Sélection de variables à l'aide de forêts aléatoires pour données de survie de grande dimension

Mélanie Huchon

2ème année Master Sciences, Technologies, Santé
Mention Santé publique - Parcours Biostatistique
ISPED - Université de Bordeaux

Promotion 2019-2021

Rapport de stage : Sélection de variables à l'aide de forêts aléatoires pour données de survie de grande dimension

---

Mélanie HUCHON

Stage réalisé du 01/03/2021 au 31/08/2021

Centre de Recherche - Bordeaux Population Heath Inserm U1219
Equipe SISTM

Maître de stage :
Robin GENUER, Maître de conférences

Remerciements

Je tiens à remercier, tout d'abord, mon maître de stage Robin Genuer pour ce stage qui fut particulièrement intéressant et enrichissant. Merci pour ton encadrement, ta bienveillance et ta disponibilité. Je te remercie également de m'avoir partagé ton expertise des forêts aléatoires ainsi que pour nos réunions et tes réflexions sur les différentes simulations.

Je tiens également à remercier Rodolphe Thiebaut et l'équipe SISTM pour leur accueil à distance pendant les réunions d'équipe en visioconférence et en présentiel, quand les conditions le permettaient, pour ceux que j'ai croisé dans le bureau John Snow. Je vous remercie également pour vos retours suite à ma présentation qui m'ont permis d'avancer sur les simulations et l'application aux données réelles.

Je remercie l'ensemble de l'équipe pédagogique de l'ISPED pour leur qualité d'enseignements et leur disponibilité durant ces 2 années de formation.

Merci à Blandine, mon acolyte depuis le lycée avec qui nos parcours universitaires se sont suivis jusqu'à aujourd'hui où la vie professionnelle commence et tu poursuis en thèse et moi en CDD mais nos bureaux resteront proches. Je te remercie pour ton soutien sans faille, autant du côté personnel qu'universitaire depuis tout ce temps, pour nos repas le midi quand nous étions en présentiel, pour tous nos autres moments et pour tout ce qui est à venir.

Je remercie également ma famille, qui de près ou de loin, me soutient et veille sur moi pour s'assurer que tout se déroule au mieux. Merci à vous sans qui je ne serais pas là aujourd'hui.

Pour finir, je tiens à remercier mon copain, Alexandre, pour ton soutien quotidien, pour m'avoir écoutée parler de forêts aléatoires, de *VSURF* et pour ta relecture de ce rapport et tes conseils malgré que ce ne soit pas ton domaine d'expertise. Merci pour tout.

Table des matières

Liste des figures

Liste des tables

Liste des abréviations

CART : Classification And Regression Trees
C-index : concordance index
iid : indépendante et identiquement distribuée
Inria : Institut National de Recherche en en Informatique et en Automatique
Inserm : Institut National de la Santé et de la Recherche Médicale
ISPED : Institut de Santé Publique, d'Epidémiologie et de Développement
OOB : out-of-bag
PBC : primary biliary cirrhosis
RF : Forêts aléatoires (Random Forest, en anglais)
RSF : Forêts aléatoires en survie (Random Survival Forest, en anglais)
SISTM : Statistiques pour la médecine translationnelle
VIMP : Variable importance
VSURF : Variable Selection Using Random Forests

1 Introduction

Dans le cadre de ma deuxième année de Master Sciences, Technologies, Santé mention Santé Publique - Parcours Biostatistique à l'Institut de Santé Publique, d'Epidémiologie et de Développement (ISPED) de Bordeaux, j'ai réalisé mon stage de fin d'étude au sein de l'équipe SISTM (Statistiques pour la médecine translationnelle) rattachée à l'unité U1219 - Bordeaux Population Health de l'Institut National de la Santé et de la Recherche Médicale (Inserm). Ce stage s'est déroulé du 1 mars au 31 août 2021 sous la responsabilité de Robin Genuer, maître de conférences.

Afin de décrire au mieux mon stage, une présentation de l'unité Inserm U1219 qui m'a accueillie pendant ces six mois de stage sera détaillée. Ensuite, le contexte de recherche dans lequel s'inscrit ce stage sera développé. Dans une autre partie, les méthodes utilisées seront présentées avant de décrire les résultats. Enfin, ces derniers seront discutés puis une conclusion sera apportée à ce rapport.

1.1 La structure d'accueil : Inserm U1219

L'Inserm, créé en 1964, est un établissement de recherche public français dédié à la santé humaine. Il est rattaché au ministère de la Santé et au ministère de la Recherche. De plus, il a une portée internationale et a été acteur de nombreuses avancées médicales majeures, notamment dans la découverte du VIH et les traitements du cancer, toujours avec l'objectif d'améliorer la santé de tous. L'Inserm dispose de plus de 350 structures de recherche en France et à l'étranger.

Dans le cadre de ce stage, j'ai intégré le Centre de recherche U1219 de l'Inserm intitulé "Bordeaux population health" localisé à Bordeaux. Ce centre de recherche associé à l'Université de Bordeaux est dirigé par le professeur Christophe Tzourio et composé de 11 équipes de recherche labellisées et 2 équipes émergentes.

Parmi ces équipes, j'ai effectué ce stage dans l'équipe de recherche SISTM (Statistiques pour la médecine translationnelle) dirigée par le professeur Rodolphe Thiebaut. Elle est labellisée par l'Inserm et l'Inria (Institut National de Recherche en en Informatique et en Automatique) et est répartie en trois axes de recherche. Ces trois axes sont nommés "Modélisation mécanistique", "Données de grande dimension" et "Vaccinologie translationnelle".
Mon stage s'est déroulé au sein de l'axe "Données de grande dimension" dans laquelle de nouvelles méthodes pour données de grande dimension comme des données omiques telles que les génomiques, transcriptomiques ou protéomiques sont développées. L'analyse de ces ensembles de données partant des voies moléculaires jusqu'à la réponse clinique d'une population de patients est aujourd'hui un défi. L'objectif de cet axe est de sélectionner les informations pertinentes ou de les résumer en vue d'une meilleure compréhension ou dans un but de prédicition.

1.2 Contexte

Avec les progrès scientifiques, techniques et informatiques réalisés ces dernières années, la quantité d'informations recueillies à chaque instant a augmenté de manière exponentielle. On peut par exemple citer le domaine de la santé et de la biologie où l'étude du génome humain amène à traiter des milliers de variables. De nombreuses données sont donc disponibles, permettant une source presque inépuisable de nouvelles connaissances, indispensables à l'innovation et aux progrès médicaux. Cependant, ces grandes quantités d'informations font place à des défis techniques majeurs

concernant leur stockage et les capacités d'exploitation. Cela implique donc un important temps de calcul et une vérification systématique des données et conditions d'application des méthodes utilisées. Pour répondre à cette problématique, il faut alors que les programmes et algorithmes informatiques et statistiques soient de plus en plus complexes.

En effet, dans de nombreuses applications médicales, on dispose d'un grand nombre de variables observées pouvant être plus grand que le nombre de patients dans l'échantillon, c'est ce qu'on appelle des données de grande dimension. Ce type de données peuvent être par exemple, des données médico-économiques, de cohortes, de registres ou encore, d'études cliniques. Dans ces bases de données, il est possible d'y trouver, des paramètres cliniques, biologiques, d'imagerie et génomiques qui introduisent un grand nombre de variables. De plus, la quantité d'intérêt dans de nombreuses études est le temps d'apparition d'un évènement au cours du temps (apparition d'une maladie, changement de stade d'une pathologie, décès, etc.). Dans ce cas, il s'agit de données de survie de grande dimension. Les données de survie se rencontrent principalement en recherche clinique dans le cadre des essais thérapeutiques et dans les études de cohorte. Dans le premier cas, on cherche à comparer l'efficacité de deux traitements avec comme critère le délai jusqu'à l'évènement d'intérêt (rechute, décès, ...). Dans le second cas, une cohorte de sujets est suivie au cours du temps pour lesquels une maladie peut ou non apparaître durant le suivi.

Avec les données de survie en grande dimension, plusieurs problématiques se posent. La première étant la prédiction du temps de l'évènement qui manque de méthodes adaptées au grand nombre de variables disponibles. Secondement, si l'objectif est d'expliquer quelles sont les variables les plus prédictrices ou associées à la survenue de l'évènement, c'est un problème de sélection de variables.

Plusieurs méthodes ont été développées pour répondre à cette problématique. Premièrement, les modèles de Cox pénalisés avec une pénalité Lasso (1) permettent d'analyser les données de survie en grande dimension mais cette méthode suppose une structure particulière due à sa modélisation. De plus, des approches dites non-paramétriques peuvent être intéressantes à mettre en œuvre pour la sélection de variables. Parmi elles, les forêts aléatoires pour données de survie, introduites par Ishwaran et al.(2) permettent de prédire des temps d'évènement à partir de données de grande dimension. A partir des forêts aléatoires en survie, la sélection de variables peut s'appliquer aussi bien en survie qu'en cas de risques compétitifs comme on peut le retrouver dans divers articles (3,4). Dans ces 2 articles, la sélection de variable se fait avec la notion de Ishwaran et al., "the minimal depth" soit la profondeur minimale (5). De plus, dans l'article de Gilhodes et al.(4), une comparaison de sélection de variables entre les forêts aléatoires en survie et une approche "boosting". D'autres auteurs comparent aussi les méthodes de sélection de variables dans le cadre de la survie. La comparaison se fait surtout entre les forêts aléatoires en survie et les modèles de Cox (6,7). Certains articles composent leur méthode de sélection de variables à partir des éléments qu'offrent les forêts aléatoires en survie. Par exemple, celui de Pang et al. (8) utilise les erreurs out-of-bag (OOB) et un seuil calculé avec un écart-type de l'erreur OOB minimale et l'article de Dietrich et al. (9) sélectionne les variables à partir des valeurs de la profondeur minimale des variables.
Dans l'article de Wang et al. (10), une revue sur les forêts aléatoires en survie a été réalisée pour données de grande dimension où est recensé les développements sur les différentes étapes autour de ces forêts. Dans le cas de la sélection de variables, on trouve notamment la version de Pang et al. (8) mais aussi d'autres méthodes comme une sélection avec un indice topologique basé sur la permutation.

Les forêts aléatoires en survie ne permettent pas de sélectionner nativement des variables, mais

fournissent un score d'importance de variables qui peut être utilisé dans une procédure de sélection de variables telle que celle de Genuer et al. (11), la suite de ce rapport présente l'adaptation de cette procédure.

1.3 Objectifs

L'objectif de ce stage est, tout d'abord, de faire un bilan sur les méthodes existantes dans le cadre de la sélection de variables pour données de survie de grande dimension puis de proposer l'adaptation d'une méthode de sélection de variables à partir des forêts aléatoires en survie pour l'analyse de données de survie en grande dimension et enfin, d'étudier son comportement et ses performances sur des simulations et des données réelles.

2 Matériel et Méthodes

2.1 Analyse de survie

L'analyse de survie est très clairement détaillée dans le document *Méthodes d'analyses de données de survie*[1] obtenu en cours et dont je me suis fortement appuyée pour cette partie.

En analyse de données de survie, la variable étudiée est la durée de survie désignant le temps écoulé jusqu'à la survenue d'un évènement précis. L'évènement d'intérêt est associé à un changement d'état, communément appelé "décès". Cependant, l'évènement n'est pas forcément la mort : il peut s'agir de l'apparition d'une maladie ou encore de la disparition de symptômes.

L'analyse des données de survie permet d'étudier le délai de survenue de cet évènement. Ce type d'analyse est utilisé dans le contexte d'études longitudinales comme les études de cohorte (suivi de patients dans le temps) et les essais thérapeutiques (tester l'efficacité d'un médicament). On cherche à estimer la distribution des temps de survie (fonction de survie : Kaplan-Meier), à comparer les fonctions de survie de plusieurs groupes de sujets (test du logrank) ou à analyser l'influence de variables explicatives sur la survie des individus (modèle de régression : Cox). Toutes les parties ne seront pas détaillées ici[1], seulement les définitions utiles pour la suite de ce rapport.

L'une des caractéristiques de l'analyse de survie concerne les données incomplètes de type censurées et/ou tronquées. La difficulté peut être l'obtention de l'information complète des temps d'apparition de l'évènement. En effet, si l'individu n'a pas subi l'évènement à sa date de dernières nouvelles, le délai n'est donc pas observé. Si cette information n'est pas disponible, le sujet est alors censuré à droite.
On notera pour l'individu i :

- son temps de survie T_i^o
- son temps de censure C_i^o
- la durée réellement observée T_i

Pour représenter les données des sujets censurés à droite, l'écriture mathématique courante est d'associer à chaque individu i un couple de variables aléatoires (T_i, δ_i) avec les définitions de la variable aléatoire T_i le délai de survie et δ_i l'indicatrice de l'évènement :

$$T_i = min(T_i^o, C_i^o)$$

et

$$\delta_i = \begin{cases} 1 & \text{si } T_i^o \leq C_i^o \text{ c'est-à-dire si l'évènement est observé (d'où } T_i = T_i^o) \\ 0 & \text{sinon c'est-à-dire si l'individu est censuré (d'où } T_i = C_i^o). \end{cases}$$

Un autre cas de données incomplètes est la troncature. Une observation est dite tronquée lorsqu'elle est conditionnelle à un autre évènement. Le cas de troncature le plus courant est celui de la troncature à gauche impliquant qu'un individu n'est observable que si sa durée de vie est supérieure à une certaine valeur. Par exemple, c'est le cas quand un suivi ne commence pas à la naissance, l'observation est donc tronquée à gauche à l'âge du début du suivi.

Concernant les fonctions liées à la survie, une fonction très utilisée en survie est la fonction de survie notée $S(t)$:

$$S(t) = P(X > t), \quad t \geq 0.$$

[1] Joly P., Alioum A., Commenges D., Jacqmin-Gadda H., Leffondré K., Le Goff M. et Proust-Lima C. Master Sciences, Technologies, Santé, Mention Santé Publique 2020-2021. Méthodes d'analyses de données de survie. Bordeaux : Université de Bordeaux, ISPED ; 2020.

Elle représente la probabilité de subir l'évènement au-delà de t, autrement dit, d'être vivant en t.

Une autre fonction utile en survie est la fonction de risque $\lambda(t)$:

$$\lambda(t) = \frac{f(t)}{S(t)}$$

$\lambda(t)\Delta t$ est la probabilité de subir l'évènement entre t et $t+\Delta t$ pour un sujet, conditionnellement au fait que ce sujet soit encore à risque à l'instant t. C'est la fonction qui est la plus souvent utilisée dans les modèles de régression. Elle est aussi parfois appelée fonction de risque instantané pour bien la distinguer de la fonction de risque cumulé définie ci-dessous.

La fonction de risque cumulé $\Lambda(t)$ est :

$$\Lambda(t) = \int_0^t \lambda(u)du.$$

La fonction de survie peut aussi s'exprimer en fonction du risque cumulé :

$$S(t) = exp(-\int_0^t \lambda(u)du) \ = e^{-\Lambda(t)}$$

Dans notre cadre, l'estimateur de Nelson-Aalen de la fonction de risque cumulé (ou *Cumulative Hazard Function - CHF*) est utilisé en présence de censure à droite et se calcule ainsi :

$$\hat{\Lambda}(t) = \sum_{j:t_j \leq t} \frac{d_j}{n_j}$$

où d_j représente le nombre d'évènements observés au temps t_j et n_j représente le nombre d'individus à risque au temps t_j. A risque signifie le nombre d'individus vivants à l'instant t_j ou qui ont un évènement en t_j. Cette fonction permet d'analyser l'évolution du risque au cours du temps.

2.2 Méthodes des forêts aléatoires

Les données de survie sont généralement analysées à l'aide de méthodes qui reposent sur des hypothèses restrictives telles que les risques proportionnels (modèles de Cox). De plus, ces méthodes sont souvent paramétriques et les effets non linéaires des variables doivent être modélisés par transformations (2).

Les forêts aléatoires en survie peuvent donc être une alternative à ce type de problème.

La méthodologie des forêts aléatoires, d'une part en régression, d'autre part en survie, va être décrite par la suite.

2.2.1 Concept de forêts aléatoires

Introduites par Leo Breiman en 2001 (12), les forêts aléatoires de l'anglais Random Forests (RF), sont une méthode d'apprentissage statistique non-paramétrique. Cette méthode permet de traiter des données de grande dimension où le nombre de variables explicatives p est supérieur à celui des observations n ($p >> n$). Elle permet aussi de gérer des données avec une relation complexe entre les variables explicatives et à expliquer.

Cet algorithme est une version d'ensemble des arbres CART (Classification And Regression Trees).

La stratégie générale des RF est la suivante :

Etape 1. Créer B échantillons bootstrap (A noter que chaque échantillon bootstrap exclut, en moyenne, 37% des données, appelées *out-of-bag* (données OOB)).

Etape 2. Pour chaque échantillon bootstrap $b = 1,\ldots, B$, appliquer un type d'arbres décisionnels correspondant au type de données considéré.

(a) A chaque nœud, un sous-ensemble de m variables prédictives est sélectionné aléatoirement.

(b) Chacune des variables sélectionnées est testée dans la division du nœud, jusqu'à obtenir celle qui optimise le critère de coupure.

Etape 3. Agréger l'information des nœuds terminaux pour obtenir le prédicteur final.

Un schéma des forêts aléatoires regroupant les 3 étapes présentées précédemment : Bootstrap, Arbres décisionnels et enfin, agrégation des prédicteurs se trouve en Figure 1. Le tirage bootstrap est désigné par Θ et le tirage aléatoire des variables l'est par Θ'.

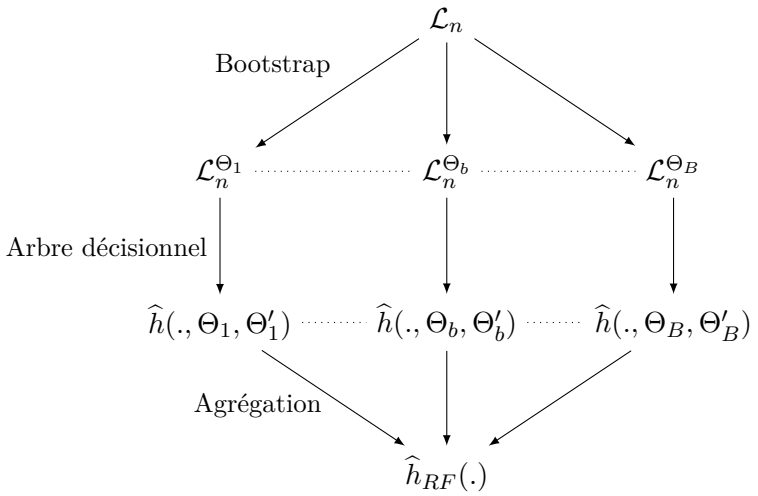

Figure 1: Schéma des forêts aléatoires.

Pour rappel, en régression, on cherche à minimiser la variance intra-groupe résultant de la découpe d'un nœud t en 2 nœuds fils t_L et t_R. En classification, l'ensemble des classes est $\{1, \ldots, C\}$, et on définit l'impureté des nœuds fils, le plus souvent par le biais de l'indice de Gini (13).

Cependant, les applications des RF sont principalement concentrées sur les problèmes de classification et de régression. C'est pourquoi depuis quelques années, des extensions de cette méthode se développent.

2.2.2 Forêts aléatoires en survie

Une des extensions des forêts aléatoires est celle pour l'analyse des données de survie censurées à droite. Elle est communément appelée forêts aléatoires en survie ou, en anglais Random Survival Forests (RSF), et a été introduite par Ishwaran et al. en 2008 (2). Les RSF fonctionnent de la même façon que dans le cadre de la régression ou de la classification. On retrouve les 3 étapes des RF,

c'est-à-dire, les échantillons bootstraps, la construction des arbres décisionnels puis l'agrégation des prédicteurs de chaque arbre. Par la suite, chaque changement entre les RF et RSF va être détaillé.

Tout d'abord, pour tenir compte de la survie dans les forêts aléatoires, les arbres décisionnels utilisés sont des arbres de survie binaires. Cette méthode d'arborescence est similaire aux arbres CART (Classification And Regression Trees) mais développée pour les données de survie censurées. Elle est basée sur la maximisation de la différence de survie entre les nœuds dans un arbre binaire (2).

Ensuite, concernant le critère de coupure des arbres des RSF, celui qui est le plus couramment utilisé est le test du *log-rank*. Il est initialement utilisé pour comparer la survie entre plusieurs groupes de sujets à un instant donné. Dans le cas des RSF, il est utilisé pour maximiser les différences de survie entre les nœuds fils.
Ce critère va être détaillé ci-dessous, dans le cas simplifié des données non-bootstrappées (14).

Soit h le nœud à diviser, les données sont désignées par (T_1,X_1,δ_1), ..., (T_n,X_n,δ_n), où X_i est le vecteur des covariables de l'individu i et T_i et δ_i sont respectivement le temps observé et l'indicateur de censure pour i. Soit X une variable des données, ici on prend le cas d'une variable quantitative pour simplifier. Une coupure proposée en utilisant X peut être $X \leq c$ et $X > c$ et divise h en nœuds fils gauche L et droit R.

Soit $t_1 < t_2 < ... < t_m$ les temps d'évènements distincts, et soit $d_{j,L}, d_{j,R}$ et $Y_{j,L}, Y_{j,R}$ le nombre de personnes qui ont subi l'évènement (décès, par exemple) et de personnes à risque au temps t_j dans les nœuds fils L et R :

$$Y_{j,L} = \#\{T_i \geq t_j, X_i \leq c\}, \quad Y_{j,R} = \#\{T_i \geq t_j, X_i > c\}$$

Définissons :

$$Y_j = Y_{j,L} + Y_{j,R}, \quad d_j = d_{j,L} + d_{j,R}$$

La valeur de la statistique de coupure de *log-rank* pour la coupure $L = \{X_i \leq c\}$ et $R = \{X_i > c\}$ est :

$$L(X,c) = \frac{\sum_{j=1}^{m}(d_{j,L} - Y_{j,L}\frac{d_j}{Y_j})}{\sqrt{\sum_{j=1}^{m}\frac{Y_{j,L}}{Y_j}(1-\frac{Y_{j,L}}{Y_j})(\frac{Y_j-d_j}{Y_j-1})d_j}}$$

La valeur $|L(X,c)|$ est une mesure de quantité de coupure des nœuds. Plus elle est élevée, plus la différence de survie entre L et R est grande et meilleure est la coupure. La meilleure division est déterminée en trouvant la variable X^* et la valeur de coupure c^* telles que $|L(X^*,c^*)| \geq |L(X,c)|$ pour tout X et c (14) .

D'autres critères peuvent être utilisés comme la conservation des évènements, le score du log-rank ou le log-rank randomisé, par exemple (2).

Dans le cas de la survie, dans chaque feuille (ou nœud terminal) de l'arbre, on calcule le prédicteur qui est l'estimateur de Nelson-Aalen de la fonction de risque cumulé, appelé aussi CHF (Cumulative Hazard Function, voir Section 2.1). Le prédicteur final est la fonction de risque cumulé d'ensemble qui est la moyenne sur l'ensemble des prédicteurs des arbres.

Pour résumer, l'algorithme des RSF est composé de la façon suivante (2) :

Etape 1. Générer *ntree* échantillons bootstrap sur les données d'origine (A noter que chaque échantillon bootstrap exclut, en moyenne, 37% des données, appelées *out-of-bag* (données OOB)).

Etape 2. Pour chacun de ces échantillons, développer un arbre de survie.

(a) A chaque nœud, *mtry* variables candidates sont sélectionnées aléatoirement. Le nœud

est divisé par la coupure qui maximise la différence de survie entre les deux nœuds fils, soit la statistique de test du *log-rank*.

(b) Développer l'arbre maximal où un nœud terminal ne doit pas avoir moins de *nodesize* évènements uniques.

Etape 3. Calculer la fonction de risque cumulé (CHF) pour chaque arbre. Faire la moyenne de ces dernières afin d'obtenir la fonction de risque cumulé d'ensemble.

Suite au développement de ces forêts, une erreur est disponible, l'erreur de prédiction. Cette erreur, aussi appelée "erreur OOB", est calculée à partir du C-index (concordance index) dans le cadre des RSF. Le C-index C utilisé ici est celui de Harrell (15), il désigne l'estimateur d'une probabilité de concordance pour les données de survie. Pour calculer C, on considère l'ensemble de toutes les paires de patients et on compare les temps de survie observés et les valeurs prédites. Les paires concordantes sont définies comme toutes les paires moins celles où la plus courte durée de survie est censurée et celles qui ont les deux durées de survie et les deux indicatrices d'évènement égales. Pour chaque paire concordante, si les prédictions sont concordantes aux observations, on compte 1 et on compte 0.5 en cas de prédictions identiques. La mesure de l'indice C est le ratio de la somme des paires concordantes sur l'ensemble des paires possibles. Ensuite, l'erreur OOB, notée $errOOB$, est définie comme :

$$errOOB = 1 - C.$$

On note que errOOB est compris entre 0 et 1 et que $errOOB = 0.5$ signifie que la prédiction n'est pas meilleure qu'une estimation aléatoire.

2.2.3 Package R randomForestSRC

Suite à cette méthode, un package a été développé par Ishwaran et al. (16), nommé *randomForestSRC*. Il permet de construire des forêts aléatoires dans différents cadres tels que la régression, la classification, en survie ou encore avec des risques compétitifs.

2.3 Notion d'importance des variables

La notion d'importance des variables permet de construire un classement des variables explicatives basé sur une quantification de l'importance des effets sur la variable réponse. Il existe plusieurs façons d'obtenir l'importance des variables comme par exemple la mesure de Gini ou l'importance par permutation introduite par Breiman et Cutler en 2004 (17). Cette dernière importance est celle qui nous intéresse pour la méthode de sélection de variable de la prochaine partie. En se basant sur Genuer et al. en 2019 (13), voici comment est calculée l'importance des variables dans leur méthode pour la classification et la régression :

On fixe $j \in \{1, \ldots, p\}$ et calcule $VI(X^j)$ l'indice d'importance de la variable X^j.

- $\mathcal{L}_n^\ell$ un échantillon bootstrap et l'échantillon OOB_ℓ associé, c'est-à-dire l'ensemble des observations qui n'apparaissent pas dans $\mathcal{L}_n^\ell$.
- On calcule $errOOB_\ell$, l'erreur commise sur OOB_ℓ par l'arbre construit sur $\mathcal{L}_n^\ell$.
- On permute aléatoirement les valeurs de la j-ième variable dans l'échantillon OOB_ℓ. Ceci donne un échantillon perturbé, noté $\widetilde{OOB_\ell}^j$.

- On calcule enfin $\widetilde{errOOB_\ell}^j$, l'erreur sur l'échantillon $\widetilde{OOB_\ell}^j$.
- On réitère ces opérations pour tous les échantillons bootstrap. L'importance de la variable X^j, $VI(X^j)$, est alors définie par la différence entre l'erreur moyenne d'un arbre sur l'échantillon OOB perturbé et celle de l'échantillon OOB :

$$VI(X^j) = \frac{1}{q}\sum_{\ell=1}^{q}(\widetilde{errOOB_\ell}^j - errOOB_\ell).$$

Ainsi, plus l'augmentation de l'erreur engendrée par les permutations aléatoires de la j-ième variable explicative est forte, plus la variable est donc importante. Inversement, si les permutations n'ont pas ou peu d'effet sur l'erreur, alors la variable est considérée comme peu importante (13).

Dans le cas des forêts aléatoires en survie, la même mesure est utilisée, l'importance par permutation (14), ce qui permet d'implémenter la survie dans la méthode suivante.

2.4 Sélection de variables et le package *VSURF*

2.4.1 Méthodologie de VSURF

La procédure de sélection de variables utilisée dans le cadre de ce stage est *VSURF (Variable Selection Using Random Forests)* de Genuer et al. introduite en 2010 (11) avec le package associé du même nom *VSURF* (18). Cette méthode est une procédure "automatique" où il n'y a aucun a priori à apporter pour faire la sélection. Elle procède en deux étapes : la première consiste à seuiller sur l'importance des variables dans le but d'éliminer un grand nombre de variables inutiles, alors que la seconde consiste à introduire les variables dans des modèles de forêts aléatoires.

Deux objectifs de sélection de variables sont distingués appelés interprétation et prédiction :

1. Dans une visée d'interprétation, on cherche à sélectionner toutes les variables X^j fortement reliées à la variable réponse Y.
2. Alors que pour un but de prédiction, on cherche à sélectionner un petit sous-ensemble de variables suffisant pour bien prédire la variable réponse.

La méthode de sélection de variables tente de satisfaire les deux objectifs précédents. Cette procédure, basée sur des forêts aléatoires comportant un très grand nombre d'arbres (typiquement *ntree*=2000), est décrite plus en détails ci-dessous (18) :

- **Etape 1.** Elimination préliminaire et classement :
  - Classer les variables par importance décroissante (obtenue en moyennant les VIMP sur 50 forêts typiquement).
  - Eliminer les variables de faible importance (soit m le nombre de variables conservées). *Afin d'éliminer les variables les moins importantes, un seuil est calculé à partir des écart-types des VIMP. Comme la variabilité des VIMP est plus grande pour les vraies variables du modèle que pour les variables non informatives, la valeur de ce seuil est trouvée par l'estimation de l'écart-type de VIMP pour les variables inutiles. Ce seuil est défini par le minimum prédit par le modèle CART où les* Y sont les écart-types des VIMP et les X sont les rangs classés par ordre croissant des VIMP. Ensuite, seules les variables dont la valeur moyennée du VIMP dépasse ce seuil sont retenues.

- **Etape 2.** Sélection de variables :

  - Pour l'*interprétation* : on construit la collection de modèles emboîtés constituée par les forêts construites sur les k premières variables, pour $k=1$ à m et on sélectionne les variables du modèle ayant la plus faible erreur OOB. Ceci conduit à considérer m' variables.

    Plus précisément, on calcule les moyennes des erreurs OOB des RF (généralement sur 25 forêts) des modèles emboîtés en commençant par celui avec uniquement la variable la plus importante et en terminant par celui comportant toutes les variables importantes conservées à l'étape 1. Les variables du modèle conduisant à l'erreur OOB la plus faible sont retenues. Pour faire face à l'instabilité, une astuce classique est utilisée : sélectionner le plus petit modèle avec une erreur inférieure à l'erreur OOB minimale augmentée de son écart-type (basée sur les 25 mêmes forêts).

  - Pour la *prédiction* : à partir des variables retenues pour l'interprétation, on construit une suite de modèles en introduisant séquentiellement, dans l'ordre d'importance croissant, et en testant itérativement les variables. Les variables du dernier modèle sont sélectionnées.

    Plus précisément, l'introduction séquentielle des variables est basée sur le test suivant : une variable est ajoutée seulement si la diminution d'erreur est supérieure à un seuil. L'idée est que cette diminution doit être significativement supérieure à la variation moyenne obtenue en ajoutant des variables non informatives. Le seuil est fixé à la moyenne des valeurs absolues des différences des erreurs OOB entre le modèle à m' variables et celui à m variables :

    $$\frac{1}{m-m'}\sum_{j=m'}^{m-1}|errOOB(j+1)-errOOB(j)|$$

    où errOOB(j) est l'erreur OOB de la forêt construite grâce aux j variables les plus importantes.

De plus, si une unique variable est sélectionnée à l'étape d'interprétation, il n'y aura pas de sélection faite pour l'étape de prédiction donc cette même variable sera aussi sélectionnée pour la prédiction.

2.4.2 Adaptation de VSURF

L'adaptation du package R *VSURF* a été réalisée à partir de la dernière version disponible sur le CRAN ou sur GitHub. Ici, la version du compte GitHub de Robin Genuer (*robingenuer*) a été importée afin d'être modifiée sur une "branche", appelée *survival*, parallèle à celle initialement créée[2].

Ensuite, l'appropriation du développement du package est nécessaire pour comprendre les objets et variables utilisées et implémentées.

Le package R de Ishwaran et al. *randomForestSRC* (16) a été utilisé, plus particulièrement la fonction *rfsrc*, pour permettre l'adaptation du package *VSURF* à la survie.

Le package est décomposé en plusieurs fichiers où chaque fichier représente une partie du package. Il y a premièrement, un fichier de description du package. Un dossier R est également

[2]Lien vers la branche *survival* : https://github.com/robingenuer/VSURF/tree/survival.

présent où chaque fichier représente une fonction créée. De plus, parmi les fichiers *R*, un fichier a été créé pour chaque partie de la méthode, c'est-à-dire, l'étape de seuillage et d'élimination préliminaire puis, celle pour l'interprétation et enfin, celle pour la prédiction. Un dernier fichier appelle ces 3 fonctions permettant d'avoir les 3 résultats avec une seule fonction. Cette fonction est donc la principale du package appelée par le même nom *VSURF*.

Chaque fonction a été modifiée afin de prendre en compte la survie, elles possèdent chacune 2 objets en fonction de la forme des variables d'entrée, le premier lorsque x et y sont rentrés indépendamment *(VSURF(x,y))* et le second quand l'entrée est une formule *(VSURF(y~x))*.

Dans le cas de la survie, le second objet avec la formule est utilisé car la survie dans *R* est écrite sous la forme : *"Surv(time, status) ~ ."*.

Les différents paramètres des fonctions ont donc été adaptés avec ceux de la survie et du package *randomForestSRC*.

2.5 Applications de la méthode

Afin d'étudier le comportement de *VSURF* sur les données de survie, plusieurs applications ont été faites. Avant chaque application de la méthode, une optimisation des paramètres des forêts aléatoires a été effectuée afin de trouver la paire de paramètres la plus optimale. Concernant les applications, différents schémas de simulation ont été réalisés afin de prendre en compte différents modèles et pourcentages de censure (30, 50 et 70%) et une application sur données réelles de grande dimension a été effectuée.

2.5.1 Optimisation des RSF

Avant d'appliquer le package *VSURF* aux simulations ou données réelles, une optimisation des paramètres des forêts a été réalisée. Il s'agit de faire varier *mtry*, le nombre de variables sélectionnées aléatoirement à chaque nœud et *nodesize*, le nombre d'évènements uniques à chaque nœud terminal. A chaque combinaison de *mtry* et *nodesize*, une fonction a été créée afin d'exécuter vingt forêts aléatoires en survie avec chacune 700 arbres et calculer une moyenne des erreurs sur les vingt forêts. Enfin, à partir des valeurs des paramètres et des erreurs, un graphique est réalisé avec en abscisse les valeurs de *nodesize*, en ordonnée les erreurs et enfin, une courbe par valeur de *mtry* est tracée. Le graphique illustre donc le comportement des RSF et permet de trouver la paire de paramètres la plus optimale.

Au vu du temps de calcul sur les différents serveurs, toutes les valeurs pour *mtry* et nodesize n'ont pas été testées. Suivant les bases de données, les valeurs des paramètres diffèrent mais les valeurs par défaut dans le package *randomForestSRC* sont testées à chaque fois. Les valeurs par défaut de *mtry* et *nodesize* sont respectivement $\sqrt{p}$ et 15.

2.5.2 Simulations

Différents schémas de simulations ont été réalisés. Le premier est réalisé à partir d'une base de données existante à laquelle des variables de bruits ont été ajoutées. Les schémas suivants ne sont pas présentés dans l'ordre qu'ils ont été réalisés mais du plus simple au plus complexe. Les simulations ont été inspirées de l'article de Hu et al. (19) pour la simulation de la section 2.4.2.3 et de l'article de Pang et al. (8) pour celle de la section 2.4.2.5. Les autres simulations sont des combinaisons des 2 articles ou alors venant de nos besoins.

Chaque simulation, hors celle avec les données existantes, est composée de 100 individus, de 600 covariables, corrélées ou non, et du temps de survie et de l'indicatrice de l'évènement étudié.

Le temps de survie pour le sujets i est calculé de la façon suivante :

$$T_i = -\frac{log(u)}{exp(X_2 + X_1 * X_2 + X_{45})}$$

où $u \sim U_{[0,1]}$. De plus, la censure est générée à partir d'une distribution exponentielle où le paramètre est défini en fonction du pourcentage de censure souhaité (30, 50 ou 70%).
Enfin, l'indicatrice de l'évènement est définie comme à la section 2.1.

Pour chaque pourcentage de censure, l'optimisation des paramètres se déroule de la façon suivante :

- la variation du paramètre *nodesize* est comprise entre 1 et 10 en saut de 1 et entre 15 et 30 en saut de 5 soit 14 valeurs.
- le paramètre *mtry* varie entre 100 et 600 en pas de 100 et avec la valeur par défaut soit 25, on a pour *mtry* 7 valeurs.

On se retrouve alors avec 84 combinaisons de *mtry* et *nodesize* où pour chaque combinaison la fonction de paramétrisation permettant de calculer les erreurs sera utilisée.

Pour les simulations, la fonction *VSURF* sera exécutée avec ses paramètres par défaut pour le nombre de forêts réalisées, soient 50 forêts de réalisées pour l'étape d'élimination préliminaire et 25 forêts pour l'étape d'interprétation et de prédiction.

2.5.2.1 PBC avec et sans variables de bruit

Les données PBC (Primary Biliary Cirrhosis) (16) sont des données disponibles dans le package *randomForestSRC* de *R*. Il s'agit d'une base constituée entre 1974 et 1984 au cours d'un essai randomisé de la Mayo Clinic pour étudier l'efficacité de la D-pénicillamine sur la cirrhose biliaire primitive du foie.

Cette base est composée de 418 patients. Parmi eux, 312 ont participé à l'essai randomisé et les 106 autres n'ont pas participé mais ont accepté de fournir différentes informations et de se faire suivre. Dans notre cas, seuls 276 patients seront considérés parmi ceux qui ont participé à l'essai car ils ne possèdent aucunes valeurs manquantes.

Initialement, deux évènements sont étudiés dans cet essai. Le premier est la transplantation d'un nouveau foie et le second le décès. Or, dans la base de *randomForestSRC*, seuls les décès sont étudiés et les patients transplantés sont donc considérés comme censurés. En plus des variables de survie (temps, évènement), 17 covariables telles que l'âge, le sexe ou des mesures biologiques comme le taux de bilirubine, d'albumine sont présentes dans la base.

Pour arriver dans un cas en grande dimension, des variables de bruit ont été ajoutées à la base de données *pbc* initiale. Cet ajout de bruit va permettre de voir si notre méthode sélectionne les bonnes variables et non celles de bruit. Trois cent quatre-vingt trois variables ont été ajoutées et suivent une distribution normale avec une moyenne de 0 et une matrice de covariance où les éléments (i, j) valent $0.9^{|i-j|}$. Cette nouvelle base sera donc composée de 400 variables au total.

Tout d'abord, l'optimisation des paramètres et l'exécution de *VSURF* ont été réalisées sur la base de données initiale et ensuite, sur celle avec les variables de bruit. Le paramètre

nodesize a été varié entre 1 et 200, de 1 à 10 en pas de un et de 50 à 200 en pas de 50, ce qui fait 15 valeurs. Pour le paramètre *mtry*, comme dans la base initiale il y a seulement 17 variables, *mtry* variera entre 1 et 17. Pour le cas des variables de bruit, *mtry* vaudra entre 50 et 400 en pas de 50 et en rajoutant 20 car il s'agit de la valeur par défaut dans *randomForestSRC*.

Enfin, *VSURF* sera réalisé avec les paramètres qui minimisent l'erreur dans l'optimisation des paramètres et ceux de la fonction définit par défaut, soient 50 forêts de réalisées pour l'étape d'élimination préliminaire et 25 forêts pour l'étape d'interprétation et de prédiction.

2.5.2.2 Simple

La simulation dite simple est composée de 600 variables indépendantes et identiquement distribuées (iid) ayant une distribution exponentielle avec un taux de 1.

Afin d'avoir les pourcentages de censure souhaités, chaque cas de censure a sa valeur du taux pour la distribution exponentielle. Pour 30% de censure, le taux vaut 4.97, pour 50%, il vaut 17 et pour 70%, la valeur est de 50.

Pour les résultats de *VSURF*, les variables de bruit considérées seront celles n'entrant pas dans le schéma de simulation.

2.5.2.3 Corrélation décroissante

La simulation avec une corrélation décroissante consiste à avoir une corrélation qui décroît entre les covariables. Les 600 variables sont générées de la façon suivante :

- les 100 premières variables suivent une distribution normale avec une moyenne valant 0 et une matrice de covariance où les éléments (i, j) valent $0.9^{|i-j|}$.
- les 500 autres variables sont iid avec une loi uniforme entre -1 et 1.

Pour obtenir les 3 pourcentages de censure, les valeurs des taux pour 30%, 50% et 70% sont respectivement 0.43, 1.3 et 8.

Pour les résultats de *VSURF*, les variables de bruit considérées seront celles qui ne sont pas présentes dans le schéma de simulation.

2.5.2.4 Par groupe

La simulation dite par groupe consiste à avoir 4 groupes de variables très corrélées entre elles. Les 600 variables sont générées de la façon suivante :

- les 100 premières variables sont divisées en 4 groupes de 25 variables qui suivent chacune une distribution normale avec une moyenne valant 0 et une matrice de covariance où les éléments (i, j) valent 0.9.
- les 500 autres variables sont iid avec une loi uniforme entre -1 et 1.

Afin d'avoir les 3 pourcentages de censure, les taux de la distribution exponentielle sont 0.41 pour 30%, 0.895 pour 50% et 3.7 pour 70%.

Dans ce cas de simulation, les variables de bruit considérées pour *VSURF* sont celles n'appartenant pas au premier et deuxième groupe de variables puisqu'au sein d'un même groupe, les variables sont très corrélées entre elles.

2.5.2.5 Forte corrélation

La simulation avec une forte corrélation possède des variables avec une corrélation élevée. Les 600 variables sont générées de la façon suivante :

- les 100 premières variables suivent une distribution normale avec une moyenne valant 0 et une matrice de covariance où tous les éléments (i, j) valent 0.9.
- les 500 autres variables sont iid avec une loi uniforme entre -1 et 1.

Les taux de la distribution exponentielle sont les suivants pour obtenir les 3 pourcentages de censure : pour 30%, le taux vaut 0.5, pour 50%, il vaut 1.535 et pour 70%, il est de 5.4.

Pour cette simulation, aucune variable de bruit ne sera considérée comme sélectionnée pour *VSURF* si elle fait partie des 100 premières variables car elles sont très corrélées entre elles. En effet, comme elles sont très corrélées, on peut considérer que si on en retrouve une, c'est une bonne variable sélectionnée.

2.5.3 Données réelles

2.5.3.1 Données Cancer du sein

Les données sur le cancer du sein sont issues de l'étude réalisée par van de Vijver et al. (20). L'expression génique a été mesurée chez 260 patientes, de l'Institut néerlandais du cancer, atteintes de carcinomes mammaires primitifs, qui sont des tumeurs du sein rares. Deux évènements sont étudiés : le décès de la patiente et l'apparition de métastases. Ici, on étudie l'apparition d'un évènement quel qu'il soit. Le temps de survie considéré est le minimum entre les temps associés à chaque évènement. De plus, dans cette base de données, on possède l'expression de 11 848 gènes.

L'optimisation des paramètres *mtry* et *nodesize* est réalisée pour les valeurs de *mtry* suivantes : 109 et de 2 000 à 9 000 en pas de 1 000 et les valeurs de *nodesize* : de 1 à 10 en pas de 1 et de 20 à 70 en pas de 10 soit au total, 198 combinaisons de variables. Certaines valeurs de *mtry* ne sont pas considérées car les exécutions ont été réalisées pour les *mtry* séparément et certains cas n'ont pas eu le temps de terminer avant la fin de la rédaction de ce rapport.

Suite à l'optimisation des paramètres, la fonction *VSURF* est exécutée avec les paramètres choisis à l'étape précédente et le nombre de forêts est modifié afin d'avoir des temps de calcul raisonnables. Pour l'étape d'élimination préliminaire, 20 forêts sont réalisées et 10 forêts pour l'étape d'interprétation et de prédiction.

2.6 Logiciel

Le package, les simulations et les analyses ont été réalisés grâce au logiciel *R* v4.1.0. Ce rapport a été rédigé sur *Rmarkdown*. Afin de réduire les temps de calcul, les simulations ont été effectuées sur des serveurs (*Curta*, *Septimi*).

3 Résultats

3.1 Simulations

Les résultats de chaque simulation sont présentés ci-dessous. Pour chaque simulation, un graphique de l'optimisation des paramètres *mtry* et *nodesize* sera présenté ainsi qu'un tableau récapitulant les variables sélectionnées par *VSURF*.

3.1.1 Données PBC sans et avec variables de bruit

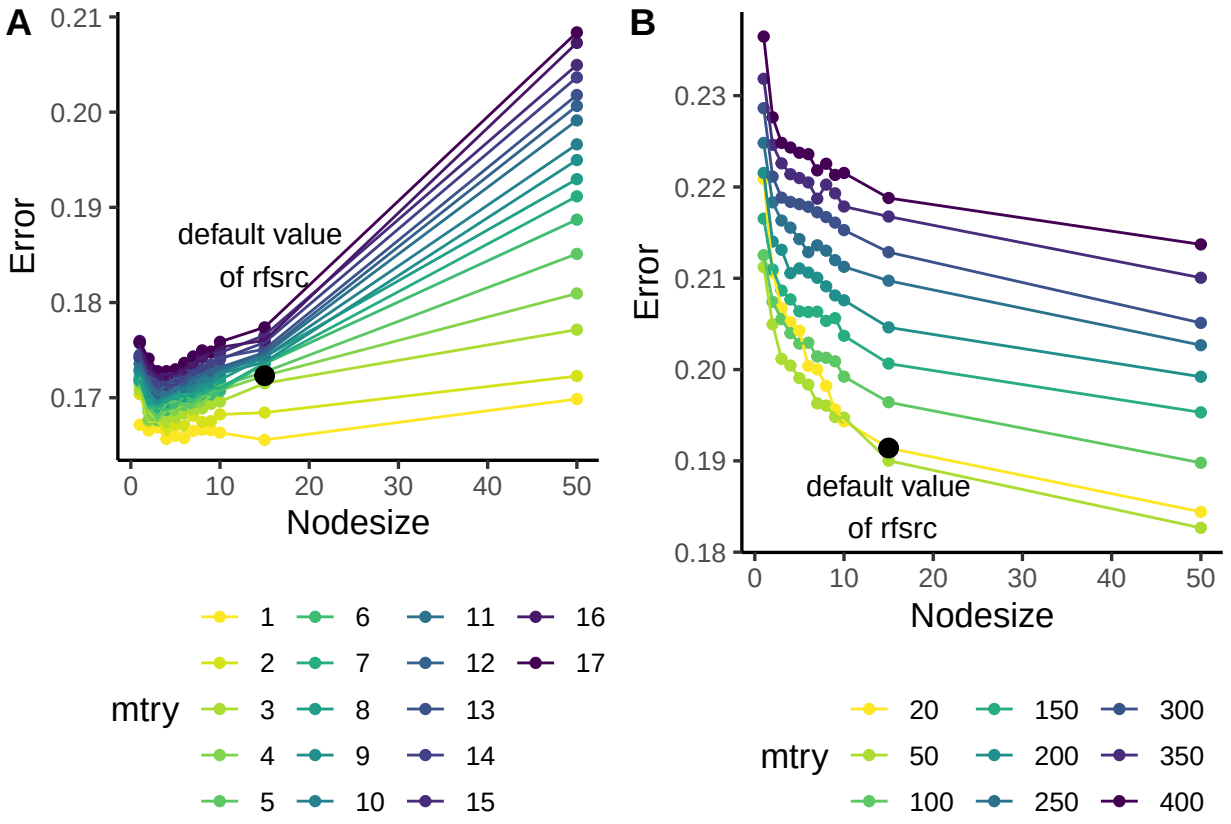

Figure 2: Graphiques de l'optimisation des paramètres mtry et nodesize pour les données PBC sans (A) et avec variables de bruit (B).

L'optimisation des paramètres *mtry* et *nodesize* pour les données de PBC avec et sans variables de bruit est présentée à la figure 2. Le graphique A représente celui pour les données sans l'ajout de variables de bruit et le B celui avec l'ajout de variables de bruit. Pour le graphique A, on remarque que les erreurs augmentent quand le *mtry* croît. Les paramètres qui minimisent l'erreur sont *mtry* égal à 1 et *nodesize* à 15. En revanche, la valeur par défaut de *rfsrc*, soit *mtry* à 4 et *nodesize* à 15 n'est pas optimale. Quant au graphique B, on voit que l'erreur décroît quand le *nodesize* augmente à l'inverse de ce qu'on peut penser et du graphique A. La combinaison de paramètres optimale est pour *mtry* faible et *nodesize* élevé à 50. Un *nodesize* élevé signifie qu'il y

aura beaucoup d'évènement unique dans le noeud terminal et que l'arbre sera peu profond et un *mtry* faible signifie qu'à chaque noeud peu de variables seront sélectionnées aléatoirement. Ici, on sait que seules 17 variables sur les 400 sont bonnes et si on sélectionne aléatoirement 50 variables sur 400, il y a moins de chance de tirer une bonne variable donc les arbres seront découpés avec les variables de bruit. De plus, on remarque que pour *mtry* égal à 400, l'écart entre *nodesize* égal à 1 et 50 est plus petit que pour *mtry* égal à 50. La valeur par défaut n'est toujours pas idéale mais n'est pas très éloignée de celle optimale.

Table 1: Sélection de variables avec VSURF pour les données PBC sans et avec variables de bruit.

Variables de bruit	Nombre de VS[1]	Nombre de VS[1] de bruit	Numéro des bonnes VS[1]
Interprétation			
Sans	11	0	2 - 4 -> 8 - 10 - 11 - 13 - 16 - 17
Avec	43	27	2 -> 17
Prédiction			
Sans	9	0	2 - 4 - 5 - 7 - 8 - 10 - 11 - 16 - 17
Avec	9	0	2 - 4 - 5 - 7 - 8 - 11 - 12 - 13 - 17

[1] Variables sélectionnées

La sélection de variables avec *VSURF* pour les données PBC sans et avec variables de bruit est récapitulée à la table 1. Pour l'étape d'interprétation, lorsqu'il n'y a pas de variables de bruit ajoutées on n'en retrouve donc pas et presque toutes les variables de PBC sont sélectionnées. En revanche quand les variables de bruit sont présentes, on retrouve 27 variables de bruit sur les 43 sélectionnées à l'étape d'interprétation. De plus, toutes les variables des données sont sélectionnées sauf la variable n°1. Concernant la prédiction, dans les 2 cas, il n'y a pas de variables de bruit et on retrouve 9 variables sélectionnées. En comparant les variables sélectionnées, 7 sont communes aux 2 cas.

3.1.2 Simple

Pour la simulation dite simple, l'optimisation des paramètres *mtry* et *nodesize* est représentée par la figure 3. On remarque que la meilleure combinaison avec l'erreur la plus faible pour chaque cas de censure est un *mtry* égal à 100 et un *nodesize* variant entre 10 et 25 en fonction des taux de censure. La valeur par défaut dans *rfsrc* est 25 pour *mtry* et 15 pour *nodesize* représentée par le point noir sur les graphiques, on voit que ces valeurs ne sont pas optimales dans cette simulation.

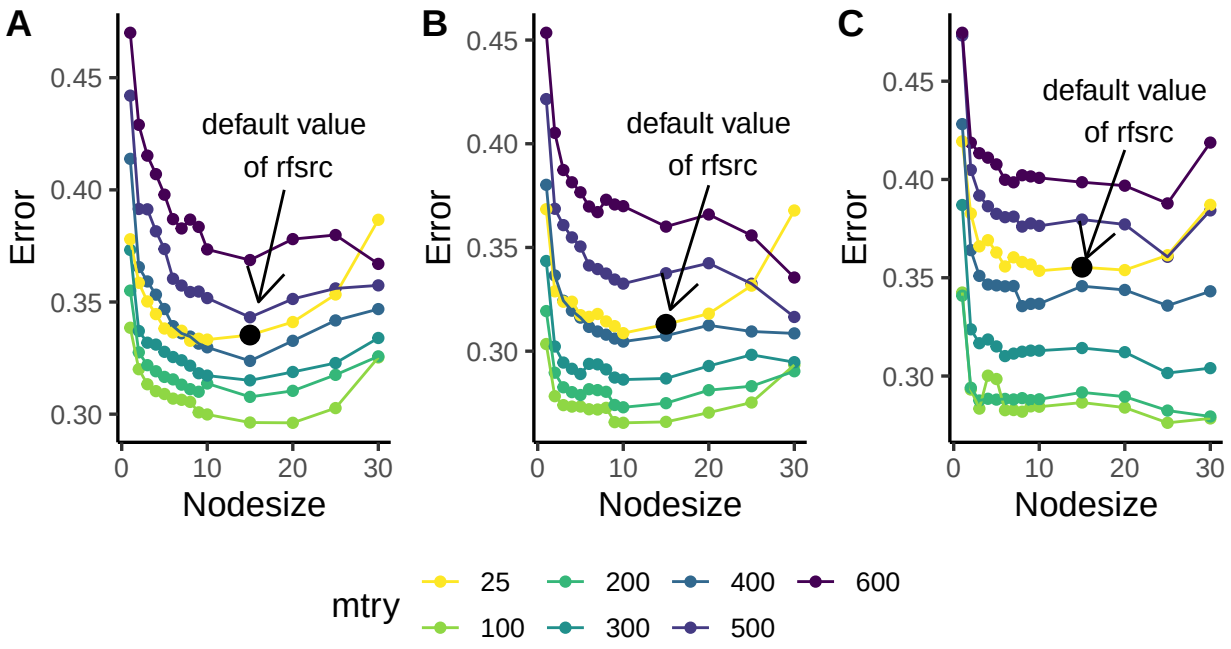

Figure 3: Graphiques de l'optimisation des paramètres mtry et nodesize pour la simulation simple pour 30% (A), 50% (B) et 70% (C) de censure.

Lorsque la fonction *VSURF* est réalisée sur les données, la table 2 récapitule les variables sélectionnées. Pour l'étape d'interprétation, on retrouve bien les variables à partir desquelles la simulation a été effectuée soit V1, V2 et V45. On remarque pour 50% de censure, il y a plus de variables de bruit sélectionnées comparées aux 30 et 70% de censure. Pour ces derniers pourcentages, les variables sélectionnées à l'étape de prédiction sont les mêmes que celles sélectionnées à celle d'interprétation.

Table 2: Sélection de variables avec VSURF pour la simulation simple pour 30, 50 et 70 de censure.

Censure (%)	Nombre de VS[1]	Nombre de VS[1] de bruit	Numéro des bonnes VS[1]
Interprétation			
30	3	1	2 - 45
50	45	42	1 - 2 - 45
70	4	2	1 - 2
Prédiction			
30	3	1	2 - 45
50	14	11	1 - 2 - 45
70	4	2	1 - 2

[1] Variables sélectionnées

3.1.3 Par groupe

Pour la simulation par groupe, la figure 4 présente l'optimisation des paramètres *mtry* et *nodesize*. Sur la figure A pour 30% de censure, on remarque que le minimum est pour *mtry* et *nodesize* respectivement égaux à 100 et 2 avec une erreur autour de 19%. La valeur par défaut de *rfsrc* n'est, ici, pas l'optimum pour ces paramètres. Pour 50% de censure (Figure B), l'erreur minimum est aussi atteinte autour de 18,5% pour *mtry* égal à 100 et *nodesize* à 2. La valeur par défaut n'est toujours pas optimale dans ce cas. Dans celui de 70% de censure (Figure C), la valeur par défaut pour *mtry* est optimale soit 25 quant à la valeur de *nodesize*, elle vaut 3.

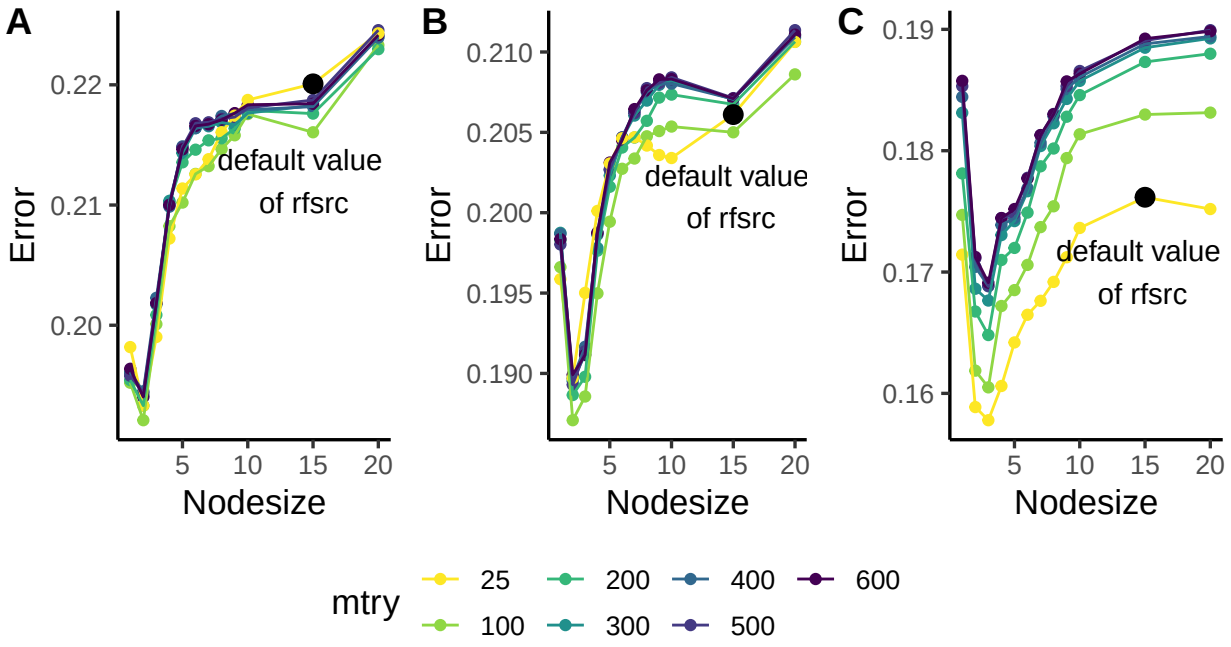

Figure 4: Graphiques de l'optimisation des paramètres mtry et nodesize pour la simulation avec une corrélation par groupe pour 30% (A), 50% (B) et 70% (C) de censure.

Le récapitulatif des variables sélectionnées par *VSURF* se trouve à la table 3. Pour l'étape d'interprétation des 30% et 50% de censure, on trouve respectivement 65 variables sélectionnées dont 15 variables de bruit et 58 dont 8 variables de bruit. Parmi les bonnes variables sélectionnées, on retrouve, dans les 2 cas de censure, les 25 variables du premier groupe (1 à 25) et les 25 du deuxième groupe (26 à 50). Pour 70% de censure, 26 variables sont sélectionnées, il n'y a aucune variable de bruit. Parmi les 26 variables sélectionnées, on retrouve les 25 variables du premier groupe et la variable n°31 appartenant au deuxième groupe. Pour l'étape de prédiction, le nombre de variables diminuent pour 30% de censure, on a 3 variables sélectionnées dont 1 de bruits, pour 50%, il y a 6 variables sélectionnées dont 2 de bruit et enfin pour 70%, on a 3 variables sélectionnées et 0 de bruit. Ici, les bonnes variables sélectionnées ne sont pas celles du modèle mais comme elles sont très corrélées entre elles, on considère qu'on trouve les bonnes variables. De plus, il y a au moins 1 variable sélectionnée du premier groupe et 1 du deuxième groupe.

Table 3: Sélection de variables avec VSURF pour la simulation par groupe pour 30, 50 et 70 de censure.

Censure (%)	Nombre de VS[1]	Nombre de VS[1] de bruit	Numéro des bonnes VS[1]
Interprétation			
30	65	15	1 -> 50
50	58	8	1 -> 50
70	26	0	1 -> 25 - 31
Prédiction			
30	3	1	16 - 26
50	6	2	21 - 23 - 31 - 37
70	3	0	7 - 20 - 31

[1] Variables sélectionnées

La figure 5 correspond à la sortie graphique de *VSURF* où chaque graphique représente une étape de la méthode. Les 2 premiers graphiques du haut correspondent à l'étape 1 soit l'élimination préliminaire. Le résultat du classement de l'importance des variables est dessiné sur le graphique en haut à gauche. On remarque que les vraies variables sont significativement plus importantes que les variables de bruit. A partir de ce classement, on procède à l'élimination des variables à l'aide des écart-types des VIMP correspondants. Une courbe des écart-types est contruite par un modèle de CART ajusté à cette courbe pour estimer une valeur de seuil pour les VIMP. Cette courbe est représentée en vert sur le graphique en haut à droite et le seuil des VIMP est fixé à la valeur minimale de cette courbe. Ce minimum est ensuite reporté sur le graphique en haut à gauche par la ligne rouge et les variables dont la valeur de VIMP est supérieure à ce seuil sont retenues. Parmi les variables retenues, on distingue bien les différents groupes où chaque escalier représente un groupe de 25 variables. On retrouve bien ces groupes dans la table 3, les variables des 2 premiers groupes et les variables suivantes sont considérées comme variables de bruit.
La figure en bas à gauche représente les erreurs OOB des forêts aléatoires en survie des modèles emboîtés à partir du modèle ne comportant que la variable la plus importante, et se finissant par celui impliquant toutes les variables sélectionnées. Sur cette figure, on remarque que l'erreur diminue en escalier. Chaque escalier représente aussi chaque groupe de variables et on sélectionne les variables dont le modèle a la plus faible erreur OOB, ici, le modèle avec 65 variables.
Pour finir avec le graphique en bas à droite pour l'étape de prédiction, on construit une suite de modèles en introduisant séquentiellement les variables où une variable n'est ajoutée que si le gain de précision dépasse un certain seuil. Dans ce cas-ci, le modèle final comporte 3 variables, V16, V26 et une variable de bruit (Table 3).

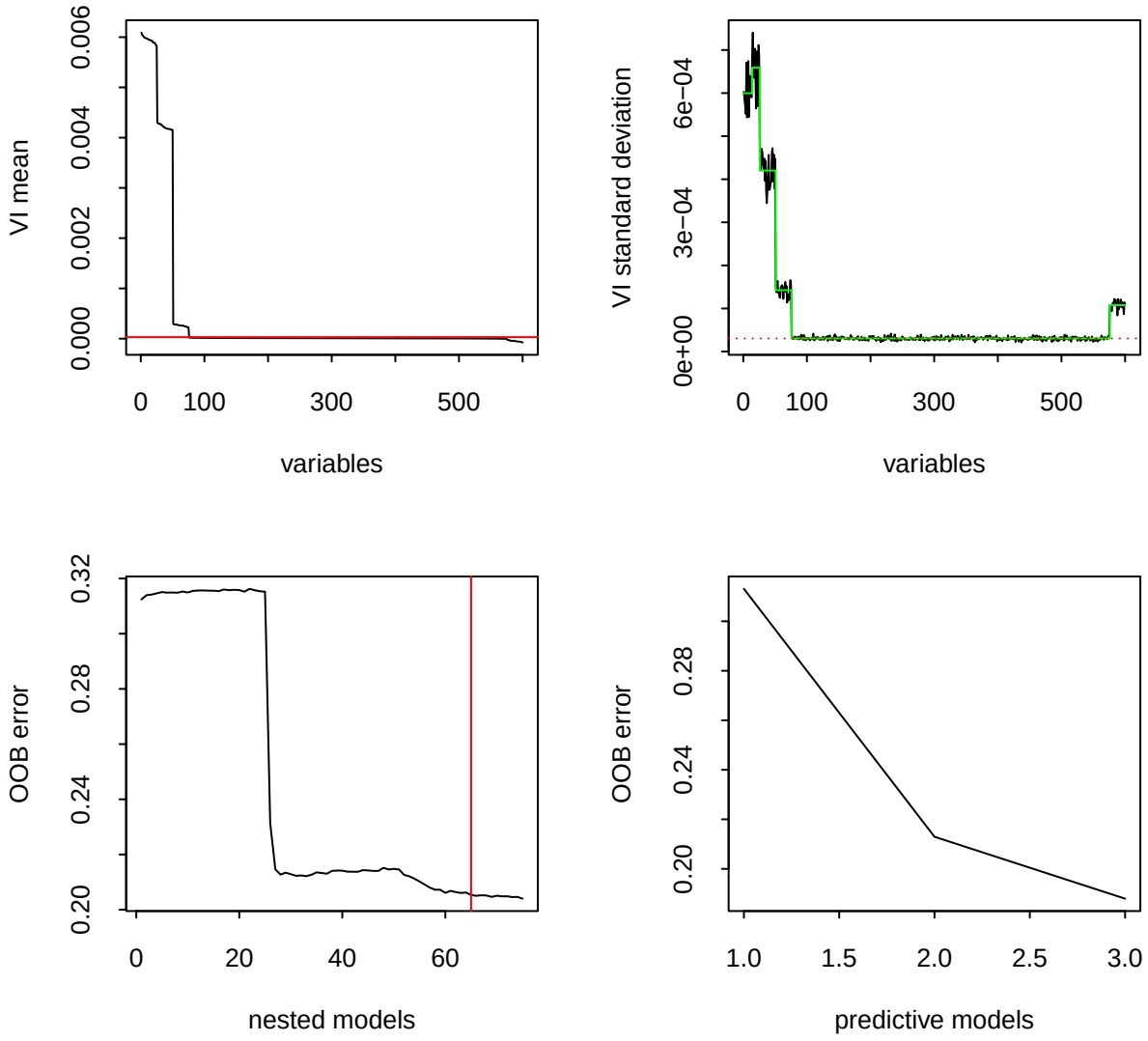

Figure 5: Sortie graphique de VSURF pour la simulation avec une corrélation par groupe pour 30% de censure.

3.1.4 Corrélation décroissante

Pour la simulation avec corrélation décroissante, l'optimisation des paramètres *mtry* et *nodesize* est présentée par la figure 6. On remarque que les erreurs pour les 3 cas de censure sont autour de 22% et la valeur par défaut de *mtry* égal à 25 est nettement plus important que pour les autres valeurs testées. Pour les censures à 30 (A) et 50% (B), le minimum de l'erreur est pour *mtry* égal à 200 et *nodesize* à 5. Pour la censure à 70% (C), le paramétrage optimum est pour *mtry* et *nodesize* valant respectivement 100 et 15.

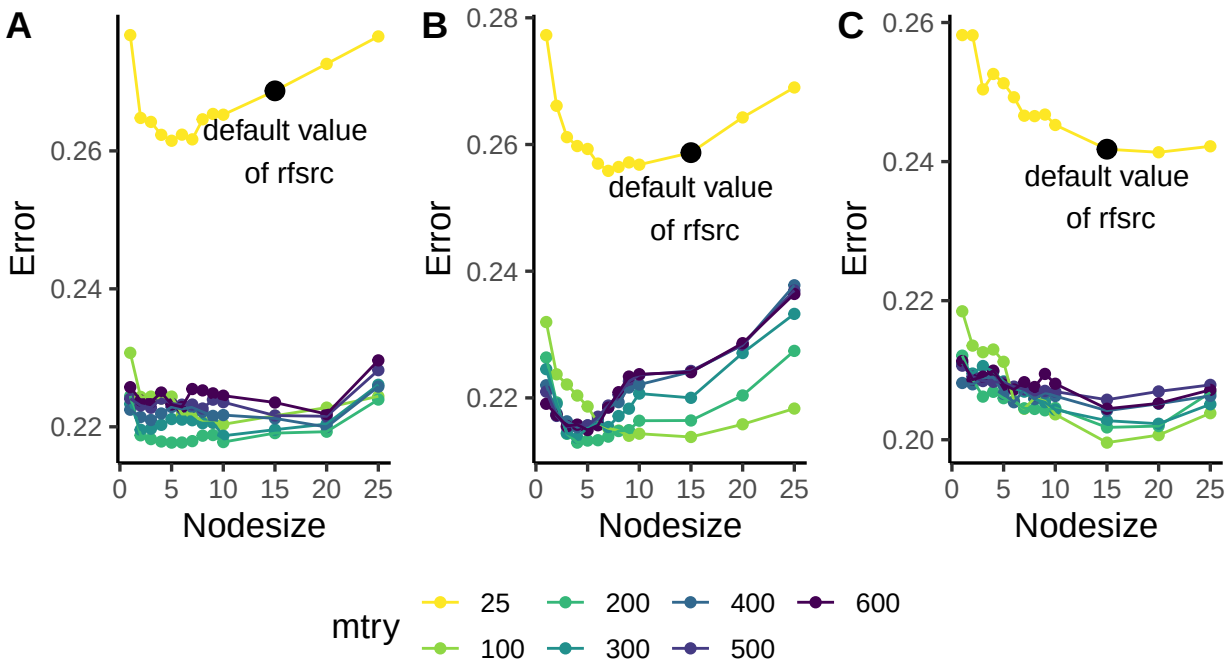

Figure 6: Graphiques de l'optimisation des paramètres mtry et nodesize pour la simulation avec une corrélation décroissante pour 30% (A), 50% (B) et 70% (C) de censure.

La table 4 présente les résultats des variables sélectionnées par VSURF pour la simulation avec une corrélation décroissante. Pour l'interprétation, les 3 cas de censure se rapprochent car on trouve presque le même nombre de variables sélectionnées et de variables de bruit sélectionnées ainsi que les mêmes bonnes variables sélectionnées. Concernant la prédiction pour 30 et 50% de censure, on trouve respectivement 7 variables sélectionnées dont 4 de bruit et 8 variables sélectionnées dont 5 de bruit. Pour ces pourcentages de censure, on récupère bien les 3 variables du schéma de simulation. En revanche pour 70% de censure, il y a 5 variables sélectionnées dont 3 de bruit et on trouve 2 bonnes variables sur les 3 intialement inclues dans le schéma.

Table 4: Sélection de variables avec VSURF pour la simulation avec une corrélation décroissante pour 30, 50 et 70 de censure.

Censure (%)	Nombre de VS[1]	Nombre de VS[1] de bruit	Numéro des bonnes VS[1]
Interprétation			
30	46	43	1 - 2 - 45
50	45	42	1 - 2 - 45
70	45	42	1 - 2 - 45
Prédiction			
30	7	4	1 - 2 - 45
50	8	5	1 - 2 - 45
70	5	3	1 - 45

[1] Variables sélectionnées

3.1.5 Forte corrélation

L'optimisation des paramètres *mtry* et *nodesize* est présentée à la figure 7 pour la simulation avec une corrélation élevée entre les variables. On remarque que la valeur par défaut de *mtry*, soit 25, est la valeur qui minimise l'erreur dans les 3 cas de censure. En ce qui concerne *nodesize*, on ne prendra pas la valeur par défaut car *nodesize* autour de 5 minimise les erreurs dans tous les cas de censure.

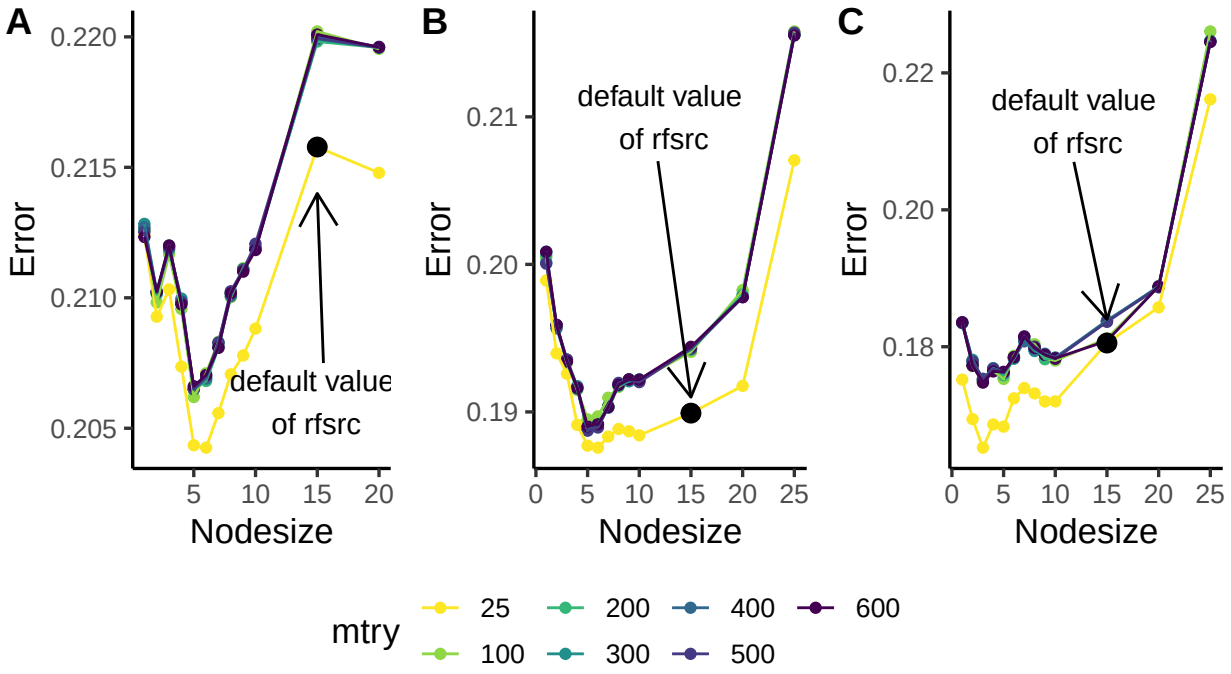

Figure 7: Graphiques de l'optimisation des paramètres mtry et nodesize pour la simulation avec une forte corrélation pour 30% (A), 50% (B) et 70% (C) de censure.

En ce qui concerne la sélection de variables avec *VSURF*, la table 5 regroupe les résultats associés. Pour 30 et 50% de censure, seule une variable a été sélectionnée pour l'interprétation et la prédiction qui est identique car il y a une forte corrélation entre les 100 premières variables. En revanche pour les 70% de censure, on récupère 109 variables sélectionnées dont 9 variables de bruit pour l'étape d'interprétation. Ici, les bonnes variables sélectionnées sont les 100 premières variables de la simulation pour lesquelles il y a une corrélation élevée. Pour l'étape de prédiction, 3 variables sont gardées dont 2 variables de bruits. La bonne variable sélectionnée est la n°44.

Table 5: Sélection de variables avec VSURF pour la simulation avec une forte corrélation pour 30, 50 et 70 de censure.

Censure (%)	Nombre de VS[1]	Nombre de VS[1] de bruit	Numéro des bonnes VS[1]
Interprétation			
30	1	0	47
50	1	0	23
70	109	9	1 -> 100.
Prédiction			
30	1	0	47
50	1	0	23
70	3	2	44

[1] Variables sélectionnées

3.2 Données Cancer du sein

L'optimisation des paramètres *mtry* et *nodesize* est présentée à la figure 8 pour les données sur le cancer du sein. On distingue deux parties, la première regroupe les valeurs de *mtry* supérieures à 2 000 et la seconde représente la valeur de *mtry* égale à 109, la valeur par défaut de *rfsrc* qui a l'erreur minimale. Pour *nodesize*, on le choisirait entre 1 et 10 avec une préférence pour 5 qui optimise le mieux l'erreur. Au vu l'écart entre les 2 groupes de courbes, on aurait tendance à vouloir tester les *mtry* variant entre 100 et 1 000.

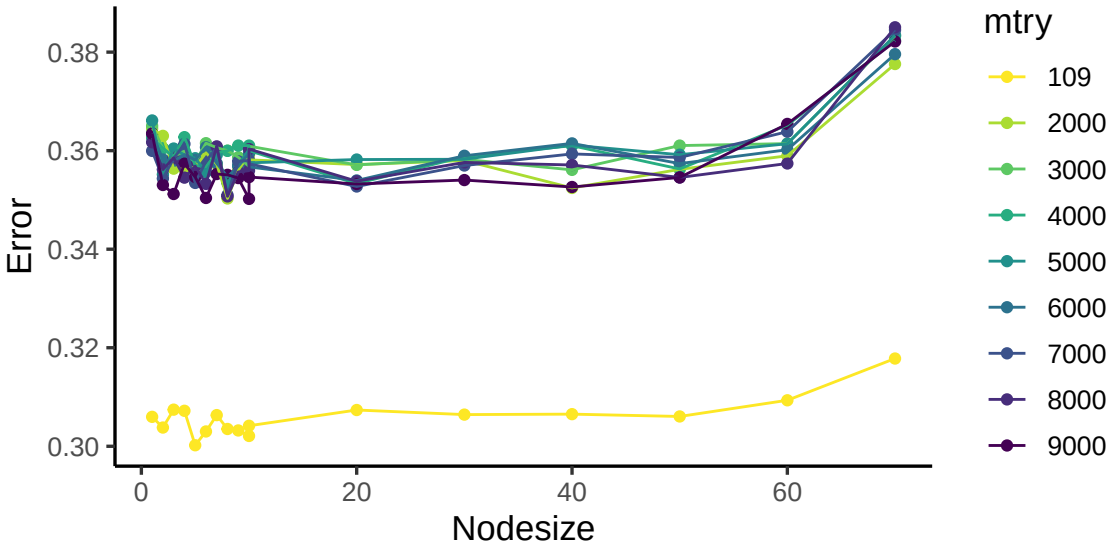

Figure 8: Graphique de l'optimisation des paramètres mtry et nodesize pour les données du cancer du sein.

4 Discussion

4.1 Résultats principaux

Ce stage portait sur l'adaptation méthodologique d'une méthode de sélection de variables (*VSURF*) pour l'analyse de données de survie en grande dimension et sur l'étude du comportement et des performances de la méthode à l'aide de simulations et d'applications sur des données réelles.

Lors de l'étude du comportement de la méthode, deux éléments ont été étudiés, le premier est l'optimisation des paramètres *mtry* et *nodesize* et le second la sélection des variables par *VSURF*.

Pour l'optimisation des paramètres, les valeurs par défaut de la fonction *rfsrc* ne sont pas optimales pour quasiment toutes les simulations même si les erreurs associées ne sont pas très éloignées des valeurs optimales. En fonction des données, on préférera un *nodesize* faible ou plus élevée et idem pour *mtry*, c'est pour quoi l'optimisation des paramètres est importante avant d'utiliser *VSURF*.
Cette étape est aussi importante afin de regarder les erreurs des forêts car si les erreurs sont supérieures à 50%, on se demande s'il est intéressant d'utiliser *VSURF* sur les données. En effet, ce fort taux d'erreur peut entrainer une mauvaise sélection de variables.

De manière générale, la fonction *VSURF* sélectionne peu de variables de bruits notamment à l'étape de prédiction qui est la plus sélective. De plus, dans presque toutes les simulations, les variables du schéma de simulation sont retrouvées soit V1, V2 et V45 et dans le cas des données PBC avec l'ajout de variables de bruit, on retrouve bien les variables initialement présentes dans les données.

4.2 Points forts et limites

Cette étude présente de nombreux avantages. Tout d'abord, le premier avantage est l'adaptation d'une méthode de sélection de variables, *VSURF*, à un type de données supplémentaires, les données de survie en grande dimension. De plus, l'étude de *VSURF* a mis en évidence une bonne sélection des variables.

Ensuite, les quatre schémas de simulation apportent une variété d'application à la méthode avec 3 cas de censure et différentes corrélations. Au total, 12 cas de simulation ont été réalisés et étudiés. De plus, l'application aux données réelles a été effectuée sur les données PBC et du cancer du sein ce qui apporte une étude supplémentaire à celles des simulations.

De plus, l'importance de l'optimisation des paramètres *mtry* et *nodesize* a été mise en avant par les graphiques de chaque simulation. Avec cette optimisation, cela permet d'avoir la meilleure combinaison de paramètres afin d'utiliser *VSURF*.

Enfin, les forêts aléatoires en survie existent depuis plusieurs années mais peu de méthodes de sélection de variables ont été développées. L'article d'Ishwaran et al. (5) a introduit une méthode de sélection de variable à partir de la profondeur minimale d'un sous-arbre maximal. L'application de *VSURF* aux RSF est donc la première méthode de sélection de variables utilisant l'importance de variables par permutation. *VSURF* peut être une alternative aux modèles de Cox pénalisés avec une pénalité de type Lasso (1) initialement utilisés pour la sélection de variables en données de grande dimension en survie.

Cependant, l'étude présente quelques limites. Pour commencer, le temps d'exécution des fonctions liées aux forêts est important notamment pour l'étape d'optimisation car on réalise 20

forêts aléatoires en survie pour chaque combinaison de *mtry* et *nodesize*. Tout d'abord, les temps de calcul ont été approximés par tâtonnement donc sur les serveurs plusieurs exécutions n'ont pas été menées jusqu'au bout par manque de temps avant de trouver le temps suffisant aux exécutions. Pour donner un ordre d'idées du temps d'exécution, on se trouve autour de 1h et 3h pour les simulations et 29h pour les données PBC avec variables de bruit. C'est pourquoi toutes les valeurs de *mtry* et *nodesize* ne sont pas utilisées pour l'optimisation mais seulement un sous-ensemble de valeurs. De plus, pour les exécutions sur le serveur Curta, il pouvait y avoir un temps d'attente avant que le "job" se réalise donc le temps d'attente et celui d'exécution cumulés entraînent un long moment avant la récupération des résultats.

Ensuite, au départ du stage, l'application aux données réelles devait être réalisée sur les données TRANSCAN, étudiant les données de patients atteints de gliomes. Or, les erreurs des forêts aléatoires en survie étaient supérieures à 50% donc *VSURF* n'a pas été effectué car cela revient à tirer aléatoirement les variables des données. Beaucoup de temps a été passé sur les données TRANSCAN donc il restait peu de temps pour réaliser l'analyse sur les données du cancer. De ce fait, seuls les résultats de l'optimisation pour ces données sont sortis. Le temps d'exécution est très long pour *VSURF* donc pour ce rapport les résultats ne sont pas présents mais il est possible que les résultats soient intégrés à l'oral.

Enfin, une étude plus poussée avec une validation croisée pourrait être réalisée. Elle permettrait de stabiliser les variables sélectionnées et d'avoir des résultats plus robustes. De plus, une étude de comparaison de méthode entre celle-ci et la méthode usuelle, les modèles de Cox pénalisés pourrait être envisagée.

5 Conclusion

5.1 Conclusion du stage

Pour conclure, l'adaptation de *VSURF* propose une méthode de sélection de variables à partir des forêts aléatoires pour les données de survie en grande dimension.
A travers les différents cas de simulation, on remarque que la sélection de variables par *VSURF* sur des données de survie en grande dimension est bonne car en générale, on retrouve bien les variables qui font partie du schéma de simulation. Cette méthode ressort donc bien les variables souhaitées dans les simulations. Néanmoins, avant de procéder à la sélection des variables, il ne faut pas oublier d'optimiser les paramètres des forêts aléatoires, *mtry* et *nodesize*, car leur optimisation peut jouer un rôle sur la sélection des variables par *VSURF*.

Cette méthode permettra, par exemple, d'identifier des gènes ayant une valeur pronostique et un pouvoir prédictif élevés chez des patients atteints de cancer ou d'autres maladies dans les essais cliniques.

5.2 Conlusion personnelle

Ce mémoire synthétise les réalisations effectuées durant mon stage. Ce stage a été très enrichissant car j'ai pu découvrir plusieurs côtés de la biostatistique. Tout d'abord, la recherche bibliographique autour du sujet est une étape importante afin d'avoir les connaissances nécessaires pour poursuivre le projet puis, les différentes étapes de la création/adaptation de package sous R aux résultats qui en ressortent. Cela a été une première pour moi de contribuer au développement d'un package R et cela m'a beaucoup intéressée. De plus, j'ai pu appliquer les méthodes vues durant mes études comme les forêts aléatoires en étoffant mes connaissances autour de cette méthode et des variantes existantes. L'application de la méthode m'a permis d'aller au bout de l'adaptation et de réellement voir l'impact de celle-ci sur les données. Toutes ces étapes m'ont permis de comprendre le cheminement autour d'un package R.

Malgré les difficultés dues aux temps de calcul, les exécutions sur serveurs ont été une nouveauté pour moi et j'ai apprécié travailler dessus car cela permet de réduire les temps de calcul de façon drastique même si dans certains cas, ils restent longs. L'estimation du temps nécessaire pour une exécution fut un peu laborieuse car pour certaines exécutions, il pouvait manquer 1 heure sur 20 heures demandées mais l'exécution s'arrêtait et il fallait recommencer du début sans compter le temps d'attente avant l'exécution. Donc cela pouvait être très frustrant lorsqu'on attendait des résultats. Les serveurs m'ont aussi permis de revoir les lignes de commande que j'avais pu rencontrer lors de ma Licence.

Références

1. Simon N, Friedman J, Hastie T, Tibshirani R. Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent. Journal of Statistical Software. 2011;39(5).

2. Ishwaran H, Kogalur UB, Blackstone EH, Lauer MS. Random survival forests. The Annals of Applied Statistics. 2008;2(3).

3. Ishwaran H, Gerds TA, Kogalur UB, Moore RD, Gange SJ, Lau BM. Random survival forests for competing risks. Biostatistics. 2014;15(4):757-73.

4. Gilhodes J, Zemmour C, Ajana S, Martinez A, Delord J-P, Leconte E, et al. Comparison of variable selection methods for high-dimensional survival data with competing events. Computers in Biology and Medicine. 2017;91:159-67.

5. Ishwaran H, Kogalur UB, Gorodeski EZ, Minn AJ, Lauer MS. High-Dimensional Variable Selection for Survival Data. Journal of the American Statistical Association. 2010;105(489):205-17.

6. Miao F, Cai Y-P, Zhang Y-T, Li C-Y. Is Random Survival Forest an Alternative to Cox Proportional Model on Predicting Cardiovascular Disease? Dans: 6th European Conference of the International Federation for Medical and Biological Engineering. Cham: Springer International Publishing; 2015. p. 740-3.

7. Gilhodes J, Dalenc F, Gal J, Zemmour C, Leconte E, Boher J-M, et al. Comparison of Variable Selection Methods for Time-to-Event Data in High-Dimensional Settings. Computational and Mathematical Methods in Medicine. 2020;2020:1-3.

8. Pang H, George SL, Hui K, Tiejun Tong. Gene Selection Using Iterative Feature Elimination Random Forests for Survival Outcomes. IEEE/ACM Transactions on Computational Biology and Bioinformatics. 2012;9(5):1422-31.

9. Dietrich S, Floegel A, Troll M, Kühn T, Rathmann W, Peters A, et al. Random Survival Forest in practice: a method for modelling complex metabolomics data in time to event analysis. International Journal of Epidemiology. 2016;45(5):1406-20.

10. Wang H, Li G. A Selective Review on Random Survival Forests for High Dimensional Data. Quantitative Bio-Science. 2017;36(2):85-96.

11. Genuer R, Poggi J-M, Tuleau-Malot C. Variable selection using Random Forests. Pattern Recognition Letters, Elsevier. 2010;31(14):2225-36.

12. Breiman L. Random forests. Machine Learning. 2001;45:5-32.

13. Genuer R, Poggi J-M. Les forêts aléatoires avec R. Rennes: Presses universitaires de Rennes; 2019.

14. Ishwaran H, Lu M. Random Survival Forests. Dans: Wiley StatsRef: Statistics Reference Online. American Cancer Society; 2019. p. 1-3.

15. Harrell FE, Califf RM, Pryor DB, Lee KL, Rosati RA. Evaluating the yield of medical tests. JAMA. 1982;247(18):2543-6.

16. Ishwaran H, Kogalur UB. randomForestSRC: Random Forests for Survival, Regression, and Classification (RF-SRC). 2017.

17. Breiman L, Cutler A. Random forest manual. 2004;

18. Genuer R, Poggi J-M, Tuleau-Malot C. VSURF: An R Package for Variable Selection Using Random Forests. The R Journal. 2015;7(2):19.

19. Hu C, Steingrimsson JA. Personalized Risk Prediction in Clinical Oncology Research: Applications and Practical Issues Using Survival Trees and Random Forests. Journal of Biopharmaceutical Statistics. 2018;28(2):333-49.

20. Vijver MJ van de, He YD, Veer LJ van 't, Dai H, Hart AAM, Voskuil DW, et al. A Gene-Expression Signature as a Predictor of Survival in Breast Cancer. New England Journal of Medicine. 2002;347(25):1999-2009.

Résumé

Aujourd'hui, la quantité d'informations recueillies à chaque instant a augmenté de manière considérable grâce aux progrès de la science. Les bases de données disponibles étant de plus en plus grandes, leurs analyses peuvent poser problème car les méthodes existantes ne sont pas toutes adaptées aux données de grande dimension. De plus, si on s'intéresse à ce type de données de survie et qu'on cherche à expliquer quelles sont les variables les plus prédictrices ou associées à la survenue de l'évènement, un problème de sélection de variable se pose et à ce jour, peu de solutions existent pour y faire face. L'objectif de ce stage est donc, tout d'abord, de faire un bilan sur les méthodes existantes dans le cadre de la sélection de variables pour données de survie de grande dimension puis de proposer l'adaptation d'une méthode de sélection de variables à partir des forêts aléatoires en survie et enfin, d'étudier son comportement et ses performances.

La méthode de sélection de variable *VSURF (Variable Selection Using Random Forests)* développée par Genuer et al. en 2010 est adaptée aux données de survie de grande dimension à l'aide des forêts aléatoires en survie (RSF) introduite par Ishwaran et al. en 2008. L'étude du comportement de la méthode est réalisée sur 4 schémas de simulation, les données PBC avec et sans ajout de variables de bruit et sur les données réelles sur le cancer du sein.

Lors de l'optimisation des paramètres des forêts aléatoires pour les diverses simulations, on remarque que chaque cas est différent et qu'il est donc important de réaliser cette étape en amont de l'exécution de *VSURF*. Pour la sélection de variables avec cette dernière, *VSURF* réussit à sélectionner les bonnes variables et peu ou pas de variables de bruit.

La méthode *VSURF* pour les données de survie de grande dimension est donc intéressante pour la sélection de variables et une méthode qui sera utile, par exemple, pour identifier des gènes ayant une valeur pronostique et un pouvoir prédictif élevés chez des patients atteints de cancer ou d'autres maladies dans les essais cliniques.

Mots-clés: survie, données de grande dimension, forêts aléatoires, sélection de variable, VSURF, simulations

---

Abstract

Today, the amount of information gathered at any given moment has increased exponentially with the advancement of science. As the databases available are increasingly large, their analysis can be problematic because not all existing methods are suitable for high-dimensional data. In addition, if we are interested in this type of survival data and we seek to explain which variables are the most predictive or associated with the occurrence of the event, a problem of variable selection arises and to date, few solutions exist to deal with it. The objective of this internship is first of all, to do a review on the existing methods of variable selection for high-dimensional survival data, then to propose an adaptation of a variable selection method with random survival forests and finally, to study its behavior and performance.

The method of variable selection *VSURF (Variable Selection Using Random Forests)* developed by Genuer et al. in 2010 is adapted to high-dimensional survival data using random survival forests (RSF) introduced by Ishwaran et al. in 2008. The study of the behavior of the method is carried out on 4 simulation schemes, the PBC data with and without addition of noise variables and on the actual data on breast cancer.

When we optimize the parameters of random forests for the various simulations, we notice that each case is different and that it is therefore important to carry out this step before of the execution of *VSURF*. For the variable selection, *VSURF* succeeds in selecting the good variables and few or no noise variables.

The *VSURF* method for high-dimensional survival data is interesting for the variable selection and a method which will be useful, for example, in identifying genes with high prognostic value and predictive power in patients with cancer or other diseases in clinical trials.

Keywords: survival, high-dimensional data, random forests, RSF, variable selection, VSURF, simulations