir les fonctions devant permettre l'obtention des limites des classes.

— ou crées spécifiquement par l'utilisateur en fonction des hypothèses à explorer : dans ce cas, libre cours est laissé à l'intuition de l'utilisateur. Il devra toutefois organiser et présenter ces matrices candidates suivant un modèle bien explicité dans le chapitre 3.

2. l'évaluation des matrices candidates : que le modèle à construire soit en deux ou en trois dimensions, la fonction utilisée est **evaluate.candidates**. Elle retourne principalement une matrice d'information avec en ligne les modèles construits à partir de l'équation (1.3.4) et en colonne les paramètres associés parmi lesquels l'AIC et le BIC. Ces critères pourront être utilisées plus tard pour comparer des modèles impliquant différentes combinaisons de matrices candidates et de paramètres de pénalisation. Ils serviront à la fois à la sélection de variables mais aussi à la sélection de modèles. C'est pour cette raison qu'une attention particulière a été portée à leur calcul ainsi qu'à leur propriétés dans le développement de la méthode.
3. le traitement post-évaluation et la construction du résultat : le résultat de l'approche est fourni par le meilleur modèle associé à la meilleure matrice candidate. Le package permet de s'intéresser à plusieurs modèles en lieu et place du meilleur fourni à une itération donnée. Pour ce faire, la fonction **coef_spicefp** fournit les coefficients de n'importe quel modèle construit autour de l'approche et les fonctions **finemeshed2d** et **finemeshed3d** permettent de rapporter les coefficients issues de bases de fonctions indicatrices différentes à une même échelle afin d'effectuer des opérations arithmétiques. Elles sont intégrées par exemple à la fonction **meancoef** qui permet d'effectuer la moyenne de coefficients de différents modèles fournis par l'approche.

1.6 Plan de la thèse

La présente thèse intitulée "Modélisation et visualisation des liens entre cinétiques de variables agro-environnementales et qualité des produits dans une approche parcimonieuse structurée" est donc principalement adossée à la recherche de solutions permettant l'analyse d'un jeu de données présenté dans la section 1.1 et mis à notre disposition. Une présentation de ce jeu de données a été produite en annexe. Dans ce chapitre introductif, nous avons présenté les données ainsi qu'un positionnement bibliographique de nos travaux. Un résumé de l'approche statistique proposée puis un bref aperçu de l'analyse des données sont aussi présentés dans ce chapitre introductif. Cette thèse a été rédigée suivant le format de présentation d'une thèse sur articles. Le deuxième chapitre est relatif au manuscrit soumis à la

revue *Computational Statistics and Data Analysis (CSDA)* et présente l'approche statistique proposée. Le troisième chapitre concerne quant à lui l'analyse des données et représente un travail de base dans le cadre d'une future soumission à une revue agronomique. Le quatrième chapitre est la vignette associée à l'implémentation de l'approche proposée sous la forme d'un package du logiciel R. Il a été rédigé en suivant les standards de la revue *Journal of Statistical Software (JSS)* et lui sera soumis dès publication de la méthode (soumise à la revue *CSDA*). Le package R est quand à lui soumis au *Comprehensive R Archive Network (CRAN)*. Un dernier et court chapitre présente une conclusion et des perspectives autour des différents travaux de cette thèse.

Chapitre 2

Identification of combined effects of functional variables using contingency tables with ordered categories - Application to agri-environmental issues

2.1 Introduction

Nowadays, several fields of activity and in particular, agriculture, are being revolutionized by the emergence of sensor data. With regard to crops, the setting up of harvest management can now be based on monitoring with the aim of including/modeling the influence of multiple environmental conditions. Specifically, water scarcity and temperature increase are two major factors which have long been analyzed and considered to be determining factors causing huge variations in crop yield. Their influences are increasing with climate change and are becoming a major concern for the sustainability of agriculture in many parts of the world. However, relationships between climatic conditions and quality of the harvest are still poorly understood and modeling approaches are still lacking. To better use newly available data from sensors, there is a need for methods able to explore which combination of climatic variables influences the quality of harvest and at which stage of plant development. Such data sets involve multivariate, longitudinal or temporal data which are handled in various ways, including the large family of functional data analysis [29].

One of the main lines of research on functional data concerns their treatment in regression problems. The regression toolbox is used to extract knowledge from input variables (functional data in our context) to predict and/or explain a scalar output or a continuous variable of interest. Directly applying machine learning and/or supervised learning with its usual black box tools (support-vector machine [22], random forest [78], neural networks [96], etc.) is not indicated in our context : all these black-box tools are based on complex combinations of the regressors whose individual effects are difficult to interpret. In [91, 89], the regression models are usually classified into 3 categories according to the role played by functional data.

A distinction is made between the "scalar-on-function", "function-on-scalar" and "function-on-function" regressions. In this paper we will focus on the "scalar-on-function" regression where the response variable is a scalar and the regressors are functions. More precisely, regressors are two functions that jointly influence the response variable.

Various methods exist to solve "scalar-on-function" regressions and the reader can refer to [90] for a review. Regression with functional data often uses pre-treatment of the data like interpolation and/or projection on a basis [89]. This kind of regression is easily implemented (see R-package FDA), but again makes the influence of regressors not easy to interpret. By contrast, the work on *"functional linear regression that's interpretable"* [53] and its Bayesian version [42] open a new research area for functional regressions where interpretation is of major interest. Unfortunately, these models do not take into account a possible combined effect of explanatory variables.

The objective of the present paper is to infer an interpretable model to study the joint influence of two functional inputs on a scalar output. Our approach is based on a transformation of the functional data that implies a change from "scalar-on-function" regression to "scalar-on-image" regression, where the "image" is a bivariate representation of both functional datasets. Scalar-on-image regression models aim to control the smoothness of non-zero estimated coefficients. Different approaches are used to solve scalar-on-image regressions, among which Bayesian approaches [61, 39], total variation penalizing approaches [121], neighborhood taken into account in the selection of variables [62] inspired by the Fused Lasso, etc. [57] proposed an approach based on the Gaussian process and compared it to the Fused Lasso. Other studies on scalar-on-image regressions are inspired, used or compared to models involving different L_1 regularization. Following this trend, we chose to use the fused lasso, and more specifically its implementation via the genlasso package [4], for identifying parsimonious and structured coefficients. The selection of the coefficients is performed using information criteria instead of cross-validation, as proposed in [125].

In the following, we present a Sparse and Structured Procedure to Identify Combined Effects of Functional Predictors, denoted SpiceFP, and its theory. Simulations and a use case based on a real issue in agri-environment are also provided.

2.2 The SpiceFP approach

This section describes the main steps of the approach ; first in §2.2.1, the originality of our approach is to transform both functional variables into categorical variables by defining joint modalities using class intervals (with bins of equal size). Several candidate partitions are defined this way, depending on the choice of the bin size. The functional model is presented in §2.2.2, from which we derived a linear multiple regression model where the regressors are the frequencies associated to the joint class intervals. As explained in [30], when faced with a high number of discretization points of functional data, the naive approach would be to consider these data as a classical multivariate sample having as dimension the number of discretization points of the functional variables. In this case, multivariate statistics meets limits, among which its failure to take into account the very strong colinearity existing between

discretized variables. By contrast, colinearity can be considered in the Fused Lasso penalized regression which is therefore retained in our approach. A "scalar on image" regression model is followed, where the "image" is a contingency table of the joint class intervals, for a fixed candidate partition, to which a graph of contiguity constraints defined in subsection §2.2.3 is associated. Identification is performed through a Generalized Fused Lasso (see §2.2.4) using each candidate contingency table as input variables. The selection of the best candidate and of its relative regression coefficients is achieved by minimizing an information criteria.

2.2.1 Transformation of both functional variables

Let us consider the observations as a triplet sample $(\mathcal{A}_i(.), \mathcal{B}_i(.), y_i)_{i=1,\dots,n}$, where n is the number of statistical individuals. The triplet consists of two explanatory functional variables $\mathcal{A}$ and $\mathcal{B}$, associated to a scalar response variable y. Both $\mathcal{A}$ and $\mathcal{B}$ are observed on the same set T of fixed observation times. It is assumed equidistant observation times with no missing values. These practical conditions of use can be released with some pre-treatment of the data (like interpolation, smoothing and imputation), see Section 2.6 for more details.

The requirement for variable transformation is intrinsically linked to the goal of the approach : identify joint class intervals of the explanatory variables that influence the response. The transformation requires the definition of joint class intervals which can be used as linear regressors to predict the response. The steps to achieve this transformation are shown in the Figure 2.2.1.

Contingency table of the joint class intervals

Let's partition the values taken by the first functional observation $\mathcal{A}$ and the second functional observation $\mathcal{B}$ into, respectively, $n_\mathcal{A}$ and $n_\mathcal{B}$ class intervals. Let's define a partition vector $u = (n_\mathcal{A}, n_\mathcal{B})$. The first partition generates $n_\mathcal{A}$+1 breaks denoted $L^u_\mathcal{A}(v),\ v = 1, \dots, n_\mathcal{A}$+1. We chose to have equidistant breaks, as defined in Equation (2.2.1) :

$$L^u_\mathcal{A}(v) = \underline{\mathcal{A}} + \frac{v-1}{n_\mathcal{A}} \left(\bar{\mathcal{A}} - \underline{\mathcal{A}}\right),\ v = 1, \dots, n_\mathcal{A} + 1, \tag{2.2.1}$$

with $\underline{\mathcal{A}} \in \mathbb{R}$ and $\bar{\mathcal{A}} \in \mathbb{R}$ the minimum and maximum based on values of $\mathcal{A}$ observed for all individuals. Therefore, the partition is the same for all individuals. The bins used for partitioning all $(\mathcal{A}_i)_{i=1\dots n}$ are $I^u_\mathcal{A}(v) = [L^u_\mathcal{A}(v), L^u_\mathcal{A}(v+1)), v = 1, \dots, n_\mathcal{A}$. The partition is the same for all i, $i = 1, \dots, n$. Using the same approach for partitioning the second explanatory variable $\mathcal{B}$, we obtain $n_\mathcal{B}$+1 breaks $L^u_\mathcal{B}(w)$ and corresponding $I^u_\mathcal{B}(w) = [L^u_\mathcal{B}(w), L^u_\mathcal{B}(w+1)), w = 1, \dots, n_\mathcal{B}$. The numbers of class intervals $n_\mathcal{A}$ and $n_\mathcal{B}$ have to be set to compute the breaks $L^u_\mathcal{A}(v)$ and $L^u_\mathcal{B}(w)$, $v = 1, \dots, n_\mathcal{A}$, $w = 1, \dots, n_\mathcal{B}$.

For all i, it is then possible to obtain the frequency bivariate histogram of $(\mathcal{A}_i, \mathcal{B}_i)$ as a contingency table C^u_i, of dimension $n_\mathcal{A} \times n_\mathcal{B}$, whose components $C^u_{i,(v,w)}$ are obtained through :

$$C^u_{i,(v,w)} = \sum_{t \in T} \mathbb{1}_{\mathcal{A}_i(t) \in I^u_\mathcal{A}(v),\ \mathcal{B}_i(t) \in I^u_\mathcal{B}(w)} = \text{Card}\ \{t \in T | \mathcal{A}_i(t) \in I^u_\mathcal{A}(v), \mathcal{B}_i(t) \in I^u_\mathcal{B}(w)\}, \tag{2.2.2}$$

for all $v = 1, \ldots, n_{\mathcal{A}}$, $w = 1, \ldots, n_{\mathcal{B}}$ and each $u = (n_{\mathcal{A}}, n_{\mathcal{B}})$, with : $\sum_{v=1}^{n_{\mathcal{A}}} \sum_{w=1}^{n_{\mathcal{B}}} C^u_{i,(v,w)} = Card(T)$.

$C^u_{i,(v,w)}$ is the number of times that the observations of $\mathcal{A}_i$ and $\mathcal{B}_i$ are at the same time in $I^u_{\mathcal{A}}(v) \times I^u_{\mathcal{B}}(w)$. $C^u_{i,(v,w)}$ can also be interpreted as a discrete approximation of the density of the time spent by the individual i with variable $\mathcal{A}$ around $L_{\mathcal{A}}(v)$ and variable $\mathcal{B}$ around $L_{\mathcal{B}}(w)$.

Part *1* of Figure 2.2.1 shows the transformation of the functional explanatory variables $\mathcal{A}$ and $\mathcal{B}$ for the fixed $u = (4, 3)$. Note that, for a fixed u, $(I^u_{\mathcal{A}}(v) \times I^u_{\mathcal{B}}(w))_{v,w}$ is a collection of 2D intervals in which the pairs $(\mathcal{A}_i, \mathcal{B}_i)$ will be projected. This is detailed in the following subsection.

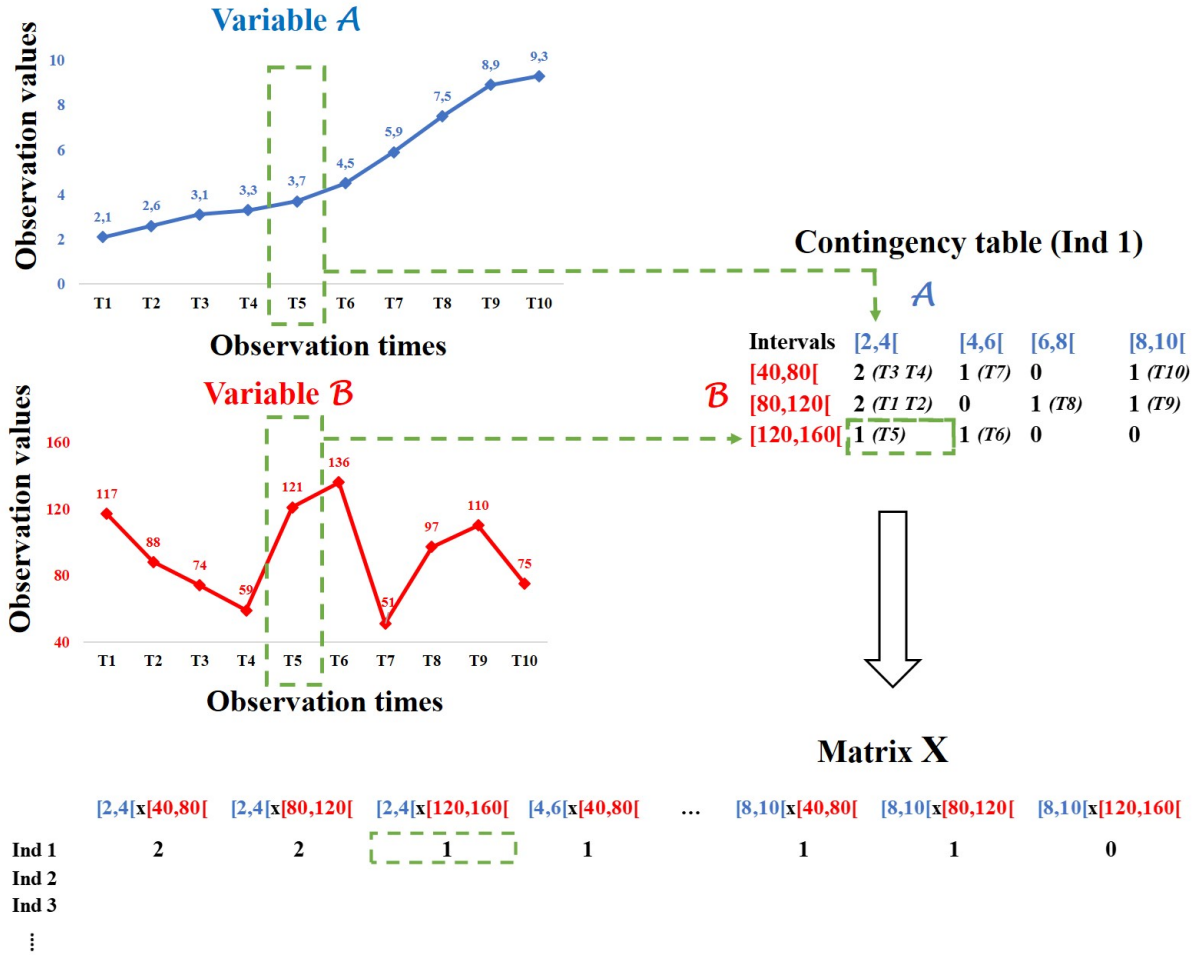


FIGURE 2.2.1 – Transformation of both functional explanatory variables for the SpiceFP approach

2.2.2 SpiceFP model

The SpiceFP model is defined, for each partition u and each individual i, by :

$$y_i = \sum_{v=1}^{n_{\mathcal{A}}} \sum_{w=1}^{n_{\mathcal{B}}} C^u_{i,(v,w)} \ \beta^u_{(v,w)} + \varepsilon_i, \tag{2.2.3}$$

where $C^u_{i,(v,w)}$ is given in Equation (2.2.2), $\beta^u_{(v,w)}$ is the coefficient to be estimated on the 2D interval $(I^u_{\mathcal{A}}(v) \times I^u_{\mathcal{B}}(w))$ and ε_i is an i.i.d. Gaussian error.

This model is a linear multiple regression model where the regressors are the frequencies associated to the joint class intervals $(I^u_{\mathcal{A}}(v) \times I^u_{\mathcal{B}}(w))_{v,w}$. To build a bridge between the SpiceFP approach and a conceptual functional model in variable $\mathcal{A}$ and $\mathcal{B}$, we give another insight of the SpiceFP model in 2.7.1. The SpiceFP model can be viewed as an approximation of the following functional model :

$$y_i = \int_{\underline{\mathcal{A}}}^{\bar{\mathcal{A}}} \int_{\underline{\mathcal{B}}}^{\bar{\mathcal{B}}} g_{\mathcal{A}_i(.),\mathcal{B}_i(.)}(a,b)\beta^u(a,b) \ da \ db + \varepsilon_i,$$

with $g_{\mathcal{A}_i(.),\mathcal{B}_i(.)}$ a quantification of the time spent by an individual i with $\mathcal{A}_i(.)$ in the vicinity of a and $\mathcal{B}_i(.)$ in the vicinity of b. Here, the relationship between the variable to be explained y_i and the observed values of the predictors $\mathcal{A}_i(.)$ and $\mathcal{B}_i(.)$ is assumed stable over time. Consequently, some ranges of cross-values have an impact on y_i and only the time spent in these ranges is important.

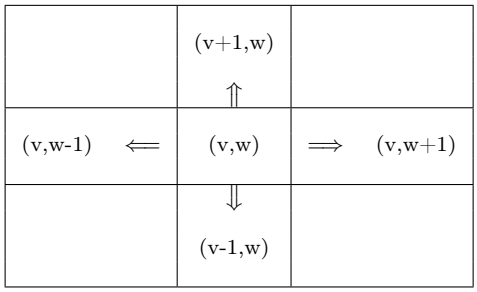
From the contingency tables C^u_i, we construct the design matrix X^u associated to model (2.2.3) as follows. After vectorization (stacking column by column) and transposition of the contingency table C^u_i (see part *2* of Figure 2.2.1), we obtain, for a fixed partition vector $u = (n_{\mathcal{A}}, n_{\mathcal{B}})$, a row vector of length $n_{\mathcal{A}}.n_{\mathcal{B}}$ $X^u_i \in \mathbb{R}^{n_{\mathcal{A}} n_{\mathcal{B}}}$:

$$X^u_i = \ \mathrm{Vect}(C^u_i)^T \,, \tag{2.2.4}$$

which represents the number of time observations t during which an individual i has been observed in each of the $n_{\mathcal{A}} \times n_{\mathcal{B}}$ levels described by the joint class intervals. The n stacked row vectors form the matrix $X^u = (X^u_1, X^u_2, \ldots, X^u_n)^T \in \mathbb{R}^{n \times n_{\mathcal{A}} n_{\mathcal{B}}}$.

2.2.3 Creation of a graph of contiguity constraints

To each matrix X^u corresponds a graph $G^u(V^u, E^u)$, which contains the contiguity constraints between modalities of the contingency table. V^u represents the columns (new variables) of the candidate matrix X^u and E^u all the edges connecting two close joint modalities. We used the Rook's case contiguity rule [85] where two joint class intervals are said to be close if the bins following the variable $\mathcal{A}$ (indexed by v) or (exclusive) the bins following the variable $\mathcal{B}$ (indexed by w) are consecutive, as shown in the following diagram :



V^u is composed of the modalities (v, w) and E^u is composed of the edges between (v, w) and $(v \pm 1, w)$ or (v, w) and $(v, w \pm 1)$.

2.2.4 Selection of class intervals and related regression coefficients

The Fused Lasso is a variant of the Lasso introduced in 2005 by [113], in order to take into account the existence of a structure in the variables. In its original form, the Fused Lasso aims not only at parsimony of coefficients but also at parsimony of differences in consecutive coefficients. This version of the Fused Lasso can be interpreted as a one-dimensional Fused Lasso (1D-Fused Lasso). The Generalized Fused Lasso (GFL) [122] aims to promote smoothness over neighboring variables on a general graph $G = (V, E)$ made of V knots and E edges. Each explanatory variable corresponds to a node on the graph and an edge symbolizes the link between a pair of separate nodes in G.

For a fixed partition vector $u = (n_{\mathcal{A}}, n_{\mathcal{B}})$, the GFL criterion to minimize is written :

$$\frac{1}{2}\sum_{i=1}^{n}(y_i - X_i^u\beta)^2 + \lambda_p \sum_{j \in V^u} |\beta_j| + \lambda_f \sum_{(j,j') \in E^u} |\beta_j - \beta_{j'}|, \tag{2.2.5}$$

with respect to β, where :

- $\beta = (\beta_{(1,1)}, \beta_{(2,1)}, \ldots, \beta_{(1,2)}, \ldots, \beta_{(n_{\mathcal{A}}, n_{\mathcal{B}})})^T \in \mathbb{R}^{n_{\mathcal{A}} n_{\mathcal{B}}}$ the unknown coefficients,
- $\lambda_p \geq 0$ and $\lambda_f > 0$ the regularization parameters (of parsimony and fusion) to be optimized,
- for $j = (v, w)$ fixed, the couples (j, j') relative to j and contained in E^u are $((v, w), (v+1, w))$ and $((v, w), (v, w+1))$. In the following and depending on the context, the index j will refer either to the pair (v, w), or to the j^{th} element of the vector obtained from the matrix stored by columns.

The argmin solution of (2.2.5), denoted $\widehat{\beta}^u$, is computed as a function of the regularization parameters λ_p and λ_f, for a fixed value of u.

The Generalized Fused Lasso in the Generalized Lasso framework

If differences of contiguous coefficients were not penalized in (2.2.5) (i.e. if λ_f was zero), then the criterion would reduce to the Lasso criterion presented by [112]. The Lasso minimizes the residual sum of squares subject to the constraint that the sum of the absolute value of the coefficients is less than a constant. In this case, there is only one regularization parameter to estimate. [27] proposed the Least-Angle Regression (LARS) algorithm able to solve the problem for all $\lambda \in [0, \infty)$, producing a full piece-wise linear solution path. The result of the

path algorithm is the finite set of increasing λ values, where each λ delimits a model reduction dimension (the number of non-zero β components). Criterion (2.2.5) has two regularization parameters λ_p and λ_f to be optimized. The path algorithm is no longer suitable to identify them.

Our proposal is to parameterize (2.2.5) with the ratio :

$$\gamma = \frac{\lambda_p}{\lambda_f}, \tag{2.2.6}$$

(choice made in the R package genlasso [4] used for implementing SpiceFP). This ratio represents a balance between parsimony and fusion. Then, for a fixed value of γ, criterion (2.2.5) can be equivalently rewritten as :

$$\frac{1}{2}||y - X^u\beta||_2^2 + \lambda||D^{u,\gamma}\beta||_1, \tag{2.2.7}$$

where $||.||_q$ designs the L_q norm, $y = (y_1, y_2, \ldots, y_n)^T \in \mathbb{R}^n$ is the response vector and $D^{u,\gamma}$ is a specified penalty matrix (see below). This model corresponds to the Generalized Lasso model, introduced by [114] as an encapsulation of statistical models using the L_1 norm to impose additional constraints. Through this new parametrization, the value of $\lambda = \lambda_f$ can be optimized with the path algorithm and a corresponding $\widehat{\beta}^{u,\gamma}(\lambda)$ can be estimated. Finally, there are as many pairs $(\lambda, \widehat{\beta}^{u,\gamma}(\lambda))$ solutions as there are u and γ parameters set. So, several models are available and we have to select one of them in order to deduce the optimal pair. Selection of the best model is done with an information criteria which requires an estimation of the degree of freedom for each model. Parameter λ_p will be deduced from γ through $\lambda_p = \gamma\lambda_f$.

In our context, the penalty matrix $D^{u,\gamma}$ is a row-binding of two sub-matrices, see details in 2.7.2. The Generalized Fused Lasso presented in criterion (2.2.5) is a 2 Dimensional - Sparse Fused Lasso (2d-SFL).

Degrees of freedom of the Generalized Fused Lasso fit

Our approach follows that of [115], which established the calculation of the degree of freedom for any Lasso-type regression written as a generalized Lasso problem as presented in (2.2.7).

We first introduce some notations. For any penalty matrix $D \in \mathbb{R}^{m\times p}$ involved in a generalized Lasso problem of type (2.2.7), let $\mathcal{S}$ be the active set corresponding to a particular solution $\hat{\beta}$, defined as :

$$\mathcal{S} = \{r \in \{1, \ldots, m\} : (D\hat{\beta})_r \neq 0\} = support(D\hat{\beta}).$$

Let $D_{-\mathcal{S}}$ be the matrix D from which were removed the rows indexed by $\mathcal{S}$. And let $null(D_{-\mathcal{S}})$ be the null space or kernel of $D_{-\mathcal{S}}$. The calculation of the degrees of freedom of the generalized Lasso fit is stated in [115, Theorem 3] and reminded in the appendix. To compute the degree of freedom (denoted $df(X^u\hat{\beta}^{u,\gamma})$), we need $\beta^{u,\gamma}$, X^u, $D^{u,\gamma}$, λ defined in Equation (2.2.7) and

γ in Equation (2.2.6). Note that, in our context, $m = n_D + n_{\mathcal{A}} n_{\mathcal{B}}$ the number of rows in $D^{u,\gamma}$, and $p = n_{\mathcal{A}} n_{\mathcal{B}}$ the number of columns in X^u.

With fused lasso constraints, we are interested in sets of parameters that share the same value. By this, we refer to connected components, which are illustrated in Figure 2.2.4 and defined as follows :

Definition 2.2.1. *A* **connected component** *cc is a set of indexes of non zero coefficients $\beta^{u,\gamma}_{v,w}$ that are linked together via the $D^{u,\gamma}$ matrix (2.7.1) and that all share the same real value.*

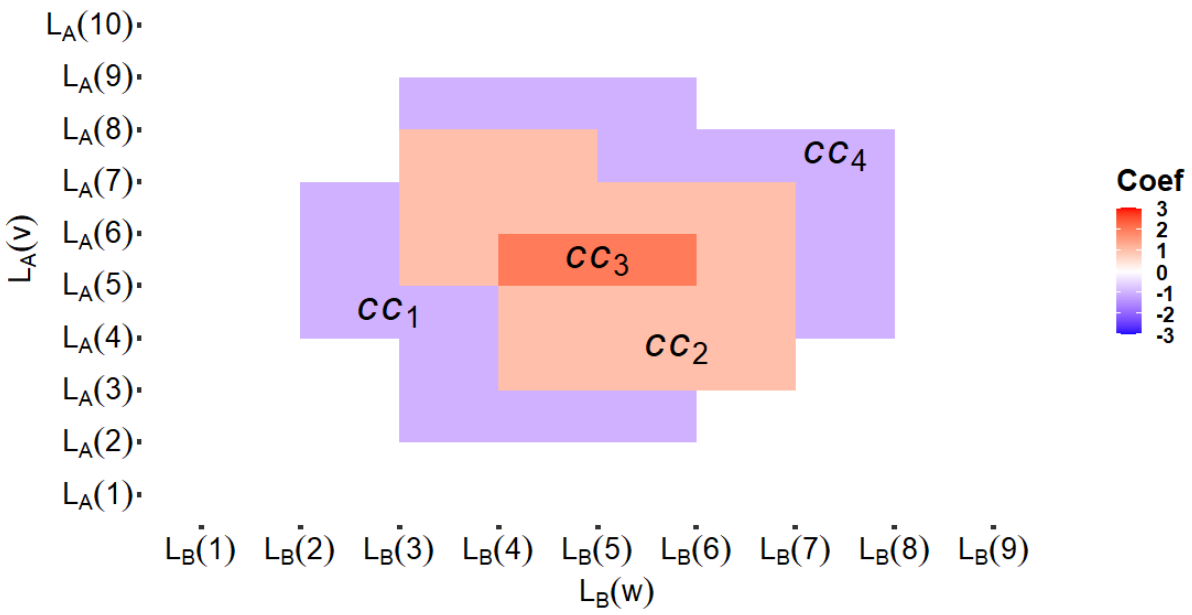


FIGURE 2.2.2 – Example of $\beta^{u,\gamma}$ coefficient values with 4 connected components (here $u = (9, 8)$).

Two coefficients $\beta^{u,\gamma}_{v,w}$ and $\beta^{u,\gamma}_{v',w'}$ are linked via the $D^{u,\gamma}$ matrix if and only if $|v' - v| = 1$ or (exclusive) $|w' - w| = 1$ (i.e., if and only if $|v' - v| + |w' - w| = 1$). Let us consider that the estimated coefficient $\widehat{\beta}^{u,\gamma}$ contains $Q^{u,\gamma}$ connected components denoted cc_q, $q = 1, 2, \ldots, Q^{u,\gamma}$. In order to identify the respective coefficients involved in each non-zero connected component, we introduce the matrix Θ through Definition 2.2.2.

Definition 2.2.2. *Let $(cc_q)_{q=1,\ldots,Q}$ be a set of connected components. The* ***connected component membership matrix*** *is a binary matrix $\Theta \in \mathbb{R}^{n_{\mathcal{A}} n_{\mathcal{B}} \times Q}$ whose q-th column $\Theta^{(q)}$ indicates the membership or not of the $\hat{\beta}^{u,\gamma}$ components to the connected component cc_q, as follows :*

$$\forall j = 1, \ldots, n_{\mathcal{A}} n_{\mathcal{B}},\ \Theta^{(q)}_j = \begin{cases} 0 & \text{if} \quad j \notin cc_q \\ 1 & \text{if} \quad j \in cc_q \end{cases},\ q = 1, \ldots, Q.$$

Let's now compute the null space of $D_{-\mathcal{S}}$, denoted null($D_{-\mathcal{S}}$), this space is generated by :

$$\{\Theta \in \mathbb{R}^{n_{\mathcal{A}} n_{\mathcal{B}} \times Q} | D_{-\mathcal{S}}\Theta^{(q)} = 0_{(m-s)}, q = 1, \ldots, Q\},$$

where $D_{-\mathcal{S}} \in \mathbb{R}^{(m-s)\times n_{\mathcal{A}} n_{\mathcal{B}}}$, $D_{-\mathcal{S}}\Theta^{(q)} \in \mathbb{R}^{m-s}$ and $s = \text{Card}(S)$.

As the $D^{u,\gamma}$ matrix has a simple structure, adapting Theorem 3 of [115] (see 2.7.3) in the context of 2d-Sparse Fused Lasso is equivalent to looking for the components of $\widehat{\beta}^{u,\gamma}$ which have different values. This is the subject of the next corollary.

Corollary 2.2.3. *The degree of freedom $\widehat{df}(X\widehat{\beta}^{u,\gamma})$ associated to the criterion (2.2.7) in the context of 2d-Sparse Fused Lasso is equal to the number of connected components $Q^{u,\gamma}$.*

Proof :
$$\begin{aligned}\widehat{df}(X\widehat{\beta}^{u,\gamma}) &= \dim\left(X^u(\text{null}(D^{u,\gamma}_{-\mathcal{S}}))\right) = \dim\left(X^u(\Theta)\right) \\ &= \dim\left(\text{Vect}\left\{X^u\Theta^{(q)},\ q = 1, \ldots, Q^{u,\gamma}\right\}\right) = \text{rank}([X^u\Theta^{(1)}, \ldots, X^u\Theta^{(Q^{u,\gamma})}]) = Q^{u,\gamma},\end{aligned}$$

where we omitted Θ dependencies on u and γ to lighten the notations in the corollary and where the notation $X^u(V)$ represents the image space of a subspace V by X^u. It is the space generated by the columns of the X^u matrix projected on V. □

Choice of the best candidate matrix and selection of its variables

SpiceFP requires the construction of different candidate explanatory matrices X^u from both functional variables and partition's vector u. Constructing a GFL for a matrix of predictors associated to a fixed u requires identifying the optimal values of the penalty parameters : λ and γ in (2.2.7). In penalized regressions, cross-validation is often used to optimize regularization parameters, but it is time consuming. We suggest using an information criterion to achieve the same purpose [33]. To that aim, we computed an adapted information criterion for each model indexed by u, λ and γ. The best model is obtained by minimizing the information criterion chosen, which yields the best partition $\widehat{u}$ and allows to select the best variables from $X^{\widehat{u}}$. The variable selection is done through the selection of $\widehat{\gamma}$ and $\widehat{\lambda}$ and the associated non-zero $\widehat{\beta}$ components are deduced.

There exist various information criteria including Akaike Information Criterion (AIC) [2] and Bayesian Information Criterion (BIC) [102]. These criteria penalize log-likelihood by the number of model parameters. The BIC also penalizes log-likelihood by the sample size. A well defined penalization is essential to compare regression models involving different explanatory matrices. These information criteria require computing the degree of freedom $Q^{u,\gamma}$ of the GFL model, obtained in Corollary 2.2.3.

For each u and γ defined on respective grids (given by the users) and λ taken from the set $(\lambda_e)_{e=1,\ldots,N_\lambda}$ delivered by the path algorithm, we considered the following information criteria :

— $AIC^{u,\gamma}_e = -2\log(L^{u,\gamma}_e) + 2Q^{u,\gamma}_e$,

— $BIC^{u,\gamma}_e = -2\log(L^{u,\gamma}_e) + \log(n)Q^{u,\gamma}_e$.

with $L_e^{u,\gamma}$ the likelihood function of the following model : $y = X^u\beta_e^{u,\gamma}+\varepsilon$ with $\varepsilon \sim \mathcal{N}(0,\ \sigma^2 I)$, associated with criterion (2.2.7). We have :

$$-2\log(L_e^{u,\gamma}) = 2n\log(\sigma) + n\log(2\pi) + \frac{1}{\sigma^2}||y - X^u\beta_e^{u,\gamma}||_2^2,$$

where the variance of the residuals σ^2 is unknown. As mentioned by [46], the same variance estimator must be used in the calculation of the criteria for all the constructed models. We thus decided to estimate σ^2 by the variance of the response variable : $\widehat{\sigma}^2 = \frac{1}{n-1}||y - \overline{y}||_2^2$. It's a biased estimator of σ^2, but this bias remains fixed for all models compared [46]. Such an estimator may lead to an overestimation of the variance, which penalizes the introduction of new coefficients in the model.

2.2.5 SpiceFP algorithm

Presentation

The SpiceFP algorithm is presented in algorithm 1. The algorithm is divided in two major steps : transformation of the input functional data and estimation with selection of the best models with respect to the Akaike criteria (AIC, BIC).

The inputs are of two types : (i) observed values of the functional variables at the same observation times and values of the response variable for all individuals $(\mathcal{A}_i(t), \mathcal{B}_i(t),\ t \in T\,;\ y_i),\ i = 1,\ldots,n)$; (ii) sets of values for the parameters : $\mathcal{U}_\mathcal{A}$,$\mathcal{U}_\mathcal{B}$ (sets of $n_\mathcal{A}$ and $n_\mathcal{B}$ values), Γ (set of γ values for the Generalized Lasso), n_λ (number of λ values to be used).

The output of the algorithm is a list of information on the estimated model : design matrix ($X^{\widehat{u}}$, partition $\widehat{u}$, estimated vector of non null regression coefficients $\widehat{\beta}$, penalization parameters $\widehat{\lambda}, \widehat{\gamma}$).

Adaptation of SpiceFP to the partitioning of functional variables according to a non-linear scale

A linear partitioning of type (2.2.1) of the functional variables is not always suitable. Other types of partitioning may have to be chosen. Assuming for example that functional variable $\mathcal{B}$ requires a partitioning according to a logarithmic scale, the following breaks can be used instead of Equation (2.2.1) :

$$L_\mathcal{B}^u(w) = \underline{\mathcal{B}} + \frac{e^{\alpha_\mathcal{B}\frac{w-1}{n_\mathcal{B}}} - 1}{e^{\alpha_\mathcal{B}} - 1}(\bar{\mathcal{B}} - \underline{\mathcal{B}}),\ w = 1,\ldots,n_\mathcal{B}+1, \tag{2.2.8}$$

where parameters $\alpha_\mathcal{B} > 0$ and $n_\mathcal{B}$ have to be set to determine $L_\mathcal{B}^u(w)$. For a fixed $n_\mathcal{B}$, high $\alpha_\mathcal{B}$ value is related to high proportion of breaks close to $\underline{\mathcal{B}}$ and vice versa. So, two partitioning parameters ($n_\mathcal{B}$ and $\alpha_\mathcal{B}$) should be optimized for partitioning $\mathcal{B}$. In this case, the partition vector is written $u = (n_\mathcal{A}, n_\mathcal{B}, \alpha_\mathcal{B})$. Let $\mathcal{V}_\mathcal{B}$ be the set containing the possible values of $\alpha_\mathcal{B}$. $\mathcal{V}_\mathcal{B}$ is an additional input to the algorithm. The only change in the core of the algorithm is to replace $u \in \mathcal{U}_\mathcal{A}.\ \mathcal{U}_\mathcal{B}$ by $u \in \mathcal{U}_\mathcal{A}.\ \mathcal{U}_\mathcal{B}.\ \mathcal{V}_\mathcal{B}$ (lines 1 to 14).

Data : $(\mathcal{A}_i(t), \mathcal{B}_i(t),\, t \in T\,;\, y_i),\, i = 1, \ldots, n$

Input : $\mathcal{U}_\mathcal{A}$,$\mathcal{U}_\mathcal{B}\,;\, \Gamma\,;\, n_\lambda\,;\, Crit \in \{AIC, BIC\}$

Output : $(X^{\widehat{u}}, \widehat{u}, \widehat{\lambda}, \widehat{\gamma}), \widehat{\beta},$

foreach $u \in \mathcal{U}_\mathcal{A} \times \mathcal{U}_\mathcal{B}$ **do**
  for $i \leftarrow 1$ **to** n **do**
    for $v \leftarrow 1$ **to** $n_\mathcal{A}$ **do**
      for $w \leftarrow 1$ **to** $n_\mathcal{B}$ **do**
        $C^u_{i,(v,w)} = \text{Card}\ \{t \in T | \mathcal{A}_i(t) \in I^u_\mathcal{A}(v), \mathcal{B}_i(t) \in I^u_\mathcal{B}(w)\}$
      end
    end
    $X^u[i,] = \text{Vect}(C^u_i)^T$
  end
  foreach $\gamma \in \Gamma$ **do**
    Construct $D^{u,\gamma}$ as shown in the Equation (2.7.1)
  end
end
foreach $u \in \mathcal{U}_\mathcal{A} \times \mathcal{U}_\mathcal{B}$ **do**
  Center y and each joint modality in X^u
  foreach $\gamma \in \Gamma$ **do**
    Find the solution path $\widehat{\beta}^{u,\gamma}(\lambda) = \underset{\beta \in \mathbb{R}^{n_\mathcal{A} n_\mathcal{B}}}{\text{argmin}}\ \frac{1}{2}||y - X^u\beta||_2^2 + \lambda||D^{u,\gamma}\beta||_1$
    Select n_λ equally spaced couples $(\lambda_e, \widehat{\beta}^{u,\gamma}_e)$ on the log scale with respect to λ over the solution path and compute $Crit^{u,\gamma}_e$ for $e = 1 \ldots n_\lambda$:
    $AIC^{u,\gamma}_e = \frac{1}{\sigma^2}||y - X^u\widehat{\beta}^{u,\gamma}_e||_2^2 + 2Q^{u,\gamma}_e$ or
    $BIC^{u,\gamma}_e = \frac{1}{\sigma^2}||y - X^u\widehat{\beta}^{u,\gamma}_e||_2^2 + \log(n)Q^{u,\gamma}_e$
  end
end
$(\widehat{u}, \widehat{\gamma}, \widehat{e}) \leftarrow \underset{u \in \mathcal{U}_\mathcal{A} \times \mathcal{U}_\mathcal{B},\ \gamma \in \Gamma,\ e \leq n_\lambda}{\text{argmin}} Crit^{u,\gamma}_e$
$\widehat{\lambda} = \lambda_{\widehat{e}}, \widehat{\beta} = \widehat{\beta}^{\widehat{u},\widehat{\gamma}}_{\widehat{e}}$

Algorithme 1 : SpiceFP algorithm

Various types of breaks (chosen between Equations (2.2.1), (2.2.8) or user-defined) can be used for partitioning explanatory functional variables and the SpiceFP algorithm can be easily adapted.

2.3 Use Case : Grapevine dataset

2.3.1 Data presentation

Data were collected during an experiment conducted in a vineyard of the Institut Agro campus in Montpellier in 2014 (Syrah vines). The aim was to study the influence of the micro-climate (temperature, solar irradiation) at the grape level on the anthocyanin content of the berries. Experts in viticulture assume that the accumulation of chemical compounds affecting the quality of the grape berry is jointly influenced by these initial explanatory variables. This assumption is reinforced by results of [111], which underlined that the anthocyanin composition of Merlot grapes was influenced by a complex combined effect of berry temperature and solar irradiation.

The experimental plot was made of three rows of vines within the vineyard, each with eight vines equipped with open-top chambers to warm the base of the plant, and eight under control conditions (without open-top chambers). The chambers were made up of 2 translucent polycarbonate panels placed on the ground at about 10 cm below the bunches on each side of the vines and inclined to form a mini two-pitched greenhouse roof open at its ridge. The greenhouse effect created during the day in the chambers generated a flow of warm air that escaped through the open top, raising the temperature of the bunches by 2 to 3°C, mimicking global warming. The microclimate at bunch level was recorded through the measurement of temperature and irradiance. According to [118], solar radiation can be characterized by three different quantifiers, including Photosynthetic Photon Flux Density (PPFD), measured in $10^{-6} mol.m^{-2}.s^{-1}$. This corresponds to the number of incident photons useful for photosynthesis, received per unit of time on a horizontal surface unit. Rows were roughly oriented south-north, and irradiance was separately measured on bunches located on the east and west side of the row. This measurement system therefore enabled the observation of different modalities of the couple (temperature, irradiance) affecting the grape berries. Temperature and Irradiance were recorded every twelve minutes throughout the maturation period when anthocyanins are known to accumulate.

So the experimental design contained : rows (1, 2, 3) $\times$ 2 sun exposure orientations (East, West) $\times$ 16 vine stocks = 96 'statistical individuals'. Anthocyanin contents were measured weekly via the Ferari Index FI_i for each individual i. This is a non-destructive measure of anthocyanin content of the bunches [1]. The objective of our study is to understand how the couple (Temperature, Irradiance) acts on a weekly variation of the Ferari Index ΔFI.

Temperature and Irradiance are variables of different natures. Temperature is a variable whose variations are regular enough to be partitioned according to Equation (2.2.1). The Irradiance variable is partitioned according to (2.2.8), as explained in the following subsection.

2.3.2 Partitioning of the Irradiance variable

Observed on a one-day scale, PPFD increases exponentially from sunrise to a daily peak (observation time t_{max}), decreases until sunset, and remains almost constant until the next sunrise. Irradiance primarily influences plant photosynthesis in a nonlinear way with a maximal reached at high irradiance. Therefore, the Irradiance variable was not partitioned according to a linear scale as proposed by Equation (2.2.1), but rather according to a logarithmic scale as in Equation (2.2.8). The use of logarithmic transformation has consistently been used in the development of models involving solar radiation [99], [79], [11].

2.3.3 Objective

The objective of our study was to identify the ranges of temperature and irradiance that jointly influence or not the accumulation of anthocyanins between sunrise and noon. This study is an appropriate application framework to test the SpiceFP method.

Our study is composed of two parts. First a simulation part is presented in section 2.4. Temperature and Irradiance, measured during the week of July 17 to 24, 2014, are the input variables of a model which simulates output variables to be predicted, using two different known β : one with two distinct patches of coefficients and another with a concentric gradient of coefficients (Table 2.3, column 1). This simulation study made it possible to evaluate the functioning and performance of the SpiceFP algorithm. Second, in section 2.5, the algorithm was tested on a complete dataset, obtained from July 24 to August 1, 2014, to understand the effect of Temperature and Irradiance interaction on ΔFI. Dataset and script are available online at https ://forgemia.inra.fr/exploratory-penalized-regression/paper-script-and-data.git.

2.4 Simulation study

We present in this section simulations that help to better understand the SpiceFP characteristics. Remember that the approach must be able to identify an optimal partition or joint class intervals used as linear regressors.

2.4.1 Simulation design and SpiceFP setting

In order to carry out the simulations, a few steps were required :

— We considered the observations of temperature ($\mathcal{A}$) and irradiance ($\mathcal{B}$) in the Vine dataset obtained between sunrise and noon during the week of July 17 to 24, 2014.

— We then arbitrarily set a partition vector $u^0 = (n_{\mathcal{A}} = 17, n_{\mathcal{B}} = 20, \alpha_{\mathcal{B}} = 4.25)$.

— From $\mathcal{A}$, $\mathcal{B}$, and u^0 we constructed X^{u^0} using Equations (2.2.2) and (2.2.4). X^{u^0} was computed based on the observed data set in order to respect realistic frequencies for the joint class intervals.

Table 2.1 – Noise simulation design

	Low noise	High noise
Simulation 1	$\sigma_\varepsilon = 1.50$, $\sigma_Y = 8.66$ ($\sigma^2_\varepsilon/\sigma^2_Y = 0.03$)	$\sigma_\varepsilon = 2.5$, $\sigma_Y = 8.82$ ($\sigma^2_\varepsilon/\sigma^2_Y = 0.08$)
Simulation 2	$\sigma_\varepsilon = 0.25$, $\sigma_Y = 2.67$ ($\sigma^2_\varepsilon/\sigma^2_Y = 0.01$)	$\sigma_\varepsilon = 1.0$, $\sigma_Y = 2.64$ ($\sigma^2_\varepsilon/\sigma^2_Y = 0.14$)

— Based on dimensions of X^{u^0}, we drew coefficients β^{u^0} from a random distribution (values are available on git@forgemia.inra.fr) and computed the response variable of the simulation :

$$Y = X^{u^0}\beta^{u^0} + \varepsilon \text{ where } \varepsilon \sim \mathcal{N}(0, \sigma^2_\varepsilon I). \tag{2.4.1}$$

After that, we used the SpiceFP approach to make the estimation.

In this simulation study, we chose to simulate two different coefficient vectors. They are as follows :

— Coefficients β^{u_0} for simulation 1 is made of two distinct patches (column 1, row 1 of the Table 2.3). Two noise levels (error level σ^2_ε nested in the response level) were used and presented in Table 2.1. The estimated standard deviation σ_Y includes variability on coefficients and measurement error.

— Coefficients β^{u_0} for simulation 2 is made of a concentric gradient of coefficients (column 1, row 2 of Table 2.3). Two noise levels were also used and presented in Table 2.1.

We thus generated four simulation datasets (1 matrix X^{u_0} × 2 simulated coefficient vectors × 2 noise levels).

The inputs required by the algorithm were as follows :

— $\mathcal{U}_\mathcal{A} = \{15, 16, \ldots, 20\}$

— $\mathcal{U}_\mathcal{B} = \{18, 19, \ldots, 22\}$

— $\mathcal{V}_\mathcal{B} = \{3.08, 4.25, 5.24, 6.33, 7.43\}$

— $\Gamma = \{0.0001, 0.05, 0.15, 0.45, 2, 8\}$

— $n_\lambda = 100$.

2.4.2 Simulation results

The results of the simulations are presented in Table 2.3 (histograms of the residuals are given in appendices). Table 2.3 is related to the estimated coefficient vectors. For each of the four response variables (one per row), three estimations are provided (rows 3 and 4, Table 2.3), computed as follows :

— row 3 : an exploratory matrix $X^{\widehat{u}}$ indexed by a partition vector $\widehat{u}$ is identified. The estimated coefficient of SpiceFP is noted $\beta^{\widehat{u}}$. The estimated response $\widehat{Y} = X^{\widehat{u}}\beta^{\widehat{u}}$ and the residuals $\varepsilon = Y - \widehat{Y}$ are computed. Figures in row 3 provide the visualization of $\beta^{\widehat{u}}$. For a suitable visualization, the vector of coefficients is transformed into a matrix of dimension $n_\mathcal{A} \times n_\mathcal{B}$, where $n_\mathcal{A}$ and $n_\mathcal{B}$ are the numbers of class intervals associated to u.

TABLE 2.2 – Slope of the regression 'predicted versus simulated y values'

	Best model		Average of 1% best models	
	Low noise	High noise	Low noise	High noise
Simulation 1	0.882	0.891	0.848	0.836
Simulation 2	0.712	0.680	0.726	0.623

— row 4 : the results presented in this row are slightly out of the scope of the SpiceFP approach. The idea here is to consider an average of the 1% best models. The best models are defined in the sense of the information criterion. Best coefficients relative to the different partitions available in these 1% best models are identified. Let's assume that there are n_m. They are then indexed by $u^{(1)}, u^{(2)}, \ldots, u^{(n_m)}$. The estimated response is $\widehat{Y}^{(1:n_m)} = \frac{1}{n_m} \sum_{m=1}^{n_m} X^{u^{(m)}} \beta^{u^{(m)}}$. All the selected models have the same weight in the computation of this average. The residuals can be obtained by $\varepsilon^{(1:n_m)} = Y - \widehat{Y}^{(1:n_m)}$. Figures in row 4 show the visualization of $\frac{1}{n_m} \sum_{m=1}^{n_m} \mathcal{C}^{u^{(m)}}(a,b)\beta^{u^{(m)}}(a,b)$ where $\mathcal{C}^u(a,b) = C^u_{v,w}$ and $\beta^u(a,b) = \beta^u_{(v,w)}$ for $a \in I^u_{\mathcal{A}}(v), b \in I^u_{\mathcal{B}}(w)$.

The black colour refers to never-observed joint modalities whereas the white color refers to joint modalities with null estimated coefficients. In order to check the goodness of fit of each estimation, Table 2.2 shows for each estimation the slope of the regression 'predicted versus simulated y values'. The estimator will be better the closer to 1 the slope is and the closer to 0 the ratio is.

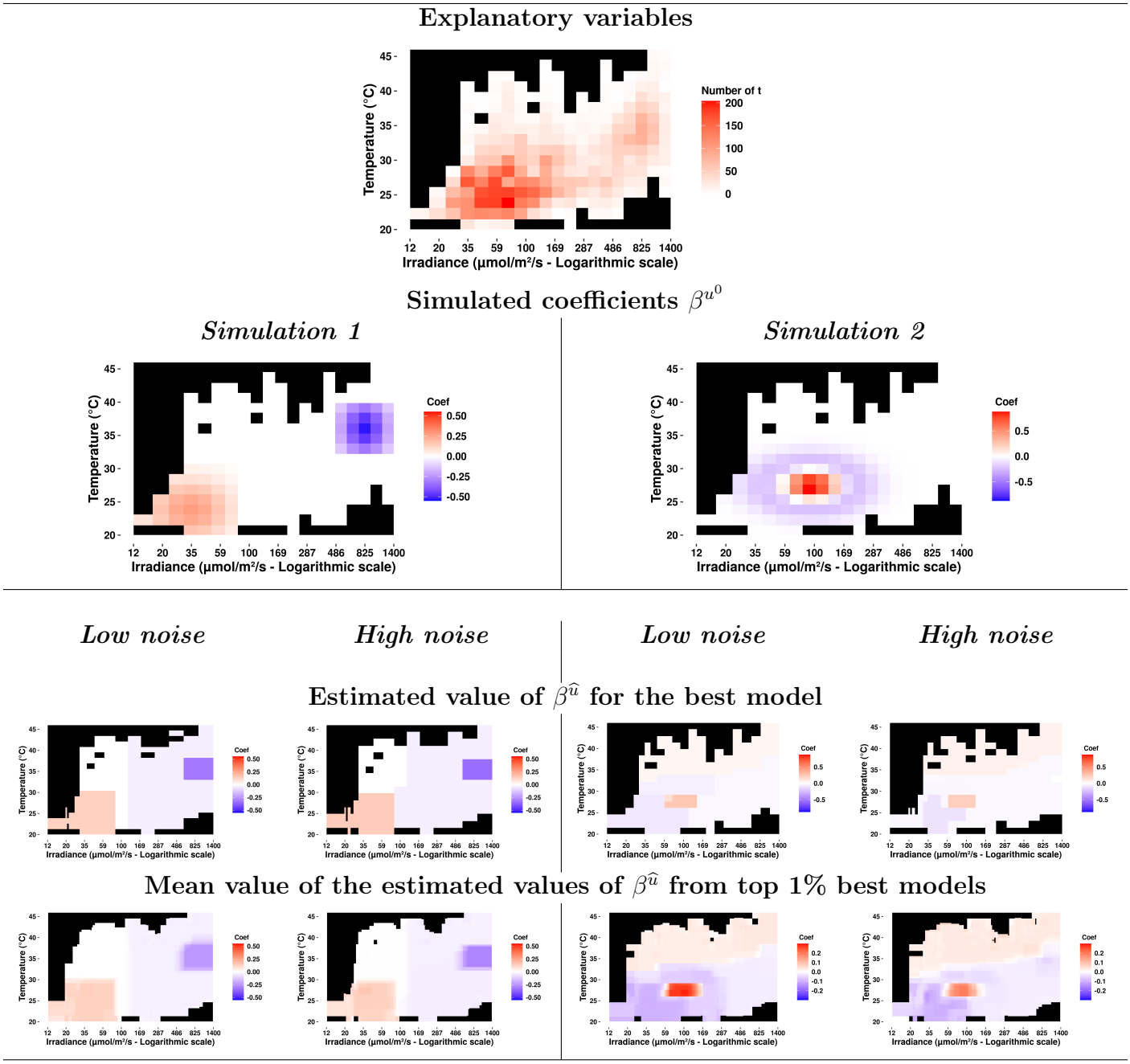
From the estimates provided by the algorithm (Table 2.3), we notice that SpiceFP effectively identifies the simulated zones of influence and assigns the right color to the coefficients : the graphics show two distinct areas, one with positive coefficients (red area), the other with negative coefficients (blue area). The approach tends to assign the same value (same color) to groups of estimated coefficients, although with a gradient within the group. This behavior can be explained on the one hand by the Generalized Fused Lasso which penalizes the difference between two related coefficients and on the other hand by the variance estimate used for computing the information criterion. This variance was overestimated, which penalized the introduction of new coefficients into the model. With respect to the noise level contained in the response variable, we note that the more noisy Y is, the more false positives are observed. The average of the 1% best models underestimates, in both simulations, the amplitude of the β values but, in simulation 2, it restores the β support much better.

2.5 Modeling the evolution of a grape berry quality index

2.5.1 Methodology used for data analysis

We focus in this section on the modeling of the Ferari Index variation ΔFI from July 24 to August 1, 2014. We selected the $n_1 = 32$ individuals which have the highest contribution

Table 2.3 – Simulation results : estimation with the SpiceFP algorithm of the two simulations coefficient β^{u^0}. Each simulation was done with two types of noise (high and low). ■ : Joint modalities that have never been observed.



to the final Ferari Index, with an initial index around 0.2 at the beginning of the week.

Most of the photosynthetic functioning of the grapevine occurs in the morning. The time period between sunrise and noon is denoted T_1 below. The following variables and parameters are the input objects of SpiceFP :

— $y_i = \Delta FI_i$, $\mathcal{A}_i(t)$, $\mathcal{B}_i(t)$, for $i = 1, \ldots, n_1$ and $t \in T_1$,

— $\mathcal{U}_\mathcal{A} = \mathcal{U}_\mathcal{B} = \{10, 11, 12, \ldots, 29, 30\}$,

— $\mathcal{V}_\mathcal{B} = \{1.50, 2.03, 2.64, 3.27, 3.92, 4.58, 5.24, 5.90, 6.57, 7.23\}$,

— $\Gamma = \{0.001, 0.01, 0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.4, 12.8, 25.6\}$,

— $n_\lambda = 20$.

2.5.2 Results

The results are presented in Table 2.4. The first column shows the estimated coefficients and the second column the histograms of residuals.

The first row of Table 2.4 contains the results observed with SpiceFP. In terms of model quality, the slope between predicted and observed values is 0.558 and the residuals follow a normal distribution centered in 0. The visualization of the coefficients indicates conditions (irradiance < 100 $\mu mol\ m^{-2}\ s^{-1}$, temperature from 15°C to 33°C) that affect the Ferari Index negatively.

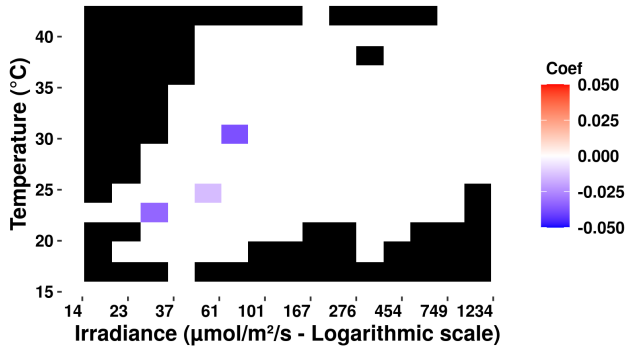
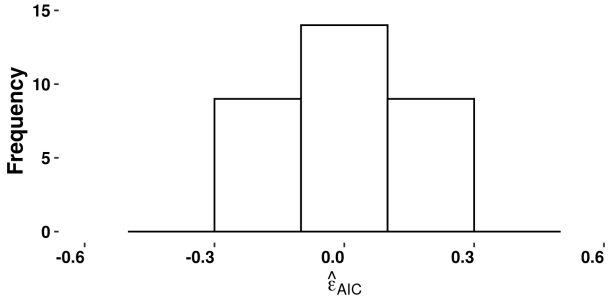
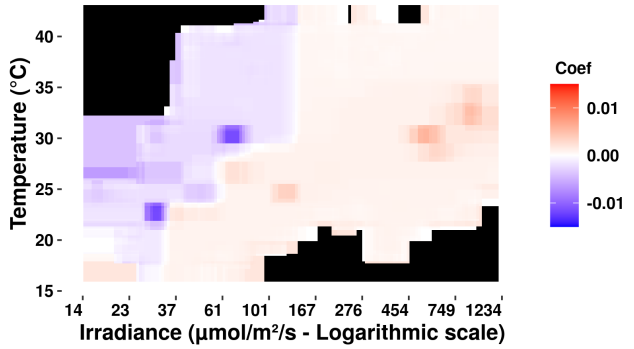
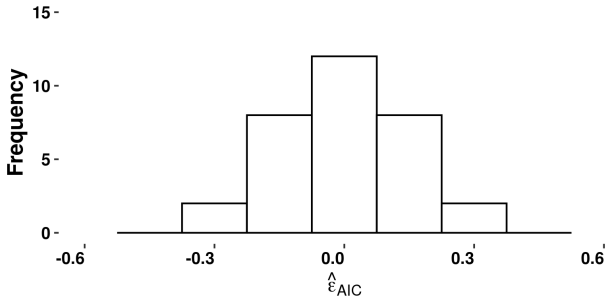
When focusing on the average of the 1% best models (presented in the second row), we remark that the quality of this model (as indicated by the slope between predicted and observed value : 0.472) is not better than that of the models chosen by SpiceFP. However, it should be noted that the model obtained has more fused coefficients, which allows the identification of a border zone between the positive and negative zones of influence.

We can note that, in the morning (sunrise to noon), for low irradiance values (< 100 $\mu mol\ m^{-2}\ s^{-1}$), there is a range of temperature values that are not suitable for an increase of the Ferari index. On the contrary, a combination of irradiance values above 150 $\mu mol\ m^{-2}\ s^{-1}$ and temperature below 30°C is suitable for increasing the Ferari index. The average of the coefficients shows, within each of the non-zero zones of influence, a relative variability of the coefficients, suggesting the importance of some temperature and irradiance amplitudes. The slope coefficients of the goodness of fit of the models are away from 1. It should be remembered that the response variable is studied in respect to the variations of temperature and irradiance only between sunrise and twelve. There is also a lot of information hidden in the residuals.

2.6 Discussion

The SpiceFP approach is a scalar-on-function approach. The response variable is real and predictors are functional variables. The approach is based on a transformation of functional variables, which yields a contingency table. To construct this table, it is assumed that no data are missing and that the times of observation are identical for both functional variables. The method takes into account the multicolinearity resulting from the auto-correlations existing

TABLE 2.4 – Visualization of the combined effects of irradiance and temperature on the Ferari index (From sunrise to twelve, ■ : joint modalities that have never been observed). Row 1 presents the results of the best model. The second row presents the average of the 1% best models .

Coefficients	*Residuals*
Estimated value of $\beta^{\widehat{u}}$ for the best model	$\sigma^2_{\hat{\varepsilon}}/\sigma^2_y = 0.327$
	
Mean value of the estimated values of $\beta^{\widehat{u}}$ from top 1% best models	$\sigma^2_{\hat{\varepsilon}}/\sigma^2_y = 0.407$
	

in the processes. Moreover, the constructed candidate explanatory matrices are not nested but they cover the same domain. That's why the model and variable selection methods (Fused Lasso, information criterion selection) had to be adapted and generalized to the framework of SpiceFP. In the implementation of the approach, all the candidate explanatory matrices are evaluated.

2.6.1 SpiceFP : a functional approach

In recent years, several studies in frequentist and Bayesian statistics, parametric as well as non-parametric, have focused on functional data analysis. The solutions developed can be used to achieve a wide range of goals (dimension reduction, regression, clustering, classification, etc.). They provide models that are predictive but often difficult to interpret. This lack

of interpretability is partly due to the pre-processing step required on the functional data, which must be projected into bases of functions (splines, kernels...). The transformation of functional variables is a fundamental step, no matter if the variable is a response variable or a predictor.

The SpiceFP approach is primarily explanatory and not necessarily predictive. The first step is a transformation of the predictors into categorical variables. The choice of this transformation was motivated by the potential to interpret the results while considering the hypothesis of a joint influence of the predictors on the response variable. Each partition is a collection of 2D intervals, see §2.2.1. We constructed linear regressors on the basis of indicator functions associated to these 2D intervals, see Equation (2.2.3). This indicator functions basis is not common compared to a polynomial basis, a Fourier basis, a wavelet basis, etc., [120]. In a previous work [70], the authors have focused on multidimensional penalized signal regression, a single surface was estimated with smooth regression coefficient using B-spline tensor products. In our case, the indicator functions facilitate the interpretation of the results, provided that both functional predictors are discretized over *the same set of equidistant observation times* (T in grape berry dataset). This constraint can be released with usual pretreatment such as :

— imputation of missing data [109, 55]

— interpolation, smoothing [88] or restriction of the functional variable to an identical set of observation times. Indeed, functions have uncountable supports.

The use of 2D class intervals is based on the assumption that the structure of the underlying process does not change over the observation variable (time in the grape berry dataset). This is what we can call *hypothesis of stationarity*. For example, in our use case on grapevines, this hypothesis of stationarity requires to work at the scale of a week but also to split the day (work with observations obtained in the morning (sunrise to noon)). When analyzing the data in section 2.5, we assumed that the underlying process is time-invariant in the mornings of the week under consideration.

The method is design-dependent : it will not be able to properly estimate the β coefficient in an area with little or no data. It is therefore necessary to have data of $(\mathcal{A}, \mathcal{B})$ in areas where something is potentially occurring. There is also a limitation due to the curse of dimensionality [36] : "If the number n of observations remains fixed while the dimension p of the observations increases, the observations get rapidly very isolated and local methods cannot work.". The more fluctuations in many directions, the more data will be needed. Our approach will work best if the shapes are simple.

2.6.2 Model specifications

The SpiceFP follows a 'scalar on image' regression model associated with contiguity constraints. In this context some authors used a total variation penalizing approach in order to force some spatial coherence on the regression parameters, see [97]. SpiceFP main concern is the detection of connected components, which does not imply sparsity. But, a second concern is the support recovery with information about the effect (positive or negative)

on the variable of interest. This second issue requires sparsity and therefore, the use of a generalized lasso constraint.

The SpiceFP model directly includes interaction terms, without distinguishing marginal effects. However these marginal effects are taken into account : the model will identify a unique marginal effect without difficulty. In the case of a unique marginal effect in each functional variable, it is also valid but not optimized as the approach will need more than two connected components. The best model might not include the simple marginal effects (as it will be more penalized in the AIC or BIC model selection approaches), but it should appear in the top best models.

The AIC model selection criteria requires the estimation of σ, which is difficult in practice. The estimate of σ used in SpiceFP is conservative. Other approaches deal with this question : the square-root lasso [8] and the quantile universal threshold [35]. The square root does not require the estimation of the variance σ^2 and is not implemented for generalized lasso with contiguity constraints. It may control very well the FDR (False Discovery Rate) but less the TPR (True Positive Rate), according to the simulation results presented in [35]. SpiceFP is constructed to uncover relations in an exploratory approach. In this context, a higher TPR was thus preferred to a perfect control of the FDR. Occasionally, many models have AIC or BIC values close to the best model. In practice, we recommend to check not only the best model, but the top best models to visually inspect the stability of the results (see for example in appendices for simulation 2 low noise output of top best models).

Acknowledgments

The authors dedicate this work to their colleague Eric Lebon, who has passed away. He participated in the very fruitful discussions that initiated the present work, and provided us with the data. We are also grateful to Nicolas Verzelen, researcher at INRAE, for his help and constructive discussions and to the referees for their insightful comments.

Data were collected during the Innovine project, which was funded by the Seventh Framework Program of the European Community (FP7/2007-2013), under Grant Agreement No. FP7-311775. The present work was supported by the French National Research Agency under the Investments for the Future Program, referred as ANR-16-CONV-0004.

2.7 Annexes

2.7.1 Functional model

The SpiceFP model is in discrete time and has the following formulation :

$$y_i = \sum_{v=1}^{n_{\mathcal{A}}} \sum_{w=1}^{n_{\mathcal{B}}} \left(\sum_{t \in T} \mathbb{1}_{\mathcal{A}_i(t) \in I^u_{\mathcal{A}}(v),\ \mathcal{B}_i(t) \in I^u_{\mathcal{B}}(w)} \right) \beta^u_{(v,w)} + \varepsilon.$$

An equivalent version of the regression model for continuous time instead of discrete time could be :

$$y_i = \sum_{v=1}^{n_{\mathcal{A}}} \sum_{w=1}^{n_{\mathcal{B}}} \left(\int_0^{t_{max}} \mathbb{1}_{\mathcal{A}_i(t) \in I^u_{\mathcal{A}}(v),\ \mathcal{B}_i(t) \in I^u_{\mathcal{B}}(w)}\ dt \right)\ \beta^u_{(v,w)} + \varepsilon,$$

where $\mathcal{A}_i(t)$, $\mathcal{B}_i(t)$ are fixed (observed) trajectories for individual i as it is commonly assumed in regression approach.

For each partition u, we may consider a piece-wise constant function β^u with value $\beta^u_{(v,w)}$ on the 2D interval $(I^u_{\mathcal{A}}(v) \times I^u_{\mathcal{B}}(w))$ for all v, w : $\beta^u(a,b) = \sum_{v=1}^{n_{\mathcal{A}}} \sum_{w=1}^{n_{\mathcal{B}}} \mathbb{1}_{a \in I^u_{\mathcal{A}}(v),\ b \in I^u_{\mathcal{B}}(w)} \beta^u_{(v,w)}$, so :

$$\begin{aligned}
y_i &= \int_{\underline{\mathcal{A}}}^{\bar{\mathcal{A}}} \int_{\underline{\mathcal{B}}}^{\bar{\mathcal{B}}} \sum_{v=1}^{n_{\mathcal{A}}} \sum_{w=1}^{n_{\mathcal{B}}} \mathbb{1}_{a \in I^u_{\mathcal{A}}(v), b \in I^u_{\mathcal{B}}(w)} \left(\int_0^{t_{max}} \mathbb{1}_{\mathcal{A}_i(t) \in I^u_{\mathcal{A}}(v),\ \mathcal{B}_i(t) \in I^u_{\mathcal{B}}(w)}\ dt \right)\ \beta^u(a,b)\ da\ db + \varepsilon_i, \\
&= \int_{\underline{\mathcal{A}}}^{\bar{\mathcal{A}}} \int_{\underline{\mathcal{B}}}^{\bar{\mathcal{B}}} \left(\int_0^{t_{max}} \sum_{v=1}^{n_{\mathcal{A}}} \sum_{w=1}^{n_{\mathcal{B}}} \mathbb{1}_{a \in I^u_{\mathcal{A}}(v), b \in I^u_{\mathcal{B}}(w)} \mathbb{1}_{\mathcal{A}_i(t) \in I^u_{\mathcal{A}}(v),\ \mathcal{B}_i(t) \in I^u_{\mathcal{B}}(w)}\ dt \right)\ \beta^u(a,b)\ da\ db + \varepsilon_i, \\
&= \int_{\underline{\mathcal{A}}}^{\bar{\mathcal{A}}} \int_{\underline{\mathcal{B}}}^{\bar{\mathcal{B}}} \left(\int_0^{t_{max}} \sum_{v=1}^{n_{\mathcal{A}}} \mathbb{1}_{a, \mathcal{A}_i(t) \in I^u_{\mathcal{A}}(v)} \sum_{w=1}^{n_{\mathcal{B}}} \mathbb{1}_{b, \mathcal{B}_i(t) \in I^u_{\mathcal{B}}(w)}\ dt \right)\ \beta^u(a,b)\ da\ db + \varepsilon_i.
\end{aligned}$$

$\sum_{v=1}^{n_{\mathcal{A}}} \mathbb{1}_{a, \mathcal{A}_i(t) \in I^u_{\mathcal{A}}(v)}$ may be approximated by $\mathbb{1}_{|\mathcal{A}_i(t) - a| \leq \Delta_{\mathcal{A}}}$ where $\Delta_{\mathcal{A}} = \dfrac{\bar{\mathcal{A}} - \underline{\mathcal{A}}}{2n_{\mathcal{A}}}$, so the SpiceFP model may be viewed as an approximation of the following functional model :

$$y_i = \int_{\underline{\mathcal{A}}}^{\bar{\mathcal{A}}} \int_{\underline{\mathcal{B}}}^{\bar{\mathcal{B}}} g_{\mathcal{A}_i(.), \mathcal{B}_i(.)}(a,b) \beta^u(a,b)\ da\ db + \varepsilon_i,$$

where $g_{\mathcal{A}_i(.), \mathcal{B}_i(.)}$ is defined by $g_{\mathcal{A}_i(.), \mathcal{B}_i(.)}(a,b) = \int_0^{t_{max}} \mathbb{1}_{|\mathcal{A}_i(t) - a| \leq \Delta_{\mathcal{A}},\ |\mathcal{B}_i(t) - b| \leq \Delta_{\mathcal{B}}}\ dt$.

2.7.2 Penalty Matrix

In our context, the penalty matrix $D^{u,\gamma}$ is a row-binding of two sub-matrices as detailed in the following :

- $D^{u,\gamma,p} \in \mathbb{R}^{n_{\mathcal{A}} n_{\mathcal{B}} \times n_{\mathcal{A}} n_{\mathcal{B}}}$: the penalty sub-matrix associated to the regularization of parsimony $\left(\lambda_p \sum_{j \in V^u} |\beta_j| \right)$,
- $D^{u,f} \in \mathbb{R}^{n_D} \times n_{\mathcal{A}} n_{\mathcal{B}}$ where $n_D = 2 n_{\mathcal{A}} n_{\mathcal{B}} - n_{\mathcal{A}} - n_{\mathcal{B}}$: the sub-matrix associated to the regularization of the fusion according to the two dimensions $\left(\lambda_f \sum_{(j,j') \in E^u} |\beta_j - \beta_{j'}| \right)$. Affecting two dimensions, $D^{u,f}$ can be subdivided into $D^{u,f1}$ and $D^{u,f2}$.

Hence $D^{u,\gamma} = \begin{pmatrix} D^{u,f1} \\ D^{u,f2} \\ D^{u,\gamma,p} \end{pmatrix} \in \mathbb{R}^{(n_D + n_{\mathcal{A}} n_{\mathcal{B}}) \times n_{\mathcal{A}} n_{\mathcal{B}}}$ with :

$$
\begin{aligned}
D^{u,f1}_{(v,w)(v',w')} &= \begin{cases} 1 & \text{if } (v',w') = (v+1,w) \\ -1 & \text{if } (v',w') = (v,w) \text{ and } v < n_{\mathcal{A}} \\ 0 & \text{if not} \end{cases}, \\
D^{u,f2}_{(v,w)(v',w')} &= \begin{cases} 1 & \text{if } (v',w') = (v,w+1) \\ -1 & \text{if } (v',w') = (v,w) \text{ and } w < n_{\mathcal{B}} \\ 0 & \text{if not} \end{cases}, \\
D^{u,\gamma,p} &= \gamma . \mathbb{I}_{n_{\mathcal{A}} . n_{\mathcal{B}}} \ ,
\end{aligned}
\tag{2.7.1}
$$

where $\gamma \geq 0$ and $\mathbb{I}_{n_{\mathcal{A}} n_{\mathcal{B}}}$ is the identity matrix.

2.7.3 Degrees of freedom of the Generalized Lasso

Theorem (Tibshirani and Taylor (2012)) *Assume that $y \in \mathbb{R}^n$ follows a normal distribution ($y \sim N(\mu, \sigma^2 I)$ with given (unknown) mean vector $\mu \in \mathbb{R}^n$ and marginal variance σ^2). For any fixed and nonrandom predictor matrix $X \in \mathbb{R}^{n \times p}$, penalty matrix $D \in \mathbb{R}^{m \times p}$ and $\lambda \geq 0$, the degree of freedom of the generalized Lasso fit can be expressed as*

$$
df(X\hat{\beta}) = E[dim(X(null(D_{-\mathcal{S}})))] \ , \tag{2.7.2}
$$

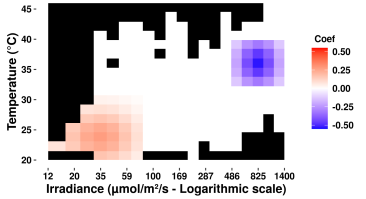
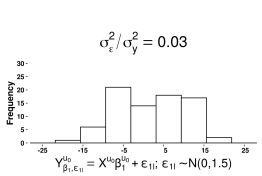
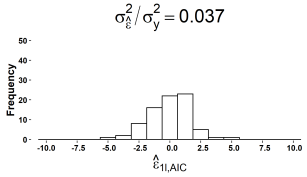
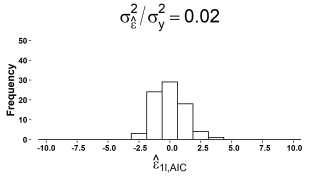
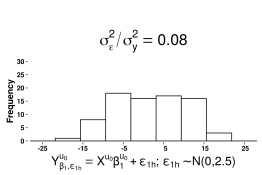
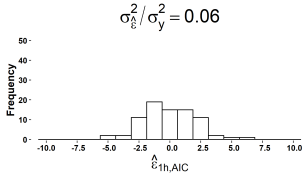
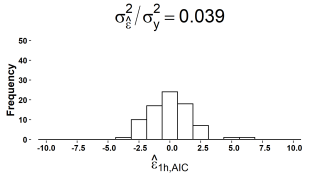
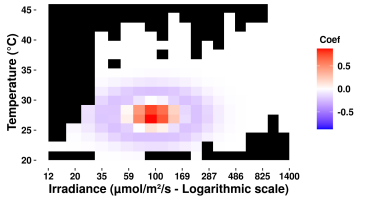
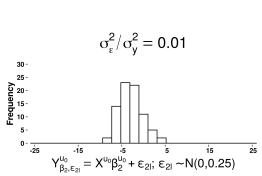
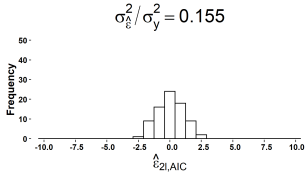
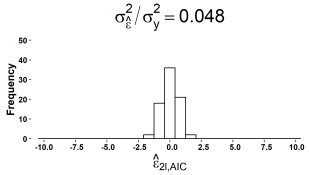
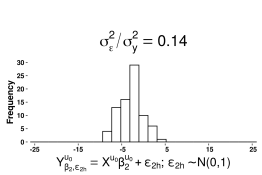
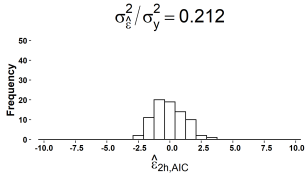
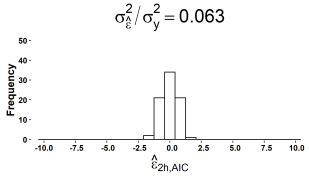
with $\mathcal{S} = \mathcal{S}(y)$ the active set corresponding to any generalized Lasso solution $\hat{\beta}(y)$ at y.

The notation $X(V)$ represents the image space of a subspace V by X. It is the space generated by the columns of the X matrix projected on V. The assumptions required for this Theorem are those usually made in regression estimations (Gaussian i.i.d. errors). No assumptions are made on matrix D nor on X. This result can thus be applied to the 2d-Sparse Fused Lasso constraint.

2.7.4 Simulation's residual histogram and quality of the estimate

The following figure completes the simulation results presented in Table 2.3,

TABLE 2.5 – Histogram of residuals

Simulations	Estimations		
	Responses	Best Model	Mean of top 1
			
			
			
			

■ : Joint modalities that have never been observed (no t counted for these joint modalities for all individuals)

2.7.5 Visual check of the top best models

The following figure illustrates the discussion at the end of section §6.2. In practice, we recommend to check not only the best model, but the top best models to visually inspect the stability of the result.

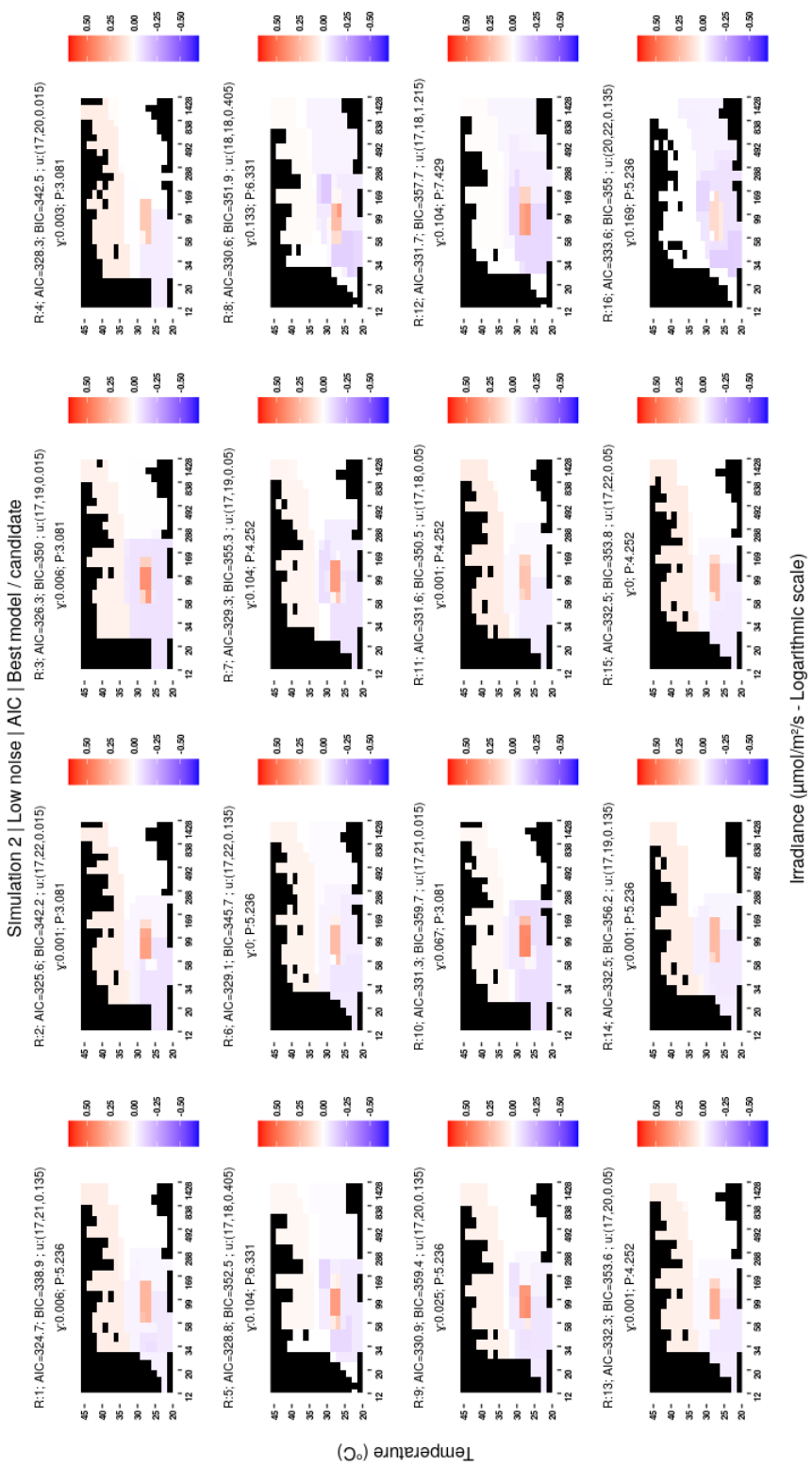


FIGURE 2.7.1 – The 16 best models.

Chapitre 3

Identification de plages d'influence conjointe de l'irradiance et de la température sur l'accumulation des anthocyanes

3.1 Introduction

La couleur d'un vin (yeux) est l'un des trois critères d'appréciation de sa qualité avec son odeur (nez) et son goût (bouche). Sa caractérisation a été codifiée [80] car elle est d'une importance cruciale pour des questions de qualité de produit, de label, de prix, etc. La couleur des vins rouges ou rosés dépend du processus de vinification et tient son origine dans la composition en anthocyanes (pigments naturels) de la baie, [20]. L'accumulation des anthocyanes a lieu au cours de la maturation (entre la véraison et les vendanges) dans la pellicule du raisin des variétés dites "noires" ou "rouges" et correspond à toute une famille de molécules voisines dérivées des chalcones [12]. Outre la couleur, les raisins connaissent à cette étape, de considérables modifications de leur composition telles que l'accumulation de sucres, la diminution de l'acidité etc. Ces modifications dépendent fortement des conditions sanitaires et environnementales lors du développement des baies. En particulier, les conditions climatiques très variables influencent la composition des baies de raisin [84]. Ces facteurs devraient prendre de plus en plus d'importance du fait du dérèglement climatique.

De précédents travaux relatifs à l'influence du climat sur la composition phénologique de la baie mettent en lien une hausse des températures et un retard d'accumulation en anthocyanes, [98], voire une diminution de leur concentration finale dans la baie, [21, 74, 76]. Certains de ces travaux renseignent même sur une sensibilité à la température. La valeur de 35°C apparaît comme un seuil au delà duquel les températures deviennent néfastes à l'accumulation d'anthocyanes dans les travaux de [73, 76, 107]. Il est cependant essentiel de souligner que les seuils peuvent varier d'un cépage à un autre, tout comme leur effets sur la

dégradation ou l'absence d'accumulation d'anthocyanes. Outre ces seuils, [94, 93] soulignent une différence de réactions physiologiques dans la baie face à un stress de température selon qu'il survient dans la journée ou dans la nuit. Il est donc essentiel de prendre en compte les différents moments de la journée auxquels surviennent ces stress. La biosynthèse des anthocyanes dépend certes de la température, mais également de nombreux autres facteurs biotiques ou abiotiques parmi lesquels l'irradiance [6, 83]. Une des difficultés est de découpler les effets de l'irradiance et de la température sur les baies de raisin. Cela s'avère délicat, puisque le rayonnement reçu est responsable d'une élévation de la température [11].

Afin de suivre l'évolution de la composition de la baie dans cette période sensible de la maturité, différentes méthodes pouvant être regroupées en deux groupes ont été développées : celles fournissant des mesures destructrices de la baie de raisin et celles fournissant des mesures non-destructrices. Une mesure non-destructrice a l'avantage de ne pas affecter le rendement tout en permettant au besoin la répétition de plusieurs mesures au fil du temps sur une même baie. En ce qui concerne les anthocyanes ou polyphénols en général, le Multiplex a été développé pour estimer l'évolution de leurs concentrations dans la pellicule du raisin au cours de la maturation [48, 19], leur mesure dans le raisin à la réception du chai ainsi que l'évaluation de la variabilité spatiale des caractéristiques du raisin. Ce Multiplex est équipé de quatre canaux d'excitation (UV, bleu, vert, rouge) et de trois canaux de détection (émission) capables de mesurer neuf signaux de fluorescence. Ces signaux sont utilisés dans le calcul des indicateurs de physiologie végétale comme la chlorophylle, les flavonols, les anthocyanes, etc. L'évolution de la teneur en anthocyanes peut être évaluée de façon non destructrice au cours du développement de la baie, en calculant un indicateur synthétique appelé l'Indice de Ferari (FI), [13]. Il est construit de sorte à être un bon indicateur de teneurs en anthocyanes.

L'objectif de ce chapitre est de mieux comprendre le processus d'accumulation des anthocyanes mesuré au cours du temps par l'Indice de Ferari. Peu de connaissances sont formalisées actuellement sur ce processus. Nous nous servirons de l'approche exploratoire SpiceFP développée au cours de cette thèse, avec comme facteurs explicatifs le micro-climat (la température et l'irradiance) observé à l'échelle de la grappe de raisins et mesurée quasiment en continu par des capteurs. La méthode proposée permet de trouver des plages limitantes ou favorables de température-irradiance, en supposant que tous les autres facteurs ne sont pas limitants (eau, azote etc.). Les données sont brièvement décrites en complément de leurs présentations dans les chapitres précédents. Pour pouvoir appliquer la méthode SpiceFP qui suppose un processus d'accumulation stable sur toute la période d'observation, une réflexion est développée dans la partie matériel et méthode pour extraire des intervalles de temps où l'on peut supposer que le phénomène d'accumulation ne varie pas. La méthode permet de dégager deux résultats principaux sur l'influence du couple température-irradiance. Ce chapitre se termine par une conclusion qui permet de restituer ces résultats dans un questionnement agronomique.

3.2 Matériel et Méthode

3.2.1 Description des données

Les données auxquelles nous nous intéressons dans ce chapitre ont été décrites dans la section "Présentation de l'expérimentation" du chapitre introductif puis dans la section "Data presentation" du chapitre 2. Nous compléterons toutefois cette présentation par une visualisation des variables fonctionnelles explicatives, observées aux mêmes instants t ainsi que celles de la courbe d'évolution des indices de Ferari pour quelques individus afin de présenter la diversité des données $(\mathcal{A}_i(t), \mathcal{B}_i(t),\ t \in T\,;\ FI_i(s),\ s \in S\,;\ i = 1, \ldots, n)$. Ces figures sont présentées dans les figures 3.5.2 et 3.5.1 en annexes de ce chapitre.

Dans le cadre de cette étude, $n = 79$. Parmi les 144 individus observés, 65 individus statistiques possédaient plus de 20% de données manquantes sur les observations des courbes de température. Ces individus n'ont pas été inclus dans l'étude. Les observations manquantes des individus disposant de moins de 20% de données manquantes ont été imputées en utilisant la méthode non-paramétrique MissForest [110, 108]. Les observations d'Indice de Ferari et d'irradiance ne présentaient pas de données manquantes. Les mesures d'Indice de Ferari ont été obtenues à l'aide du Multiplex Force A [9], permettant de déterminer de manière non-destructive les quantités en anthocyanes par des mesures de fluorescence. Les mesures ont été réalisées chaque semaine à 9 reprises sur les même grappes (durant 8 semaines) à partir de la véraison (changement de couleur du raisin) jusqu'à maturité en respectant le plan d'expérience mis en place. Nous disposons donc d'un suivi temporel de la teneur en anthocyanes.

3.2.2 Proposition de découpages nécessaires à l'utilisation de SpiceFP

Dans cette section, nous nous intéresserons à différents découpages des données dans le but d'utiliser l'approche proposée à des fins d'identification de conditions d'accumulation d'anthocyanes. Ces découpages sont nécessaires du fait de l'approche utilisée, SpiceFP, qui ne permet d'identifier des plages d'influence du microclimat que si cette influence est suffisamment stable tout au long de la période d'observation. Ce qui implique dans notre cas d'étude de s'intéresser à des données où la structure sous-jacente du processus d'accumulation d'anthocyanes ne change pas au fil du temps. Or, l'observation des courbes d'indice de Ferari (figures 3.5.2 et 3.5.1) montre qu'elles ont une allure sigmoïde et permet de définir trois phases :

— phase 1 : une première phase de plateau ou de faible croissance relative (semaine 1). Les variations au cours de cette phase sont quasi-nulles ou très faibles, ne permettant pas de distinguer aisément information et bruit comme on peut l'apercevoir sur la figure 3.2.1

— phase 2 : une phase de forte croissance relative (semaines 2 et 3) avant l'arrivée sur le plateau (semaine 4). Cette phase de croissance est généralement bien marquée pour tous les individus.

— phase 3 : une deuxième phase de faible croissance relative ou de plateau chez certains individus ou encore de décroissance chez d'autres individus (semaines 5 à 8). Les variations observées au cours de cette dernière étape témoignent bien de la différence des parcours au cours des semaines écoulées. Aussi, nous suspectons que d'autres phénomènes (différents de ceux observés durant l'accumulation) entrent en jeu.

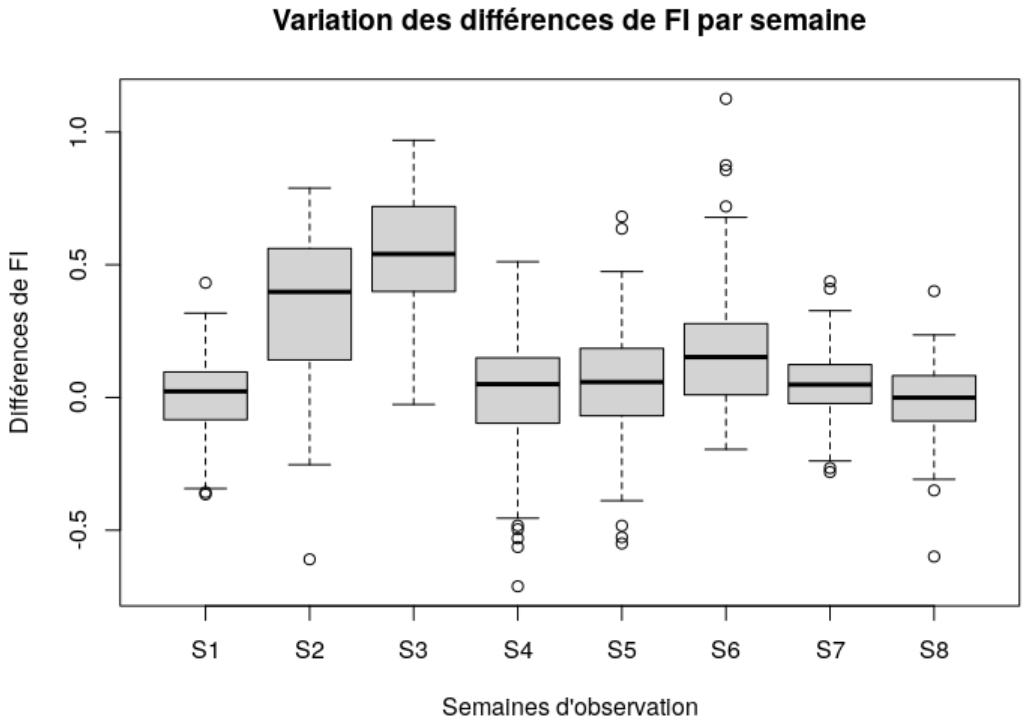


Figure 3.2.1 – Boîtes à moustaches des différences d'Indice de Ferari (différence entre la valeur de fin de semaine et la valeur de début de semaine) semaine par semaine

Tout au long des huit semaines, nous pouvons supposer ainsi une variation du processus sous-jacent. Ce qui nous a amené à une modélisation *phase par phase* ou encore *semaine par semaine*. Par prudence, nous avons opté pour la modélisation semaine par semaine et nous avons ciblé plus particulièrement la semaine 3 (coeur de la phase 2). Tous les individus n'étant pas au même niveau de maturité en début des semaines 3 et 4, nous avons conditionné les observations à modéliser en différents groupes selon le taux d'accumulation (IF) obtenant ainsi des *retardés* et des *avancés*. La figure 3.2.2-A illustre bien la séparation des données de la semaine 3 en fonction du niveau d'accumulation d'anthocyanes en début de cette semaine.

À partir d'un jeu de données découpé semaine par semaine et conditionné par les niveaux d'accumulation d'anthocyanes, nous nous sommes intéressés à la variabilité de l'irradiance

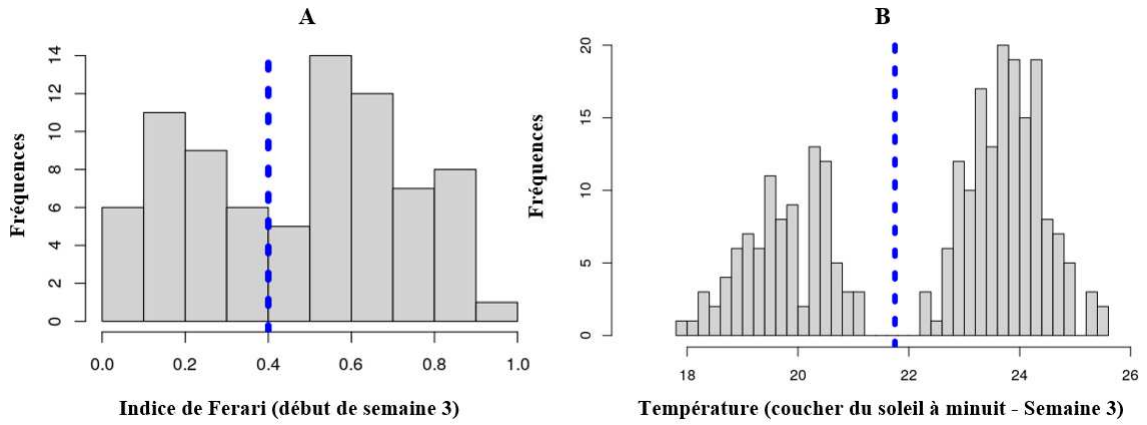


FIGURE 3.2.2 – A : valeurs d'Indice de Ferari en début de semaine 3 (la barre permet de définir les individus avancés et les individus retardés) - B : Température de la nuit d'après au cours de la semaine 3 (la barre permet de séparer les nuits chaudes et les nuits froides)

et de la température ainsi qu'à leurs effets à l'échelle d'une journée. Ces deux variables sont corrélées et toutes deux influencées par le parcours du soleil durant une journée. On observe généralement une élévation de la température et de l'irradiance avec le lever du soleil, un pic aux environs de midi (variable pour l'irradiance en fonction de l'orientation) et un déclin progressif de l'après-midi jusqu'au coucher du soleil. La nuit, l'irradiance est quasiment constante et considérée comme nulle tandis que de légères variations sont toujours observées en ce qui concerne la température.

D'une nuit à l'autre, la température peut varier d'où l'intérêt de prendre en compte l'influence de la température de la nuit comme souligné par de précédents travaux [93]. Nous avons séparé le jeu de données de la semaine 3 - retardés en deux jeux de données en fonction de la température de la "nuit d'après" (période allant du coucher du soleil à minuit ; soirée suivant la matinée d'observation). Ainsi, les journées d'observation à l'échelle d'une même semaine ont été divisées en deux groupes que nous désignerons par semaine 3 - retardés - nuits chaudes, et semaine 3 - retardés - nuits froides. Le graphique 3.2.2-B présente une visualisation des températures subies par les individus statistiques au cours de la nuit d'après. Ce graphique nous permet d'identifier un seuil de séparation de la nuit qui est représenté par la barre verticale bleue à 21,75°C. À partir de la température moyenne journalière de la nuit d'après et de ce seuil, nous avons identifié les journées avec les nuits chaudes et celles avec les nuits froides. Sur ces 4 jeux de données, semaine 3 - retardés, semaine 3 - avancés, semaine 3 - retardés - nuits chaudes, et semaine 3 - retardés - nuits froides, nous nous sommes intéressés à la période de la journée du lever du soleil à midi (suite à une étude préliminaire [24] sur le même jeu de données qui s'était intéressé à l'influence de la température sur l'accumulation des anthocyanes). Le schéma récapitulatif présenté dans la figure 3.2.3 résume le découpage des données et le plan d'analyse, afin d'étudier des phénomènes que nous supposons stables au fil du temps.

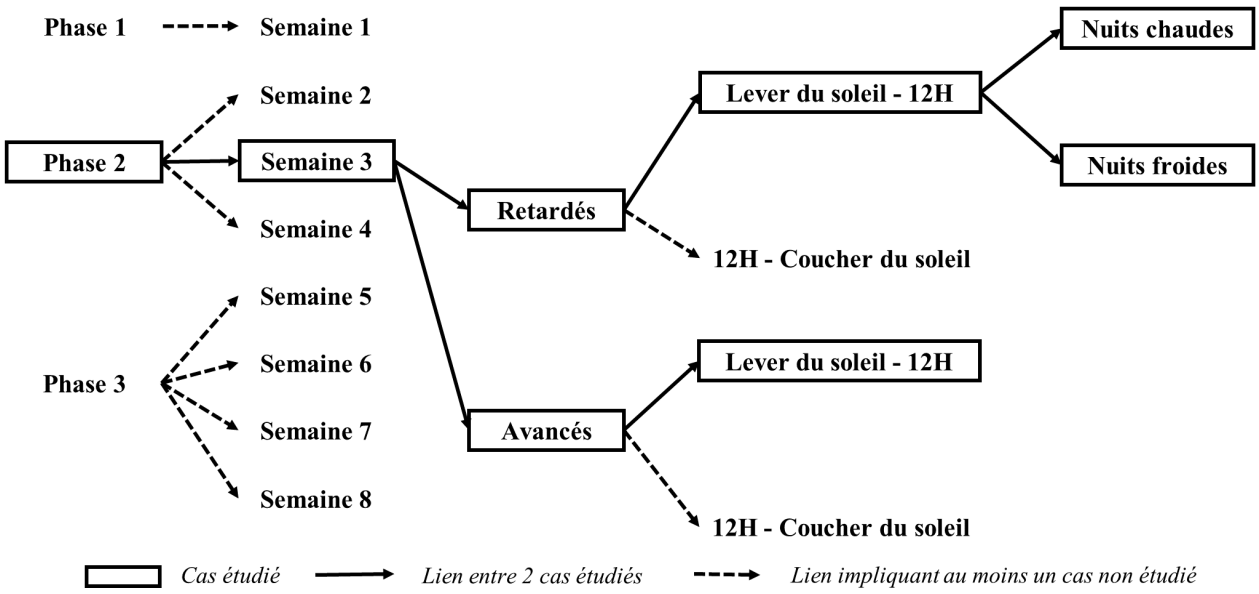


FIGURE 3.2.3 – Schéma récapitulatif du plan d'analyse des données

3.2.3 Utilisation de l'approche SpiceFP

Dans le cadre de la mise en oeuvre de l'approche SpiceFP sur les matinées des groupes semaine 3 - retardés, semaine 3 - avancés, semaine 3 - retardés - nuits chaudes, et semaine 3 - retardés - nuits froides, les classes de température ont été crées suivant une échelle linéaire et celle d'irradiance suivant une échelle logarithmique grâce à la fonction **logbreaks** du package (**SpiceFP**). L'utilisation de l'approche proposée a impliqué la construction et l'évaluation 490 matrices candidates. Ces dernières sont construites à partir de la combinaison des paramètres appartenant aux ensembles $\mathcal{U}_{\mathcal{A}}$, $\mathcal{U}_{\mathcal{B}}$, et $\mathcal{V}_{\mathcal{B}}$ présentés dans la liste ci-dessous. Pour chaque matrice candidate, 10 valeurs de ratio de paramètres de pénalisation (parcimonie/fusion) fournis par Γ ont été utilisées. Et pour chaque régression impliquant une matrice candidate et un ratio $\gamma \in \Gamma$, nous avons retenus 100 modèles, chacun associé à une valeur de paramètre λ fournie par l'approche. C'est ainsi que parmi 490 000 modèles, nous avons choisi le meilleur sélectionné par l'AIC et le meilleur sélectionnée par le BIC. Pour nous assurer de la stabilité du meilleur modèle suivant un critère donné, nous nous sommes intéressés, comme suggéré dans [37], à plusieurs meilleurs modèles (figures présentées en annexes de ce chapitre) dans le cadre de certaines analyses. Explicitement, les valeurs des paramètres sont :

— $\mathcal{U}_{\mathcal{A}} = \{9, 11, 13, 15, 17, 19, 21\}$
— $\mathcal{U}_{\mathcal{B}} = \{9, 12, 15, 18, 21, 24, 27\}$,
— $\mathcal{V}_{\mathcal{B}} = \{0.0025, 0.0039, 0.0060, 0.0094, 0.0147, 0.0229, 0.0357, 0.0556, 0.0868, 0.1353\}$,
— $\Gamma = \{1/200, 1/100, 1/50, 1/25, 1/12.5, 1/6.25, 1, 6.25, 12.5, 25\}$,
— $n_\lambda = 100$.

Outre ces analyses, les extensions de l'approche (SpiceFP 3D à 2 itérations, extensions présentées dans le chapitre 4) ont été utilisées afin d'examiner de manière conjointe au sein d'une même analyse, l'influence de la température et de l'irradiance en matinée, ainsi que l'influence de la température de la nuit sur l'accumulation d'anthocyanes pour le groupe semaine 3 - retardés. À cet effet, les variables explicatives dans les régressions sont les modalités obtenues lors de la création d'un tableau de contingence en 3 dimensions : les deux premières dimensions sont irradiance et température de la matinée, la troisième correspond à la température de la nuit. Pour obtenir une évaluation de la qualité d'ajustement de nos estimations, nous nous sommes servis de la pente de la régression $\hat{Y} \sim Y$.

3.3 Résultats

3.3.1 Sans conditionnement à la température de la nuit

L'approche SpiceFP a été utilisée indépendemment pour analyser les données des deux groupes : avancés et retardés. Selon les résultats fournis par l'approche SpiceFP et présentés dans la figure 3.3.1 pour les avancés, on observe une mauvaise qualité d'ajustement du modèle (pente entre prédictions et observations de la variable réponse de l'ordre de 0,15 pour la sélection du modèle par le BIC et 0.23 pour la sélection du modèle par l'AIC). Pour ce qui concerne donc les individus avancés de la semaine 3, nous ne nous fierons pas aux zones d'influence fournies par SpiceFP.

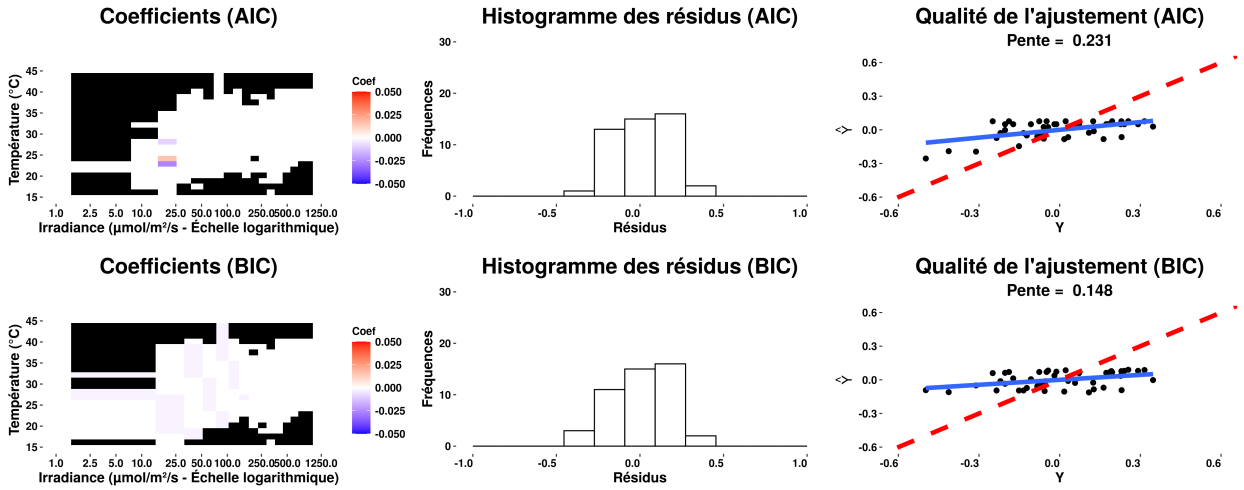


Figure 3.3.1 – Résultats de l'analyse SpiceFP pour les observations sans conditionnement à la température de la nuit - Groupe des individus avancés - Sélection du modèle selon l'AIC (graphes du haut) ou le BIC (graphes du bas)

En ce qui concerne les individus retardés de la semaine 3 par contre, la qualité d'ajustement est bien meilleure (pente entre prédictions et observations de la variable réponse de

l'ordre de 0,46 pour la sélection du modèle par le BIC et l'AIC) et on obtient une sélection similaire des plages d'influence ainsi que de leurs effets quel que soit le critère d'information utilisé. On observe sur la figure 3.3.2 un effet négatif des irradiances faibles (<100 $\mu mol.m^{-2}.s^{-1}$) et températures élevées (entre 25°C et 35°C) le matin (entre le lever du soleil et midi) sur l'accumulation des anthocyanes. La matérialisation de la zone d'influence permet bien de distinguer une séparation du domaine des variables en deux, suivant une ligne oblique.

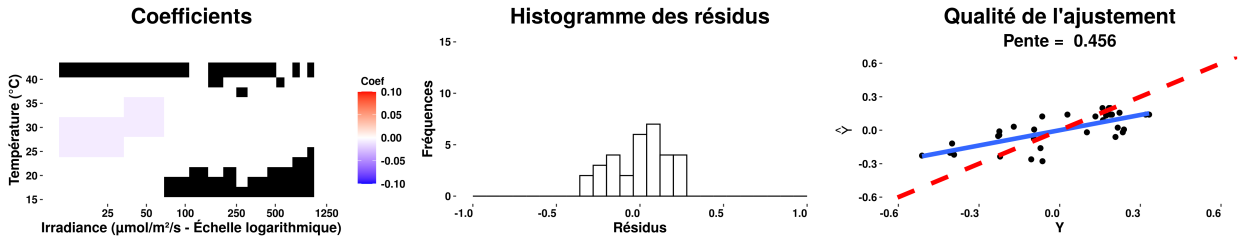


FIGURE 3.3.2 – Résultats de l'analyse SpiceFP pour les observations sans conditionnement à la température de la nuit - Groupe des individus retardés - Sélection du même modèle selon l'AIC ou le BIC

3.3.2 Conditionnement en fonction de la température de la nuit

Conditionnement en fonction de la température de la nuit, deux analyses séparées par SpiceFP 2D

Dans ce premier conditionnement par rapport à la température de la nuit, nous nous intéressons séparément à l'influence des nuits chaudes et nuits froides. Pour chacune de ces nuits et chacun des critères d'information utilisés, les pentes obtenues sont supérieures à 0,35 pour les meilleurs modèles choisis par nos deux critères d'informations. En ce qui concerne les nuits chaudes, la figure 3.3.3 permet de bien observer une ligne de séparation oblique entre des coefficients négatifs et des coefficients non significatifs. Suivant cette ligne, les températures élevées (entre 22.5°C et 40°C) à faible irradiance (<100 $\mu mol.m^{-2}.s^{-1}$) ont ici un impact négatif sur l'accumulation d'anthocyanes. Une partie du domaine des valeurs ne contient pas d'observations (matrice des observations creuse, présentée en 3.3.4), notamment celles des fortes irradiances et températures, ce qui explique l'arrêt de la ligne de séparation.

En ce qui concerne les nuits froides, les résultats présentés dans la figure 3.3.5 permettent d'observer les mêmes effets que les nuits chaudes mais en beaucoup moins marqués. Il faut noter que la matrice des observations (figure 3.3.4) est encore plus creuse et restreinte que celle des nuits chaudes.

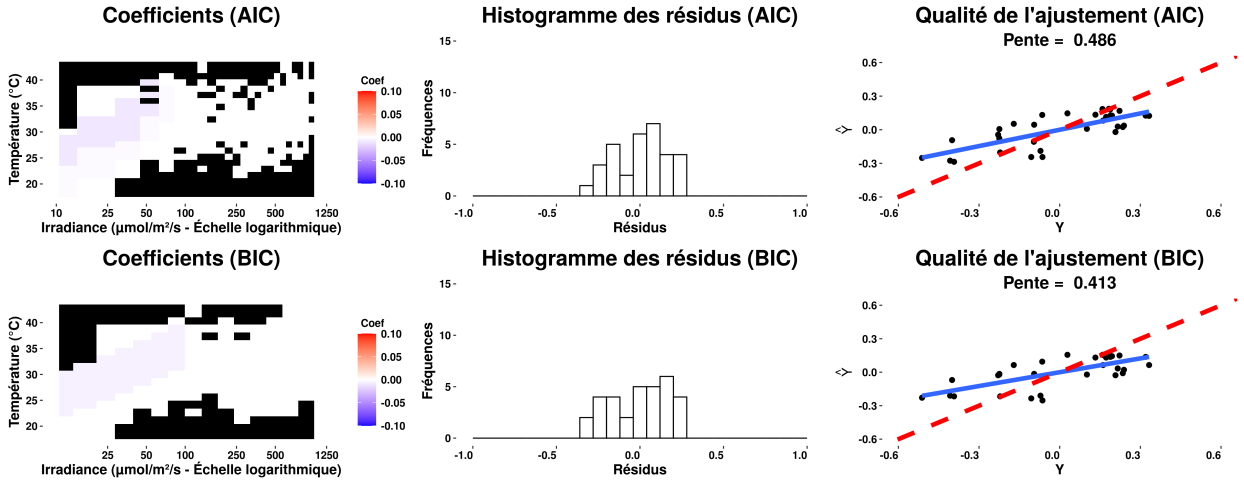


FIGURE 3.3.3 – Résultats de l'analyse SpiceFP pour les observations avec un conditionnement en fonction de la température de la nuit - Groupe des individus retardés et soirées chaudes - Sélection du modèle selon l'AIC (graphes du haut) ou le BIC (graphes du bas).

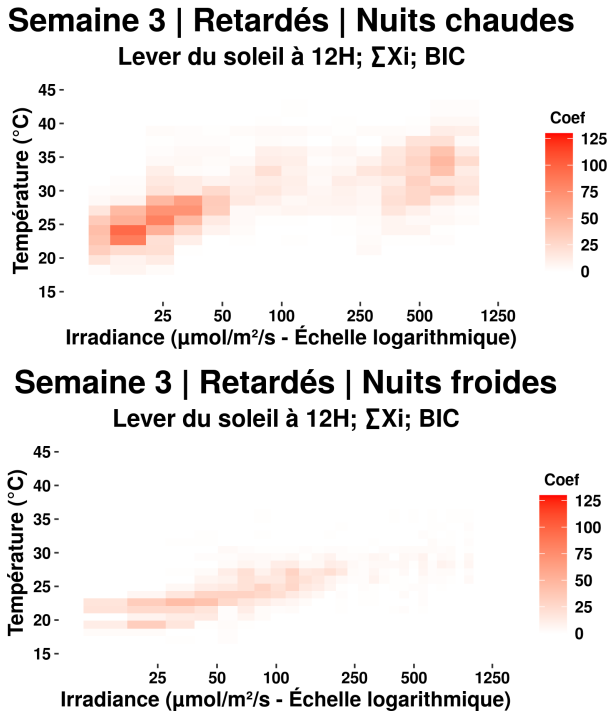


FIGURE 3.3.4 – Représentation de la matrice des observations (Somme des X_i pour un vecteur de paramètre $u \in U_{\mathcal{A}}.U_{\mathcal{B}}.V_{\mathcal{A}}$)

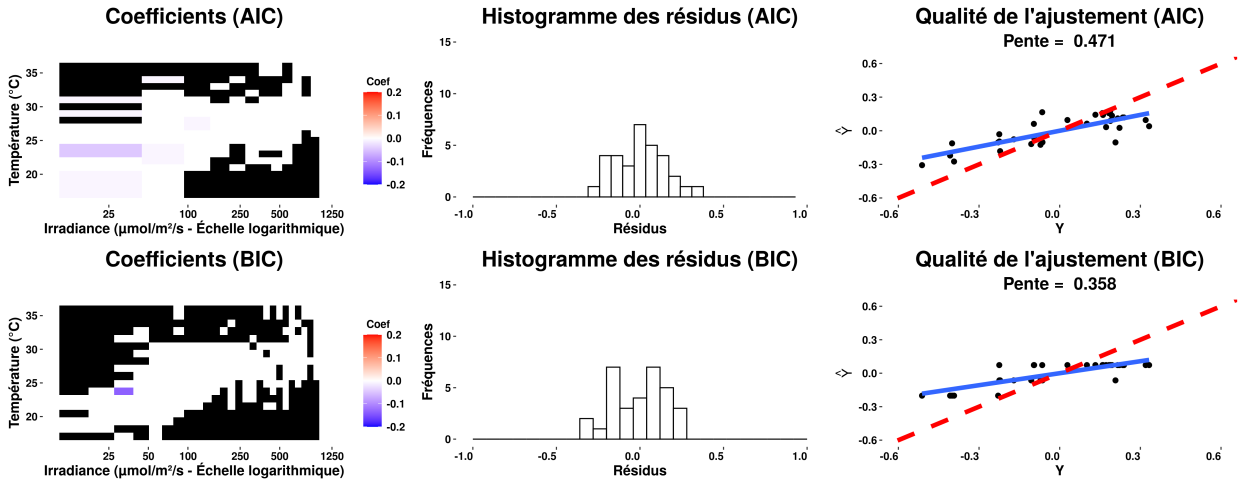


FIGURE 3.3.5 – Résultats de l'analyse SpiceFP pour les observations avec un conditionnement en fonction de la température de la nuit - Groupe des individus retardés et soirées froides - Sélection du modèle selon l'AIC (graphes du haut) ou le BIC (graphes du bas)

Prise en compte du conditionnement directement dans le même modèle par une analyse SpiceFP 3D

Les résultats fournis par l'analyse SpiceFP 3D et présentés dans la figure 3.3.6 présentent une qualité d'ajustement de 0,578 pour le meilleur modèle choisi par les critères AIC et BIC (un même modèle a été sélectionné). Cette pente est meilleure que toutes celles obtenues pour les groupes semaine 3 - retardés, semaine 3 - retardés - nuits chaudes, et semaine 3 - retardés - nuits froides. L'idée d'intégrer la prise en compte de la catégorisation de la nuit au sein d'un même modèle permet bien de mieux expliquer l'accumulation des anthocyanes en matinée de la semaine 3. Il faut noter toutefois que ce modèle retient moins de paramètres même s'il semble indiquer la même tendance que celle obtenue via deux analyses séparées. Ce qui apporte une cohérence entre les résultats 2D et 3D de SpiceFP.

Synthèse du message agronomique

En résumé, nous pouvons retenir de ces diverses analyses que lors de la phase de forte croissance des anthocyanes, les températures élevées combinées aux faibles irradiances en matinée ont un impact négatif qui va retarder l'accumulation d'anthocyanes. Cet impact négatif s'observe généralement pour des niveaux de température qui augmentent avec le niveau d'irradiance, dessinant une droite de séparation oblique dans la matrice des intervalles température-irradiance sur la figure 3.3.3. Cette séparation varie en fonction de la température de la nuit. Ces résultats nous montrent ainsi l'importance des combinaisons irradiance - température et le besoin d'études complémentaires pour analyser le découplage possible entre température et irradiance.

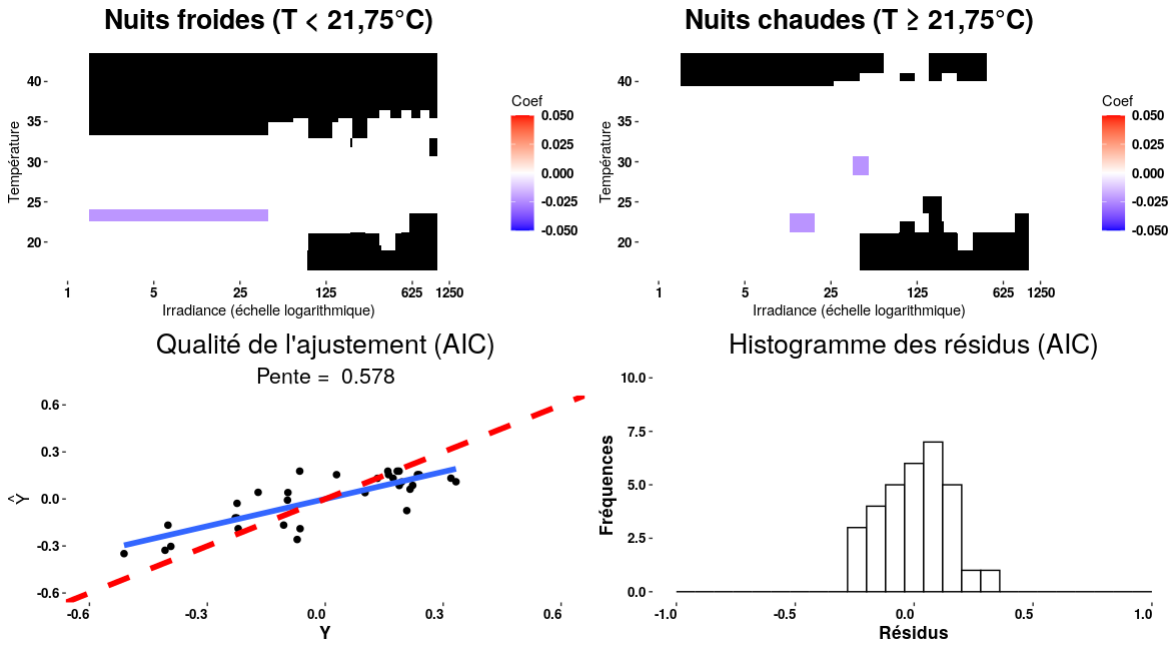


FIGURE 3.3.6 – Résultats de l'analyse SpiceFP pour les observations avec un conditionnement en fonction de la température de la nuit pris directement en compte dans une analyse SpiceFP 3D - Groupe des individus retardés - Sélection du même modèle selon l'AIC ou le BIC

3.3.3 Modèle basé sur la droite de séparation

On considère la période de la journée, comprise entre le lever du soleil et 12h / semaine 3. Les prédicteurs fonctionnels sont la température $\mathcal{T}_i$ et l'irradiance $\mathcal{I}_i$ pour chaque individu i et sont mesurées à des dates $t_k, k = 1, ..T$.

À partir des résultats de l'analyse SpiceFP pour les groupes d'individus retardés ou avancés, au dessus d'une droite dans le plan $(T, \log(I))$ (figures 3.3.2 et 3.3.1), on remarque des coefficients β négatifs pour les meilleures solutions au sens des critères AIC ou BIC. Dans certains cas mais très limités, quelques coefficients positifs sont présents en dessous de cette droite. Ces deux remarques nous inspirent un modèle où les coefficients sont de signes différents de part et d'autre d'une droite de pente θ : $T = T_0 + \theta \log \frac{I}{I_0}$, potentiellement négatif au dessus de cette droite et positif en dessous. La variation de l'indice de Ferari est donnée par :

$$\Delta F_i = \int_T \int_I g_i(T, I)\ \beta(T, I)\ dT\ dI \tag{3.3.1}$$

avec $g_i(T, I) = \sum_k \mathbb{1}_{|\mathcal{T}_i(t_k)-T|\leq\Delta_T,\ |\mathcal{I}_i(t_k))-I|\leq\Delta_I}$. $g(T, I)$ un indicateur relatif au temps passé autour de la température T et de l'irradiance I.

$$\beta(T,I) = \begin{cases} \beta_- & \text{if } T > T_0 + \theta \log \dfrac{I}{I_0} \ \&\ I_0 < I < I_1 \\ \beta_+ & \text{if } (T < T_0 + \theta \log \dfrac{I}{I_0} \ \&\ I_0 < I < I_1) \text{ or } I > I_1 \\ 0 & \text{if } I < I_0 \end{cases}$$

Les paramètres sont : $\beta_- < 0 < \beta_+, T_0, \theta > 0, I_0 \leq I_1$. De manière équivalente à l'équation (3.3.1) :

$$\begin{aligned} \Delta F_i =& \beta_- \sum_k \mathbb{1}_{\mathcal{T}_i(t_k) > T_0 + \theta \log \frac{\mathcal{I}_i(t_k)}{I_0} \ \&\ I_0 < \mathcal{I}_i(t_k) < I_1} \\ &+ \beta_+ \sum_k \mathbb{1}_{\mathcal{I}_i(t_k) < I_0 \text{ or } (\mathcal{T}_i(t_k) < T_0 + \theta \log \frac{\mathcal{I}_i(t_k)}{I_0} \ \&\ \mathcal{I}_i(t_k) > I_0) \text{ or } \mathcal{I}_i(t_k) > I_1} \end{aligned}$$

D'où le modèle linéaire dans les paramètres β_- et β_+ d'une part et non linéaire dans les paramètres T_0, θ, I_0, I_1 d'autre part :

$$\Delta F_i = \mu + \beta_- N_-^i(T_0, \theta, I_0, I_1) + \beta_+ N_+^i(T_0, \theta, I_0, I_1), \tag{3.3.2}$$

où :

- $N_-^i(T_0, \theta, I_0, I_1) = \text{Card}\{k | \mathcal{T}_i(t_k) > T_0 + \theta \log \dfrac{\mathcal{I}_i(t_k)}{I_0} \ \&\ I_0 < \mathcal{I}_i(t_k) < I_1\}$,
- $N_+^i(T_0, \theta, I_0, I_1) = \text{Card}\{k | (\mathcal{T}_i(t_k) < T_0 + \theta \log \dfrac{\mathcal{I}_i(t_k)}{I_0} \ \&\ \mathcal{I}_i(t_k) > I_0) \text{ or } \mathcal{I}_i(t_k) > I_1\}$,
- et où $\mathcal{T}_i(t_k)$ est la température de l'individu i au temps t_k, $\mathcal{I}_i(t_k)$ est l'irradiance de l'individu i au temps t_k et $\text{Card}\{k \mid \text{condition}\}$ est le cardinal de l'ensemble des pas de temps t_k où la condition est vérifiée.

Les paramètres de ce modèle sont : $\beta_- < 0 < \beta_+, T_0, \theta > 0, I_0 \leq I_1$. Pour la calibration de ces paramètres nous utilisons le critère des moindres carrés :

$$\min_{T_0, \theta, I_0 < I_1, \beta_-, \beta_+} \sum_i (\Delta F_i - \beta_- N_-^i(T_0, \theta, I_0, I_1) - \beta_+ N_+^i(T_0, \theta, I_0, I_1))^2$$

où $N_-^i(T_0, \theta, I_0, I_1), N_+^i(T_0, \theta, I_0, I_1)$ et ΔF_i ont été centrés. Une illustration de la position de ces paramètres sur le domaine étudié (température-irradiance) est présentée dans la figure 3.3.7. La lecture des résultats de l'analyse SpiceFP nous incite à prendre comme paramètres initiaux : $T_0 = 20\,°\text{C}, \theta = 6.5\,°\text{C}, I_0 = 9.97\,\mu\text{mol}\,\text{m}^{-2}\,\text{s}^{-1}, I_1 = 99.48\,\mu\text{mol}\,\text{m}^{-2}\,\text{s}^{-1}$. Nous utiliserons ces paramètres initiaux ainsi que divers autres pour nous assurer d'obtenir le minimum global. La routine utilisée pour l'optimisation est la fonction *optim* de R.

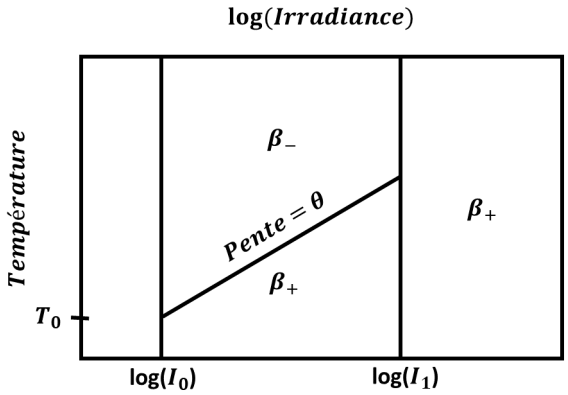


FIGURE 3.3.7 – Illustration des paramètres du modèle basé sur la droite de séparation oblique.

Pour la calibration, on peut récrire le problème d'optimisation en distinguant les paramètres dépendant linéairement :

$$\min_{T_0,\theta,I_0<I_1} J(T_0,\theta,I_0,I_1)$$

où : $J(T_0,\theta,I_0,I_1) = \min_{\beta_-,\beta_+} \sum_i (\Delta F_i - \beta_- N_-^i(T_0,\theta,I_0,I_1) - \beta_+ N_+^i(T_0,\theta,I_0,I_1))^2$.

$J(T_0,\theta,I_0,I_1)$ est obtenu via une régression linéaire pour chaque vecteur de paramètres (T_0,θ,I_0,I_1).

Pour les retardés (32 individus), les paramètres optimaux sont : $\beta_- = -.000429, \beta_+ = .00504, T_0 = 24.42\,^\circ\mathrm{C}, \theta = 7.768\,^\circ\mathrm{C}, I_0 = 42.65\,\mu\mathrm{mol\,m^{-2}\,s^{-1}}, I_1 = +\infty, Crit/N = .02596$.

Pour les avancés (45 individus), : $\beta_- = -.06019, \beta_+ = .003751, T_0 = 26.52\,^\circ\mathrm{C}, \theta = 25.68\,^\circ\mathrm{C}, I_0 = 139.5\,\mu\mathrm{mol\,m^{-2}\,s^{-1}}, I_1 = +\infty, Crit/N = .02727$

On peux également subdiviser les avancés en intermédiaires et très avancés. Pour les intermédiaires (33 individus) : $\beta_- = -.01843, \beta_+ = .00424, T_0 = 25.58\,^\circ\mathrm{C}, \theta = 8.557\,^\circ\mathrm{C}, I_0 = 141.7\,\mu\mathrm{mol\,m^{-2}\,s^{-1}}, I_1 = +\infty, Crit/N = .01868$

Pour les très avancés (14 individus) : $\beta_- = -.00865, \beta_+ = -.01587, T_0 = 22.19\,^\circ\mathrm{C}, \theta = 1.093\,^\circ\mathrm{C}, I_0 = 8.390\,\mu\mathrm{mol\,m^{-2}\,s^{-1}}, I_1 = +\infty, Crit/N = .01841$

Dans chaque cas, I_1 est rejeté à l'infini, donc la condition associée n'est pas opérationnelle. La valeur négative de β au dessus de la droite de séparation est confirmée. La nouveauté est la mise en évidence d'une valeur positive en dessous de cette même droite. Seul dans le cas très avancé, cette valeur positive est remise en cause mais avec peu d'individus.

3.4 Discussion

À partir d'une partie des données du projet Innovine, l'utilisation de l'approche SpiceFP complétée par un modèle de séparation linéaire a permis d'identifier un impact négatif des combinaisons matinales de faible irradiance (inférieure à environ $100\,\mu mol\,m^{-2}\,s^{-1}$ ou $45\,\mu mol\,m^{-2}\,s^{-1}$ selon l'état avancé-retardé) et température élevée (supérieure à environ 25 °C). Ces combinaisons sont corrélées à un retard d'accumulation d'anthocyanes en 3e semaine, pendant la phase d'accumulation rapide des anthocyanes. Plus précisément, pour les individus dont la valeur de l'indice de Ferari est considérée comme retardée en début de 3e semaine (FI<0.4), on observe une séparation le long d'une ligne entre des conditions plus favorables et moins favorables. Cette séparation est identifiée par SpiceFP (figure 3.3.3) nettement dans le cas où les nuits sont relativement chaudes (entre 22 °C et 26 °C) et confirmée par l'estimation du modèle de séparation linéaire. Nos résultats concordent pour des conditions où l'on n'est pas dans des situations qui favorisent la dégradation en même temps que la photosynthèse est limitée. De plus, nos résultats sont cohérents avec la littérature [28] qui suggère qu'en environnement naturel, les températures élevées sur une longue période ont un impact négatif sur la voie de biosynthèse ([107, 26]) et qu'en revanche les températures basses associées à une forte irradiance favorisent l'accumulation d'anthocyanes [21].

L'utilisation de SpiceFP va plus loin que les résultats classiques car l'approche tient compte de la complexité des relations entre irradiance et température sur l'accumulation d'anthocyanes. Raisonner sur des seuils de température et/ou d'irradiance n'est pas assez précis. De même, l'utilisation de modèles simples qui considèrent les effets découplés des facteurs température et irradiance ([107]) ne permettent pas de rendre compte de cette complexité car il néglige les effets combinés de la température en fonction de l'irradiance et inversement. De plus, nous avons constaté que selon l'état d'avancement de la maturité (mesurée par l'IF), les modèles obtenus montraient des différences et qu'il n'y avait pas un modèle unique mais une évolution au cours de la maturité de la sensibilité face aux conditions climatiques.

Les résultats obtenus sont cohérents et stables entre une approche 2D dans laquelle la distinction entre nuits chaudes ou moins chaudes est fixée par l'utilisateur et l'approche 3D qui optimise ce découpage en plus des partitions de température-irradiance. Dans le cadre d'Innovine, les températures nocturnes sont obtenues en conditions naturelles et ne vont pas dans les gammes de valeurs généralement citées dans la littérature, citons notamment [107] et [94] (nuit chaude >35 °C et nuit froide <15 °C). Cette limitation explique l'impact très relatif du conditionnement selon la température nocturne. De plus, le protocole expérimental (open top, orientation des baies et ombrage/couverture foliaire) a entraîné un découplage modéré et naturel des facteurs température et irradiance, contrairement aux expérimentations de [28] . Aussi, dans le contexte des données Innovine, nous n'avons pas observé une importance plus grande du facteur température (qui inhiberait/arrêterait la voie de biosynthèse au delà d'un certain seuil, souvent posé autour de 35 °C) sur le facteur irradiance pour l'accumulation d'anthocyanes.

Les résultats sont conditionnés aux conditions expérimentales qui ont généré les valeurs observées pour la température et l'irradiance. Quand on étudie la répartition de ces valeurs (figure 3.3.4), on constate beaucoup de zones non observées et deux zones/pics où se concentrent les valeurs observées (design déséquilibré). La conséquence de cette distribution des valeurs est une matrice de design/expérimentation X creuse. Intuitivement, on comprend bien qu'il sera plus facile statistiquement de détecter une influence dans les zones denses plutôt que dans les zones rarement observées. Il est possible que certains effets soient masqués ou non détectables dans les plages combinées de valeurs de température et irradiance peu observées.

Afin de ne pas déséquilibrer encore plus notre design et mélanger les effets, nous n'avons considéré que l'impact des facteurs climatiques du matin. En effet, la température et l'irradiance de nuit ne changent pas ou très peu, entraînant une sur-représentation (un gros pic d'observation) de cette plage de faibles valeurs. La température de la nuit étant tout de même considérée comme importante, nous l'avons prise en compte dans une approche 2D conditionnelle à la température de la nuit ou dans une approche 3D. Les conditions climatiques de l'après-midi n'ont pas été prises en compte dans notre analyse. L'existence dans l'après-midi, d'un décalage dans le temps des valeurs température-irradiance des baies selon leur orientation complexifiait l'interprétation des résultats. Or, une première étude sur le jeu de données Innovine qui incluait uniquement la température, [24], a révélé une influence potentiellement positive des heures au petit matin (les heures entre 06h et 10h). C'est peut-être une période où les températures ne sont pas trop élevées et favoriseraient la production d'anthocyanes. En revanche, il y aurait potentiellement une influence négative durant les heures de l'après-midi et début de soirée (entre 15h et 20h) [24]. Le déterminisme de l'accumulation des anthocyanes, à savoir une synthèse, dépendrait de la photosynthèse réalisée dans la journée et d'une dégradation qui serait plus forte la nuit à forte température. Aussi, dans l'approche combinant température et irradiance, nous avons préféré distinguer l'impact des conditions matinales et ensuite vérifié si cet impact variait selon la température de début de soirée en utilisant le conditionnement par rapport à la température de début de soirée (et nocturne).

Un résultat intéressant fourni par l'approche SpiceFP est la différence de sensibilité entre les individus dit retardés et avancés. Le taux de sucre dans la baie est un indicateur de maturité utilisé fréquemment. N'ayant pas accès à cette mesure (destructrice), les états avancés ou retardés ont été définis à partir des valeurs observées d'indice de Ferari. Même si cette mesure de l'état d'avancement n'est pas la plus fiable/fréquente, elle donne une indication. Une hypothèse expliquant la différence de sensibilité entre les deux états serait que la couleur de la baie pourrait être due à une accélération de son cycle physiologique incluant l'accumulation des anthocyanes indépendante des facteurs climatiques. Pour les individus avancées, on aurait peut être un cycle physiologique plus rapide et ils auraient atteint un état moins sensible aux facteurs climatiques. La diminution de la sensibilité au cours de la maturité pourrait provenir d'un amortissement de l'accumulation déterminé par la "fin" d'un processus physiologique (similaire à la fin d'une croissance de feuille qui devient

insensible à la T°C). Une autre hypothèse serait une plus grande difficulté à détecter les effets des facteurs climatiques pour les individus dit avancés. En effet, l'erreur de mesure sur l'indice de Ferari pourrait être assez forte car ce n'est pas un suivi strictement individuel, dans le sens où ce n'est pas forcément le même groupe de baies qui est suivie d'une mesure à l'autre (c'est un groupe de baies du même pied dans la même condition expérimentale). Aussi, pour les individus dit avancés, les variations dans l'indice de Ferari sont plus faibles (dès qu'ils ont dépassé le point d'inflexion dans la courbe de l'indice de Ferari en fonction du temps) que pour les individus dit retardés (qui sont encore dans la phase rapide). Les variations dans les valeurs d'indice sont plus faibles entre individus avancés alors que l'erreur de mesure peut être forte, ce qui pourrait expliquer la mauvaise qualité des ajustements fournis par SpiceFP pour les avancés et le peu de résultats significatifs.

3.5 Annexes

3.5.1 Visualisation des courbes de quelques individus statistiques

Nous présentons dans cette annexe deux figures (3.5.1, 3.5.2) relatives aux observations d'Indice de Ferari, de Température et d'Irradiance pour quelques individus statistiques. Les individus sont organisés en colonne sur ces deux figures et les variables en ligne. Les individus sous le dispositif OpenTop sont présentés en figure 3.5.2 et ceux sans ce dispositif sont présentés en figure 3.5.1. Des observations à l'ouest et à l'est sont présentes sur ces figures.

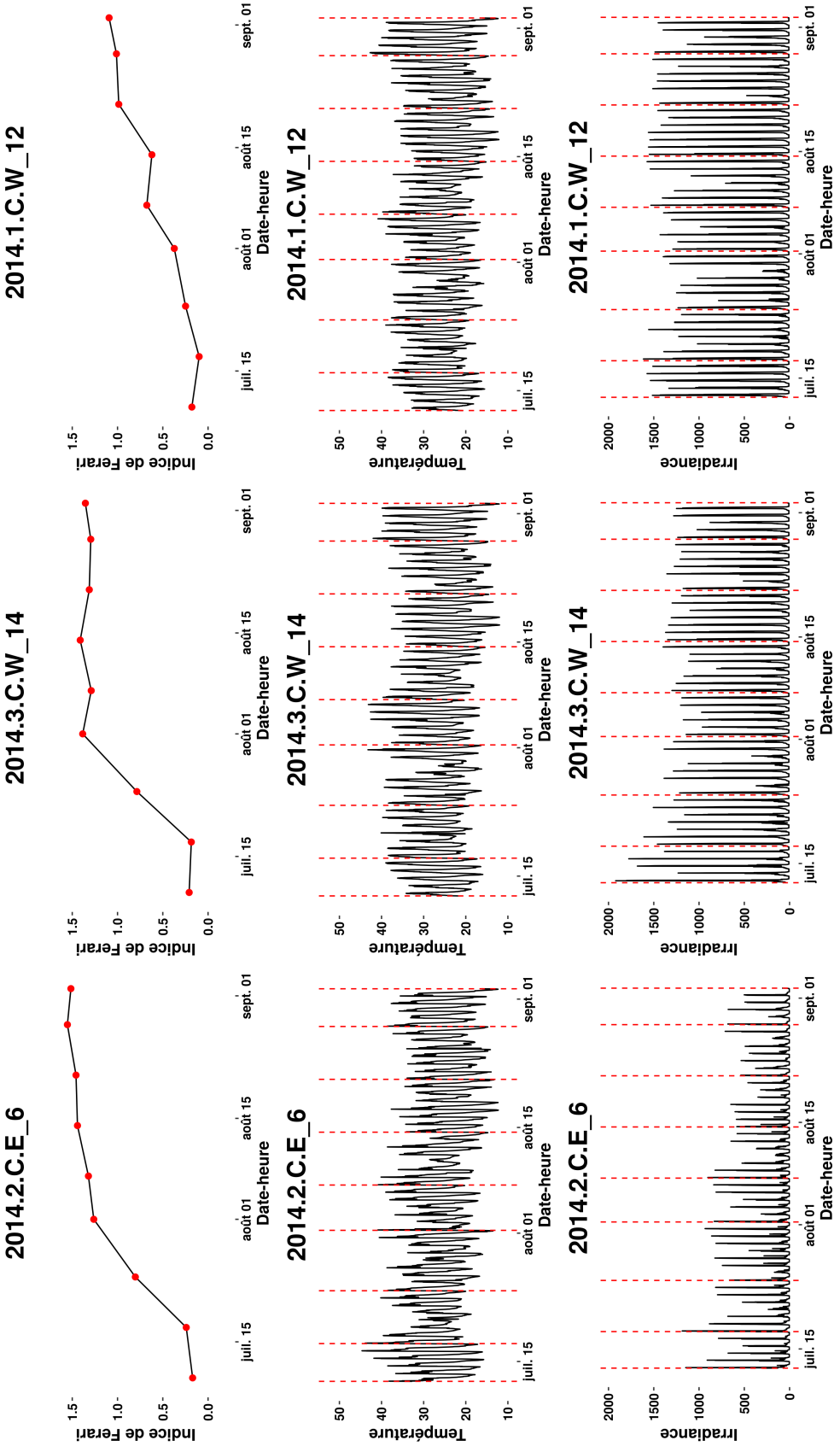


FIGURE 3.5.1 – Courbes d'Indice de Ferari, de température et d'irradiance de quelques individus statistiques sans dispositif OpenTop

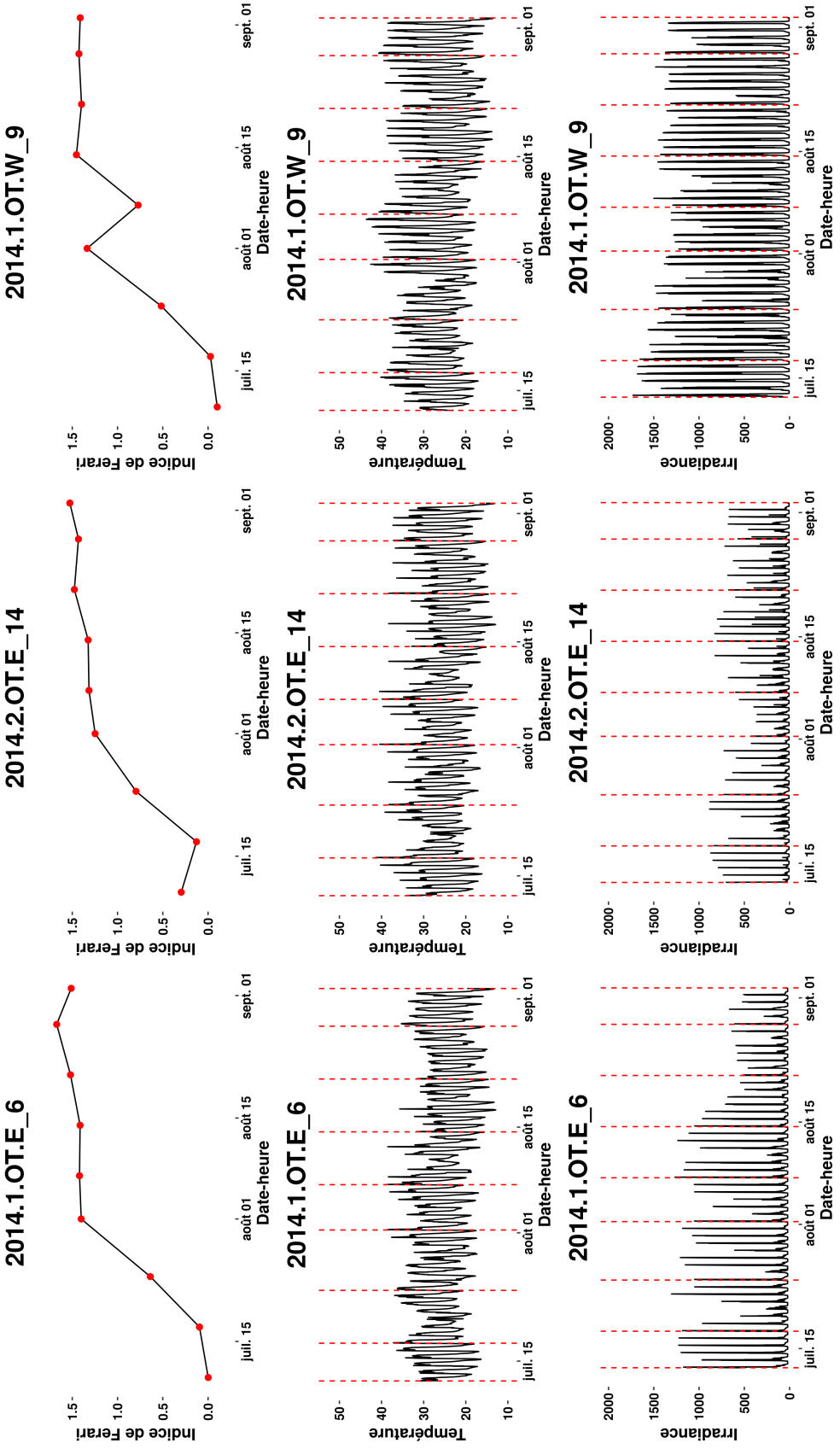


FIGURE 3.5.2 – Courbes d'Indice de Ferari, de température et d'irradiance de quelques individus statistiques sous le dispositif OpenTop

3.5.2 Visualisation des meilleurs modèles pour les observations Semaine 3 - Retardés - Nuits chaudes

En guise d'illustration, nous présentons les 25 meilleurs modèles suivant les critères AIC (figure 3.5.3) et BIC (figure 3.5.4) issus de l'analyse SpiceFP pour les observations Semaine 3 - Retardés - Nuits chaudes. On y voit une légère variabilité des plages d'influences (pour une bonne visibilité des plages d'influence, différentes échelles sont utilisées pour la légende). Malgré dette légère variabilité, on remarque que :

— toutes les plages sont relatives à une même aire dans le graphique et que le signe des coefficients dans cette aire reste le même (négatif)

— la ligne de séparation oblique est identifiable sur presque tous les graphiques.

Aussi, on note sur le dernier graphique (modèle n°25) de la figure 3.5.3, l'apparition de coefficients positifs dans une nouvelle aire du graphique. Bien que ce modèle présente une meilleure pente que plusieurs autres modèles qui le précèdent, il occupe la 25^{eme} place. Cela illustre bien la forte pénalisation de l'entrée de nouveaux coefficients par les critères d'information utilisés comme discuté dans le chapitre 2.

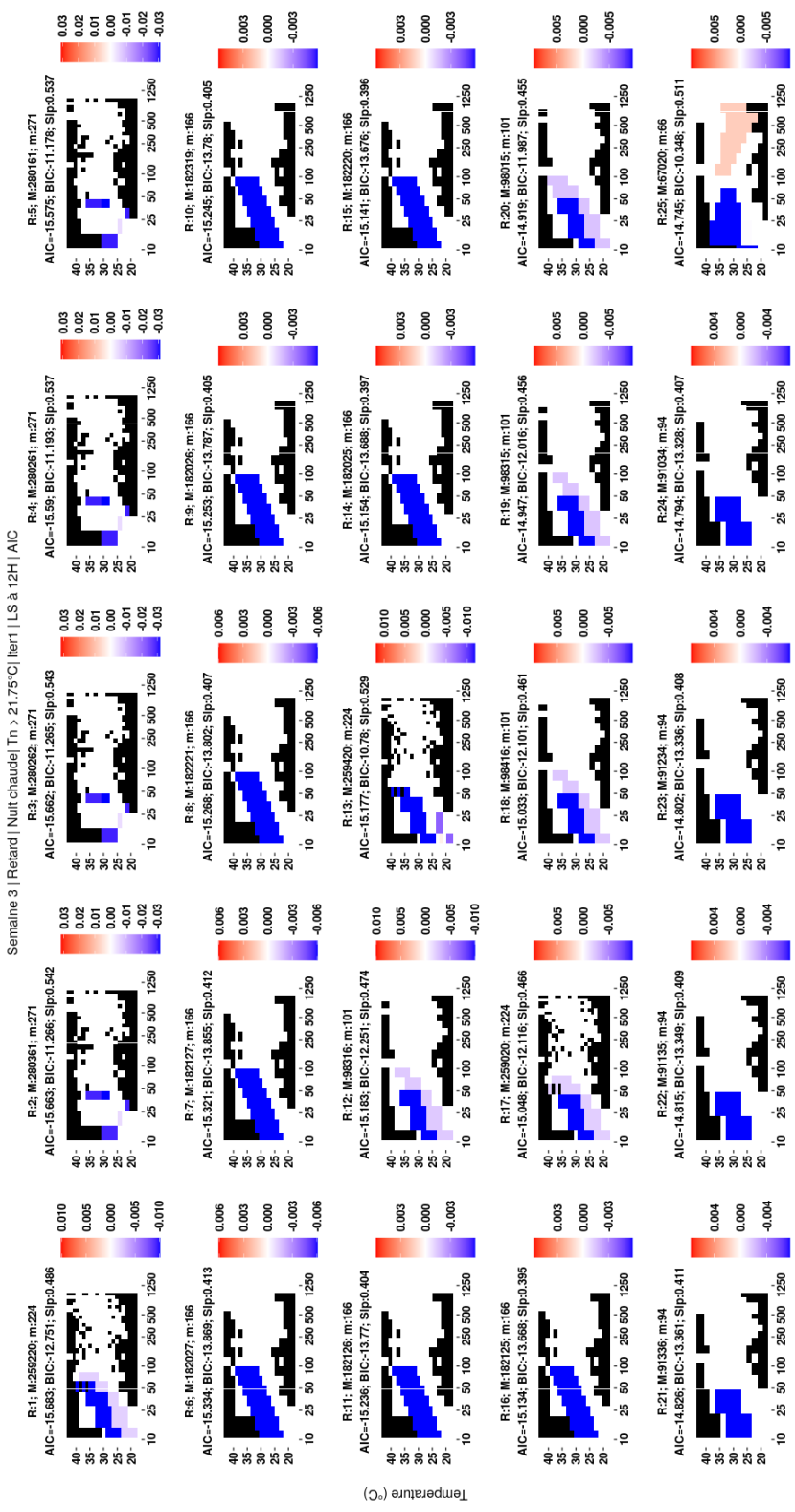


FIGURE 3.5.3 – Semaine 3 - Retardés - Nuits chaudes. 25 meilleurs modèles suivant l'AIC. R :Rang du modèle, m : matrice candidate, M : modèle, Slp : pente (slope)

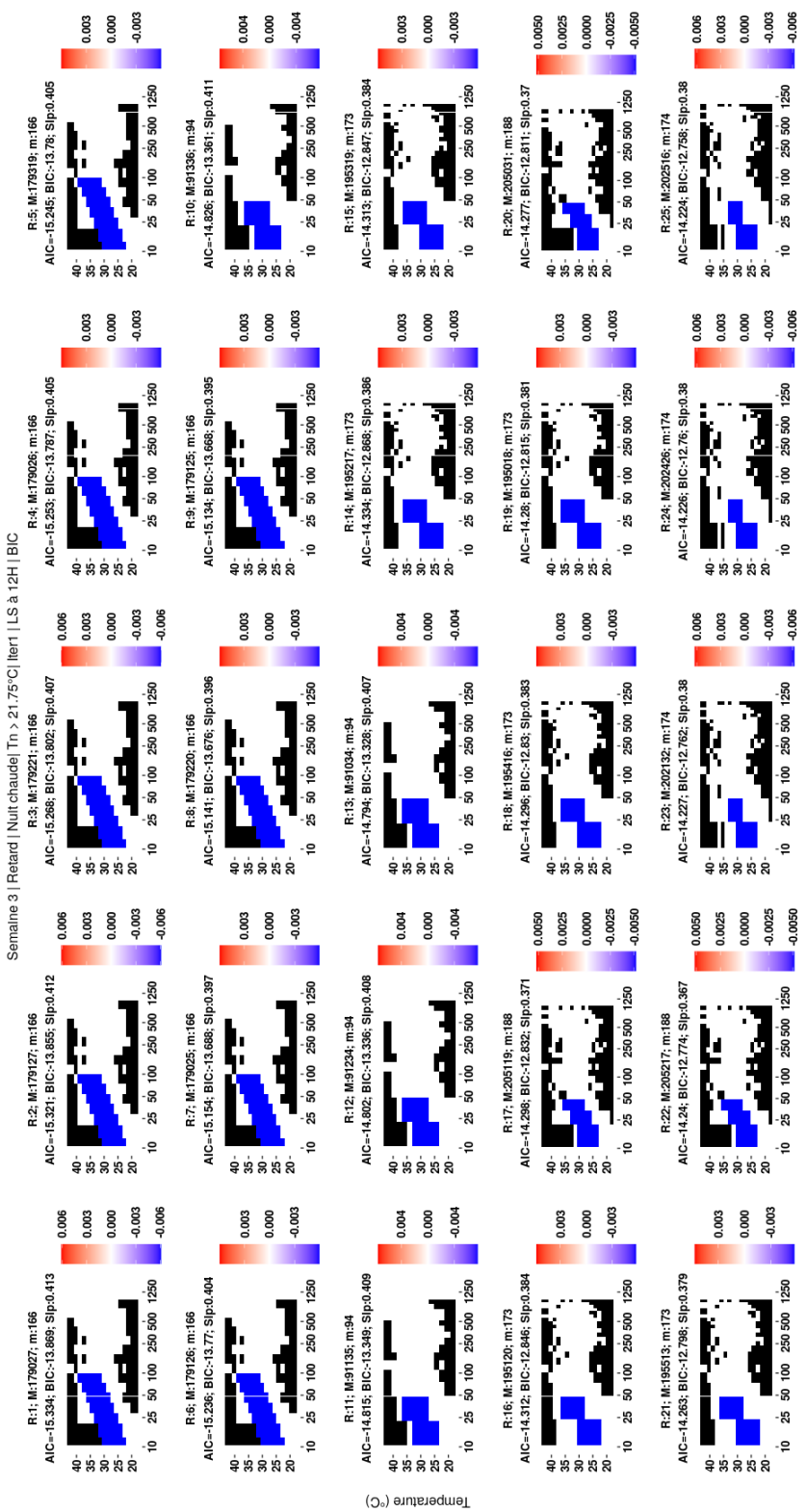


FIGURE 3.5.4 – Semaine 3 - Retardés - Nuits chaudes. 25 meilleurs modèles suivant le BIC. R :Rang du modèle, m : matrice candidate, M : modèle, Slp : pente (slope)

Chapitre 4

R Package SpiceFP : a Sparse and Structured Procedure to Identify Combined Effects of Functional Predictors

4.1 Introduction

The development of information technologies, observed over the last decades, has not occurred without some changes in the various data collected. The flow of information, sometimes obtained continuously, generates new types of data (curves, images, surfaces, etc.). In order to extract knowledge from these data or to use them in the prediction of a variable of interest, it is necessary to propose new statistical solutions or to adapt existing ones. In this context, the regression with functional data is commonly used [89]. In most production systems (industry, agronomy, health etc.), the variable of interest is observed at the final point (as the yield, at the end of the process). This final value is conditioned and impacted by past values of covariables we are able now to observe continuously (during the production process). This context is usually modeled with 'scalar-on-function' approaches [90, 66]. In a different context, the development of available spatialised data (from medical imagery, GPS, image from satellite or drone etc.) have led to specific approaches for 'scalar-on-image' or 'spatial functional' regressions like [38], [43]. Both contexts offer new tools for multiple functional regression, we can cite as an example FAME (Functional Adaptive Model Estimation) from [51]. Most of the linear or generalized linear modeling also included a penalized estimation like [70] or [34] for examples. Some R packages were developed for multivariate functional models.

According to [117], 'a common assumption made by all the above mentioned models is that the effects of the functional predictors are additive, thus any interaction between the functional covariates is not taken into account'. Ignoring interaction effect in linear models may lead to biased estimation and incorrect conclusions. Therefore [117] proposed an interac-

tion model for functional regression. Their proposition is a generalization of what is usually done in multiple linear regression with a product (convolution) between functional predictors to model the interaction term. Another way to account for interaction is to define combinations of ranges of functional predictors values and identify the combinations that have an impact on the variable of interest. This way offers an alternative to the scalar-on-function regression that is more interpretable. The notion of interpretability for functional linear regression has grown up with [53] (R code available). A Bayesian extension was proposed by [42] with the R package **bliss** available on CRAN. The 'scalar-on-function' multiple linear or generalized linear regressions are implemented in R packages (**FDA**, **funcreg** or **refund**), no identification of interaction terms is provided. The package **SpiceFP** offers to fill this gap. The full description of the method is available in [37].

The originality of the Sparse and Structured Procedure to Identify Combined Effects of Functional Predictors (SpiceFP) is to partition the range of values of each functional predictor into intervals (with bins of equal size). The idea is to transform each functional variable into a categorical variable and then to create a contingency table (which is in fact a partition of the predictors' observation domain). It is assumed that only the time spent in a combined class of intervals affects the variable of interest. This assumption is supported by the idea of a certain kind of independence and stationarity during the production process : same causes, same effects, not conditioned by the past. Several candidate partitions are defined this way, depending on the choice of the bin size. Each candidate partition forms a candidate design matrix which is used in a linear multiple regression model. A contiguity constraint was added between adjacent class intervals to manage possible colinearity. Identification is performed through a Generalized Fused Lasso using each candidate matrix as input variables. The selection of the best candidate and of its relative regression coefficients is achieved by minimizing an information criteria.

In a nutshell, the package **SpiceFP** offers a new functional approach whose main objective is the interpretability of the result. Under the hypothesis of combined influence of univariate functional predictors on a scalar response, the SpiceFP approach provides a surface (support) describing areas of influence and non-influence.

This paper is organised as follows. Section 2 briefly presents the methodology of this approach. The reader may refer to [37] for an in-depth presentation of the method. Details about the implementation and the options available in the package **SpiceFP** are presented in section 3. Some examples of use are presented in section 4.

4.2 Presentation of the approach

The SpiceFP algorithm is based on a transformation of functional variables into categorical variables by defining joint modalities from which we derived a collection of multiple regression models, where the regressors are the frequencies associated to the joint modalities. Regressors are candidate matrices under contiguity constraints. Generalized Fused Lasso regressions are performed in order to identify the best model. The **SpiceFP** package provides a set of easy-to-use functions to implement the algorithm on a dataset. To combine our explo-

ration objectives with technical constraints (such as a potentially small amount of data, for example), we also propose an iterative extension of the SpiceFP algorithm, which explores a larger space of solutions, possibly at the cost of a slight overestimation.

The 4 main steps of the procedure are described in this section :

1. The functional variables $\mathcal{A}$ and $\mathcal{B}$ are transformed into categorical variables which requires the construction of a contingency table (counting t) at the scale of each individual i, $i = 1, ..., n$, see figure 4.2.1. The information contained in the n contingency tables is used to construct a matrix of predictors with n rows.
2. Generalized Lasso Regression is used to estimate the coefficients of the joint modalities. The generalized version allows a constraint on the model estimation to enforce continuity between adjacent joint modalities.
3. Model selection is done with respect to Akaike or Bayesian information criteria
4. The iterative step, optional, relies on the residuals : once an iteration is completed, the residuals are used as the response variable in a new iteration.

The overall approach is outlined in the remainder of this section.

4.2.1 Transformation of functional predictors into a set of candidate matrices

For n statistical individuals, consider the samples of two explanatory functional variables $\mathcal{A}$ and $\mathcal{B}$ expressed as $(\mathcal{A}_i)_{i=1,...,n}$ and $(\mathcal{B}_i)_{i=1,...,n}$, where i stands for an individual and $(y_i)_{i=1,...,n}$ the sample of the corresponding scalar response variable y.

Both $\mathcal{A}$ and $\mathcal{B}$ are observed on the same set T of fixed observation variables. We note $\mathcal{A}_i(t)$, respectively $\mathcal{B}_i(t)$, the observations of $\mathcal{A}$, respectively $\mathcal{B}$, at $t \in T$ (t can be time, wavelength or other observation variable) for an individual i.

Contingency table per individual

Using $n_{\mathcal{A}}+1$ (*resp.* $n_{\mathcal{B}}+1$) breaks, the explanatory variables $\mathcal{A}_i$ (*resp.* $\mathcal{B}_i$) are partitioned in $n_{\mathcal{A}}$ (*resp.* $n_{\mathcal{B}}$) class intervals, the same for all i, according to a linear scale (equidistant breaks). These breaks, denoted $L_{\mathcal{A}}(v),\ v = 1, \ldots, n_{\mathcal{A}} + 1$ (*resp.* $L_{\mathcal{B}}(w),\ w = 1, \ldots, n_{\mathcal{B}} + 1$) are computed as follows :

$$L_{\mathcal{A}}(v) = \underline{\mathcal{A}} + \frac{v-1}{n_{\mathcal{A}}} \left(\bar{\mathcal{A}} - \underline{\mathcal{A}}\right),\ v = 1, \ldots, n_{\mathcal{A}} + 1, \tag{4.2.1}$$

with $\underline{\mathcal{A}} \in \mathbb{R}$ and $\bar{\mathcal{A}} \in \mathbb{R}$ the minimum and maximum of the observed values of $(\mathcal{A}_i)_i$. The breaks $L_{\mathcal{B}}(w)$ can equivalently be obtained by using $\underline{\mathcal{B}} \in \mathbb{R}$ and $\bar{\mathcal{B}} \in \mathbb{R}$ the minimum and maximum of the observed values of $(\mathcal{B})_i$. The class intervals obtained for partitioning the $(\mathcal{A})_i$ are $I_{\mathcal{A}}(v) = [L_{\mathcal{A}}(v), L_{\mathcal{A}}(v+1)[,\ v = 1, \ldots, n_{\mathcal{A}}$ and those for partitioning the $(\mathcal{B})_i$ are $I_{\mathcal{B}}(w) = [L_{\mathcal{B}}(w), L_{\mathcal{B}}(w+1)[,\ w = 1, \ldots, n_{\mathcal{B}}$. The numbers of class intervals $n_{\mathcal{A}}$ and $n_{\mathcal{B}}$ have to be set to compute these breaks. Let $u = (n_{\mathcal{A}}, n_{\mathcal{B}})$ denote the partition vector. For all

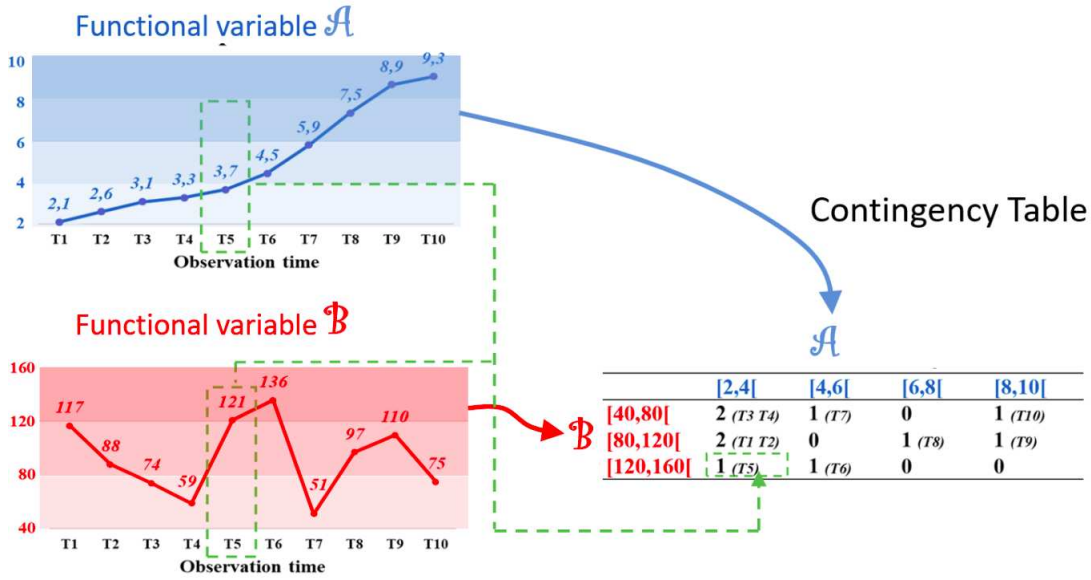


FIGURE 4.2.1 – Contingency table per individual

couples $(\mathcal{A}_i, \mathcal{B}_i)$, we obtain the frequency bivariate histogram as a contingency table C_i^u, of dimension $n_\mathcal{A} \times n_\mathcal{B}$, whose components $C^u_{i,(v,w)}$ are obtained through :

$$C^u_{i,(v,w)} = \sum_{t \in T} \mathbb{1}_{\mathcal{A}_i(t) \in I^u_\mathcal{A}(v),\ \mathcal{B}_i(t) \in I^u_\mathcal{B}(w)} = Card\ \{t \in T | \mathcal{A}_i(t) \in I^u_\mathcal{A}(v), \mathcal{B}_i(t) \in I^u_\mathcal{B}(w)\}, \quad (4.2.2)$$

for all $v = 1, \ldots, n_\mathcal{A}$, $w = 1, \ldots, n_\mathcal{B}$ and each $u = (n_\mathcal{A}, n_\mathcal{B})$, with $\sum_{v=1}^{n_\mathcal{A}} \sum_{w=1}^{n_\mathcal{B}} C^u_{i,(v,w)} = Card(T)$. From a theoretical point of view, when t is a time step, $C^u_{i,(v,w)}$ represents a discrete approximation of the density of the time spent by the individual i with variable $\mathcal{A}_i$ in $L_\mathcal{A}(v)$ and variable $\mathcal{B}_i$ in $L_\mathcal{B}(w)$. In practice, it is the number of times that the observations of $\mathcal{A}_i$ and $\mathcal{B}_i$ are at the same time in $I^u_\mathcal{A}(v) \times I^u_\mathcal{B}(w)$. These combinations will be referred to as joint modalities hereafter.

Predictor and penalty matrices per partition vector

For each fixed partition vector u, a matrix of predictors X^u is obtained by vectorization (stacking column by column) and transposition of the contingency tables :

$$X^u_i = {}^t Vect(C^u_i)\,,\ X^u_i \in \mathbb{R}^{n_\mathcal{A} n_\mathcal{B}}. \quad (4.2.3)$$

Each row X^u_i of X^u contains all the information relative to one individual and each column those relative to one joint modality. X^u has n rows and $n_\mathcal{A} \times n_\mathcal{B}$ columns. For each partition vector u, X^u is called a candidate matrix. To this matrix, we add the information about the contiguity constraints between the joint modalities by creating a graph $G^u(V^u, E^u)$ where V^u represents the columns (new variables) of the candidate matrix X^u and E^u the set of edges

connecting two close joint modalities. The definition proposed by default in the **SpiceFP** package is that two joint modalities are said to be close if the classes following the variable $\mathcal{A}$ (indexed by v) or (exclusive) the classes following the variable $\mathcal{B}$ (indexed by w) are consecutive. The package offers the user the possibility to define the proximity between the joint modalities in a different way.

4.2.2 Models and estimation

The model under SpiceFP is defined, for each partition u and each individual i, by :

$$y_i = X_i^u \beta^u + \varepsilon_i, \tag{4.2.4}$$

where X_i^u is given in Equation (4.2.3), $\beta^u_{(v,w)}$ is the coefficient to be estimated on the 2D interval $(I^u_{\mathcal{A}}(v) \times I^u_{\mathcal{B}}(w))$ and ε_i is an i.i.d. Gaussian error. Investigating the suitability of a partition vector u requires the construction of a matrix of predictors X^u associated to a graph G^u, the estimation of the coefficients in (4.2.4) and finally the evaluation of the goodness of fit of the model related to u.

Set of candidate matrices to be evaluated

There will be as many candidate matrices as partition vectors to investigate. The set of all partition vectors is defined from the set of possible values of each parameter involved in u ($n_{\mathcal{A}}$ and $n_{\mathcal{B}}$ in the default case, as discussed in the present section). For example, if we set 15 possible values for $n_{\mathcal{A}}$ and 20 values for $n_{\mathcal{B}}$, we will have 300 partition vectors to investigate or 300 candidate matrices.

Fitting of all candidate models

The estimation issue here is similar to a well-known topic in statistics called the scalar-on-image regression [57, 61]. The objective is to control the smoothness of the non-zero estimated coefficients. In the present case, the "image" (2D image) is a bivariate representation (indexed by v and w) of both functional variables. To estimate coefficients of scalar-on-image regression, you can choose among Bayesian approaches [39], total variation penalizing approaches [121] or approaches that take L_1 regularization or neighborhood into account, as [62]. We selected the Generalized Fused Lasso (GFL) [122] which promotes smoothness and sparsity over neighboring variables by using constraints on the parameter differences. In SpiceFP, the coefficients of this Generalized Fused Lasso are estimated using the framework of the generalized Lasso model, introduced by [114] as an encapsulation of statistical models using the L_1 norm to impose additional constraints. This estimator is as follows, for a fixed partition vector u :

$$\widehat{\beta}^{u,\gamma}(\lambda) = \underset{\beta \in \mathbb{R}^{n_{\mathcal{A}}\, n_{\mathcal{B}}}}{\operatorname{argmin}} \ \frac{1}{2}||y - X^u\beta||_2^2 + \lambda||D^{u,\gamma}\beta||_1, \tag{4.2.5}$$

where :

— $y = {}^t(y_1, y_2, \ldots, y_n) \in \mathbb{R}^n$ is the response vector.

— $X^u \in \mathbb{R}^{n \times n_{\mathcal{A}} n_{\mathcal{B}}}$ is the matrix of predictors defined in Equation (4.2.3).

— $\lambda \in \mathbb{R}$ is a penalty parameter that controls the smoothness of the coefficients.

— $\gamma \in \mathbb{R}^+$ is a ratio that controls the sparsity among the coefficients. If $\gamma = 0$, there is no sparsity i.e pure fusion of the coefficients. The higher its value, the more parsimonious the model is.

— $\widehat{\beta}^{u,\gamma}(\lambda) \in \mathbb{R}^{n_{\mathcal{A}} n_{\mathcal{B}}}$ is the vector of estimated coefficients for fixed values of u, γ and λ.

— $D^{u,\gamma} = \begin{pmatrix} D^{u,f1} \\ D^{u,f2} \\ D^{u,\gamma,p} \end{pmatrix} \in \mathbb{R}^{(3n_{\mathcal{A}} n_{\mathcal{B}} - n_{\mathcal{A}} - n_{\mathcal{B}}) \times n_{\mathcal{A}} n_{\mathcal{B}}}$ is a specified penalty matrix for fixed values of u and γ with : $((v,w)(v',w')$ defines one of the edge in E^u for the graph $G^u)$

$$
\begin{aligned}
D^{u,f1}_{(v,w)(v',w')} &= \begin{cases} 1 & \text{if} \quad (v',w') = (v+1,w) \\ -1 & \text{if} \quad (v',w') = (v,w) \text{ and } v < n_{\mathcal{A}} \\ 0 & \text{if} \quad \text{not} \end{cases} \\
D^{u,f2}_{(v,w)(v',w')} &= \begin{cases} 1 & \text{if} \quad (v',w') = (v,w+1) \\ -1 & \text{if} \quad (v',w') = (v,w) \text{ and } w < n_{\mathcal{B}} \\ 0 & \text{if} \quad \text{not} \end{cases} \\
D^{u,\gamma,p} &= \gamma . \mathbb{I}_{n_{\mathcal{A}} . n_{\mathcal{B}}}
\end{aligned}
\tag{4.2.6}
$$

where $\gamma \geq 0$ and $\mathbb{I}_{n_{\mathcal{A}} . n_{\mathcal{B}}}$ is the identity matrix. $D^{u,f1}$ and $D^{u,f2}$ are relative to the smoothness of the coefficients following both functional variables and $D^{u,\gamma,p}$ is relative to sparsity among them.

4.2.3 Selection among the candidate models

Let's n_u be the number of candidate matrices, Γ the set of γ values, and n_λ the number of λ values. We are thus interested in the best models obtained from the $n_u \times Card(\Gamma) \times n_\lambda$ estimated models through a model selection procedure. Table 4.1 presents three criteria that are provided in the package. Once the criteria are computed, the best candidate matrix $X^{\widehat{u}}$ according to a specified criterion is the one involved in the best model identified by this criterion.

These criteria require the computation of degrees of freedom. A theorem is proposed by [115] to compute the degrees of freedom in Lasso problems. The computation of the degree of freedom for the Generalized Fused Lasso is presented in [37]. Its value is equivalent to the number of sets of indexes of non zero $\widehat{\beta}^{u,\gamma}_{v,w}$ that are linked together via the $D^{u,\gamma}$ matrix (4.2.6)

TABLE 4.1 – Criteria used to select a model. Those criteria needs the computation of the Residual Sum of Squares $RSS = \|y - X^u \widehat{\beta}^{u,\gamma}(\lambda)\|_2^2$, where X^u is a candidate matrix, $\widehat{\beta}^{u,\gamma}(\lambda)$ its estimated coefficients, *df* the degree of freedom and where σ^2 is assumed to be the known variance of y.

Criteria	Formula
Akaike Information Criterion [2]	$AIC = n\ \log(2\pi\sigma^2) + \frac{RSS}{\sigma^2} + 2\text{df}$
Bayesian Information Criterion [102]	$BIC = n\ \log(2\pi\sigma^2) + \frac{RSS}{\sigma^2} + \log(n)\text{df}$
Mallows's C_p [68]	$C_p = RSS + 2\sigma^2\text{df}$

and that all share the same real value. We will refer to these sets as connected components. In other words, a connected component is a set of joint modalities linked by $D^{u,\gamma}$ and having a common influence on the response variable.

The criteria presented in Table 4.1 also require the knowledge of σ^2, the variance of y. Since the variance must remain the same for all the models to be compared, we decided to estimate it by the empirical variance of y : $\widehat{\sigma}^2 = \frac{1}{n-1}||y - \overline{y}||_2^2$. It's a biased estimator of σ^2, but this bias remains fixed for all models compared [46]. Such an estimator may lead to an overestimation of the variance, which penalizes the introduction of new coefficients in the model. This bias can be partly offset by an iterative approach. Therefore, we set up an iterative extension where the residuals are used as a new response variable and so on, until stopping conditions are verified.

4.2.4 An optional iterative extension

To capture all potential non-zero coefficients, one should take a fine partition with values of $n_{\mathcal{A}}$ and $n_{\mathcal{B}}$ large enough. This could imply a very low or even zero number of points in the joint class intervals (making the method ineffective) and prohibitively long computation times [67]. As a trade off between thinness and work-ability we chose to develop an iterative approach to explore a large space of solutions (that allows addition of different thinness of partition).

After identifying $X^{\widehat{u}}$ at one iteration, the residuals of the best model according to the same criterion may be computed and used as response variable at a next iteration. The iterative process is stopped when the vector of estimated coefficients is the null vector, or when the maximum number K of iterations is reached. The final model is the sum of all the models estimated at each iteration.

4.2.5 Sum up

The figure 4.2.2 illustrates the overall approach.

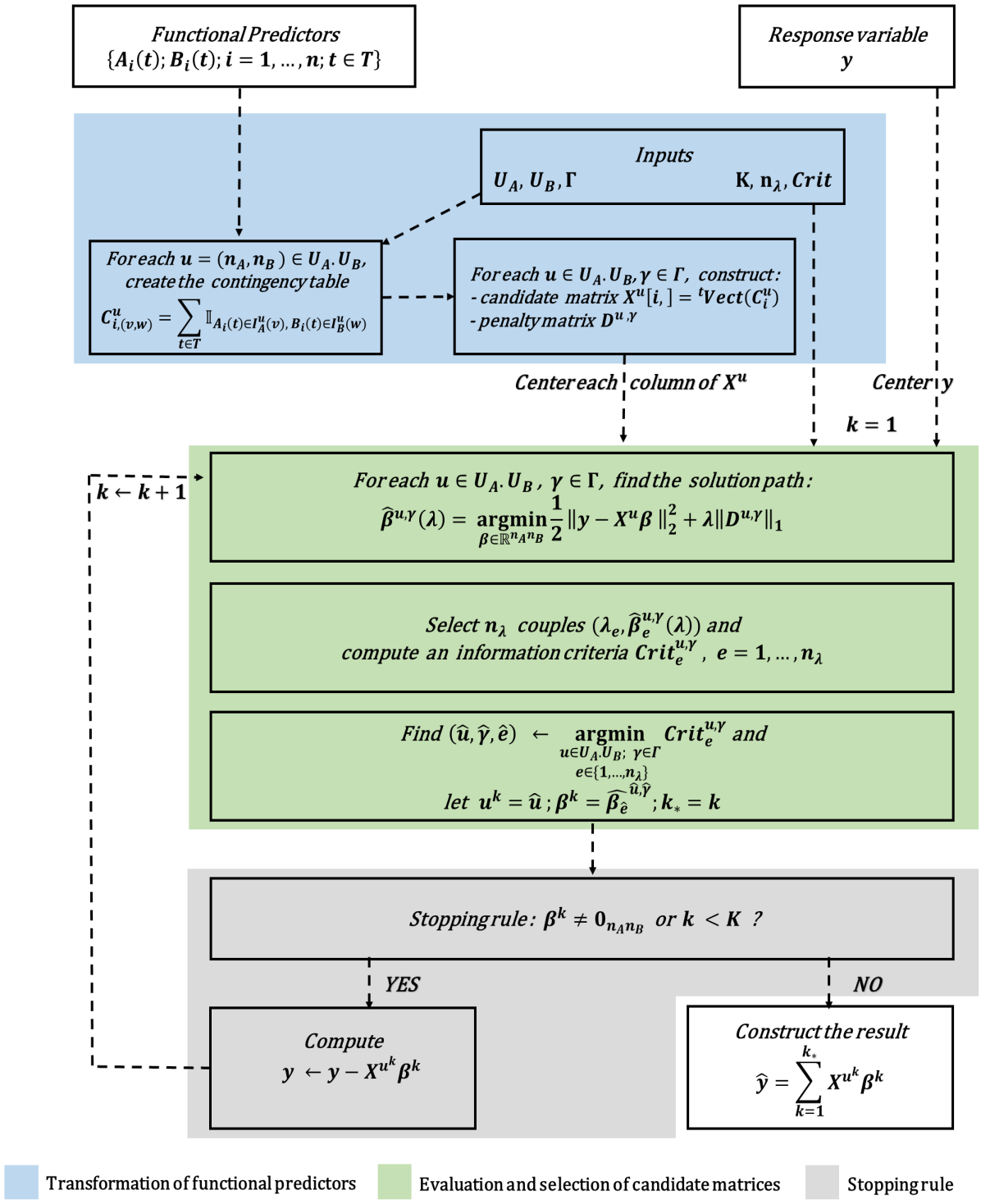


FIGURE 4.2.2 – Summary diagram of the SPICEFP approach.

The input data are the functional explanatory variables $\mathcal{A}_i$ and $\mathcal{B}_i$ discretized on a grid T and a response variable y_i with $i = 1, \ldots, n$. Other elements are also required : Γ, a set of positive reals representing γ ratios of regularization parameters, $\mathcal{U}_\mathcal{A}$ and $\mathcal{U}_\mathcal{B}$ the sets of numbers of class intervals n_A and n_B, n_λ the selected number of pairs (among N_λ) (λ, $\widehat{\beta}^{u,\gamma}(\lambda)$), the information criterion to be used, and K the maximum number of iterations to explore. The n_λ values of λ are chosen equally spaced on the log scale (see Genlasso package [4]).

SpiceFP constructs for each couple (u, γ), a matrix of explanatory variables X^u and a penalty matrix $D^{u,\gamma}$. This first step is performed only once and the set of candidate explanatory matrices remains unchanged. For each candidate matrix X^u a solution path (λ, $\widehat{\beta}^{u,\gamma}(\lambda)$) is obtained from the second step of the algorithm. Once the criterion $Crit_e^{u,\gamma}$ associated with each model is computed, the optimal triplet $(\widehat{u}, \widehat{\gamma}, \widehat{e})$ is the argument that minimizes $Crit_e^{u,\gamma}$, where e is the index of the solution path-coefficients $(\lambda_e, \hat{\beta}_e^{u,\gamma})$. $X^{\widehat{u}}$, $D^{\widehat{u},\widehat{\gamma}}$ and $\widehat{\beta}_{\widehat{e}}^{\widehat{u},\widehat{\gamma}}$ respectively represent the optimal matrices of the explanatory, penalty and coefficient variables.

At each iteration k, we denote $u^k = \widehat{u}$ and $\beta^k = \widehat{\beta}_{\widehat{e}}^{\widehat{u},\widehat{\gamma}}$. SpiceFP then checks if the selected coefficients according to the criterion $Crit_e^{u,\gamma}$ correspond to a zero vector or if the maximum number of iterations K is reached. The algorithm is stopped when at least one of these conditions is verified. When none of these conditions are verified, the residuals of the optimal model are computed and used as a response variable at the next iteration of the algorithm.

The final prediction is the sum of all the predictions obtained at each iteration : $\widehat{y} = \sum_{k=1}^{k_*} X^{u^k} \beta^k$, where k_* is the final number of iterations.

We remark that each vector β^k at each iteration may have different length $dim(u^k)$.

4.3 Implementation in R

The SpiceFP implementation in R follows the S3 methods. The main function `spicefp` allows to execute all the approach and returns estimated coefficients which are visualized as a 2D image. An extension including a third explanatory variable (combination of 3 class intervals) is also provided. The package allows the user to define his own partition of variables. The `meancoef` function enables to extend the approach while using coefficients estimated in the framework of **SpiceFP**. The user can access various functions required to achieve some of the approach's steps such as `candidates`, `evaluate.candidates` or `coef_spicefp` functions. Data are also provided in the package to test the functions.

4.3.1 Transformation of functional predictors : the "candidates" function

There is two main ways to construct candidate matrices. One is based on the framework defined by the **SpiceFP**'s functions and the other allows the user to construct the matrices as he wishes, before submitting them to the `spicefp` function via the `candmatrices` argument. These two options are detailed hereafter.

Construction of candidate matrices in the package framework

The **SpiceFP** package constructs candidate matrices through the function `candidates`. Its philosophy is to return matrices of new predictors with in columns combinations of two (three) class intervals related to two (three) functional predictors. To achieve this goal, the function requires as inputs each functional predictor, an R function that will be used to partition this functional predictor and arguments of this function. The other requirements inform about the number of cores to use (parallelization with **doParallel** package [72]) and define whether or not the columns of candidate matrices (joint modalities) should be centered or scaled. Each functional predictor is presented as one numerical matrix with in columns observations of one statistical individual. Due to the fact that the functional predictors should be observed with the same set T of time steps, the numerical matrices contain $Card(T)$ rows and each row is relative to the observation variable $t \in T$. Statistical individuals keep the same order through the matrices to be provided for `fp1`, `fp2` or `fp3` arguments.

In order to partition the observations of the functional predictors, the functions (`fun1`, `fun2` or `fun3`) must comply with certain constraints. The only output of these functions must be a numerical vector of breaks that will be used to generate the class intervals. They must also consider at least two separate arguments. The first argument is a numerical vector (the vector to be partitioned). The second is a list of partitioning parameters to be optimized by the approach. The parameters to be optimized are the components of the partition vector u. The parameters that do not need to be optimized can be defined as additional arguments and set by default. To illustrate, let's look at two partition functions. The first one, `linbreaks`, presented below, allows to partition a numerical vector according to a linear scale. Its use in the construction of the `example1` vector allows the identification of 13 breaks that will be used to formulate 12 consecutive class intervals.

```
R> linbreaks <- function(x,n){
+     round(seq(trunc(min(x)),
+                     ceiling(max(x)),
+                     length.out = unlist(n)+1),
+                 1)
+ }
R> set.seed(284)
R> example1 <- linbreaks(rpois(1000,100), list(12))
R> example1

 [1]  67.0  72.2  77.5  82.8  88.0  93.2  98.5 103.8 109.0 114.2 119.5 124.8
[13] 130.0
```

For the second example, we focus on the `logbreaks` function in the **SpiceFP** package which allows to get breaks according to a logarithmic scale. The reader can refer to the help page of this function for more details. To compute the breaks, `logbreaks` requires two essential parameters (`alpha` and `J`) provided by the argument `parlist` as well as other additional arguments. One way to use `logbreaks` in the **SpiceFP** approach is to create a

new function `Logbreaks` setting the additional arguments by default as presented below. This framework therefore offers the user a high degree of flexibility in the way functional predictors are partitioned.

```
R> Logbreaks <- function(x, Parameterlist){
+   logbreaks(x, Parameterlist,  round_breaks = 1, plot_breaks = FALSE,
+             effect.threshold.begin = NA,   effect.threshold.end = NA)
+ }
R> set.seed(284)
R> example2 <- Logbreaks(rpois(1000,100), list(0.05,12))
R> example2

 [1]  67.0  69.5  72.4  75.5  79.1  83.2  87.7  92.9  98.6 105.2 112.5 120.7
[13] 130.0
```

Once functional predictors and partition functions are defined, another important argument of `candidate` function is `parlists`. It is a list. Its length equals the number of functional predictors involved in the model (i.e. 2 or 3). Each element of this list is related to one functional predictor and its partition function. The length of each of these elements is exactly the number of candidate matrices to create. More precisely, each element in `parlists` represents a list with all the second arguments related to one partition function that will help to create various candidate matrices. In order to illustrate this, let us consider the creation of the argument `parlists` required to transform two functional predictors `var1` and `var2` into a set of candidate matrices. Suppose that one parameter is needed to partition `var1` namely `var1.nclass` (4 values are possible : 10, 12, 14, 16) and two parameters for `var2` : `var2.nclass` (3 possible values : 20, 22, 24) and `var2.alpha` (3 possible values : 0.01, 0.02, 0.03). To construct $4 \times 3 \times 3 = 36$ candidate matrices covering the different combinations of parameters, `parlists` can be constructed using the following commands :

```
R> var1.nclass <- c(10, 12, 14, 16)
R> var2.nclass <- c(20, 22, 24)
R> var2.alpha <- c(0.01, 0.02, 0.03)
R> p2 <- expand.grid(var1.nclass, var2.alpha, var2.nclass)
R> parlist.var1 <- split(p2[,1], seq(nrow(p2)))
R> parlist.var1 <- lapply(parlist.var1, function(x){list(x[[1]])})
R> parlist.var2 <- split(p2[,2:3], seq(nrow(p2)))
R> parlist.var2 <- lapply(parlist.var2, function(x){list(x[[1]],x[[2]])})
R> parlists <- list(parlist.var1,  parlist.var2)
R> length(parlists[[1]])

[1] 36
```

Then, the `candidates` function uses the partition function related to each functional predictor to compute the breaks that should be used to construct its class intervals. These

breaks together with the observations of the functional variables ($\mathcal{A}_i$, $\mathcal{B}_i$, etc.), are the required inputs of the functions `hist_2d` or `hist_3d`, used to construct, for each individual i, histograms or contingency tables C_i^u. Each of these n contingency tables will be transformed into X_i^u, as presented in the equation (4.2.3). All these matrices of explanatory variables X^u are saved in a list with the same length as each element of `parlists`. Attention was also paid to associate with each of these matrices, a numerical vector giving information on the index of the associated vector u and the number of class intervals of each functional predictor in the order `fp1`, `fp2` (and `fp3`, if three functional predictors are used). These numbers of class intervals will be used to generate the penalty matrix $D^{u,\gamma}$.

Construction of candidate matrices independently of the package framework

There is one constraint to respect in the construction of the candidate matrices that will be provided directly as inputs to the function `spicefp` : the organization of these candidate matrices as well as their presentation must be similar to that provided by the function `candidates`. TIn order to remove this constraint, we present here in details the `candidates` function output. This output is a list of eleven named elements respectively and organized as follows : to help complying with this constraint, we present here in details the `candidates` function output. This output is a list of eleven elements, named and organized as follows :

— `spicefp.dimension` : scalar number equal to 2 or 3, which represents the number of class intervals that compose a joint modality (in column of the X^u matrix).

— `candidates` : list in which each element is a list containing a matrix X^u and a vector Z^u, both relative to the same candidate. The matrix, the first element of the list, always contains n rows. The number of columns varies according to the candidates. The columns are named by the combination of class intervals separated by the symbol "underscore (_)". As example, for the two functional variables $\mathcal{A}$ and $\mathcal{B}$, we get $[L_{\mathcal{A}}(v), L_{\mathcal{A}}(v+1)[\_[L_{\mathcal{B}}(w), L_{\mathcal{B}}(w+1)[,\ v = 1, \ldots, n_{\mathcal{A}},\ w = 1, \ldots, n_{\mathcal{B}}$. The order of the class intervals in editing the combination is important. Here, $\mathcal{A}$ is considered the first and $\mathcal{B}$ the second. This order appears in the construction of the vector Z^u ($Z^u = c(\text{match}(u, U), n_{\mathcal{A}}, n_{\mathcal{B}})$). Its first element is an identification key that allows to associate the X^u matrix with the parameters used to create it. The rest of the elements Z^u are the numbers of class intervals created by functional predictor arranged in the same order.

— `fp1, fp2, fp3, fun1, fun2, fun3, parlists, xcentering, xscaling` : these terms respectively represent the functional predictors, the associated partition functions, the partition vectors and the logical parameters that indicate whether the candidate columns should be centered or scaled. They correspond to the elements used or not in the construction of the candidates. The user can assign them the `NULL` value but he must provide a value for each term.

4.3.2 Evaluation of candidate models by generalized fused lasso : the "evaluate.candidates" function

After constructing the candidate matrices, the second step of the approach is implemented by the `evaluate.candidates` function. This function mainly returns a matrix of information criteria values, with in rows the models and in columns the associated parameters. This function consists in estimating the parameters $(\lambda, \widehat{\beta}^{u,\gamma}(\lambda))$ of equation (4.2.5) and computing, for each model, different information criteria [46]. To achieve this, we start by defining a set Γ containing all the γ values. For each candidate X^u, we construct $Card(\Gamma)$ penalty matrices $D^{u,\gamma}$. Each $D^{u,\gamma}$ is generated using the argument `penfun` of `evaluate.candidates`. `penfun` is a function taking as inputs all elements of the Z^u vector except the first one. By default, it takes the `NULL` value. In this case, either the `getD2dSparse` function of the **genlasso** package [5] (when `spicefp.dimension` $= 2$) or the `getD3dSparse` function of the **SpiceFP** package (when `spicefp.dimension` $= 3$) are used. `getD2dSparse` allows to define a rook's case contiguity between the joint modalities. `getD3dSparse` is an extension of `getD2dSparse` to a third dimension. More precisely, it is the part of $D^{u,\gamma}$ (see equation (4.2.6)) penalizing the fusion of the coefficients that is constructed by `penfun`. It can be noticed that γ is not part of its arguments. The creation of $D^{u,\gamma,p}$, the part that penalizes parsimony, is automatically done in the next step by the `fusedlasso` function (**genlasso**).

In statistics, they are many fast algorithms for solving linear regression with l_1 penalty at a single value of the tuning parameter λ. But when the solution is requested for many values of the tuning parameter, the least-angle regression algorithm (LARS) [27] provides a computational advantage in the sense that it help to solve linear regression with l_1 penalty for all $\lambda \in [0, \infty[$. The algorithm in **genlasso** is derived from the LARS path algorithm in order to solve problems that use the l_1 norm to enforce structural constraints instead of pure sparsity on the coefficients in a linear regression [114]. Since **SpiceFP** has to estimate a tuning parameter as well as coefficients for different candidate matrices, this computational advantage is welcome. The **genlasso** package is designed to compute the solution path of the generalized lasso problem, which minimizes a criterion similar to the one presented in equation (4.2.5). This solution path is computed by solving the equivalent Lagrange dual problem and its implementation is presented in [3]. For a candidate X^u and a penalty matrix $D^{u,\gamma}$, `fusedlasso` returns N_λ couples $(\lambda, \widehat{\beta}^{u,\gamma}(\lambda))$. The values of lambda in these couples are those at which the solution path changes slope. By default, $N_\lambda = 2000$ when using **genlasso** [5]. But we will only work with $n_\lambda < N_\lambda$ pairs chosen according to a logarithmic scale in the path solution (argument `nknots`) or from an *a priori* knowledge of the number of expected areas of influence (argument `appropriate.df`).

4.3.3 Post-evaluation treatment and result construction

The **SpiceFP** package returns areas of influence of non null coefficients.`spicefp` is the main function of the **SpiceFP** package. It allows to implement the SpiceFP approach from already constructed candidate matrices to be provided as inputs (argument `candmatrices`) or candidate matrices to be constructed via functional predictors (`fp1, fp2, fp3`) and as-

TABLE 4.2 – Statistics used to summarize the information associated with a model involving a response variable y, a candidate matrix X^u, estimated coefficients $\beta^{u,\gamma}(\lambda)$, degree of freedom df and σ^2 the variance of y.

Statistics	Formula
Residual Sum of Squares	$RSS = \|y - X^u\beta^{u,\gamma}(\lambda)\|_2^2$
Generalized Cross Validation [23]	$GCV = \frac{1}{n}\frac{RSS}{(1-\mathrm{df}/n)^2}$
Slope of regression $\widehat{y} \sim y$; $(\widehat{y} = X^u\beta^{u,\gamma}(\lambda))$	$Slope = ({}^t yy)^{-1}\ {}^t y\widehat{y}$
Variance ratio of $\widehat{y}$ and y	$R_{var} = \sigma^2_{\widehat{y}}/\sigma^2$

sociated categorization elements (`fun1, fun2, fun3, parlists`). In this latter case, the function `candidates` is used to construct these matrices. At each iteration, the candidates are evaluated and the best model is selected via the AIC or the BIC. As a result, vectors of coefficients of different lengths (i.e. from different meshes) can be selected at the various iterations. This situation does not pose any difficulty in estimating the predicted values $\widehat{y}$. It is obtained by summing the products $X^u\beta^{u,\gamma}(\lambda)$ of the models retained at each iteration. In order to visualize the identified areas of influence, it is essential to be able to sum the coefficients retained at each iteration. The functions `finemeshed2d` and `finemeshed3d` respectively allow to transform a vector of coefficients (named with combinations of classes) into a 2 and 3 dimensional table with an extremely fine mesh. Once the same fine mesh is used to transform all the coefficient vectors, it is easy to sum up them and to visualize the areas of influence. This process allows to approximate the area of coefficients (continuous) from discrete estimates. All these procedures are already implemented in the function `spicefp`.

The function `coef_spicefp`, provides from the outputs of `spicefp`, the possibility to reconstruct any model evaluated during the approach, regardless of the iteration. The candidate matrices as well as the result of their evaluation, the coefficients retained at each iteration, and the sum of the fine meshes associated with these coefficients are the main outputs of `spicefp`.

Remark : It is possible to obtain other results from the `spicefp` outputs. Instead of successive iterations, we can focus only on the first iteration and make an average of the best models obtained at this iteration. The number of best models to be averaged has to be defined by the user. In the simulations presented in [37], this average of coefficients performs slightly less than the approach itself, but has the advantage of reconstructing the contours of the areas of influence much more faithfully. This model average can be implemented by the function `meancoef` of the package **SpiceFP**.

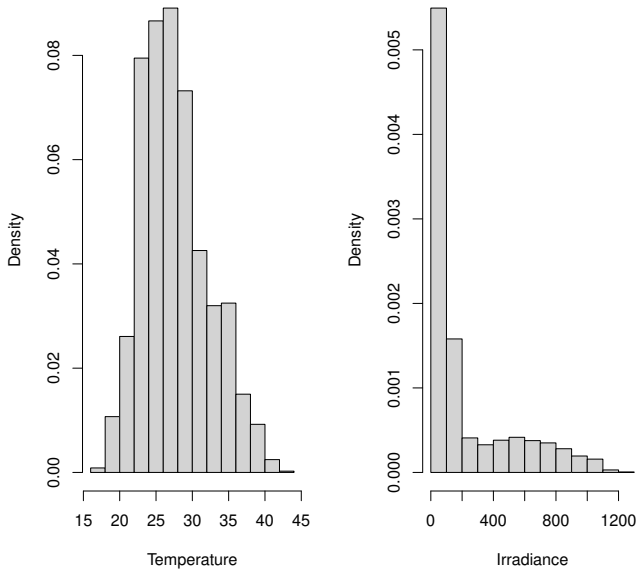


FIGURE 4.4.1 – Distribution of Temperature and Irradiance values according to a linear scale

4.4 Example

In this section, we are interested in exploring the joint influence of two functional variables (`Temperature` and `Irradiance`) on a response variable (`FerariIndex_Difference`) giving information on the quality of the grape berry. The data are available and described in the package **SpiceFP**. The hypothesis of a joint influence is made because both temperature and irradiance are affected by solar radiation. But before exploring this joint influence, let us first look at each of the functional predictors in order to better partition them. The distribution of the Temperature values (see figure 4.4.1) suggests that a linear scale is suitable for its partitioning. If this scale was used for Irradiance, the majority of observations would be distributed in very few classes.

We then propose to use a logarithmic scale to partition the Irradiance values. This scale has the advantage of expanding the low values and compressing the high ones. The function `logbreaks` of **SpiceFP** allows to obtain such results. It is necessary to provide the argument of this function : the parameter α. For a fixed number of breaks, this parameter controls the proportion of breaks chosen from the low values. By visualizing some Irradiance distributions made from a logarithmic scale (figure 4.4.2), the user can get an idea of the values to scan to build the different candidate matrices.

Once all this information is available, we decide to build candidate matrices with the number of Temperature classes (linear scale) varying between 10 and 18 and the number of Irradiance classes (logarithmic scale) between 15 and 25. The values of the parameter `alpha`, necessary for the logarithmic scale will vary between e^{-5} and e^{-1}. At this step, the user can

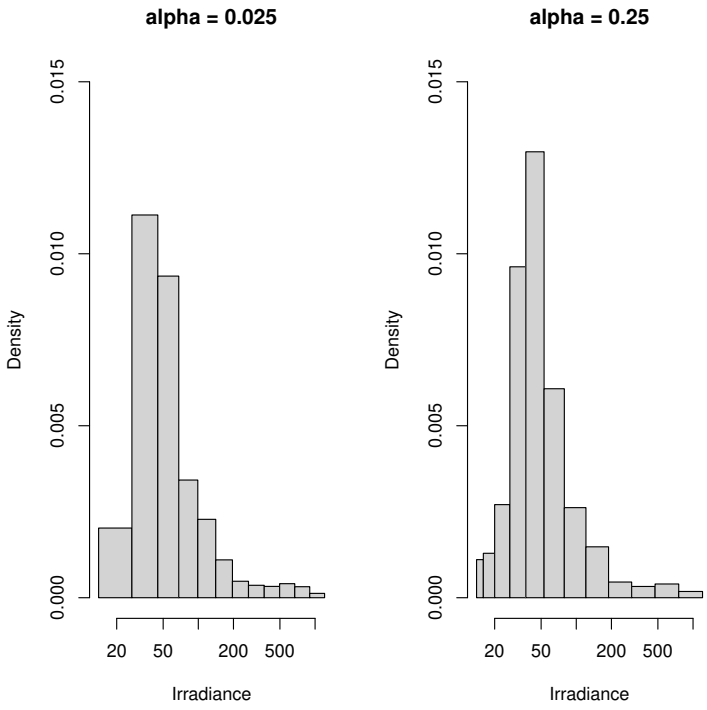


FIGURE 4.4.2 – Distribution of Irradiance values according to a logarithmic scale

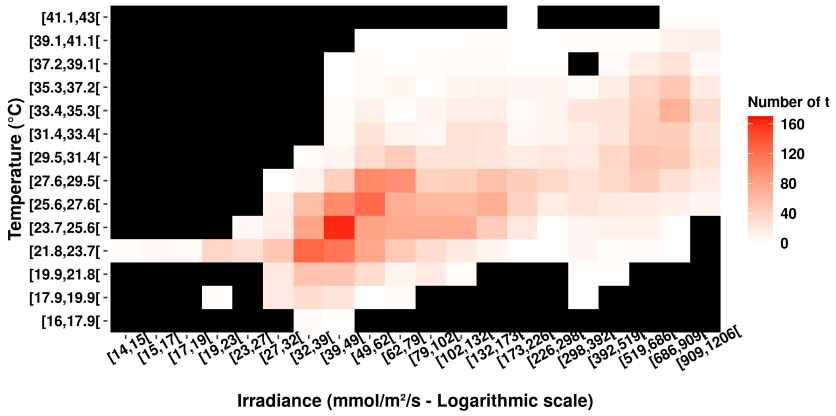


FIGURE 4.4.3 – Candidate matrix with 14 Temperature classes on linear scale and 12 Irradiance classes on log scale. ■ : Joint modalities that have never been observed.

visualize, with the function `hist_2d` of **SpiceFP**, a distribution of the observations in two dimensions (Figure 4.4.3). This provides insight into areas of relative importance, if any.

In this example, the construction of 120 candidate matrices allowed us to roughly browse the value ranges of the three parameters constituting the partition vector u. The user should keep in mind that the execution time of the approach is correlated to the number of candidate matrices to be evaluated, as well as to the complexity of the links between the joint modalities of the candidate matrices. With respect to the γ ratio between parsimony and fusion regulation parameters, 6 ratios were used. The approach can be performed using the following commands :

```
R> ## Data and inputs
R> tpr.nclass=seq(10,18,2)
R> irdc.nclass=seq(15,25,3)
R> irdc.alpha=round(exp(seq(-5,-1,length.out=6 )),4)
R> p2<-expand.grid(tpr.nclass, irdc.alpha, irdc.nclass)
R> parlist.tpr<-split(p2[,1], seq(nrow(p2)))
R> parlist.irdc<-split(p2[,2:3], seq(nrow(p2)))
R> parlist.irdc<-lapply(
+    parlist.irdc,function(x){
+    list(x[[1]],x[[2]])}
+ )
R> start_time_sp <- Sys.time()
R> ex_sp<-spicefp(y = FerariIndex_Difference$fi_dif,
+                 fp1 = m.irdc, fun1 = logbreaks,
+                 fp2 = m.tpr,  fun2 = linbreaks,
+                 parlists = list(parlist.irdc, parlist.tpr),
+                 penratios = c(1/100, 1/10, 1, 10),
+                 K = 2, criterion = "AIC_",
```

```
+                nknots = 100, ncores = 4,
+                dim.finemesh = c(1000, 1000),
+                write.external.file = TRUE)
R> duration_sp <- Sys.time() - start_time_sp
```

About one hour and fifty minutes were needed to perform the calculations (2 iterations) using 4 cores (Intel Core i7-7600U CPU, 2.80GHz × 4). The argument `write.external.file = TRUE` allowed the library to generate as an external file (available in the working directory) the summary table of the model evaluation at each iteration. The final result is shown in figure 4.4.4.

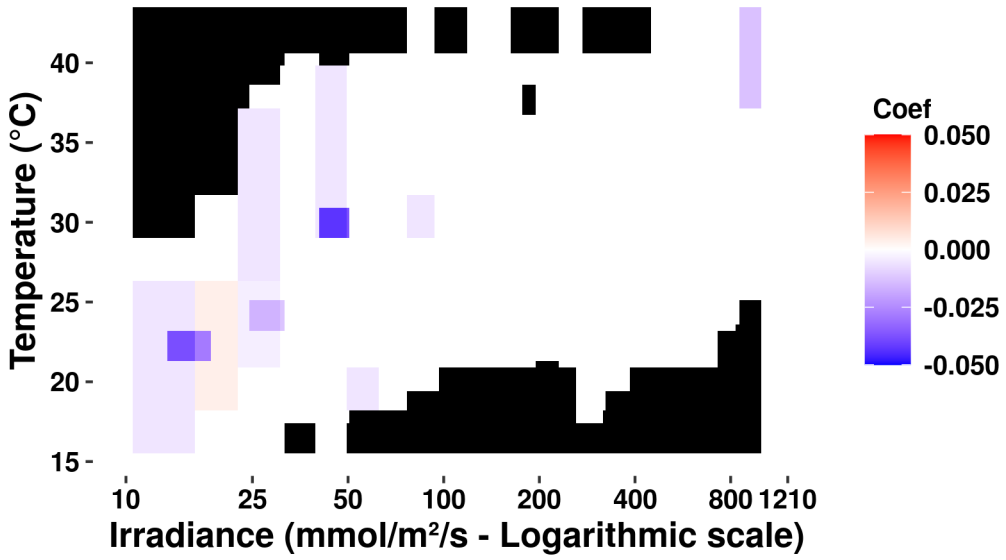


FIGURE 4.4.4 – Visualization of the SpiceFP result (2 iterations, AIC criterion). ■ : Joint modalities that have never been observed.

From the figure 4.4.4, one can observe the joint modalities for which no observation is made (value :`NA`, color black). This makes it possible to differentiate between unobserved and observed but non-influential areas. As an interpretation, we can retain that for low irradiance values ($< 100\ \mu mol.m^{-2}.s^{-1}$), there is a gradient according to temperature which is not suitable for an increase of the Ferari index. All the information on the models retained in the two iterations are accessible via the commands `ex_sp$Evaluations[[1]]` and `ex_sp$Evaluations[[2]]`. Their characteristics are as follows :

```
R> # Itération 1
R> t(ex_sp$Evaluations[[1]]$thecandidate.parameters)
```

```
     Candidate_id Pen_ratio PenPar_fusion Df_ RSS_     AIC_     BIC_
[1,] 108          10        0.1648765     3   13.53096 214.5332 218.9305
```

```
     AICc_    Cp_      GCV_      Slope_    Var_ratio
[1,] 56.41074 13.86642 0.5148522 0.5650574 0.327378
```

```
R> # Itération 2
R> t(ex_sp$Evaluations[[2]]$thecandidate.parameters)
```

```
     Candidate_id Pen_ratio PenPar_fusion Df_ RSS_     AIC_     BIC_
[1,] 101          0.1       0.3617341     4   13.35619 668.4776 674.3406
     AICc_    Cp_      GCV_      Slope_    Var_ratio
[1,] 53.37039 13.50262 0.5451506 0.2108834 0.6919936
```

An estimate of the goodness of fit of the final estimate can be obtained : *i)* statistically by computing the correlation between the response variable y and the predictions $\widehat{y}$,

```
R> xbeta <- c(ex_sp$Evaluations[[1]]$XBeta + ex_sp$Evaluations[[2]]$XBeta)
R> cor(FerariIndex_Difference$fi_dif , xbeta, method = "pearson")
```

```
[1] 0.8884187
```

ii) or graphically (figure 4.4.5) by representing y and $\widehat{y}$ in the same scatterplot.

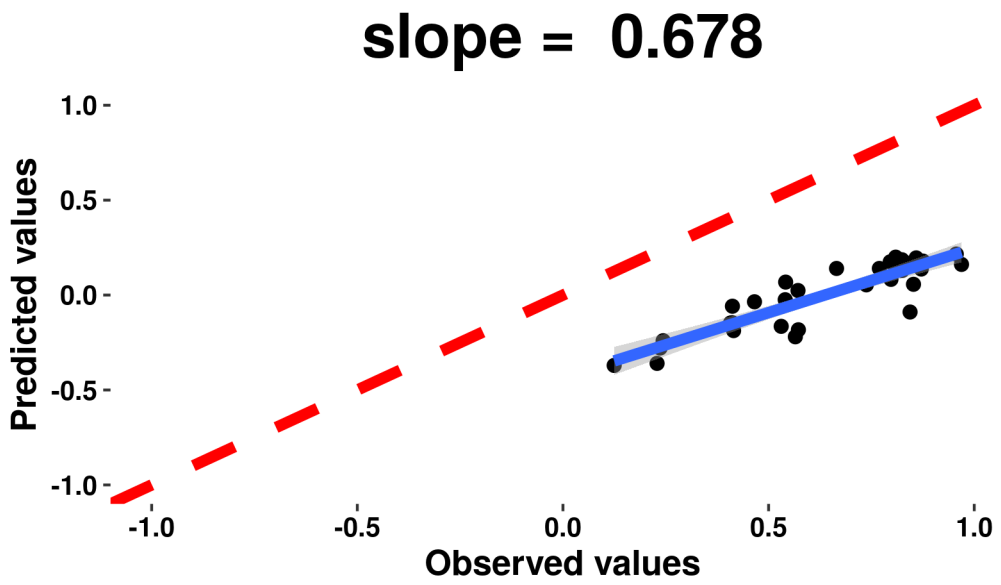


FIGURE 4.4.5 – Quality of the SPICEFP estimation

Let's take a closer look at the top 10 models among those selected by the AIC in order to make an average.

```
R> # Models at iteration 1
R> models_iter1 <- read.table(ex_sp$Evaluations[[1]]$
                      Evaluation.results$evaluation.result, header = TRUE)
R> dim(models_iter1)
```

```
[1] 48000    12

R> OrderbyAIC_iter1 <- models_iter1[order(models_iter1$AIC_) ,]
R> head(OrderbyAIC_iter1, n = 10)

      Candidate_id Pen_ratio PenPar_fusion Df_     RSS_     AIC_     BIC_
43122          108     10.00     0.1648765   3 13.53096 214.5332 218.9305
43121          108     10.00     0.1846918   3 13.54702 214.8206 219.2178
43120          108     10.00     0.2068887   3 13.56718 215.1811 219.5783
43119          108     10.00     0.2317533   3 13.59248 215.6336 220.0308
43009          108      1.00     0.7389190   3 13.59921 215.7539 220.1511
43118          108     10.00     0.2596061   3 13.62422 216.2013 220.5985
40725          103     10.00     0.1380610   4 13.51566 216.2597 222.1226
15526           38     10.00     0.1519366   3 13.63613 216.4143 220.8115
8851            24      0.01     0.9533787   3 13.64396 216.5543 220.9515
15525           38     10.00     0.1697173   3 13.64786 216.6240 221.0212
         AICc_      Cp_      GCV_    Slope_ Var_ratio
43122 56.41074 13.86642 0.5148522 0.5650574 0.3273780
43121 56.44872 13.88249 0.5154635 0.5477581 0.3366471
43120 56.49630 13.90265 0.5162305 0.5283798 0.3482779
43119 56.55591 13.92795 0.5171930 0.5066726 0.3628724
43009 56.57175 13.93467 0.5174491 0.4948788 0.3667548
43118 56.63055 13.95969 0.5184008 0.4823565 0.3811858
40725 53.75021 13.96295 0.5516597 0.5671179 0.3185532
15526 56.65851 13.97160 0.5188539 0.5229218 0.3880567
8851  56.67687 13.97942 0.5191518 0.5076475 0.3925728
15525 56.68601 13.98332 0.5193001 0.5093079 0.3948225
```

The reconstruction of the models is done through the function `coef_spicefp` and the visualization of their mean is presented in figure 4.4.6.

```
R> # Estimation of coefficients
R> AICtopten_iter1 <- coef_spicefp( spicefp.result = ex_sp,
+                                   iter_ = 1,
+                                   criterion = "AIC_",
+                                   nmodels = 10,
+                                   model.parameters = NULL,
+                                   dim.finemesh = c(1000, 1000),
+                                   ncores = 4,
+                                   write.external.file = TRUE )
R> # Compute the mean of the coefficients
R> mean_AICtopten_iter1 <- meancoef(coef.list = AICtopten_iter1$coef.list ,
+                                   weight = rep(1,10))
```

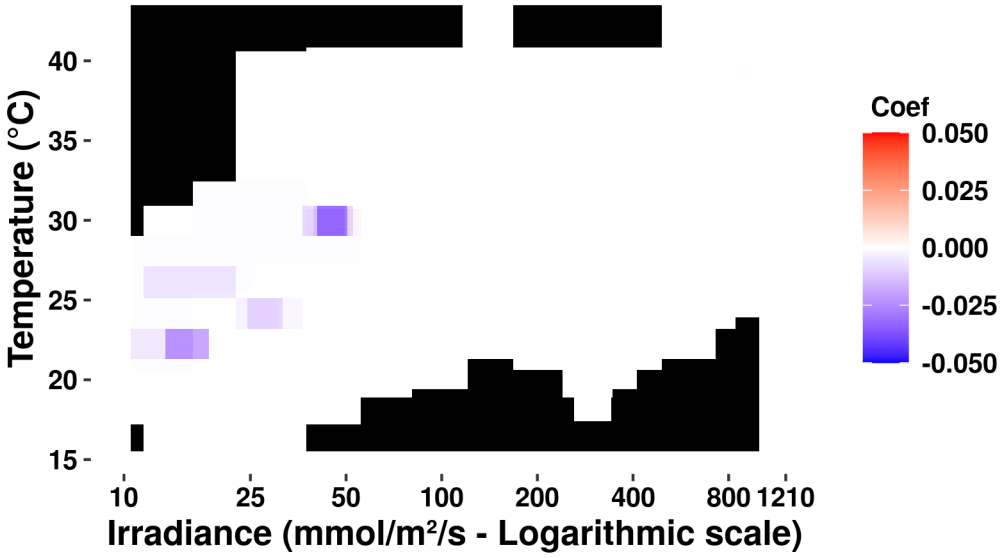


FIGURE 4.4.6 – Visualization of coefficient mean of the 10 best models selected by the AIC at iteration 1. ■ : Joint modalities that have never been observed.

The goodness of fit of the model associated with the mean of these coefficients can also be obtained by computing the correlation between y and $\widehat{y}$.

```
R> cor(FerariIndex_Difference$fi_dif , mean_AICtopten_iter1$y.estimated,
+     method = "pearson")

[1] 0.8312705
```

4.5 Conclusion

The **SpiceFP** library offers various functions allowing the implementation of the exploratory approach of the same name. Although exploratory, this approach requires the use of inferential statistics tools. The functional model that supports this approach takes functional observations as predictors and a scalar variable as a response. This falls within the framework of "scalar-on-function" functional models. It implies a joint influence of the predictors and is designed for exploration in this direction. Thus, it allows the identification of areas of influence (derived from combinations of classes of predictors).

Acknowledgments

This work was supported by the French National Research Agency under the Investments for the Future Program, referred as ANR-16-CONV-0004. The data used were acquired

during the Innovine project, funded by the Seventh Framework Programme of the European Community (FP7/2007-2013), under Grant Agreement No. FP7-311775.

Chapitre 5

Conclusion et perspectives

Les travaux de cette thèse ont permis de répondre à des questions soulevées par l'essor de l'agriculture numérique. Nous avons développé des outils méthodologiques et logiciels permettant d'extraire de l'information de séries climatiques pour améliorer la connaissance de processus biologiques (maturation de baies de raisin). Cela a entraîné la résolution d'un problème de type scalar-on-functions (2 ou 3 variables fonctionnelles explicatives) tout en privilégiant l'obtention de résultats interprétables dans un contexte d'influence conjointe de prédicteurs fonctionnels. L'outil développé est dénommé "Sparse and Structured Procedure to Identify Combined Effects of Functional Predictors" (SpiceFP). Il a été inspiré par les contraintes liées à l'extraction de connaissances d'un jeu de données fonctionnelles issu d'une expérimentation viticole. Ces données ont servi de cas d'étude tout au long des différents chapitres de cette thèse. Le problème que posait ce jeu de données n'était pas un problème de type scalar-on-functions mais plutôt function-on-functions car tant les variables explicatives que la variable réponse étaient des courbes observées respectivement toutes les douze minutes et quasiment chaque semaine. L'application agronomique pouvait se simplifier par l'étude d'une variable réponse scalaire, résultant de la quantité hebdomadaire d'anthocyanes accumulée dans la baie de raisin, étudiée à travers la mesure de l'Indice de Ferari (FI). Cet état de choses nous incite donc à catégoriser les perspectives de cette thèse en deux groupes : celles induites par les contraintes de l'application et celles associées à l'approche proposée.

Les contraintes associées à l'analyse du jeu de données utilisé dans l'application peuvent se résumer comme suit :

1. *une très grande différence d'échelle d'observation entre les prédicteurs fonctionnels et les courbes de la variable réponse :* environ une semaine pour l'Indice de Ferari contre 12 minutes pour la température et l'irradiance. Au bout de huit semaines de collecte de données, les courbes de FI sont constituées de neuf points illustrant une cinétique d'accumulation d'anthocyanes tandis que celles de température et d'irradiance sont constituées de plus de six milles points illustrant des variations intra et inter-journalières. Au lieu de travailler sur des différences de FI, une perspective aurait été de générer de nouveaux points de FI (entre 20 et 40 points) à partir de ceux existant pour chaque individu statistique afin de pouvoir envisager l'utilisation des méthodes fonctionnelles

d'analyse de données de type function-on-functions. Nous n'avons pas utilisé une telle approche parce qu'elle induisait une interpolation qui aurait été une approximation trop grossière.

2. *les résultats issus de l'analyse devraient être interprétables c'est-à-dire fournir des connaissances agronomiques qui pourraient servir de leviers à l'amélioration de la qualité à la vendange :* l'interprétabilité des résultats est un reproche habituellement fait aux modèles fonctionnels mais il existe des modèles fonctionnels interprétables comme celui de [42]. Mais généralement dans ces modèles, les coefficients sont dépendants du temps et leur interprétation permet d'identifier des périodes de temps ou des dates ayant une influence. Or la question d'interprétabilité associée à notre jeu de données réside dans l'identification de conditions de température et d'irradiance influentes, c'est à dire des coefficients dépendant de "modalités" des variables explicatives. Nous avons opté pour l'utilisation d'un tableau de contingence qui peut se représenter sous forme d'histogramme bivarié. Une autre approche aurait été de s'intéresser à ce que pourrait apporter, dans une telle modélisation, l'utilisation de densités de copules. Nous n'avons pas opté pour ce choix parce les variables de température et d'irradiance ont des distributions très différentes qui complexifient l'approche par copules. De plus, l'aspect interprétable et le mélange entre variables catégorielles, discrètes et continues ne semblent pas aussi immédiats.
3. *les courbes de l'Indice de Ferari auraient pu bénéficier d'un recalage en fonction de l'état de maturité des baies. De même, les données du microclimat sont décalées à l'échelle d'une journée selon l'orientation (dispositif expérimental, est-ouest)* : une possibilité serait de s'intéresser à l'apport des techniques de déformation temporelle dynamique (Dynamic Time Warping) pour le recalage des courbes avant leur analyse statistique. Pour le microclimat, cela n'est pas forcément pertinent car les processus biologiques de l'après-midi ne sont pas forcément ceux du matin et ont potentiellement des sensibilités différentes au climat. Pour les courbes d'indice de Ferari, nous disposions de trop peu d'observations et de connaissances. D'un point de vue agronomique, ce recalage se fait à partir du taux de sucre dans la baie [95]. Cette mesure étant destructrice, nous n'y avons pas eu accès pendant la dynamique.

La principale contribution de l'approche SpiceFP réside dans l'utilisation d'une collection de bases de fonctions indicatrices afin d'identifier convenablement un partitionnement du domaine des variables fonctionnelles explicatives, mais aussi une ou plusieurs zones d'influence à l'intérieur de cette partition. L'utilisation de telles bases, qui facilitent l'interprétation du résultat, impose toutefois d'obtenir une discrétisation sans données manquantes des variables fonctionnelles explicatives sur le même ensemble de points d'observation équidistants T. Cette condition permet d'avoir une bonne prise en compte de tout le domaine. Si les prédicteurs ne sont pas observés suivant un tel design, un pré-traitement par le biais d'une interpolation ou encore l'utilisation des techniques de lissage / ré-échantillonnages [88] permettra d'obtenir un nombre voulu de points de discrétisation équidistants. En cas de données manquantes dans l'observation des courbes, les diverses techniques d'imputation disponibles permettront d'imputer le jeu de données. Dans le cadre de notre application,

nous nous sommes servis de la méthode d'imputation non paramétrique MissForest [109] via son implémentation sous le logiciel **R**. Toutefois, il est essentiel de préciser que fournir une discrétisation telle que précisée des courbes explicatives avant la création des tableaux de contingence ne suffit pas. Il faut en effet un jeu de données échantillonnées avec une fréquence suffisante. L'approche proposée est sensible au design des tableaux de contingence et plus précisément à la répartition des comptages sur le domaine d'étude. Les données d'expérimentation d'Innovine utilisées dans l'application étaient réparties de façon déséquilibrées (cf. figure 3.3.4). L'approche SpiceFP n'a pu identifier des effets significatifs que dans les zones bien observées. En conséquence, il devient essentiel donc d'avoir suffisamment de points d'observation à l'échelle d'un "phénomène" ou d'une portion du domaine des variables explicatives pour qu'il ressorte dans les analyses.

Une des perspectives de SpiceFP est d'ordre méthodologique et s'articule autour de la sélection de modèles. Les critères d'information utilisés représentent un bon compromis entre estimation convenable du degré de liberté et surestimation de la variance. Un autre critère utilisé dans les applications de type scalar on image est la variation totale [121]. Ce critère n'a pas été retenu tel quel car il ne propose pas le côté parcimonieux, important dans notre approche, pour permettre l'interprétation des résultats. Dans l'approche SpiceFP, l'utilisateur peut faire varier l'importance entre les deux pénalités (fusion et parcimonie). Une autre perspective irait dans le sens de la conception et de l'implémentation de nouvelles fonctions de création des limites de classes (breaks) mais aussi dans la définition des contraintes de contiguïté (terme de fusion) qui interviennent dans les matrices de pénalité. Cela offrirait plus de liberté dans la modélisation (choix de structures et types de liens).

Deux extensions de l'approche ont été proposées : la première étant une extension de l'approche à la troisième dimension et la deuxième permettant d'itérer SpiceFP 2D ou SpiceFP 3D de manière à construire un résultat final constitué de plusieurs modèles. Cette dernière extension part du postulat qu'en présence de zones d'influence complexes, non aisément reproductibles par des rectangles dans le tableau de contingence des variables explicatives, les résidus d'un premier modèle pourraient toujours contenir de l'information (plus que du bruit). D'où l'idée d'aller à une deuxième itération en utilisant comme variable réponse cette fois ci, les résidus d'un premier modèle. Une telle initiative, bien qu'utile pour certaines données (cas de la simulation 2, chapitre 2), n'est pas sans risque. En guise de perspective, il serait intéressant de construire un indicateur facilitant le choix du nombre d'itérations.

SpiceFP est une nouvelle contribution de type scalar-on-function aux outils d'analyse des données fonctionnelles. Il prend en compte les interactions et retourne un résultat interprétable. Plus spécifiquement, il pourrait être d'une grande aide dans le domaine de l'agriculture numérique où l'on a fréquemment besoin de comprendre des interactions complexes impliquant des observations en grande dimension.

Bibliographie

[1] G. Agati, S. Meyer, P. Matteini, and Z. G. Cerovic. Assessment of anthocyanins in grape (vitis vinifera l.) berries using a noninvasive chlorophyll fluorescence method. *Journal of agricultural and food chemistry*, 55(4) :1053–1061, 2007.

[2] H. Akaike. *Information Theory and an Extension of the Maximum Likelihood Principle*, pages 199–213. Springer New York, New York, NY, 1973.

[3] T. B. Arnold and R. J. Tibshirani. Efficient implementations of the generalized lasso dual path algorithm. *Journal of Computational and Graphical Statistics*, 25(1) :1–27, 2016.

[4] T. B. Arnold and R. J. Tibshirani. *genlasso : Path algorithm for generalized lasso problems*, 2019. Package version 1.4 for R version 3.6.1.

[5] T. B. Arnold and R. J. Tibshirani. *genlasso : Path Algorithm for Generalized Lasso Problems*, 2020. R package version 1.5.

[6] A. Azuma, H. Yakushiji, Y. Koshita, and S. Kobayashi. Flavonoid biosynthesis-related genes in grape skin are differentially regulated by temperature and light conditions. *Planta*, 236 :1067–80, 05 2012.

[7] A. Baíllo, A. Cuevas, and F. Ricardo. Classification methods for functional data. *The Oxford Handbook of Functional Data Analysis*, pages 259–297, 01 2011.

[8] A. Belloni, V. Chernozhukov, and L. Wang. Square-root lasso : pivotal recovery of sparse signals via conic programming. *Biometrika*, 98(4) :791–806, 2011.

[9] N. Ben Ghozlen, Z. Cerovic, C. Germain, S. Toutain, and G. Latouche. Non-destructive optical monitoring of grape maturation by proximal sensing. *Sensors (Basel, Switzerland)*, 10 :10040–68, 11 2010.

[10] R. Bendel and A. Afifi. Comparison of stopping rules in forward "stepwise" regression. *Journal of the American Statistical Association*, 72 :46–53, 1977.

[11] J. Bergqvist, N. Dokoozlian, and N. Ebisuda. Sunlight exposure and temperature effects on berry growth and composition of cabernet sauvignon and grenache in the

central san joaquin valley of california. *American Journal of Enology and Viticulture*, 52(1) :1–7, 2001.

[12] P. Boss, C. Davies, and S. P. Robinson. Anthocyanin composition and anthocyanin pathway gene expression in grapevine sports differing in berry skin colour. *Australian Journal of Grape and Wine Research*, 2(3) :163–170, 1996.

[13] R. Bramley, M. Le Moigne, S. Evain, J. Ouzman, L. Florin, E. Fadaili, C. Hinze, and Z. Cerovic. On-the-go sensing of grape berry anthocyanins during commercial harvest : development and prospects. *Australian Journal of Grape and Wine Research*, 17(3) :316–326, 2011.

[14] L. Breiman. Random forests. *Machine Learning*, 45(1) :5–32, 2001.

[15] S. Brockhaus, M. Melcher, F. Leisch, and S. Greven. Boosting flexible functional regression models with a high number of functional historical effects. *Statistics and Computing*, 27(4) :913–926, July 2017.

[16] S. Brockhaus, D. Rügamer, and S. Greven. Boosting functional regression models with fdboost. *Journal of Statistical Software, Articles*, 94(10) :1–50, 2020.

[17] H. Cardot, F. Ferraty, and P. Sarda. Functional linear model. *Statistics and Probability Letters*, 45 :11–22, 02 1999.

[18] H. Cardot, F. Ferraty, and P. Sarda. Spline estimators for the functional linear model. *Statistica Sinica*, 13 :571–591, 07 2003.

[19] Z. Cerovic, N. Moise, G. Agati, G. Latouche, N. Ben Ghozlen, and S. Meyer. New portable optical sensors for the assessment of winegrape phenolic maturity based on berry fluorescence. *Journal of Food Composition and Analysis*, 21 :650–654, 12 2008.

[20] V. Cheynier. La couleur des vins rouges. *Wine Internet Technical Journal*, 2003.

[21] S. Cohen, J. Tarara, and J. Kennedy. Assessing the impact of temperature on grape phenolic metabolism. *Analytica chimica acta*, 621 :57–67, 08 2008.

[22] C. Crambes, A. Gannoun, and Y. Henchiri. Support vector machine quantile regression approach for functional data : Simulation and application studies. *Journal of Multivariate Analysis*, 121 :50 – 68, 2013.

[23] P. Craven and G. Wahba. Smoothing noisy data with spline functions : Estimating the correct degree of smoothing by the method of generalized cross-validation. *Numer. Math.*, 31 :377–403, 1979.

[24] B. Cuer. Étude de l'impact de la température sur la maturité du raisin pour la filière vigne et vin. Technical report, Rapport de stage de Master 2 Biostatistique. Université de Montpellier, 2017.

[25] A. Delaigle and P. Hall. Methodology and theory for partial least squares applied to functional data. *Ann. Statist.*, 40(1) :322–352, 02 2012.

[26] M. Downey, N. Dokoozlian, and M. Krstic. Cultural practice and environmental impacts on the flavonoid composition of grapes and wine : A review of recent research. *American Journal of Enology and Viticulture*, 57 :257–268, 09 2006.

[27] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Ann. Statist.*, 32(2) :407–499, 04 2004.

[28] A. Fernandes de Oliveira, L. Mercenaro, A. Del Caro, P. Luca, and G. Nieddu. Distinctive anthocyanin accumulation responses to temperature and natural uv radiation of two field-grown vitis vinifera l. cultivars. *Molecules*, 20, 01 2015.

[29] F. Ferraty and P. Vieu. *Nonparametric Functional Data Analysis : Theory and Practice (Springer Series in Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[30] F. Ferraty and P. Vieu. Richesse et complexité des données fonctionnelles. *Revue MODULAD*, 43 :25–43, 2011.

[31] P. Filzmoser, M. Gschwandtner, and V. Todorov. Review of sparse methods in regression and classification with application to chemometrics. *Journal of Chemometrics*, 26(3-4) :42–51, 2012.

[32] J. H. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software, Articles*, 33(1) :1–22, 2010.

[33] M. G. Garcia, M. C. Medeiros, and G. F. Vasconcelos. Real-time inflation forecasting with high-dimensional models : The case of brazil. *International Journal of Forecasting*, 33(3) :679 – 693, 2017.

[34] J. Gertheiss, A. Maity, and A.-M. Staicu. Variable selection in generalized functional linear models. *Stat*, 2(1) :86–101, 2013.

[35] C. Giacobino, S. Sardy, J. Diaz-Rodriguez, and N. Hengartner. Quantile universal threshold. *Electronic Journal of Statistics*, 11 :4701–4722, 01 2017.

[36] C. Giraud. *Introduction to High-Dimensional Statistics*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis, 2014.

[37] G. Gnanguenon Guesse, P. Loisel, B. Fontez, T. Simonneau, and N. Hilgert. Identification of combined effects of functional variables using contingency tables with ordered categories - application to agri-environmental issues. *to appear*, XX, 202X.

[38] J. Goldsmith, J. Bobb, C. M. Crainiceanu, B. Caffo, and D. Reich. Penalized functional regression. *Journal of Computational and Graphical Statistics*, 20(4) :830–851, 2011. PMID : 22368438.

[39] J. Goldsmith, L. Huang, and C. M. Crainiceanu. Smooth scalar-on-image regression via spatial bayesian variable selection. *Journal of Computational and Graphical Statistics*, 23(1) :46–64, 2014. PMID : 24729670.

[40] B. Gregorutti, B. Michel, and P. Saint-Pierre. Grouped variable importance with random forests and application to multiple functional data analysis. *Computational Statistics and Data Analysis*, 90 :15 – 35, 2015.

[41] P.-M. Grollemund. *Régression linéaire bayésienne sur données fonctionnelles*. Theses, Université de Montpellier, Nov. 2017.

[42] P.-M. Grollemund, C. Abraham, M. Baragatti, and P. Pudlo. Bayesian functional linear regression with sparse step functions. *Bayesian Analysis*, 14(1) :111 – 135, 2019.

[43] S. Guillas and M.-J. Lai. Bivariate splines for spatial functional regression models. *Journal of Nonparametric Statistics*, 22(4) :477–497, 2010.

[44] N. E. Haouij, J. Poggi, R. Ghozi, S. Sevestre, and M. Jaïdane. Random forest-based approach for physiological functional variable selection for driver's stress level classification. *Statistical Methods and Applications*, 28 :157–185, 2019.

[45] M. Hebiri. *Quelques questions de sélection de variables autour de l'estimateur LASSO*. Theses, Université Paris-Diderot - Paris VII, June 2009.

[46] K. Hirose, S. Tateishi, and S. Konishi. Tuning parameter selection in sparse regression modeling. *Computational Statistics & Data Analysis*, 59 :28 – 40, 2013.

[47] A. E. Hoerl and R. W. Kennard. Ridge regression : Biased estimation for nonorthogonal problems. *Technometrics*, 12(1) :55–67, 1970.

[48] R. Jackson. *Wine Science : Principles and Applications*. ISSN. Elsevier Science, 2014.

[49] J. Jacques and C. Preda. Functional data clustering : a survey. *Advances in Data Analysis and Classification*, 8(3) :231–255, 2014.

[50] G. M. James. Generalized linear models with functional predictors. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 64(3) :411–432, 2002.

[51] G. M. James and B. W. Silverman. Functional adaptive model estimation. *Journal of the American Statistical Association*, 100(470) :565–576, 2005.

[52] G. M. James, J. Wang, and J. Zhu. Functional linear regression that's interpretable. *The Annals of Statistics*, 37(5A) :2083 – 2108, 2009.

[53] G. M. James, J. Wang, and J. Zhu. Functional linear regression that's interpretable. *Ann. Statist.*, 37(5A) :2083–2108, 10 2009.

[54] P. Jones and P. Thornton. The potential impacts of climate change on maize production in Africa and Latin America in 2055. *Global Environmental Change*, 13(1) :51–59, Apr. 2003.

[55] J. Josse and F. Husson. missmda : A package for handling missing values in multivariate data analysis. *Journal of Statistical Software, Articles*, 70(1) :1–31, 2016.

[56] J. Jouan. Les AOC viticoles face au changement climatique : exploration des voies d'adaptation par la prospective et l'analyse économique. Master, AGROCAMPUS OUEST, FRA., 2014.

[57] J. Kang, B. J. Reich, and A.-M. Staicu. Scalar-on-image regression via the soft-thresholded Gaussian process. *Biometrika*, 105(1) :165–184, 01 2018.

[58] P. Kokoszka and M. Reimherr. *Introduction to Functional Data Analysis*. Chapman and Hall/CRC, 09 2017.

[59] C. Laurin, D. Boomsma, and G. Lubke. The use of vector bootstrapping to improve variable selection precision in lasso models. *Statistical Applications in Genetics and Molecular Biology*, 15(4) :305–320, Aug. 2016.

[60] A.-L. Lereboullet, G. Beltrando, and D. K. Bardsley. Socio-ecological adaptation to climate change : A comparative case study from the mediterranean wine industry in france and australia. *Agriculture, Ecosystems & Environment*, 164 :273 – 285, 2013.

[61] F. Li, T. Zhang, Q. Wang, M. Z. Gonzalez, E. L. Maresh, and J. A. Coan. Spatial bayesian variable selection and grouping for high-dimensional scalar-on-image regression. *Ann. Appl. Stat.*, 9(2) :687–713, 06 2015.

[62] Y. Li, H. Sun, X. Deng, C. Zhang, H.-P. B. Wang, and R. Jin. Manufacturing quality prediction using smooth spatial variable selection estimator with applications in aerosol jet® printed electronics manufacturing. *IISE Transactions*, 52(3) :321–333, 2020.

[63] H. Lian. Shrinkage estimation and selection for multiple functional regression. *Statistica Sinica*, 23(1) :51–74, 2013.

[64] F. Ludwig, S. Milroy, and S. Asseng. Impacts of recent climate change on wheat production systems in western australia. *Climatic Change*, 92(3-4) :495–517, 2009. Online first.

[65] R. Luo and X. Qi. Function-on-function linear regression by signal compression. *Journal of the American Statistical Association*, 112(518) :690–705, 2017.

[66] W. Ma, L. Xiao, B. Liu, and M. A. Lindquist. A functional mixed model for scalar on function regression with application to a functional MRI study. *Biostatistics*, 10 2019. kxz046.

[67] J. Mairal and B. Yu. Complexity analysis of the lasso regularization path. In J. Langford and J. Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, ICML '12, pages 353–360, New York, NY, USA, July 2012. Omnipress.

[68] C. L. Mallows. Some comments on cp. *Technometrics*, 15(4) :661–675, 1973.

[69] B. D. Marx and P. H. C. Eilers. Generalized linear regression on sampled signals and curves : A p-spline approach. *Technometrics*, 41(1) :1–13, 1999.

[70] B. D. Marx and P. H. C. Eilers. Multidimensional penalized signal regression. *Technometrics*, 47(1) :13–22, 2005.

[71] A. Matese and S. Di Gennaro. Technology in precision viticulture : A state of the art review. *International Journal of Wine Research*, 7 :69–81, 05 2015.

[72] Microsoft-Corporation and S. Weston. *doParallel : Foreach Parallel Adaptor for the 'parallel' Package*, 2019. R package version 1.0.15.

[73] K. Mori, N. Goto-Yamamoto, M. Kitayama, and K. Hashizume. Loss of anthocyanins in red-wine grape under high temperature. *Journal of experimental botany*, 58 :1935–45, 02 2007.

[74] K. Mori, S. Sugaya, and H. Gemma. Decreased anthocyanin biosynthesis in grape berries grown under elevated night temperature condition. *Scientia Horticulturae - Sci Hort Amsterdam*, 105 :319–330, 07 2005.

[75] J. S. Morris. Functional regression. *Annual Review of Statistics and Its Application*, 2(1) :321–359, 2015.

[76] N. Movahed, C. Pastore, A. Cellini, G. Allegro, G. Valentini, S. Zenoni, E. Cavallini, E. D'Incà, G. b. Tornielli, and I. Filippetti. The grapevine vviprx31 peroxidase as a candidate gene involved in anthocyanin degradation in ripening berries under high temperature. *Journal of Plant Research*, 129, 01 2016.

[77] M. R. Mozell and L. Thach. The impact of climate change on the global wine industry : Challenges & solutions. *Wine Economics and Policy*, 3(2) :81 – 89, 2014.

[78] A. Möller, G. Tutz, and J. Gertheiss. Random forests for functional covariates. *Journal of Chemometrics*, 30(12) :715–725, 2016.

[79] T. G. O'Connor. Acacia karroo invasion of grassland : Environmental and biotic effects influencing seedling emergence and establishment. *Oecologia*, 103(2) :214–223, 1995.

[80] OIV. *Recueil international des méthodes d'analyses. Caractéristiques Chromatiques - Méthode OIV-MA-AS2-11*, 2006.

[81] N. Ollat and J.-M. Touzard. Etude des impacts à long terme du changement climatique et de l'adaptation de la filière viti-vinicole française : projet Laccave. In *11. Journée technique du CIVB : S'adapter aux défis de demain et garantir la qualité de nos vins, de la pratique à l'innovation*, 11. Journée technique du CIVB. S'adapter aux défis de demain et garantir la qualité de nos vins, de la pratique à l'innovation : les actes du colloque, page 160 p., Bordeaux, France, Feb. 2013. Conseil interprofessionnel des Vins de Bordeaux (CIVB). Bordeaux, FRA.

[82] T. Park and G. Casella. The bayesian lasso. *Journal of the American Statistical Association*, 103(482) :681–686, 2008.

[83] P. Pérez-García, Y. Ma, M. J. Yanovsky, and P. Mas. Time-dependent sequestration of rve8 by lnk proteins shapes the diurnal oscillation of anthocyanin biosynthesis. *Proceedings of the National Academy of Sciences*, 112(16) :5249–5253, 2015.

[84] J. Pillet. *Impact du microclimat sur le métabolisme de la baie de raisin*. Theses, Université de Bordeaux Ségalen (Bordeaux 2), Dec. 2011.

[85] R. Plant. *Spatial Data Analysis in Ecology and Agriculture Using R*. Taylor & Francis, 2012.

[86] X. Qiao, S. Guo, and G. M. James. Functional graphical models. *Journal of the American Statistical Association*, 114(525) :211–222, 2019.

[87] R. Rahman, S. Dhruba, S. Ghosh, and R. Pal. Functional random forests with applications in dose response predictions. *Scientific reports*, Feb. 2019.

[88] J. Ramsay, G. Hooker, and S. Graves. *Functional Data Analysis with R and MATLAB*. Use R ! Springer New York, 2009.

[89] J. Ramsay and B. Silverman. *Functional Data Analysis*. Springer Series in Statistics. Springer, 2005.

[90] P. T. Reiss, J. Goldsmith, H. L. Shang, and R. T. Ogden. Methods for scalar-on-function regression. *International Statistical Review*, 85(2) :228–249, 2017.

[91] P. T. Reiss, L. Huang, and M. Mennes. Fast function-on-scalar regression with penalized basis expansions. *The international journal of biostatistics*, 6(1) :Article 28, 2010.

[92] P. T. Reiss and R. T. Ogden. Functional generalized linear models with images as predictors. *Biometrics*, 66(1) :61–69, 2010.

[93] M. Rienth, L. Torregrosa, M. T. Kelly, N. Luchaire, A. Pellegrino, J. Grimplet, and C. Romieu. Is transcriptomic regulation of berry development more important at night than during the day ? *PLoS ONE*, 9(2) :np, 2014.

[94] M. Rienth, L. Torregrosa, N. Luchaire, R. Chatbanyong, D. Lecourieux, M. T. Kelly, and C. Romieu. Day and night heat stress trigger different transcriptomic responses in green and ripening grapevine (vitis vinifera) fruit. *BMC Plant Biology*, 14 :108, 2014.

[95] M. Rienth, L. Torregrosa, G. Sarah, M. Ardisson, J.-M. Brillouet, and C. Romieu. Temperature desynchronizes sugar and organic acid metabolism in ripening grapevine fruits and remodels their transcriptome. *BMC Plant Biology*, 16, 07 2016.

[96] F. Rossi, N. Delannay, B. Conan-Guez, and M. Verleysen. Representation of functional data in neural networks. *Neurocomputing*, 64 :183 – 210, 2005. Trends in Neurocomputing : 12th European Symposium on Artificial Neural Networks 2004.

[97] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D : Nonlinear Phenomena*, 60(1) :259–268, 1992.

[98] V. Sadras, M. Moran, and M. Bonada. Effects of elevated temperature in grapevine. i berry sensory traits. *Australian Journal of Grape and Wine Research*, 19(1) :95–106, 2013.

[99] R. Salminen, P. Hari, S. Kellomaki, E. Korpilahti, M. Kotiranta, and R. Sievanen. A measuring system for estimating the frequency distribution of irradiance within plant canopies. *Journal of Applied Ecology*, 20(3) :887–895, 1983.

[100] S. A. Saseendran, K. K. Singh, L. S. Rathore, S. Singh, and S. K. Sinha. Effects of climate change on rice production in the tropical humid climate of kerala, india. *Climatic Change*, 44 :495–514, 2000.

[101] F. Scheipl, A.-M. Staicu, and S. Greven. Functional additive mixed models. *Journal of Computational and Graphical Statistics*, 24(2) :477–501, 2015.

[102] G. Schwarz. Estimating the dimension of a model. *Ann. Statist.*, 6(2) :461–464, 03 1978.

[103] B. Seguin. Le changement climatique : conséquences pour les végétaux. *Quaderni*, 71 :27–40, 2010.

[104] H. L. Shang. A survey of functional principal component analysis. *AStA Advances in Statistical Analysis*, 98, 04 2011.

[105] B. W. Silverman. Smoothed functional principal components analysis by choice of norm. *Ann. Statist.*, 24(1) :1–24, 02 1996.

[106] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani. A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2) :231–245, 2013.

[107] S. Spayd, J. Tarara, D. Mee, and J. C. Ferguson. Separation of sunlight and temperature effects on the composition of vitis vinifera cv. merlot berries. *American Journal of Enology and Viticulture*, 53 :171–182, 2002.

[108] D. J. Stekhoven. *missForest : Nonparametric Missing Value Imputation using Random Forest*, 2013. R package version 1.4.

[109] D. J. Stekhoven and P. Bühlmann. MissForest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1) :112–118, 10 2011.

[110] D. J. Stekhoven and P. Bühlmann. MissForest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1) :112–118, 10 2011.

[111] J. M. Tarara, J. Lee, S. E. Spayd, and C. F. Scagel. Berry temperature and solar radiation alter acylation, proportion, and concentration of anthocyanin in merlot grapes. *American Journal of Enology and Viticulture*, 59(3) :235–247, 2008.

[112] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1) :267–288, 1996.

[113] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 67(1) :91–108, 2005.

[114] R. J. Tibshirani and J. Taylor. The solution path of the generalized lasso. *Ann. Statist.*, 39(3) :1335–1371, 06 2011.

[115] R. J. Tibshirani and J. Taylor. Degrees of freedom in lasso problems. *Ann. Statist.*, 40(2) :1198–1232, 04 2012.

[116] B. Tisseyre, H. Ojeda, and J. Taylor. New technologies and methodologies for site-specific viticulture. *OENO One*, 41(2) :63–76, Jun. 2007.

[117] J. Usset, A.-M. Staicu, and A. Maity. Interaction models for functional regression. *Computational Statistics and Data Analysis*, 94 :317–329, 2016.

[118] C. Varlet-Grancher, G. Gosse, M. Chartier, H. Sinoquet, R. Bonhomme, and J. Allirand. Mise au point : rayonnement solaire absorbé ou intercepté par un couvert végétal. *Agronomie*, 9(5) :419–439, 1989.

[119] H. Wang and C. Leng. A note on adaptive group lasso. *Computational Statistics and Data Analysis*, 52(12) :5277 – 5286, 2008.

[120] J.-L. Wang, J.-M. Chiou, and H.-G. Müller. Functional data analysis. *Annual Review of Statistics and Its Application*, 3(1) :257–295, 2016.

[121] X. Wang, H. Zhu, and for the Alzheimer's Disease Neuroimaging Initiative. Generalized scalar-on-image regression models via total variation. *Journal of the American Statistical Association*, 112 :519 :1156–1168, 06 2017.

[122] B. Xin, Y. Kawahara, Y. Wang, and W. Gao. Efficient generalized fused lasso and its application to the diagnosis of alzheimer's disease. *Proceedings of the National Conference on Artificial Intelligence*, 3 :2163–2169, 01 2014.

[123] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B*, 68 :49–67, 02 2006.

[124] Y. Zhao, R. Ogden, and P. Reiss. Wavelet-based lasso in functional linear regression. *Journal of computational and graphical statistics : a joint publication of American Statistical Association, Institute of Mathematical Statistics, Interface Foundation of North America*, 21 :600–617, 07 2012.

[125] H. Zhou and L. Li. Regularized matrix regression. *Journal of the Royal Statistical Society. Series B, Statistical methodology*, 76 2 :463–483, 2014.

[126] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 67(2) :301–320, 2005.

[127] H. Zou and H. H. Zhang. On the adaptive elastic-net with a diverging number of parameters. *Ann. Statist.*, 37(4) :1733–1751, 08 2009.

Annexes

Package 'SpiceFP'

July 23, 2021

Type Package

Title Sparse and Structured Procedure to Identify Combined Effects of Functional Predictors

Version 0.1.0

Author Girault Gnanguenon Guesse [aut, cre],
Patrice Loisel [aut],
Benedicte Fontez [aut],
Nadine Hilgert [aut],
Thierry Simonneau [ctr],
Isabelle Sanchez [ctr]

Maintainer Girault Gnanguenon Guesse <girault.gnanguenon@gmail.com>

Description A set of functions allowing to implement the SpiceFP approach which is iterative. It involves transformation of functional predictors into several candidate explanatory matrices (based on contingency tables), to which relative edge matrices with contiguity constraints are associated. Generalized Fused Lasso regression are performed in order to identify the best candidate matrix, the best class intervals and related coefficients at each iteration. The approach is stopped when the maximal number of iterations is reached or when retained coefficients are zeros. Supplementary functions allow to get coefficients of any candidate matrix or mean of coefficients of many candidates.

License GPL-3

Encoding UTF-8

LazyData true

Depends R (>= 3.6.0)

Imports doParallel, foreach, stringr, tidyr, Matrix, genlasso, purrr, gplots

Suggests rmarkdown, knitr, fields

RoxygenNote 7.1.1

NeedsCompilation no

R topics documented:

`SpiceFP-package` *A Sparse and Structured Procedure to Identify Combined Effects of Functional Predictors*

Description

A set of functions allowing to implement the SpiceFP approach which is iterative. It involves transformation of functional predictors into several candidate explanatory matrices (based on contingency tables), to which relative edge matrices with contiguity constraints are associated. Generalized Fused Lasso regression are performed in order to identify the best candidate matrix, the best class intervals and related coefficients at each iteration. The approach is stopped when the maximal number of iterations is reached or when retained coefficients are zeros. Supplementary functions allow to get coefficients of any candidate matrix or mean of coefficients of many candidates.

Details

The main function of the package is the `spicefp` function. It directly performs the three main steps of the SpiceFP approach, by using intermediate functions of the package.
1) At he first step, contingency tables are constructed by defining joint modalities using class intervals or bins. Several candidate partitions are then defined. For each statistical individual i and each candidate partition (denoted u here), the 2 (resp. 3) functional predictors are transformed into frequency bi(resp. tri)-variate histograms (or contingency tables), stored as row vectors. The combination of these row vectors for all individuals enables the construction of a candidate explanatory matrix indexed by u (denoted here X^u). The function `candidates` is designed to build these candidate matrices.
2) At the second step, for each candidate explanatory matrix, an edge matrix is defined to represent the contiguity constraints between modalities of the contingency table.
3) Finally at the last step, the best class intervals and related regression coefficients are defined by:

i) performing a Generalized Fused Lasso using each candidate explanatory matrix. The SpiceFP model is the following

$$y_i = X_i^u \beta^u + \varepsilon_i,$$

where β^u is the coefficient to be estimated on the 2D (resp. 3D) intervals. The estimator of β is obtained as follows:

$$\hat{\beta}^{u,\gamma}(\lambda) = argmin \frac{1}{2} \|y - X^u \beta\|_2^2 + \lambda \|D^{u,\gamma} \beta\|_1,$$

where λ is a penalty parameter that controls the smoothness of the coefficients, and γ is the ratio between the regularization parameters of parsimony and fusion. ii) choosing the best candidate matrix and selecting its variables using an information criterion and checking the shutdown conditions to stop the approach. Indeed, SpiceFP may be used in an iterative way. It therefore allows to identify up to K best candidate matrices and related coefficients.

Author(s)

Maintainer: Girault Gnanguenon Guesse <girault.gnanguenon@gmail.com>

Authors:

- Patrice Loisel <patrice.loisel@inrae.fr>
- Benedicte Fontez <benedicte.fontez@supagro.fr>
- Nadine Hilgert <nadine.hilgert@inrae.fr>

Other contributors:

- Thierry Simonneau <thierry.simonneau@inrae.fr> [contractor]
- Isabelle Sanchez <isabelle.sanchez@inrae.fr> [contractor]

`candidates` *candidates*

Description

The "candidates" function essentially provides the candidate matrices and their characteristics. These candidate matrices can be constructed from 2 or 3 functional predictors.

Usage

```
candidates(
  fp1,
  fp2,
  fp3 = NULL,
  fun1,
  fun2,
  fun3 = NULL,
  parlists,
  ncores = parallel::detectCores() - 1,
  xcentering = TRUE,
  xscaling = FALSE
)
```

Arguments

fp1	numerical matrix with in columns observations of one statistical individual to partition. Each column corresponds to the functional predictor observation for one statistical individual. The order of statistical individuals is the same as in fp2. It is assumed that no data are missing and that all functional predictors are observed on an equidistant (time) scale.
fp2	numerical matrix with the same number of columns and rows as fp1. Columns are also observations. The order of statistical individuals is the same as in fp1.
fp3	NULL by default. numerical matrix with the same number of columns and rows as fp1 and fp2. The order of statistical individuals is the same as in fp1 and fp2.
fun1	a function object with 2 arguments. First argument is fp1 and the second is a list of parameters that will help to partition fp1, such as the number of class intervals, etc. For example, the list of parameters for using the logbreaks function is equivalent to list(alpha, J). All arguments to be varied for the creation of different candidate matrices must be stored in the parameter list. The other arguments must be set by default.
fun2	a function object with 2 arguments. First argument is fp2 and the second is a list of parameters.
fun3	NULL by default. Same as fun1 and fun2, a function with 2 arguments fp3 and a list of parameters.
parlists	list of 2 elements when fp3 and fun3 are equal to NULL or of 3 elements when fp3 and fun3 are provided. All elements of parlists are lists that have the same length. Each list contains all the lists of parameters required to create different candidates. The first element of parlists concerns the list of parameters required for fun1, the second element is relative to fun2 and the third to fun3. See Example 2 below.
ncores	numbers of cores that will be used for parallel computation. By default, it is equal to detectCores()-1.
xcentering	TRUE by default. Defined whether or not the variables in the new candidate matrices should be centered.
xscaling	FALSE by default. Defined whether or not the variables in the candidate matrices should be scaled.

Details

The function begins by partitioning each of the functional predictors using the function and associated parameter lists. Once the class intervals are obtained for each predictor, a contingency table is created for each statistical individual. This table counts the components of the observation variable (time for time series). The contingency table is then transformed into a row vector that corresponds to a row of the candidate matrix created. The number of candidate matrices is equal to the length of each element contained in parlists. For a fixed index, the functional predictors (fp1, fp2, fp3), the functions (fun1, fun2, fun3) and the lists of parameters associated to the index in each element of parlists allow to create a single candidate matrix. In addition to constructing the candidate matrices, the function associates with each matrix a vector containing the index and the numbers of class intervals used per predictor.

Value

The function returns a list with:

spicefp.dimension the dimension of the approach. Equal to 2 if fp3=NULL and 3 if not

candidates a list that has the same length as the elements of parlists. Each element of this list contains a candidate matrix and a vector with index and the numbers of class intervals used per predictor

fp1, fp2, fp3, fun1, fun2, fun3, parlists, xcentering, xscaling same as inputs

Examples

```
##linbreaks: a function allowing to obtain equidistant breaks
linbreaks<-function(x,n){
    sort(round(seq(trunc(min(x)),
         ceiling(max(x)+0.001),
         length.out =unlist(n)+1),
         1)
    )
}

p<-expand.grid(c(12,15),c(15,20))
pl<-list(split(p[,1], seq(nrow(p))),
         split(p[,2], seq(nrow(p))))

# Setting ncores=2 for this example check purpose
test<-candidates(fp1=matrix(rnorm(1000,52,15),ncol=10),
                 fp2=matrix(rpois(1000,50),ncol=10),
                 fun1=linbreaks,
                 fun2=linbreaks,
                 parlists=pl,
                 xcentering = FALSE,
                 xscaling = FALSE,
                 ncores=2)
str(test)
names(test)

# Example 2 from the spiceFP data
tpr.nclass=seq(10,16,2)
irdc.nclass=seq(20,24,2)
irdc.alpha=c(0.01,0.02,0.03)
p2<-expand.grid(tpr.nclass, irdc.alpha, irdc.nclass)
parlist.tpr<-split(p2[,1], seq(nrow(p2)))
parlist.irdc<-split(p2[,2:3], seq(nrow(p2)))
parlist.irdc<-lapply(
  parlist.irdc,function(x){
    list(x[[1]],x[[2]])}
)
m.irdc <- as.matrix(Irradiance[,-c(1)])
m.tpr <- as.matrix(Temperature[,-c(1)])
test2<-candidates(fp1=m.irdc,
                  fp2=m.tpr,
```

```
                    fun1=logbreaks,
                    fun2=linbreaks,
                    parlists=list(parlist.irdc,
                                  parlist.tpr),
                    xcentering = TRUE,
                    xscaling = FALSE,
                    ncores=2)
length(test2$candidates)
class(test2$candidates)
#View(test2$candidates[[1]][[1]])
dim(test2$candidates[[1]][[1]])
test2$candidates[[1]][[2]]

# Closing the connections for the example check purpose
closeAllConnections()
```

`coef_spicefp` *coef_spicefp*

Description

This function allows to obtain the coefficients of a model (involving a candidate matrix and 2 regularization parameters). There are two possible options to use this function: 1/ by minimizing an information criterion and selecting a number of model (option by default), or 2/ directly by providing the parameters of the model(s) that the user wishes to reconstruct.

Usage

```
coef_spicefp(
  spicefp.result,
  iter_,
  criterion = "AIC_",
  nmodels = 1,
  model.parameters = NULL,
  dim.finemesh = NULL,
  ncores = parallel::detectCores() - 1,
  write.external.file = TRUE
)
```

Arguments

`spicefp.result`	List. Outputs of the spicefp function.
`iter_`	integer. number of the iteration of interest.
`criterion`	character. One of "AIC_", "BIC_", "Cp_". Can be NULL, "AIC_" by default. If specified, nmodels must also be provided.
`nmodels`	integer. Equal to 1 by default. Represents the number of best models, according to the information criterion used. Should be NULL if criterion = NULL.

`model.parameters`
: data.frame. NULL by default. One or more rows contained in the file where the model statistics were stored. Be careful to use the file related to the selected iteration. Names used in model.parameters shoud be the same in the file.

`dim.finemesh`
: numeric vector of length 2 or 3. This vector informs about the dimension of the fine-mesh arrays (or matrices).

`ncores`
: numbers of cores that will be used for parallel computation. By default, it is equal to detectCores()-1.

`write.external.file`
: logical. indicates whether the result table related to each iteration has been written as a file (txt) in your working directory. This argument must be equal to the argument with the same name in the spicefp function.

Details

By providing criterion and nmodels, the function returns the coefficients of the nmodels best models chosen by the selected information criterion. When model.parameters is instead provided, it returns the coefficients of the models described on each row of the data.frame.

Value

Returns a list of 2 elements:

Model.parameters data.frame where each row contains statistics related to the models of interest. Same as input if model.parameters is provided.

coef.list List of length nmodels or the number of rows in Model.parameters. Each element of this list contains the model results as provided by the genlasso package, its coefficients without and with NA, a fine-mesh array with the coefficients, and the estimation of $X\beta$. Coefficients with NA are coefficient vector where the coefficient value of never-observed joint modalities is NA.

Examples

```
##linbreaks: a function allowing to obtain equidistant breaks
linbreaks<-function(x,n){
       sort(round(seq(trunc(min(x)),
                ceiling(max(x)+0.001),
                length.out =unlist(n)+1),
            1)
           )
}
# In this example, we will evaluate 2 candidates with 14 temperature
# classes and 15 irradiance classes. The irradiance breaks are obtained
# according to a log scale (logbreaks function) with different alpha
# parameters for each candidate (0.005, 0.01).
## Data and inputs
tpr.nclass=14
irdc.nclass=15
```

```
irdc.alpha=c(0.005, 0.01)
p2<-expand.grid(tpr.nclass, irdc.alpha, irdc.nclass)
parlist.tpr<-split(p2[,1], seq(nrow(p2)))
parlist.irdc<-split(p2[,2:3], seq(nrow(p2)))
parlist.irdc<-lapply(
     parlist.irdc,function(x){
     list(x[[1]],x[[2]])}
)
m.irdc <- as.matrix(Irradiance[,-c(1)])
m.tpr <- as.matrix(Temperature[,-c(1)])

# For the constructed models, only two regularization parameter ratios
# penratios=c(1/25,5) is used. In a real case, we will have to evaluate
# more candidates and regularization parameters ratio.
ex_sp<-spicefp(y=FerariIndex_Difference$fi_dif,
               fp1=m.irdc,
               fp2=m.tpr,
               fun1=logbreaks,
               fun2=linbreaks,
               parlists=list(parlist.irdc,
                             parlist.tpr),
               penratios=c(1/25,5),
               appropriate.df=NULL,
               nknots = 100,
               ncores =2,
               write.external.file = FALSE)

# coef_spicefp
## coefficients based on the parameters of the model
## focus on model selected by Mallows's Cp at iteration 1

start_time_spc <- Sys.time()
results.eval.iter1<-ex_sp$Evaluations[[1]]$Evaluation.results$evaluation.result
c.mdl <- coef_spicefp(ex_sp, iter_=1,
                      criterion =NULL,
                      nmodels=NULL,
  model.parameters=results.eval.iter1[which.min(results.eval.iter1$Cp_),],
                      ncores = 1,
                      write.external.file =FALSE)

g1<-c.mdl$coef.list$'231'$Candidate.coef.NA.finemeshed
g1.x<-as.numeric(rownames(g1))
g1.y<-as.numeric(colnames(g1))
duration_spc <- Sys.time() - start_time_spc

library(fields)
plot(c(10,2000),c(15,45),type= "n", axes = FALSE,
     xlab = "Irradiance (mmol/m2/s - Logarithmic scale)",
     ylab = "Temperature (deg C)",log = "x")
rect(min(g1.x),min(g1.y),max(g1.x),max(g1.y), col="black", border=NA)
image.plot(g1.x,g1.y,g1, horizontal = FALSE,
           col=designer.colors(64, c("blue","white")),
           add = TRUE)
```

```
axis(1) ; axis(2)

## Let's visualize the same model from other arguments of coef_spicefp
c.crit <- coef_spicefp(ex_sp, iter_=1,
                       criterion ="Cp_",nmodels=1,
                       ncores = 1,
                       write.external.file =FALSE)
g2<-c.crit$coef.list$'231'$Candidate.coef.NA.finemeshed
g2.x<-as.numeric(rownames(g2))
g2.y<-as.numeric(colnames(g2))
plot(c(10,2000),c(15,45),type= "n", axes = FALSE,
     xlab = "Irradiance (mmol/m2/s - Logarithmic scale)",
     ylab = "Temperature (deg C)",log = "x")
rect(min(g2.x),min(g2.y),max(g2.x),max(g2.y), col="black", border=NA)
image.plot(g2.x,g2.y,g2, horizontal = FALSE,
           col=designer.colors(64, c("blue","white")),
           add = TRUE)
axis(1) ; axis(2)
closeAllConnections()
```

`evaluate.candidates` *evaluate.candidates*

Description

This function performs for each candidate matrix, a Generalized Fused Lasso (sparse fused lasso 2d or 3d) and computes various statistics and information criteria related to the constructed model.

Usage

```
evaluate.candidates(
  candmatrices,
  y,
  penratios,
  nknots,
  appropriate.df = NULL,
  ncores = parallel::detectCores() - 1,
  penfun = NULL,
  file_name = "parametertable",
  write.external.file = TRUE
)
```

Arguments

`candmatrices`	List. Output of the "candidates" function. The spicefp dimension is the first element. The second contains many lists of one candidate matrix and related vector with index and numbers of class intervals used per predictor. The other

elements of the lists are the inputs of "candidates" function. If the user does not need the "candidates" function for the creation of candmatrices, it is possible to build a list provided that it respects the same structure as well as the names of the outputs of the "candidates" function. In this case only the first two elements of the list are essential: spicefp.dimension and candidates. The remaining elements can be NULL.

`y` numerical vector. Contains the dependent variable. This vector will be used as response variable in the construction of models involving each candidate matrix.

`penratios` numeric vector with values greater than or equal to 0. It represents the ratio between the regularization parameters of parsimony and fusion. When penratios=0, it corresponds to the pure fusion. The higher its value, the more parsimonious the model is.

`nknots` integer. For one value in penratios vector, it represents the number of models that will be constructed for each candidate matrix. It is the argument "nlam" of `coef.genlasso` function. This argument can also be NULL. In this case, the argument appropriate.df must be provided.

`appropriate.df` (appropriate degree of freedom) NULL by default. Numerical vector with values greater than or equal to 1. The degree of freedom of generalized fused problem is equal the number of connected components. A connected component gives information on a group of non-zero coefficients sharing the same value and connected by a contiguity matrix. More simply, it can be interpreted as a group of coefficients that have a unique influence. When the user has a prior idea of the number of zones of influence that the desired solution could contain, it is advisable to provide appropriate.df, a vector of appropriate degrees of freedom. In this case, nknots must be NULL.

`ncores` numbers of cores that will be used for parallel computation. By default, it is equal to detectCores()-1.

`penfun` function with 2 arguments (dim1, dim2) when dealing with 2 dimensional spiceFP or 3 arguments (dim1, dim2, dim3) when dealing with 3 dimensional spiceFP. The argument order in the penalty function is associated with the order of numbers of class intervals used per predictor in the second element of candmatrices argument. NULL by default. When penfun=NULL, getD2dSparse of genlasso or getD3dSparse is used according to the dimension of spiceFP.

`file_name` character. It is the name of the file in which the evaluation summary of all the candidate matrices is stored. This file is saved in your working directory.

`write.external.file`
logical. Indicates whether the result table should be written as a file (txt) in your working directory. It is recommended to use write.external.file=TRUE when evaluating a large number of candidate matrices (more than 100) in order to keep memory available.

Details

This function mainly returns statistics on the models built based on the candidate matrices. For each candidate matrix, length(penratios) x nknots or length(penratios) x length(appropriate.df) models are constructed in order to estimate the regularization parameters and to perform a variable selection. The computed statistics provide information on the quality of the models. For obvious reasons

of memory management, the coefficients related to each of these models are not stored. The statistics are stored in a file named via the argument file_name and can be consulted to get an idea of the state of progress of the program. The genlasso package is used for the implementation of the Generalized Fused Lasso.

Value

The output is a list with :

evaluation.result Same as file_name. The file contains a matrix with in columns : the candidate index (Candidate_id), the value of penratios used for this model (Pen_ratio), the parameter that penalizes the difference in related coefficients (PenPar_fusion), the degree of freedom of the model (Df_), the residual sum of squares (RSS_), the Akaike information criterion (AIC_), the Bayesian information criterion (BIC_), the Mallows' Cp (Cp_), the Generalized Cross Validation (GCV_), the slope of the regression lm($y \sim X\beta$) (Slope_), the ratio $var(y - X\beta)/var(y)$ (Var_ratio).

response.variable, penalty.ratios, nknots, appropriate.df, penalty.function Exactly the inputs y, penratios, nknots, appropriate.df, penfun

Examples

```
# Constructing 2 candidates for spiceFP data (temperature and Irradiance)
linbreaks<-function(x,n){
     sort(round(seq(trunc(min(x)),
                ceiling(max(x)+0.001),
                length.out =unlist(n)+1),
            1)
          )
}
# In this example, we will evaluate 2 candidates (each having 10
# temperature classes and respectively 10 and 20 irradiance classes).
# Only one value is used for alpha (logbreaks argument)
tpr.nclass=10
irdc.nclass=c(10,20)
irdc.alpha=0.005
p2<-expand.grid(tpr.nclass, irdc.alpha, irdc.nclass)
parlist.tpr<-split(p2[,1], seq(nrow(p2)))
parlist.irdc<-split(p2[,2:3], seq(nrow(p2)))
parlist.irdc<-lapply(
   parlist.irdc,function(x){
   list(x[[1]],x[[2]])}
)
m.irdc <- as.matrix(Irradiance[,-c(1)])
m.tpr <- as.matrix(Temperature[,-c(1)])
test2<-candidates(fp1=m.irdc,
                  fp2=m.tpr,
                  fun1=logbreaks,
                  fun2=linbreaks,
                  parlists=list(parlist.irdc,
                                parlist.tpr),
                  xcentering = TRUE,
```

```
                xscaling = FALSE,
                ncores=2)
# Evaluating candidates
# For the constructed models, only one regularization parameter ratio
# penratios=c(1) is used. In a real case, we will have to evaluate
# more candidates and regularization parameters ratio.
start_time_ev <- Sys.time()
evcand<-evaluate.candidates(candmatrices = test2,
                            y=FerariIndex_Difference$fi_dif,
                            penratios=c(1),
                            appropriate.df=NULL,
                            nknots = 100,
                            ncores=2,
                            write.external.file = FALSE)
duration_ev <- Sys.time() - start_time_ev
tab_res<-evcand$evaluation.result
dim(tab_res)
tab_res[which.min(tab_res$AIC_),]

closeAllConnections()
```

`FerariIndex_Difference`

FerariIndex_Difference of vine dataset

Description

Data were collected during an experiment conducted on a vineyard of the INRAE/Institut Agro campus at Montpellier in 2014 (Syrah vines). The objective of the experiment was to study the influence of the micro-climate (temperature and irradiance) at the grape level on the anthocyanin contents of the berries indicated by the Ferari index. This dataset contains Ferari index differences between August 01, 2014 at 09:00 am and July 24th, 2014 at 09:00 am. The individuals are in rows. The individuals' names (Indiv1,...,Indiv32) are used to name the rows. The same individuals are also present in the irradiance and temperaure datasets.

Usage

```
FerariIndex_Difference
```

Format

A data frame with 32 observations and 1 variable.

fi_dif numeric. Ferari index differences between July 24th, 2014 at 09:00 am and August 01, 2014 at 09:00 am.

Source

These data were acquired during the Innovine project, funded by the Seventh Framework Programme of the European Community (FP7/2007-2013), under Grant Agreement No. FP7-311775.

`finemeshed2d` *finemeshed2d*

Description

Function that helps to transform a vector into a matrix (with a fine mesh). In the implementation of the spiceFP approach, it allows to transform matrices of coefficients having different dimensions into matrices of the same dimension in order to perform arithmetic operations. In practice, the matrix to be transformed is associated with a contingency table, which implies numerical variables for which classes have been created.

Usage

```
finemeshed2d(
  x,
  n.breaks1 = 1000,
  n.breaks2 = 1000,
  round.breaks1 = 9,
  round.breaks2 = 9
)
```

Arguments

`x`	vector or one column matrix to scale. This vector comes from the vectorization of the matrices to be transformed. x is named using the concatenation of the names of the rows and the names of the columns of the matrix to be transformed, as shown in the example below.
`n.breaks1`	integer. Number of breaks needed for the first variable. The variable for which classes are in first position when constructing x's names is the first variable.
`n.breaks2`	integer. Number of breaks needed for the second variable. The variable for which classes are in second position when constructing x's names is the second variable.
`round.breaks1`	integer. Number of decimals for breaks of the first variable.
`round.breaks2`	integer. Number of decimals for breaks of the second variable.

Details

This function is designed to return a fine meshed matrix and breaks associated. In order to obtain a fine mesh, a high number of breaks must be fixed.

Value

Returns:

finemeshed.matrix Matrix of dimension n.breaks2 x n.breaks1. The row and column names of finemeshed.matrix are the breaks created from each variable and the associated n.breaks. Each value of finemeshed.matrix is equal to the value of x indexed by the classes containing the row and column names of finemeshed.matrix

finemeshed.values1 First variable breaks

finemeshed.values2 Second variable breaks

Examples

```
set.seed(45)
count_table<- hist_2d(x = rnorm(1000),
                      y = rnorm( 1000,5,0.1),
                      breaks_x = seq(-4, 4, by =1),
                      breaks_y = seq(2, 8, by =1))$Hist.Values

df.x<-as.data.frame.table(count_table)
x<-df.x$Freq
names(x)<-paste0(df.x$Var1,"_",df.x$Var2)

res.fm2d <- finemeshed2d(x,100,100)
dim(res.fm2d$finemeshed.matrix)
```

`finemeshed3d` *finemeshed3d*

Description

Function that helps to transform a vector into a 3 dimensional array (with a fine mesh). In the implementation of the spiceFP approach, it allows to transform matrices of coefficients having different dimensions into matrices of the same dimension in order to perform arithmetic operations. In practice, the 3d array to be transformed is associated with a contingency table, which implies numerical variables for which classes have been created.

Usage

```
finemeshed3d(
  x,
  n.breaks1 = 10,
  n.breaks2 = 1000,
  n.breaks3 = 500,
  round.breaks1 = 9,
  round.breaks2 = 9,
  round.breaks3 = 9
)
```

Arguments

x	vector or one column matrix to scale. This vector comes from the vectorization of the 3d array to be transformed. x is named using the concatenation of the names of the dimension of the array to be transformed, as shown in the example below.
n.breaks1	integer. Number of breaks needed for the first variable The variable for which classes are in first position when constructing x's names is the first variable.
n.breaks2	integer. Number of breaks needed for the second variable. The variable for which classes are in second position when constructing x's names is the second variable.
n.breaks3	integer. Number of breaks needed for the third variable. The variable for which classes are in third position when constructing x's names is the third variable.
round.breaks1	integer. Number of decimals for breaks of the first variable.
round.breaks2	integer. Number of decimals for breaks of the second variable.
round.breaks3	integer. Number of decimals for breaks of the third variable.

Details

This function is designed to return a 3d fine meshed array and breaks associated. In order to obtain a fine mesh, a high number of breaks must be fixed.

Value

Returns:

finemeshed.array Array of dimension n.breaks1 x n.breaks2 x n.breaks3. The dimension names of finemeshed.array are the breaks created from each variable and the associated n.breaks. Each value of finemeshed.array is equal to the value of x indexed by the classes containing the row and column names of finemeshed.array

finemeshed.values1 First variable breaks

finemeshed.values2 Second variable breaks

finemeshed.values3 Third variable breaks

Examples

```
set.seed(4)
count_table<-hist_3d(x = rnorm(1000),
                     y = rnorm( 1000,5,0.1),
                     z = rnorm( 1000,2,1),
                     breaks_x = seq(-4, 4, by =1),
                     breaks_y = seq(2, 8, by =1),
                     breaks_z = seq(-3, 6, by =1))$Hist.Values

df.x<-as.data.frame.table(count_table)
x<-df.x$Freq
names(x)<-paste0(df.x$Var1,"_",df.x$Var2,"_",df.x$Var3)
```

```
res.fm3d<- finemeshed3d(x,10,50,100)
dim(res.fm3d$finemeshed.array)
```

---

getD3dSparse *getD3dSparse*

Description

getD3dSparse is a function that helps to construct generalized lasso penalty matrix D when using the fusedlasso function over a 3 dimensional grid

Usage

```
getD3dSparse(dim1, dim2, dim3)
```

Arguments

dim1	positive integer. Based on a 3 dimensional grid, dim1 represents the number of units represented on the first dimension
dim2	positive integer which represents the number of units represented on the second dimension
dim3	positive integer which represents the number of units represented on the third dimension

Details

The function returns a sparse penalty matrix providing information on the connections between the variables during the implementation of a generalizad fused lasso.

Value

a matrix with dim1 x dim2 x dim3 columns. Each row represents an edge (a link between 2 variables) and is constructed with the couple (-1, 1), relative to these 2 variables and 0 for all others. In the context of a generalized fused lasso, this matrix penalizes only the differences in coefficients (fusion). To obtain parsimony in addition to the fusion, a diagonal matrix with the same number of columns must be bound to the penalty matrix constructed by getD3dSparse. This matrix will contain diagonally the ratio: parsimony penalty parameter on fusion penalty parameter. When using fusedlasso function, this operation is performed when you provide the argument gamma.

Examples

```
library(genlasso)
library(Matrix)
D<-getD3dSparse(2,3,2)
plot(getGraph(D))
```

`hist_2d` *hist_2d*

Description

This function results from a modification of the `hist2d` function of the gplots package in order to build the 2D histogram with breaks directly provided as inputs of the new function.

Usage

```
hist_2d(
  x,
  y,
  breaks_x,
  breaks_y,
  same.scale = FALSE,
  na.rm = TRUE,
  FUN = base::length
)
```

Arguments

`x`	either a numerical vector to be partitioned or a matrix of 2 numerical columns to be partitioned.
`y`	a numerical vector to be partitioned. Not required if x is a matrix.
`breaks_x`	a numerical vector. Contains the breaks related to x for the histogram
`breaks_y`	a numerical vector. Contains the breaks related to y for the histogram
`same.scale`	logical. Default to FALSE. If TRUE, breaks_x will be used for x and y
`na.rm`	logical. Default to TRUE. Indicates whether missing values should be removed
`FUN`	function used to summarize bin contents.

Details

The default function used for the argument FUN is the function length. When another function is used, it is applied on x, or on the first column of x if this is a two-column matrix. The lower limit of each class interval is included in the class and the upper limit is not.

Value

Using a given set of breaks per each variable, the function returns :

Hist.Values a matrix with in rows class intervals of x and in columns class intervals of y. Contingency table is returned if FUN=length

breaks_x, breaks_y same as the inputs of the function

Midpoints.x, Midpoints.y the midpoints for each bin per variable

nobs.x , nobs.y number of observations of x and y

n.bins vector of 2 elements containing the number of bins for x and y

Examples

```
set.seed(45)
hist_2d(x = rnorm(1000),
        y = rnorm( 1000,5,0.1),
        breaks_x = seq(-4, 4, by =1),
        breaks_y = seq(2, 8, by =1))
```

`hist_3d` *hist_3d*

Description

This function can be used in order to construct a 3D histogram based on 3 variables and relative breaks directly provided as inputs.

Usage

```
hist_3d(
  x,
  y,
  z,
  breaks_x,
  breaks_y,
  breaks_z,
  same.scale = FALSE,
  na.rm = TRUE,
  FUN = length
)
```

Arguments

`x`	either a numerical vector to be partitioned or a matrix with 3 numerical columns to be partitioned.
`y`	a numerical vector to be partitioned. Not required if x is a matrix.
`z`	a numerical vector to be partitioned. Not required if x is a matrix
`breaks_x`	a numerical vector. Contains the breaks related to x for the histogram
`breaks_y`	a numerical vector. Contains the breaks related to y for the histogram
`breaks_z`	a numerical vector. Contains the breaks related to z for the histogram
`same.scale`	logical. Default to FALSE. If TRUE, breaks_x will be used for x, y and z
`na.rm`	logical. Default to TRUE. Indicates whether missing values should be removed
`FUN`	function used to summarize bin contents.

Details

The default function used for the argument FUN is the function length. When another function is used, it is applied on x or on the first column of x if this is a three-column matrix. The lower limit of each class interval is included in the class and the upper limit is not.

Value

Using a given set of breaks per each variable, the function returns :

Hist.Values a 3 dimensional array. The 1st (respectively 2nd, 3rd) dimension is related to the class intervals of x (resp. y, z). Contingency table is returned if FUN=length

breaks_x, breaks_y, breaks_z same as the inputs of the function

Midpoints.x, Midpoints.y, Midpoints.z the midpoints for each bin per variable

nobs.x , nobs.y, nobs.z number of observations of x, y and z

n.bins vector of 3 elements containing the number of bins for x, y and z

Examples

```
set.seed(4)
hist_3d(x = rnorm(1000),
        y = rnorm( 1000,5,0.1),
        z = rnorm( 1000,2,1),
        breaks_x = seq(-4, 4, by =1),
        breaks_y = seq(2, 8, by =1),
        breaks_z = seq(-2, 6, by =1))
```

Irradiance — *Photosynthetic Photon Flux Density PPFD (PPFD) measurements of vine dataset*

Description

Data were collected during an experiment conducted on a vineyard of the INRAE/Institut Agro campus at Montpellier in 2014 (Syrah vines). The objective of the experiment was to study the influence of the micro-climate (temperature and irradiance) at the grape level on the anthocyanin contents of the berries indicated by the Ferari index. This dataset is related to irradiance measurements in the morning (sunrise to twelve am) between July 24th, 2014 at 09:00 am and August 01, 2014 at 09:00 am. These observations are made at the same time (every 12 minutes) as the temperature observations. The individuals are in columns while the observation times are in rows. The same individuals are also present in the Temperature and FerariIndex_Difference datasets.

Usage

```
Irradiance
```

Format

A data frame (of one functionnal variable) with 127 rows (observation times) and 33 columns: the 1st one is a character vector which corresponds to date-time in format "yyyy-mm-dd hh:mm:ss", the others are numeric vectors made of the observations of irradiance (PPFD) measured in $10^{-6}mol.m^{-2}.s^{-1}$ on each of the 32 statistical individuals Indiv1,...,Indiv32. Irradiance corresponds to the number of incident photons useful for photosynthesis, received per unit of time on a horizontal surface unit.

Source

These data were acquired during the Innovine project, funded by the Seventh Framework Programme of the European Community (FP7/2007-2013), under Grant Agreement No. FP7-311775.

`logbreaks` *logbreaks*

Description

A function that allows to obtain histogram class limits following a logarithmic scale. It also has a parameter that allows to set the scale at your convenience.

Usage

```
logbreaks(
  x,
  parlist = list(alpha, J),
  round_breaks = 0,
  plot_breaks = FALSE,
  effect.threshold.begin = NA,
  effect.threshold.end = NA
)
```

Arguments

`x`	either a numeric vector to be partitioned or a numeric vector containing the minimum and maximum of the vector to be partitioned.
`parlist`	a list of 2 elements. The first one is alpha, a numeric and positive value. It is a parameter affecting the number of breaks closed to the minimum. The second one is J. It is a nonnegative and nonzero integer and represent the selected number of classes.
`round_breaks`	a nonnegative integer. Equal to 0 by default, it is the number of decimal values of the breaks.
`plot_breaks`	logical. FALSE by default. If TRUE, the breaks are plotted.
`effect.threshold.begin`	NA by default. Numeric value between the minimum and maximum of x. If it isn't NA, the first class is created with xmin and effect.threshold.begin.
`effect.threshold.end`	NA by default. Numeric value between the minimum and maximum of x. If it isn't NA, the last class is created with xmax and effect.threshold.end.

Details

The breaks are obtained as follows:

$$L(w) = \min(x) + \frac{e^{\alpha \frac{w-1}{J}} - 1}{e^{\alpha} - 1}(\max(x) - \min(x)),\ w = 1, \ldots, J+1.$$

Value

The return is a numeric vector of length J+1 with the breaks obtained following a log scale.

Examples

```
logbreaks(c(10,1000), parlist=list(0.2,5))
logbreaks(c(10,1000), parlist=list(0.2,5),plot_breaks=TRUE)
```

`meancoef` *meancoef*

Description

This function can be used to compute the mean of coefficients from different partitions in the context of the spicefp approach.

Usage

```
meancoef(coef.list, weight)
```

Arguments

`coef.list`	list. The second element of the coef_spicefp function outputs. It has the same name as the argument.
`weight`	a numerical vector of weights with the same length as coef.list.

Details

Here, the fine-mesh coefficients are weighted and a weighted mean is deduced. If the user wishes, he can use as weights the slopes associated with the qualities of the models concerned.

Value

Returns a list of :

weighted_mean fine-mesh matrix or array with the weighted mean of the coefficients

y.estimated weighted estimation of $X\beta$

coefficients.array An array with all the fine-mesh coefficients that will be used to compute the weighted mean

weight same as inputs

Examples

```
##linbreaks: a function allowing to obtain breaks linearly
linbreaks<-function(x,n){
     sort(round(seq(trunc(min(x)),
                ceiling(max(x)+0.001),
                length.out =unlist(n)+1),
           1)
          )
}
# In this example, we will evaluate 2 candidates with 14 temperature
# classes and 15 irradiance classes. The irradiance breaks are obtained
# according to a log scale (logbreaks function) with different alpha
# parameters for each candidate (0.005, 0.01).
## Data and inputs
tpr.nclass=14
irdc.nclass=15
irdc.alpha=c(0.005, 0.01)
p2<-expand.grid(tpr.nclass, irdc.alpha, irdc.nclass)
parlist.tpr<-split(p2[,1], seq(nrow(p2)))
parlist.irdc<-split(p2[,2:3], seq(nrow(p2)))
parlist.irdc<-lapply(
   parlist.irdc,function(x){
   list(x[[1]],x[[2]])}
)
m.irdc <- as.matrix(Irradiance[,-c(1)])
m.tpr <- as.matrix(Temperature[,-c(1)])

# For the constructed models, only two regularization parameter ratios
# penratios=c(1/25,5) is used. In a real case, more candidates
# and regularization parameter ratios should be evaluated.
ex_sp<-spicefp(y=FerariIndex_Difference$fi_dif,
               fp1=m.irdc,
               fp2=m.tpr,
               fun1=logbreaks,
               fun2=linbreaks,
               parlists=list(parlist.irdc,
                             parlist.tpr),
               penratios=c(1/25,5),
               appropriate.df=NULL,
               nknots = 100,
               ncores =2,
               write.external.file = FALSE)

## Focus on the 2 best models retained by the AIC criterion at iteration 1
c.mdls <- coef_spicefp(ex_sp, iter_=1, criterion ="AIC_",
                       nmodels=2, ncores = 2,
                       dim.finemesh=c(1000,1000),
                       write.external.file = FALSE)

# meancoef
# Compute the mean of the coefficients of these models
```

```
mean.c.mdls<-meancoef(c.mdls$coef.list,
                      weight = c.mdls$Model.parameters$Slope_)
g3<-mean.c.mdls$weighted_mean
g3.x<-as.numeric(rownames(g3))
g3.y<-as.numeric(colnames(g3))


library(fields)
plot(c(10,2000),c(15,45),type= "n", axes = FALSE,
     xlab = "Irradiance (mmol/m2/s - Logarithmic scale)",
     ylab = "Temperature (deg C)",log = "x")
rect(min(g3.x),min(g3.y),max(g3.x),max(g3.y), col="black", border=NA)
image.plot(g3.x,g3.y,g3, horizontal = FALSE,
           col=designer.colors(256, c("blue","white","red")),
           add = TRUE)
axis(1) ; axis(2)

closeAllConnections()
```

---

spicefp *spicefp*

Description

This function is used to implement the spiceFP approach. This approach transforms 2 (by default) or 3 functional predictors into candidate explonatory matrices in order to identify joint classes of influence. It can take functional predictors and partitioning functions as inputs in order to create candidate matrices to be evaluated. The user can choose among the existing partitioning functions (as logbreaks) or provide his own partitioning functions specific to the functional predictors under consideration. The user can also directly provide candidate matrices already constructed as desired.

Usage

```
spicefp(
  y,
  fp1,
  fp2,
  fp3 = NULL,
  fun1,
  fun2,
  fun3 = NULL,
  parlists,
  xcentering = TRUE,
  xscaling = FALSE,
  candmatrices = NULL,
  K = 2,
```

```
  criterion = "AIC_",
  penratios = c(1/10, 1/5, 1/2, 1, 2, 5, 10),
  nknots = 50,
  appropriate.df = NULL,
  penfun = NULL,
  dim.finemesh = c(1000, 1000),
  file_name = paste0("parametertable", 1:2),
  ncores = parallel::detectCores() - 1,
  write.external.file = TRUE
)
```

Arguments

y	a numerical vector. Contains the dependent variable. This vector will be used as response variable in the construction of models involving each candidate matrix.
fp1	a numerical matrix with in columns observations of one statistical individual to partition. Each column corresponds to the functional predictor observation for one statistical individual. The order of the statistical individuals is the same as in fp2. It is assumed that no data are missing and that all functional predictors are observed on an equidistant (time) scale.
fp2	a numerical matrix with the same number of columns and rows as fp1. Columns are also observations. The order of the statistical individuals is the same as in fp1.
fp3	NULL by default. A numerical matrix with the same number of columns and rows as fp1 and fp2. The order of the statistical individuals is the same as in fp1 and fp2.
fun1	a function object with 2 arguments. First argument is fp1 and the second is a list of parameters that will help to partition fp1, such as the number of class intervals, etc. For example using the logbreaks function, the list of parameters is equivalent to list(alpha, J). All the arguments to be varied for the creation of different candidate matrices must be stored in the parameter list. The other arguments must be set by default.
fun2	a function object with 2 arguments. First argument is fp2 and the second is a list of parameters.
fun3	NULL by default. Same as fun1 and fun2, a function with 2 arguments fp3 and a list of parameters.
parlists	a list of 2 elements when fp3 and fun3 are equal to NULL or of 3 elements when fp3 and fun3 are provided. All the elements of parlists are lists that have the same length. Each list contains all the lists of parameters that have to be used to create different candidates. The first element of parlists concerns the first functional predictor fp1, the second element is relative to fp2 and the third to fp3.
xcentering	TRUE by default. Defined whether or not the variables in the new candidate matrices should be centered.
xscaling	FALSE by default. Defined whether or not the variables in the candidate matrices should be scaled.

`candmatrices` NULL by default. List. Output of the "candidates" function. The spiceFP dimension is its first element. The second contains many lists of one candidate matrix and related vector with index and numbers of class intervals used per predictor. The other elements of the lists are the inputs of "candidates" function. If the user does not need the "candidates" function for the creation of candmatrices, it is possible to build a list while making sure that it respects the same structure as well as the names of the outputs of the "candidates" function. In this case, only the first two elements of the list are essential: spicefp.dimension and candidates. The remaining elements can be NULL.

`K` number of iterations of the spiceFP approach. Equal to 2 by default.

`criterion` character. One of "AIC_", "BIC_", "Cp_". The criterion to be used in each iteration in order to identify the best candidate matrix and to estimate the regulation parameters. This criterion is used to perform model selection as well as variable selection.

`penratios` a numeric vector with values greater than or equal to 0. It represents the ratio between the regularization parameters of parsimony and fusion. When penratios=0, it corresponds to the pure fusion. The higher its value, the more parsimonious the model is.

`nknots` integer. For one value in penratios vector, it represents the number of models that will be constructed for each candidate matrix. It is the argument "nlam" of `coef.genlasso` function. This argument can be also NULL. In this case, the argument appropriate.df must be provided.

`appropriate.df` (appropriate degree of freedom) NULL by default. When used, nknots must be NULL. It is the argument "df" of `coef.genlasso` function. When the user has a prior idea of the number of zones of influence that the solution could contain, it is advisable to provide appropriate.df, a vector of appropriate degrees of freedom. appropriate.df is a numerical vector with values greater than or equal to 1. The degree of freedom of generalized fused Lasso models is equal to the number of connected components. A connected component gives information on a group of non-zero coefficients sharing the same value and connected by a contiguity matrix. More simply, it can be interpreted as a group of coefficients that have a unique influence.

`penfun` function with 2 arguments (dim1, dim2) when dealing with 2 dimensional spiceFP, or with 3 arguments (dim1, dim2, dim3) when dealing with 3 dimensional spiceFP. The argument order in the penalty function is associated with the order of numbers of class intervals used per predictor in the second element of candmatrices argument. NULL by default. When penfun=NULL, getD2dSparse of genlasso or getD3dSparse is used according to the dimension of spiceFP.

`dim.finemesh` numeric vector of length 2 or 3. This vector informs about the dimension of the fine-mesh arrays (or matrices) that will be used for the visualization of the sum of the coefficients selected at different iterations.

`file_name` character vector. Of length K, it contains the list of names that will be used to name the files containing informations on the candidate matrix models

`ncores` numbers of cores that will be used for parallel computation. By default, it is equal to detectCores()-1.

write.external.file
: logical. indicates whether the result table related to each iteration should be written as a file (txt) in your working directory. It is recommended to use write.external.file=TRUE when evaluating a large number of candidate matrices (more than 100) in order to keep memory available.

Details

Three main steps are involved to implement spiceFP: transformation of functional predictors, creation of a graph of contiguity constraints and identification of the best class intervals and related regression coefficients.

Value

Returns a list with:

Candidate.Matrices a list with candidate matrices and their characteristics. same as candmatrices if it has been provided.

Evaluations List of length less than or equal to K. Each element of the list contains information about an iteration. Contains the results related to the evaluation of the candidate matrices. These include the name of the file where the model information is stored, the best candidate matrix and related coefficients, the partition vector that indexes it, the $X\beta$ estimation, the residuals, etc.

coef.NA List of length less than or equal to K. For each iteration, it contains the coefficient vector where the coefficient value of never-observed joint modalities is NA

coef.NA.finemeshed List of length less than or equal to K. For each iteration, the coefficient vector is transformed into fine-mesh array or matrix allowing arithmetic operations to be performed between coefficients coming from different partitions

spicefp.coef fine-mesh array or matrix. Sum of the coefficients selected at all iterations

Examples

```
##linbreaks: a function allowing to obtain breaks linearly
linbreaks<-function(x,n){
    sort(round(seq(trunc(min(x)),
                ceiling(max(x)+0.001),
                length.out =unlist(n)+1),
            1)
        )
}

# In this example, we will evaluate 2 candidates with 14 temperature
# classes and 15 irradiance classes. The irradiance breaks are obtained
# according to a log scale (logbreaks function) with different alpha
# parameters for each candidate (0.005, 0.01).
## Data and inputs
tpr.nclass=14
irdc.nclass=15
irdc.alpha=c(0.005, 0.01)
```

```
p2<-expand.grid(tpr.nclass, irdc.alpha, irdc.nclass)
parlist.tpr<-split(p2[,1], seq(nrow(p2)))
parlist.irdc<-split(p2[,2:3], seq(nrow(p2)))
parlist.irdc<-lapply(
   parlist.irdc,function(x){
   list(x[[1]],x[[2]])}
)
m.irdc <- as.matrix(Irradiance[,-c(1)])
m.tpr <- as.matrix(Temperature[,-c(1)])

# For the constructed models, only two regularization parameter ratios
# penratios=c(1/25,5) are used. In a real case, we will have to evaluate
# more candidates and regularization parameters ratio.
start_time_sp <- Sys.time()
ex_sp<-spicefp(y=FerariIndex_Difference$fi_dif,
               fp1=m.irdc,
               fp2=m.tpr,
               fun1=logbreaks,
               fun2=linbreaks,
               parlists=list(parlist.irdc,
                             parlist.tpr),
               penratios=c(1/25,5),
               appropriate.df=NULL,
               nknots = 100,
               ncores =2,
               write.external.file=FALSE)

duration_sp <- Sys.time() - start_time_sp
# View(ex_sp$Evaluations[[1]]$Evaluation.results$evaluation.result)
# View(ex_sp$Evaluations[[2]]$Evaluation.results$evaluation.result)
# Visualization of the coefficients
g<-ex_sp$spicefp.coef
g.x<-as.numeric(rownames(g))
g.y<-as.numeric(colnames(g))

library(fields)
plot(c(10,2000),c(15,45),type= "n", axes = FALSE,
     xlab = "Irradiance (mmol/m²/s - Logarithmic scale)",
     ylab = "Temperature (°C)",log = "x")
rect(min(g.x),min(g.y),max(g.x),max(g.y), col="black", border=NA)
image.plot(g.x,g.y,g, horizontal = FALSE,
           col=designer.colors(256, c("blue","white","red")),
           add = TRUE)
axis(1) ; axis(2)

closeAllConnections()
```

`Temperature` *Temperature measurements of vine dataset*

Description

Data were collected during an experiment conducted on a vineyard of the INRAE/Institut Agro campus at Montpellier in 2014 (Syrah vines). The objective of the experiment was to study the influence of the micro-climate (temperature and irradiance) at the grape level on the anthocyanin contents of the berries indicated by the Ferari index. This dataset is related to temperature measurements in the morning (sunrise to twelve am) between July 24th, 2014 at 09:00 am and August 01, 2014 at 09:00 am. These observations are made at the same time (every 12 minutes) as the irradiance observations. The individuals are in columns while the observation times are in rows. The same individuals are also present in the Irradiance and FerariIndex_Difference datasets.

Usage

```
Temperature
```

Format

A data frame (of one functionnal variable) with 127 rows (observation times) and 33 columns: the 1st one is a character vector which corresponds to date-time in format "yyyy-mm-dd hh:mm:ss", the others are numeric vectors made of the observations of temperature measured in degree celsius on each of the 32 statistical individuals Indiv1,...,Indiv32.

Source

These data were acquired during the Innovine project, funded by the Seventh Framework Programme of the European Community (FP7/2007-2013), under Grant Agreement No. FP7-311775.

Index