HEC MONTRÉAL

Mesure de l'importance de variables à partir de forêt aléatoire : Applications à la génétique

par

Antoine Main

Sciences de la gestion
(Option Analytique d'affaires)

Mémoire présenté en vue de l'obtention
du grade de maîtrise ès sciences
(M. Sc.)

Septembre 2018

HEC MONTRÉAL

Comité d'éthique de la recherche

Le 07 septembre 2017

À l'attention de :
Antoine Main, HEC Montréal

Objet : Approbation éthique de votre projet de recherche

 # Projet : 2018-2853

Titre du projet de recherche : Utilisation d'outils statistiques complexes dans le but de caractériser les interactions entre les variations génétiques et les fonctions cognitives dans la population générale.

Votre projet de recherche a fait l'objet d'une évaluation en matière d'éthique de la recherche avec des êtres humains par le CER de HEC Montréal.

Un certificat d'approbation éthique qui atteste de la conformité de votre projet de recherche à la *Politique relative à l'éthique de la recherche avec des êtres humains* de HEC Montréal est émis en date du 07 septembre 2017. Prenez note que ce certificat est **valide jusqu'au 01 septembre 2018.**

Vous devrez obtenir le renouvellement de votre approbation éthique avant l'expiration de ce certificat à l'aide du formulaire *F7 - Renouvellement annuel*. Un rappel automatique vous sera envoyé par courriel quelques semaines avant l'échéance de votre certificat.

Si des modifications sont apportées à votre projet avant l'échéance du certificat, vous devrez remplir le formulaire *F8 - Modification de projet* et obtenir l'approbation du CER avant de mettre en oeuvre ces modifications. Si votre projet est terminé avant l'échéance du certificat, vous devrez remplir le formulaire *F9 - Fin de projet ou F9a - Fin de projet étudiant*, selon le cas.

Notez qu'en vertu de la *Politique relative à l'éthique de la recherche avec des êtres humains de HEC Montréal*, il est de la responsabilité des chercheurs d'assurer que leurs projets de recherche conservent une approbation éthique pour toute la durée des travaux de recherche et d'informer le CER de la fin de ceux-ci. De plus, toutes modifications significatives du projet doivent être transmises au CER avant leurs applications.

Vous pouvez dès maintenant procéder à la collecte de données pour laquelle vous avez obtenu ce certificat.

Nous vous souhaitons bon succès dans la réalisation de votre recherche.

Le CER de HEC Montréal

HEC MONTRÉAL

Comité d'éthique de la recherche

CERTIFICAT D'APPROBATION ÉTHIQUE

La présente atteste que le projet de recherche décrit ci-dessous a fait l'objet d'une évaluation en matière d'éthique de la recherche avec des êtres humains et qu'il satisfait aux exigences de notre politique en cette matière.

Projet # : 2018-2853

Titre du projet de recherche : Utilisation d'outils statistiques complexes dans le but de caractériser les interactions entre les variations génétiques et les fonctions cognitives dans la population générale.

Chercheur principal :
Antoine Main, étudiant M. Sc.
HEC Montréal

Cochercheurs :
Jean-Louis Martineau; Guillaume Huguet

Directeur/codirecteurs :
Aurelie Labbe (HEC Montréal); Sébastien Jacquemont (CHU Ste-Justine)

Date d'approbation du projet : 07 septembre 2017

Date d'entrée en vigueur du certificat : 07 septembre 2017

Date d'échéance du certificat : 01 septembre 2018

Maurice Lemelin
Président du CER de HEC Montréal

HEC MONTRÉAL

Comité d'éthique de la recherche

Le 16 août 2018

À l'attention de :
Antoine Main

Projet # 2018-2853 – Variations génétiques et les fonctions cognitives

Titre : Mesure de l'importance de variables à partir de forêt aléatoire : Applications à la génétique

---

Bonjour,

Pour donner suite à votre demande de renouvellement, le certificat d'approbation éthique pour le présent projet a été renouvelé en date du 01 septembre 2018.

Ce certificat est valide jusqu'au 01 septembre 2019.

Vous devez donc, avant cette date, obtenir le renouvellement de votre approbation éthique à l'aide du formulaire *F7 - Renouvellement annuel.* Un rappel automatique vous sera envoyé par courriel quelques semaines avant l'échéance de votre certificat.

Si votre projet est terminé avant cette échéance, vous devrez remplir le formulaire *F9 - Fin de projet.*

Si des modifications importantes sont apportées à votre projet avant l'échéance du certificat, vous devrez remplir le formulaire *F8 - Modification de projet.*

Prenez également note que tout nouveau membre de votre équipe de recherche devra signer le formulaire d'engagement de confidentialité et que celui-ci devra nous être transmis lors de votre demande de renouvellement.

Nous vous souhaitons bon succès dans la poursuite de votre recherche.

Cordialement,

Le CER de HEC Montréal

HEC MONTRÉAL

Comité d'éthique de la recherche

RENOUVELLEMENT DE L'APPROBATION ÉTHIQUE

La présente atteste que le projet de recherche décrit ci-dessous a fait l'objet d'une évaluation en matière d'éthique de la recherche avec des êtres humains et qu'il satisfait aux exigences de notre politique en cette matière.

Projet # : 2018-2853 - Variations génétiques et les fonctions cognitives

Titre du projet de recherche : Mesure de l'importance de variables à partir de forêt aléatoire : Applications à la génétique

Chercheur principal :
Antoine Main, étudiant M. Sc.
HEC Montréal

Cochercheurs :
Jean-Louis Martineau; Guillaume Huguet

Directeur/codirecteurs :
Aurélie Labbe; Sébastien Jacquemont

Date d'approbation du projet : 07 septembre 2017

Date d'entrée en vigueur du certificat : 01 septembre 2018

Date d'échéance du certificat : 01 septembre 2019

Maurice Lemelin
Président du CER de HEC Montréal

HEC MONTRÉAL

Comité d'éthique de la recherche

ATTESTATION D'APPROBATION ÉTHIQUE COMPLÉTÉE

La présente atteste que le projet de recherche décrit ci-dessous a fait l'objet des approbations en matière d'éthique de la recherche avec des êtres humains nécessaires selon les exigences de HEC Montréal.

La période de validité du certificat d'approbation éthique émis pour ce projet est maintenant terminée. Si vous devez reprendre contact avec les participants ou reprendre une collecte de données pour ce projet, la certification éthique doit être réactivée préalablement. Vous devez alors prendre contact avec le secrétariat du CER de HEC Montréal.

Projet # : 2018-2853 **-** Variations génétiques et les fonctions cognitives

Titre du projet de recherche : Mesure de l'importance de variables à partir de forêt aléatoire : Applications à la génétique

Chercheur principal :
Antoine Main, étudiant M. Sc.
HEC Montréal

Cochercheurs :
Guillaume Huguet; Jean-Louis Martineau

Directeur/codirecteurs :
Aurélie Labbe; Sébastien Jacquemont

Date d'approbation initiale du projet : 07 septembre 2017

Date de fermeture de l'approbation éthique : 31 août 2018

Maurice Lemelin
Président du CER de HEC Montréal

Résumé

Contexte

L'algorithme des forêts aléatoires de Breiman (2001) est l'une des méthodes statistiques les plus utilisées et l'une des plus efficaces pour effectuer des prédictions. Il est possible d'utiliser cet algorithme pour évaluer l'importance de chaque variable explicative (VIM) utilisée dans le modèle de prédiction. Il existe plusieurs méthodes de mesure de VIM. L'objectif principal de notre étude est de synthétiser les avantages et les limites de ces différentes méthodes à travers une étude de simulations. Nous proposons par la suite d'appliquer les meilleures méthodes à un cas réel en génétique. Le but de cette application consiste à classer par ordre d'importance, des variables génétiques dans la prédiction d'une mesure d'intelligence générale (le facteur G).

Méthode

Nous avons dans un premier temps évalué 8 méthodes de mesure d'importance de variables à partir de forêts aléatoires à l'aide de 21 scénarios de simulation. Nous avons évalué les 8 méthodes en fonction de leur proximité avec une définition commune et intuitive d'importance de variable que nous avons calculé de manière théorique. Au cours des simulations, nous avons testé la fiabilité des méthodes VIM en fonction de 5 questions développées à partir des problématiques récurrentes associées aux données génétiques : (1) est-ce que les méthodes de VIM sont robustes à une augmentation du nombre de variables

non explicatives dans le modèle ? (2) Est-ce que les méthodes de VIM sont robustes à une augmentation du nombre de variables explicatives dans le modèle ? (3) Est-ce que les méthodes de VIM sont fiables quand les relations entre les variables explicatives et la variable d'intérêt sont linéaires ? (4) Est-ce que les méthodes de VIM sont fiables quand les relations entre les variables explicatives et la variable d'intérêt sont non linéaires ? (5) Comment se comportent les méthodes de VIM lorsque les variables explicatives sont corrélées entre elles ?

Nous avons ensuite évalué les différentes méthodes de VIM sur 3 jeux de données simulés fréquemment utilisés dans la littérature.

Dans un second temps, nous avons appliqué les deux mesures d'importance de variables ayant été les plus robustes dans l'étude des scénarios de simulation à une étude portant sur la génétique d'une certaine mesure d'intelligence. Pour cela, nous avons utilisé le jeu de données "**Generation of Scotland**". Le jeu de données comprend de l'information génétique, environnementale et phénotypique provenant de 12,743 individus.

Conclusion

Notre étude de simulation montre que les méthodes de VIM peuvent être appliquées en toute confiance dans la majorité des scénarios simulés. Nous constatons cependant que les méthodes de VIM ne sont pas fiables lorsqu'il y a de la corrélation entre les variables explicatives. De plus, nous constatons que les méthodes de VIM sont moins robustes lorsqu'un nombre élevé de variables explicatives ont une relation non linéaire avec la variable d'intérêt.

Les résultat des deux méthodes de VIM utilisées dans le cas de l'application en génétique se sont révélés concordants. Les deux méthodes ont permis d'observer quel groupe de variables génétiques est le plus importants pour prédire le facteur G. Elles ont également permis de montrer que l'environnement dans lequel sont conditionnés les individus a un impact non négligeable sur la mesure d'intelligence générale.

Mots-clés : Forêt aléatoire, mesure d'importance de variables (VIM), apprentissage automatique, génétique.

Table des matières

Liste des tableaux

Liste des figures

Liste des abréviations

CART Arbre de classification et de régression proposé par Breiman en 1984. L'abréviation provient de l'anglais "*Classification And Regression Trees*".

CTREE Arbres d'inférence conditionnelles. L'abréviation provient de l'anglais "*Conditional Inference Trees*".

VIM Mesure de l'importance des variables. L'abréviation provient de l'anglais "*Variable importance measure*".

VIM_CART_Standardisée Méthode de mesure de l'importance des variables calculée selon la permutation de Breiman standardisée construite à partir d'une forêt aléatoire composée d'arbres CART.

VIM_CART_Non_Standardisée Méthode de mesure de l'importance des variables calculée selon la permutation de Breiman non standardisée construite à partir d'une forêt aléatoire composée d'arbres CART.

VIM_Impureté Méthode de mesure de l'importance des variables calculée selon la méthode de l'impureté construite à partir d'une forêt aléatoire composée d'arbres CART.

VIM_Ishwaran_Aléatoire Méthode de mesure de l'importance des variables calculée selon la méthode d'ajout du bruit aléatoire de Ishwaran.

VIM_Ishwaran_Noeud_Opposé Méthode de mesure de l'importance des variables calculée selon la méthode de sélection du nœud opposé d'Ishwaran.

VIM_CTREE Méthode de mesure de l'importance des variables calculée selon la permutation de Breiman non standardisée construite à partir d'une forêt aléatoire composée d'arbres d'inférences conditionnelles.

VIM_CTREE_Strobl Méthode de mesure de l'importance des variables calculée selon la permutation de Breiman non standardisée construite à partir d'une forêt aléatoire composée d'arbres d'inférences conditionnelles avec les mêmes hyperparamètres que les forêts aléatoires de l'article de Strobl et al. (2007).

VIM_CTREE_Corrélation Méthode de mesure de l'importance des variables calculée selon la permutation de Breiman non standardisée construite à partir d'une forêt aléatoire composée d'arbres d'inférences conditionnelles corrigés pour la corrélation entre les variables explicatives.

ADN Acide DésoxyriboNucléique .

CNV Variation du nombre de copies d'un segment d'ADN entre individus de la même espèce. L'abréviation provient de l'anglais "*Copy Numbers Variant*".

Remerciements

Ce travail n'aurait pas vu le jour sans l'exigeante bienveillance de Docteur Aurélie Labbe que je remercie chaleureusement. Non seulement elle m'a introduit au laboratoire du Docteur Jacquemont au CHU de Sainte-Justine mais grâce à elle j'ai pris conscience de l'importance des statistiques dans les modèles d'apprentissage automatique. Je lui sais gré de sa disponibilité pour m'aiguiller dans mes recherches, pour répondre à mes interrogations et de la rigueur de sa supervision.

J'aimerais également témoigner ma reconnaissance au Docteur Guillaume Huguet, associé de recherche au centre de recherche de Sainte-Justine à qui je dois toutes mes connaissances en génétique. A l'aide de métaphores bien trouvées et de son expertise il a su me transmettre sa passion pour ce domaine.

Un grand merci aussi au Docteur Sébastien Jacquemont qui m'a ouvert les portes de son laboratoire, pour sa curiosité et l'intérêt qu'il a porté à mes recherches. Sans son ouverture d'esprit je n'aurais pas eu la chance de côtoyer des chercheurs de grand talent et mener à bien ce travail.

Ma reconnaissance va également au Docteur Catherine Schram qui m'a fourni de précieux conseils grâce à son expertise en statistiques et à Monsieur Martineau Jean-Louis qui a effectué une partie de la préparation des données et dont les conseils et le coaching m'ont été précieux en optimisation de code.

Je tiens également à remercier chaleureusement tous les autres chercheurs et membres du personnel du laboratoire qui ont su instaurer une ambiance bienveillante, chaleureuse et propice au travail de recherche.

J'adresse également mes remerciements à HEC Montréal qui en m'octroyant la bourse de recherche pour étudiant étranger m'a permis de mener à bien ce travail.

Enfin merci à Isabelle Main et Nafissatou Sory pour leur patient travail de relecture du mémoire.

Chapitre 1

Introduction aux concepts d'arbres de décision et de forêts aléatoires

L'objectif de ce chapitre est de présenter l'algorithme des forêts aléatoires et le concept de mesure d'importance des variables.

L'algorithme des forêts aléatoires, développé par Breiman (2001), est une des méthodes statistiques les plus utilisées et l'une des plus efficaces pour effectuer des prédictions. Au-delà de sa capacité de prédiction, cette méthode permet également de trier les variables en fonction de leur importance pour prédire la variable d'intérêt souhaitée. Mesurer l'importance de certaines variables dans le cadre des forêts aléatoires permet d'élargir l'analyse faite avec la régression linéaire classique en étudiant des relations non linéaires entre les variables explicatives et la variable à expliquer. Cet algorithme permet également d'inclure de manière implicite des relations d'interaction entre les variables explicatives et la variable dépendante lors de l'analyse (Rodenburg et al. (2008)).
L'excellente capacité de prédiction des forêts aléatoires rend cet algorithme pluridisciplinaire. Il est en effet possible de retrouver cette méthode dans divers domaines tels que la chimio-informatique (Svetnik et al. (2003)), l'écologie (Cutler et al. (2007)), le marketing (Larivière and Van den Poel (2005)), la reconnaissance d'objet (Shotton et al. (2011)), la génétique (Boulesteix et al. (2011)) et bien d'autres encore (Biau and Scornet (2016)).

Dans un premier temps, nous verrons la structure de cet algorithme en introduisant les modèles d'arbres de régression et le concept de Bagging (Breiman (1996)). Par la suite, nous verrons comment mesurer l'importance des variables explicatives dans un problème donné à partir de cet algorithme. Enfin, nous étudierons des méthodes qui permettent de sélectionner un nombre de variables limité à partir de ces mesures d'importance.

1.1 Introduction aux arbres de décision

Tout d'abord, il est important de différencier deux types d'arbre de décision : les arbres de classification et les arbres de régression.

Les arbres de classification sont utilisés lorsque la variable à prédire est de type catégoriel. Les arbres de régression sont quant à eux utilisés dans le cas où la variable cible est de type continue. Dans le cas de notre application, la variable d'intérêt est de type continue, c'est pourquoi nous étudierons en détail uniquement le cas des arbres de régression.

Les premières publications relatant des arbres de décision datent des années 60 (Morgan and Sonquist (1963)) ("*Automated Interaction Detection*"(AID)), ce n'est pourtant qu'après la publication des arbres de classification et de régression ("*Classification And Regression Trees*") (CART) (Breiman et al. (1984)) que l'étude des arbres de régression s'est démocratisée (Loh (2014)). Parmi les différentes approches de construction d'arbres de régression, il est possible de citer le "*CHi-squared Automatic Interaction Detector*" (CHAID) (Kass (1980)), la méthode ID3 (Quinlan (1986)), le "*Fast and Accurate Classification Tree*" (FACT) (Loh and Vanichsetakul (1988)), le C4.5 (Quinlan (1993)) (Quinlan (2014)), le "*Smoothed and Unsmoothed Piecewise POlynomial Regression Trees*" (SUPPORT) (Chaudhuri et al. (1994)), le "*Quick, Unbiased and Efficient Statistical Tree*" (QUEST) (Loh and Shih (1997)), le "*Classification Rule with Unbiased Interaction Selection and Estimation*" (CRUISE) (Kim and Loh (2001)), les arbres d'inférence conditionnelles ("*Conditional Inference Trees*") (CTREE) (Hothorn et al. (2006b)) ou encore la méthode "*Generalized, Unbiased, Interaction Detection and Estimation*" (GUIDE) (Loh (2009)).

L'objectif de notre recherche n'étant pas de résumer toutes les méthodes de construction d'arbres de régression, nous étudierons en détail uniquement deux types de méthodes d'arbres de régression fréquemment utilisées dans la littérature. La méthode des arbres de classification et de régression (CART) et les arbres d'inférence conditionnelle (CTREE). Pour plus d'informations concernant les autres méthodes de construction d'arbres, veuillez-vous référer aux articles les concernant ou aux articles de revues suivants : Loh (2008), Loh (2011) et Loh (2014).

1.1.1 Concepts de base

L'objectif d'un arbre de décision est de prédire une variable Y à partir de covariables X. Le principe de construction d'un arbre de décision consiste à séparer successivement les données en sous-groupes en fonction de caractéristiques mesurées par les covariables X permettant de prédire Y. Au sein de l'arbre, le meilleur choix de séparation de chaque sous-groupe est appelé un **nœud**. Lorsqu'un nœud découle d'un autre nœud on l'appelle un **nœud fils**. On appelle le nœud en amont le **nœud parent**. Lorsque les nœuds de l'arbre séparent le jeu de données en deux sous-groupes on dit que les embranchements de l'arbre sont binaires.

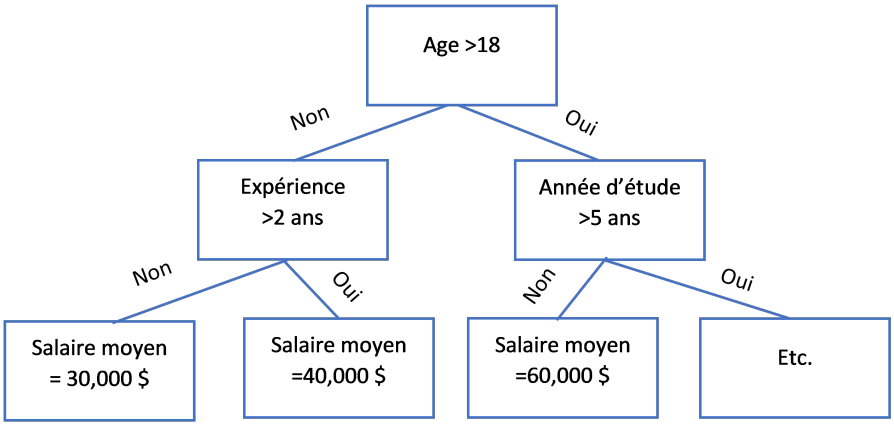

FIGURE 1.1 – Exemple d'arbre de décision

Illustrons ces propos à l'aide de la figure 1.1. Dans cette situation, l'objectif est de

prédire le salaire de chaque individu à partir de variables X telles que l'âge, l'expérience et le nombre d'années d'étude. Dans ce cas, le premier nœud de l'arbre consiste à séparer les données en fonction de l'âge des individus. Les observations initiales sont séparées en deux sous-groupes, le premier sous-groupe comprend tous les individus dont l'âge est inférieur ou égal à 18 ans et le second sous-groupe comprend tous les individus dont l'âge est supérieur à 18 ans. Dans ce cas on dit que 18 est le **seuil de séparation** du nœud utilisant la variable âge. Notre arbre de décision stipule aussi que parmi les individus étant âgés de moins de 18 ans, ceux dont l'expérience est inférieure à 2 ans auront en moyenne un salaire annuel de 30,000 dollars. Dans ce cas, le nœud divisant les données en fonction de l'expérience de l'individu est considéré comme le nœud fils du nœud séparant les données en fonction de l'âge. Les nœuds finaux de chaque arbre indiquant la valeur prédite de Y sont communément appelés les feuilles de l'arbre. Notons que les noeuds des arbres sont souvent représentés à l'aide d'un vecteur w de longueur n (nombre de nœuds dans l'arbre) prenant les valeurs 0 ou 1 en fonction de la présence ou non des individus étudiés dans ce nœud. Par exemple, si l'individu 1 testé est un adulte possédant plus de 2 ans d'expérience alors w_1 sera égal à 1 pour le nœud "*expérience >2 ans*".

En séparant le jeu de données en sous-groupes de plus en plus petits, il est possible de prédire le salaire de chaque individu en utilisant la moyenne des observations présentes dans les feuilles de l'arbre. Il est par exemple possible que la feuille prédisant un salaire moyen de 30,000 dollars comprenne trois individus gagnant respectivement 20,000 dollars, 30,000 dollars et 40,000 dollars.

Supposons que cet exemple simpliste soit applicable dans la vraie vie, il serait alors possible de prédire le salaire d'un individu pris au hasard uniquement en fonction de son âge, de son expérience de travail et de ses années d'études en suivant les embranchement de l'arbre. Par exemple, il serait possible de dire qu'un individu âgé de moins de 18 ans possédant plus de 2 années d'expérience gagne en moyenne un salaire de 40,000 dollars.

Il est important de comprendre qu'en pratique les arbres sont beaucoup plus massifs, ils possèdent en général plus de nœuds. A ce stade, il faut également savoir qu'un arbre de décision peut être trop spécifique aux individus utilisés lors de sa création et qu'il ne

permet donc pas de bien prédire le Y pour de nouveaux individus. On parle alors de sur-apprentissage (*overfitting*). Ce phénomène se produit lorsque l'arbre de décision devient trop grand et sépare les sous-jeux de données à partir de critères non pertinents. Reprenons l'exemple dont le but était de prédire le salaire des individus et supposons que l'arbre utilise aussi la couleur du T-shirt des individus pour séparer le sous-groupe des individus mineurs de moins de 2 ans d'expérience. Il donc possible qu'un individu portant un T-shirt rouge lors de l'étude et donc lors de la construction de l'arbre gagne un salaire de 25,500 dollars. Il est cependant peu probable que toutes les personnes non majeures ayant moins de 2 ans d'expérience portant un T-shirt rouge aient un salaire de 25,500 dollars. Ce critère est trop spécifique et si l'on veut utiliser cet arbre pour prédire le salaire d'un individu pris au hasard, le nœud séparant les individus en fonction de la couleur de leur T-shirt va diminuer la qualité de la prédiction. Pour répondre à ce problème, des critères d'arrêts sont mis en place. Le nombre minimal d'observations présentes dans chaque feuille de l'arbre est un exemple de critère d'arrêt. Il est important de préciser que le risque de sur-apprentissage dépend du type d'arbre utilisé. Nous reviendrons sur ces points plus en détail tout au long de ce chapitre.

1.1.2 Arbre CART (Classification And Regression Trees)

Dans l'algorithme des arbres CART, tous les embranchements de l'arbre sont binaires. Pour chaque nœud de l'arbre, le choix de la variable pour séparer les données en sous-groupes se fait en fonction du seuil de séparation (parmi toutes les variables) permettant d'améliorer le plus la qualité de prédiction de l'arbre. Dans ce cas ,plus la prédiction de la variable cible Y obtenue à l'aide de l'arbre est proche de sa vraie valeur, plus on considère que la prédiction de l'arbre est de bonne qualité.

Les seuils de séparation sont obtenus de manière itérative. Pour les variables de type continu ou ordinal, les seuils de séparations sont obtenus en utilisant toutes les valeurs composant les variables en conservant l'ordre des valeurs. Pour les variables catégorielles, les seuils de séparations sont obtenus en essayant toutes les combinaisons de valeurs pos-

sibles comprises dans la variable.

Illustrons cette idée à l'aide d'un exemple où l'on cherche le point de séparation de la variable X_1. Dans notre exemple, X_1 peut être égal à 1, 2, 3 ou 4. Si X_1 est de type ordinal ou continu, les point de séparation à évaluer sont les suivants :

— $X_1 \in \{1\}$ vs $X_1 \in \{2,3,4\}$

— $X_1 \in \{1,2\}$ vs $X_1 \in \{3,4\}$

— $X_1 \in \{1,2,3\}$ vs $X_1 \in \{4\}$

Si par contre X_1 est de type catégoriel alors les points de séparation potentiels sont les suivants :

— $X_1 \in \{1\}$ vs $X_1 \in \{2,3,4\}$

— $X_1 \in \{1,2\}$ vs $X_1 \in \{3,4\}$

— $X_1 \in \{1,3\}$ vs $X_1 \in \{2,4\}$

— $X_1 \in \{1,4\}$ vs $X_1 \in \{2,3\}$

— $X_1 \in \{1,2,3\}$ vs $X_1 \in \{4\}$

— $X_1 \in \{1,3,4\}$ vs $X_1 \in \{2\}$

— $X_1 \in \{1,2,4\}$ vs $X_1 \in \{3\}$

L'impact des seuils de séparation de chaque variable sur la qualité de la prédiction de l'arbre est évalué à l'aide d'une fonction notée $I(.)$. Cette fonction permet de mesurer l'erreur de prédiction de l'arbre de régression et nous permet d'évaluer la diminution de l'erreur relative à la découpe d'un nœud t en 2 nœuds fils t_L et t_R. Les notations t_L et t_R proviennent respectivement de l'anglais "*left daughter nodes*" et "*right daughter nodes*".

Dans le cas d'arbres de régression, la fonction $I(t)$ est simplement la somme des carrés des résidus telle que $I(t) = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ (Loh (2008)). De manière plus formelle, la construction de l'arbre de régression consiste simplement à maximiser la décroissance de l'erreur de prédiction de l'arbre :

$$\max I(t) - P_L I(t_L) - P_R I(t_R),$$

où $I(t)$ correspond à la mesure de l'erreur au nœud parent t. L'erreur est calculée seulement parmi les observations présentes dans le nœud t, $I(t_L)$ correspond à la mesure de l'erreur relative au nœud fils gauche, $I(t_R)$ correspond à la mesure de l'erreur relative au nœud fils droit et P_L et P_R sont respectivement les proportions du nombre de données présentes dans les nœuds fils gauche et droit.

Notons que cette façon d'évaluer les points de coupure est équivalente à maximiser la somme des carrés entre les groupes dans une analyse de variance. La librairie R `Rpart` créée par Therneau et al. (1997) permet d'implémenter ce type d'arbre.

En général, le type d'arbre que nous venons de décrire entraine des problèmes de surapprentissage, c'est pourquoi il est souvent pertinent d'avoir recours à l'élagage (Pruning) des arbres à l'aide de données de validation ou en ayant recours à la validation croisée. Ce thème n'est pas traité en détail dans le cadre de ce mémoire, veuillez vous référer au chapitre 9 du livre *"The elements of statistical learning"* de Friedman et al. (2001) pour une brève introduction. De plus, selon certains auteurs, ce type d'arbre pourrait engendrer des biais dans la sélection des variables pour séparer chaque nœud. Strobl et al. (2007) proposèrent une alternative afin de résoudre ce type de problème ; l'utilisation des arbres d'inférence conditionnelle.

1.1.3 Arbres d'inférence conditionnelle

L'idée principale des arbres d'inférence conditionnelle (CTREE) de Hothorn et al. (2006b) est de déterminer les variables explicatives à utiliser ainsi que leur seuil de séparation à partir de tests statistiques. Dans cet algorithme, comme dans l'arbre CART, tous les embranchements sont binaires et les variables utilisées peuvent être continues, nominales ou ordinales. Cet algorithme peut également prendre en compte des valeurs manquantes.

Tests de permutation

Il faut savoir que les valeurs-p des tests statistiques utilisés dans la construction des CTREE sont obtenues à partir de tests de permutation (Legendre and Legendre (2012), Good (2013), LaFleur and Greevy (2009)). Il est nécessaire de bien comprendre ce concept pour pouvoir comprendre la construction des CTREE, et c'est ce que nous détaillons ici.

L'idée générale de ce test est bien illustrée à travers l'exemple introduit dans le livre "*Numerical écologie*" de Legendre and Legendre (2012). L'objectif dans leur situation est de tester la corrélation ρ_{YX} entre deux variables Y et X. La première étape consiste à formuler l'hypothèse nulle :

$$H_0 : \rho_{YX} = 0,$$

et l'hypothèse alternative :

$$H_1 : p_{YX} \neq 0.$$

La seconde étape consiste à calculer la statistique de test $t^* = \sqrt{n-2}\frac{\rho_{YX}}{\sqrt{1-\rho_{YX}^2}}$. La statistique de test est calculée sur l'échantillon original. La distribution de t^* sous H_0 est obtenue à partir de nb_{perm} échantillons permutés où la statistique de test t^i_{perm} est recalculée sur le i^{ieme} échantillon permuté. L'échantillon permuté correspond à l'échantillon original à l'exception près que la variable Y est permutée aléatoirement afin de déstructurer la relation entre Y et X. La valeur-p se calcule comme suit :

$$\text{valeur-p} = \frac{\sum_{i=1}^{nb_{perm}} g(i)}{nb_{perm}},$$

où :

$$g(i) = \begin{cases} 1 & \text{si } (t^i_{perm} > t^*), \\ 0 & \text{si } (t^i_{perm} \leq t^*). \end{cases}$$

L'avantage d'effectuer des tests de permutation est qu'aucune hypothèse concernant la distribution des variables testées n'est nécessaire. Le temps de calcul peut cependant être extrêmement long lorsque beaucoup de permutations (nb_{perm}) sont effectuées.

Strasser and Weber (1999) ont développé une technique générale permettant de couvrir les cas où les variables explicatives proviennent de plusieurs classes et que la variable d'intérêt est centrée ou non. Cette nouvelle formulation permet de comparer la valeur-p de variables de différents types. Elle est en ce sens extrêmement utile pour les tests utilisés dans la construction des CTREE que nous allons voir maintenant.

Construction d'un arbre d'inférence conditionnelle

La construction de l'arbre d'inférence conditionnelle défini par Hothorn et al. (2006b) peut se résumer en 3 étapes : (1) l'étape d'arrêt, (2) l'étape du choix des variables et (3) l'étape du choix du meilleur point de séparation. Ces étapes sont résumées ci-dessous :

1. **Critère d'arrêt**

   Pour chaque nœud, chaque variable X_j est testée à partir de m hypothèses partielles $H_0^j : D(Y \mid X_j) = D(Y)$ et d'une hypothèse nulle globale $H_0 = \cap_{j=1}^{m} H_0^j$. Dans notre situation $D(Y)$ correspond à la distribution de Y et $D(Y \mid X_j)$ correspond à la distribution conditionnelle de Y sachant X_j. Lorsque l'hypothèse nulle globale n'est plus rejetée (seuil de significativité défini à l'avance), la récursion est arrêtée.

   L'association entre Y et X_j est mesurée à l'aide du test d'indépendance unifié construit à partir de la moyenne de la distribution conditionnelle de la statistique linéaire dans le contexte de tests de permutation développés par Strasser and Weber (1999). Ces tests de permutation sur chaque variable X_j permettent de comparer la valeur-p de n'importe quel type de variable. Pour l'hypothèse nulle globale, il est possible d'effectuer une correction de Bonferroni.

2. **Sélection de variables**

   Dans le cas où l'hypothèse nulle globale est rejetée, la variable X_j qui possède la plus grande association avec Y et donc la plus faible valeur-p est sélectionnée.

3. **Meilleur point de séparation**

   Après avoir sélectionné la variable explicative X_j^* utilisée à l'embranchement, le meilleur point de séparation est déterminé en sélectionnant le sous-ensemble d'ob-

servation A^* permettant de maximiser la valeur de la statistique de test en fonction de tous les sous-ensembles A possibles.

Les étapes 1, 2, 3 sont répétées pour chaque sous-arbre crée.

Détails sur la statistique de test utilisée dans la construction des CTREE

Nous présentons ici la situation où la variable Y est numérique et univariée. De plus, nous considérons le cas où les variables X_j sont numériques. La mesure T_j de l'association entre Y et X_j dans un ensemble d'entrainement L_n de n individus est alors définie par Strasser and Weber (1999) telle que :

$$T_j(L_n, \mathbf{w}) = \sum_{i=1}^{n} w_i Y_i X_{ij},$$

où L_n correspond à l'ensemble d'entrainement de n individus et w est un vecteur de taille n tel que $w_i = 1$ si l'observation i est utilisée dans le nœud évalué et 0 sinon. Notons que dans cette situation spécifique la mesure d'association est équivalente à la corrélation de Pearson. L'extension au cas où X est catégorielle est triviale. L'avantage principal de ce test est qu'il ne requiert pas de supposition sur la distribution des observations. Il est possible d'implanter les CTREE à partir des librairies R `party` de Hothorn et al. (2006a) et de `partykit` de Hothorn et al. (2015).

1.2 Introduction aux forêts aléatoires

Les arbres de régressions étudiés ont l'avantage d'être simples à interpréter, simples à implanter, et nécessitent (en général) peu de temps de calcul (Ziegler and König (2014)). En revanche, leur instabilité constitue un de leur défaut majeur puisqu'une petite variation au niveau des variables explicatives peut avoir un impact important sur la prédiction. Leo Breiman (2001) propose une solution à ce problème à travers l'algorithme des forêts aléatoires. Pour résumer cela de manière courte, une forêt aléatoire consiste à agréger la prédiction de plusieurs arbres. L'idée derrière cette technique est de regrouper la moyenne

(dans le cas de la régression) des prédictions afin de diminuer la variance associée à la prédiction.

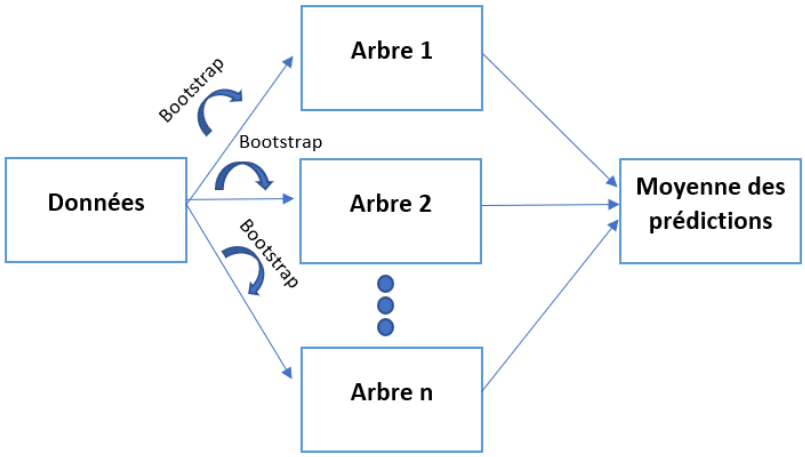

FIGURE 1.2 – Construction d'une forêt aléatoire basique. Le principe consiste à agréger la prédiction de plusieurs arbres de régression différents.

Il est possible d'illustrer cette idée à travers un exemple. Considérons un concours dont le but est d'estimer le poids d'un panier garni. Certains participants auront une prédiction du poids du panier trop élevée et d'autres auront une prédiction trop faible. L'idée consiste à dire qu'il y a moins de chance d'obtenir une prédiction très éloignée du vrai poids du panier si la moyenne des prédictions de chaque individu est utilisée. Il convient de stipuler que cette analogie a été démontrée mathématiquement par Breiman (2001). Ce principe, qui définit la structure d'une forêt aléatoire est illustré à la figure 1.2.

Afin de rassembler la prédiction de plusieurs arbres de régression différents, deux techniques complémentaires sont utilisées. Il est important de préciser que ces deux techniques sont utilisées conjointement.

La première technique consiste à effectuer un ré-échantillonnage des observations originales. Un processus de bootstrap est utilisé à cet effet (Efron and Tibshirani (1994)). L'idée derrière cette technique est d'entraîner chaque arbre de la forêt à partir d'un jeu de données légèrement différent, composé d'un échantillon avec remise du jeu de données

initial. Ainsi, certains individus pourraient être présent plusieurs fois dans l'échantillon bootstrap tandis que d'autres ne seront pas présent du tout (voir figure 1.3).

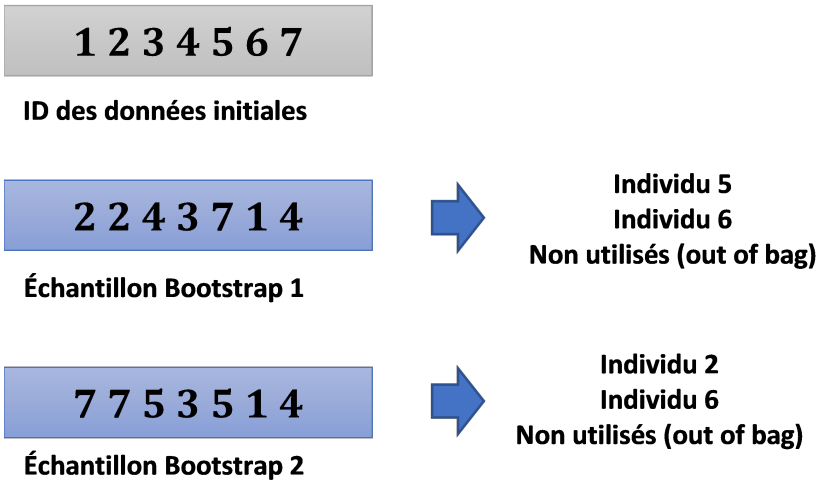

FIGURE 1.3 – Illustration du processus de Bootstrap. Dans cette situation, le jeu de données initial comprend 7 individus. L'échantillon bootstrap 1 est créé en tirant successivement et aléatoirement un individu provenant du jeu de données initial. Il est important de préciser que tous les individus sont susceptibles d'être pigés lors des tirages y compris ceux ayant déjà été sélectionnés. Ceci entraine la présence de doublons dans les échantillons bootstrap. Il est possible de constater par exemple que dans l'échantillon bootstrap 1 de l'illustration les individus 2 et 4 sont présents 2 fois. Un autre phénomène intéressant à mentionner est le fait que certains individus provenant de l'échantillon original ne sont pas présents dans les échantillons bootstrap. C'est le cas des individus 5 et 6 dans l'échantillon bootstrap 1 de l'exemple.

Les observations non utilisées à la suite du processus bootstrap sont appelées les observations **out-of-bag (OOB)**. Il est possible d'évaluer la performance de chaque arbre pour prédire la variable d'intérêt à partir de ces observations et l'erreur associée à cette prédiction s'appelle **l'erreur Out-of-Bag**. Le processus d'agrégation de plusieurs processus bootstrap s'appelle le bagging (*Bootstrap Aggregating*) Breiman (1996).

La seconde technique consiste à sélectionner aléatoirement un nombre de variables explicatives candidates parmi toutes les variables pour chaque nœud de chaque arbre composant la forêt. Le nombre de variables à sélectionner aléatoirement est défini avant l'entrainement du modèle, il fait partie des hyperparamètres du modèle. Supposons que notre

jeu de données soit composé de 10 variables explicatives, X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9, X_{10} et que le nombre de variables à sélectionner aléatoirement soit de 3. Lors de la construction de l'arbre, il sera possible de séparer les données en sous-groupes uniquement à partir de 3 variables sélectionnées aléatoirement parmi les 10 variables. Ce processus est effectué à chaque embranchement de l'arbre. Il sera par exemple possible d'avoir X_2, X_3, X_9 comme variables candidates pour le premier nœud et X_7, X_2 et X_6 comme variables candidates pour le second nœud et ainsi de suite...

Le pseudo-code des forêts aléatoires est le suivant :

Data: $Z_{entrainement}$=(X,Y) où $X = (X_1...X_p)$
Soit M le nombre d'arbres utilisés dans la forêt.
$i = 0$
while i<M **do**
  –Tirage d'un échantillon Bootstrap Z_i à partir de $Z_{entrainement}$.
  –Entrainement de l'arbre i sur cet échantillon avec sélection aléatoire des variables explicatives X candidates à chaque nœud.
  $i = i + 1$
end
–Calcul de la moyenne des prédictions des M arbres de la forêts.

A l'origine, Breiman (2001) proposa des arbres CART pour construire les forêts aléatoires, il est cependant possible de construire une forêt aléatoire à partir d'arbres CTREE. La procédure est similaire.

1.3 Mesure d'importance des variables

Il est possible d'utiliser des forêts aléatoires pour évaluer quelles sont les variables qui ont le plus d'importance pour prédire une variable cible Y. Dans le cadre de ce mémoire, nous avons considéré la qualité des méthodes de VIM en fonction de leur proximité avec une définition commune et intuitive d'importance de variable que nous avons calculé de manière théorique. Ainsi, d'après notre définition, une variable qui a beaucoup d'importance est une variable qui, si elle n'était pas utilisée pour prédire la variable cible, engendrerait une augmentation de l'erreur de prédiction et cela de manière proportionnelle à

son importance. Nous sommes conscients que cette définition de la qualité des méthodes de VIM est critiquable. Notre définition ne prend pas en compte le rôle des variables dans le processus de construction de l'arbre.

Nous présentons dans cette section différentes approches pour mesurer l'importance de variables à partir d'une forêt aléatoire. Nous verrons dans un premier temps comment mesurer l'importance des variables à partir de permutations, nous étudierons dans un deuxième temps comment mesurer l'importance des variables à partir de la fonction $I(.)$ présentée lors de la présentation de l'arbre CART, puis nous verrons enfin différentes applications où ces mesures d'importance peuvent être utilisées pour sélectionner un nombre restreint de variables.

1.3.1 Mesure d'importance des variables à partir de permutations

Concepts de base

Le calcul de la mesure d'importance de chaque variable à partir de permutations a été proposé par Breiman (2001). L'idée générale de cette méthode est de déstructurer de manière itérative la relation entre la variable testée X_k et les autres variables (variables explicatives et variables cibles) et d'observer l'impact de cette déstructuration sur la précision de la prédiction du modèle. Ainsi, si la précision du modèle se dégrade à la suite de la perturbation, c'est que la variable est importante. En revanche, si aucun changement n'est observé à la suite de la permutation de la variable, c'est que la variable n'est pas importante.

Pour comprendre plus en détail la façon dont la mesure d'importance est calculée, il est nécessaire de comprendre le principe d'échantillon Out-of-Bag (OOB) présenté à la section 1.2. Pour rappel, l'échantillon OOB correspond aux données non utilisées pour entraîner un arbre lors du processus de la construction de la forêt aléatoire (James et al. (2013)), (Genuer et al. (2010)). L'erreur OOB_t de chaque arbre t est simplement l'erreur de prédiction de chaque arbre t calculée à partir des observations OOB de cet arbre. Le processus de calcul de l'importance de chaque variable est le suivant. Pour chaque

arbre t, l'erreur OOB_t est calculée et stockée en mémoire. Par la suite, pour chaque variable testée X_k, les observations OOB_t de chaque arbre t sont aléatoirement permutées en X_k. L'erreur OOB_{tk} de l'arbre t calculée après la permutation de X_k est stockée en mémoire. La mesure d'importance de chaque variable X_k et pour chaque arbre t (VIM_{tk}) correspond à la différence entre l'erreur OOB_{tk} et l'erreur OOB_t. Ce processus est effectué pour chaque variable X_k et chaque arbre t. L'importance de chaque variable X_k (VIM_k) consiste simplement en la moyenne des différentes VIM_{tk} obtenues à partir de tous les arbres de la forêt. Cette méthode est fréquemment appelée "*Mean decrease in accuracy*" ou la méthode de permutation de Breiman. Le pseudo-code de la mesure d'importance effectué à l'aide de la méthode de permutation de Breiman est le suivante :

Étape 1 : La forêt aléatoire est entraînée

Étape2 : **while** *t< nombre d'arbres dans la forêt (M)* **do**

L'erreur OOB_t est calculée pour chaque arbre t

$k = 0$

while *k<nombre de variables* **do**

La variable k est permutée aléatoirement dans l'échantillon OOB_t.

L'erreur out of bag résultant de la permutation est calculée (OOB_{tk}).

$VIM_{tk} = OOB_{tk} - OOB_t$

$k = k + 1$

end

$t = t + 1$

end

Étape 3 :

while *k<nombre de variables* **do**

$VIM_k = \frac{1}{M} \sum_{t=1}^{M} VIM_{tk}$

$k = k + 1$

end

Dans la méthode originale, la mesure d'importance moyenne de chaque variable X_k est divisée par l'erreur standard des VIM_{tk} de chaque arbre t. Cependant, les résultats des études empiriques de Díaz-Uriarte and De Andres (2006), Strobl and Zeileis (2008),

Nicodemus et al. (2010), Nicodemus (2011) montrent qu'il est préférable d'utiliser la VIM non standardisée. D'après nos connaissances, aucune étude théorique traitant de ce sujet de manière spécifique n'a été publiée.

Plus formellement, il est possible d'écrire le calcul de la mesure d'importance $VI(X_k)$ pour une variable X_k avec la méthode de permutation de Breiman de la sorte :

$$VI(X_k) = \frac{1}{M}\sum_{t=1}^{M}\frac{1}{|OOB_t|}\sum_{i\in OOB_t} I(y_i - \widehat{y}_{ti}) - I(y_i - \widehat{y}_{ti\pi_k}),$$

où $VI(X_k)$ est l'importance mesurée de la variable k, M correspond au nombre d'arbres dans la forêt, OOB_t correspond à l' échantillon *out of bag* de l'arbre t, I() correspond à la fonction de mesure de performance, dans notre cas il s'agit de la somme des carrés des résidus, $\widehat{y}_{ti}$ correspond à la prédiction de l'observation i de l'arbre t avant avoir permuté la variable X_k et $\widehat{y}_{ti\pi_k}$ correspond à la prédiction de l'observation i de l'arbre t après avoir permuté les valeurs de la variable X_k dans les échantillons OOB de chaque arbre.

L'implémentation de la permutation classique est disponible dans presque tous les packages implémentant des forêts aléatoires. Certains packages comme `ranger` crée par Wright and Ziegler (2015) ne permettent cependant pas d'effectuer plus d'une permutation. Nous avons utilisé pour nos analyses le package R `Random Forest` crée par Liaw and Wiener (2002) et le package R `RandomForestSRC` crée par Ishwaran and Kogalur (2014).

Mesure d'importance à partir de la permutation conditionnelle

Il existe des variantes à la méthode de permutation classique proposée par Léo Breiman. Strobl et al. (2007) ont démontré que la méthode de VIM mesurée à partir de forêts aléatoires classiques est biaisée lorsque des variables explicatives catégorielles sont utilisées. Ils proposent notamment d'utiliser des arbres d'inférences conditionnels (CTREE) pour construire la forêt aléatoire. D'après leur étude, construire une forêt aléatoire à partir de CTREE serait plus fiable que les forêts aléatoires classiques pour mesurer l'importance des variables. L'étude empirique de Auret and Aldrich (2011) vient supporter ces

résultats. Il faut préciser que ce résultat s'applique uniquement dans le cas où le processus bootstrap n'a pas été utilisé dans le processus de construction des forêts. Dans ce cas, chaque jeu de données utilisé pour entraîner chaque arbre est obtenu à partir de la sélection aléatoire de 2/3 des observations.

Strobl et al. (2008) identifient un nouveau problème dans le calcul de la VIM à l'aide de la méthode de permutation classique lorsque les variables explicatives sont corrélées. Il démontrent que la VIM originale calculée avec la méthode de permutation de Breiman surestime l'importance des variables non corrélées avec la variable dépendante, mais qui sont fortement corrélées avec des variables possédant un lien avec la variable à expliquer. Cela vient du fait que la VIM calculée à partir de la méthode de permutation de Breiman déstructure la relation entre Y et la variable testée X_k, mais aussi, par conséquent, la relation entre la variable testée et les autres variables explicatives corrélées à X_k. En présence de corrélation entre les variables, la diminution de la précision de l'arbre suite à la permutation de X_k peut être due à une dépendance de X_k avec Y ou à une dépendance de X_k avec les autres variables explicatives.

Pour corriger ce problème, Strobl et al. (2008) proposent une méthode de permutation conditionnelle aux variables corrélées. Le principe est le même que pour la méthode de permutation classique, la différence se situe au niveau de la permutation des variables. Soit X_k une variable à permuter, et $Z_{[X_k]} = (X_1, \ldots X_{(k-1)}, X_{(k+1)}, \ldots, X_p)$ correspondant aux autres variables explicatives. L'idée consiste à sélectionner les variables comprises dans $Z_{[X_k]}$ étant corrélées avec X_k et de déterminer une grille de permutation avec laquelle X_k sera permutée. X_k ne sera donc pas permutée en fonction de toutes ses valeurs, mais en fonction de sous groupes d'observation. Plus formellement il est possible d'écrire avec les mêmes notations utilisées précédemment :

$$VI(X_k) = \frac{1}{M}\sum_{t=1}^{M}\frac{1}{|OOB_t|}\sum_{i\in OOB_t} I(y_i - \widehat{y}_{ti}) - I(y_i - \widehat{y}_{ti\pi_{k|Z}}),$$

où $\widehat{y}_{ti\pi_k|Z_{[X_k]}}$ correspond à la prédiction de l'observation i de l'arbre t après avoir effectué π permutations de la variable X_k dans les échantillons OOB de chaque arbre sachant la grille de permutation $Z_{[X_k]}$.

Les observations faites par Strobl et al. (2007) ne font pas l'unanimité dans la communauté scientifique. Suite à cette publication Nicodemus and Malley (2009) observent que la VIM calculée à partir de la permutation classique est robuste aux variables corrélées et qu'uniquement la VIM calculée à partir de la décroissance GINI (voir ci-dessous) est biaisée par cette corrélation. Meng et al. (2009) déterminent que la corrélation entre les variables explicatives affecte les résultats. Enfin, Nicodemus et al. (2010) examinent la question en profondeur et concluent que l'utilisation de l'importance conditionnelle est préférable lorsqu'une analyse est effectuée avec peu de variables explicatives corrélées. Plus récemment, Gregorutti et al. (2017) proposèrent une alternative à la VIM conditionnelle en proposant une élimination récursive des variables basée sur la méthode de permutation de Breiman. L'implantation de la VIM conditionnelle est disponible à l'aide du package `Party` crée par Hothorn et al. (2006a).

Mesure d'importance proposée par Ishwaran

Ishwaran et al. (2007) proposent une approche basée sur la même idée que la permutation originale, mais plus facile à analyser théoriquement. Les mesures d'importance sont maintenant calculées comme la différence entre l'erreur de prédiction de chaque arbre avec et sans bruit. Ils proposent pour cela d'introduire les bruits directement dans la structure des arbres.

La libraire R `RandomForestSRC` crée par Ishwaran and Kogalur (2014) permet de calculer l'importance des variables de deux nouvelles façons :

— Dans le premier cas, une observation Out-of-bag est assignée de manière aléatoire à chaque nœud fils lorsque que la variable testée est utilisée en tant que nœud parent (VIM_Ishwaran_Aléatoire).

— Dans le second cas, la variable testée est assignée au nœud opposé à chaque fois que la variable testée est utilisée en tant que nœud parent (VIM_Ishwaran_Noeud_opposé).

Ces deux mesures d'importance sont illustrées à l'aide de la figure 1.4. Ishwaran et al. (2008) proposent également une manière accélérée pour calculer l'importance de chaque

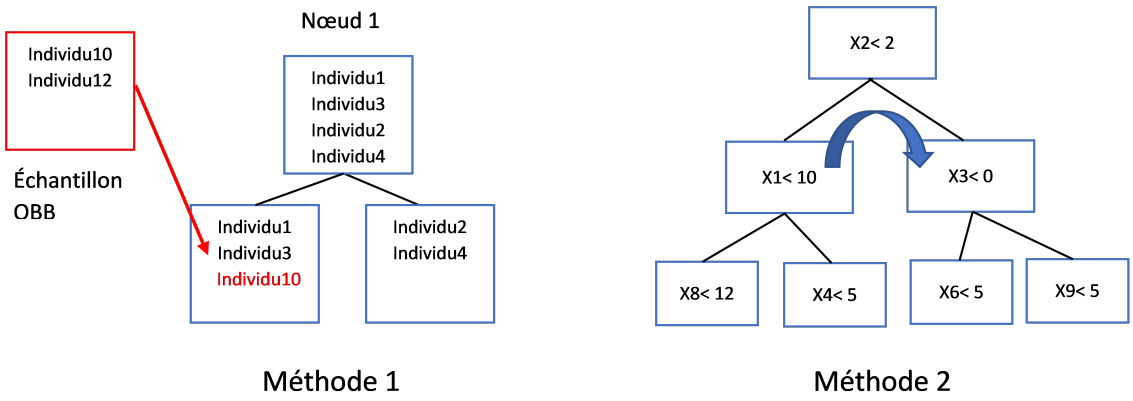

FIGURE 1.4 – Illustration des mesures d'importance disponibles à l'aide du package `randomforestSRC`. Dans cette illustration, nous supposons que le but est d'évaluer l'importance de la variable X_1. L'exemple de la méthode 1 correspond au premier cas et le nœud 1 correspond aux individus séparés en sous groupe par la variable X_1. Dans cette situation, un individu de l'échantillon OOB est simplement ajouté à l'un des deux nœuds fils à chaque fois que la variable X_1 est utilisée pour fragmenter le jeu de données. Dans l'exemple de la méthode 2, la variable testée est assignée au nœud opposé à chaque fois que la variable testée est utilisée pour séparé les données. Dans ce cas, X_1 est assigné a la place du nœud "$X_3 < 0$". Le seuil de séparation sera celui qui minimise l'erreur de l'arbre.

variable en calculant la différence non plus directement depuis les arbres de régression, mais directement depuis les forêts. Ces méthodes sont disponibles à partir du package `randomForestSRC` crée par Ishwaran and Kogalur (2014).

1.3.2 Diminution moyenne de « l'impureté »

La diminution moyenne de « l'impureté » (VIM_Impureté) est une approche proposée par Breiman et al. (1984) pour calculer l'importance de chaque variable à partir d'arbres CART. Cette méthode a beaucoup été utilisée pour les arbres de classification. Les arbres de classification utilisent l'index GINI (ou entropie) comme fonction pour mesurer l'erreur de prédiction dans le processus de construction des arbres CART. Lorsque ces indices sont utilisés, on dit qu'ils mesurent la *pureté* de l'arbre, c'est de là que provient le nom

diminution moyenne de l'impureté. Il est d'ailleurs fréquent d'observer dans la littérature cette mesure d'importance sous le nom de *Gini importance* ou de *diminution moyenne Gini*. Dans le cas d'arbres de régression, la mesure de *pureté* de l'arbre est simplement la somme des carrés des résidus (voir sous section 1.1.2). Il est alors préférable d'appeler cette méthode « Mean deacrease in impurity » (Louppe et al. (2013)).

Dans le cas de forêts aléatoires ayant pour but d'effectuer une prédiction d'une variable continue, cette mesure d'importance consiste simplement à évaluer l'apport moyen de chaque variable dans la diminution de la somme des carrés des résidus lors de leur apparition dans l'arbre. Cette diminution de la somme des carrés des résidus est pondérée en fonction du "moment" de l'apparition de la variable dans l'arbre. Plus la variable est utilisée tôt dans la construction de l'arbre, plus le gain de prédiction associé à sa présence sera valorisé.

Plus formellement, il est possible d'écrire (en utilisant les mêmes notations que précédemment) :

$$VI(X_k) = \frac{1}{M} \sum_{t=1}^{M} \sum_{l \in L:(v(s_l)=X_k)} p(l) \nabla I(s_l, l),$$

où L correspond à l'ensemble des nœuds dans l'arbre t, $p(l)$ correspond à la proportion de l'échantillon atteignant le nœud l, s_l correspond au seuil de séparation ("*split*") du nœud l, $v(s_l)$ correspond à la variable utilisée dans la division du nœud l, et $\nabla \mathrm{I}(\mathrm{s}_l, l)$ correspond à la différence entre la somme des carrés des résidus avant et après la séparation du noeud l à l'aide du seuil de séparation s_l.

L'avantage de cette méthode est qu'elle est beaucoup moins coûteuse en temps de calcul que les méthodes présentées jusqu'à présent. Le désavantage de cette méthode est qu'elle a tendance à privilégier les prédicteurs de type continu ou avec beaucoup de catégories. Ceci vient du fait que ces types de variables sont plus souvent sélectionnés dans le processus de construction des arbres CART, (Strobl et al. (2007), Gregorutti et al. (2017), Gregorutti et al. (2017), Nicodemus (2011), Nicodemus and Malley (2009)).

1.3.3 Méthode de sélection des variables basée sur l'importance des variables

Dans cette sous-section, nous présentons différentes méthodes de sélection de variables basées sur l'importance des variables. Cette sous-section ne fait pas l'objet des analyses présentées dans ce mémoire mais elle peut cependant inspirer d'éventuels travaux futurs.

Méthode de Altmann et al.

La première méthode proposée par Altmann et al. (2010) propose d'associer une valeur-p aux mesures d'importance. La distribution des mesures d'importance dans le cas où il n'y a aucune relation entre les variables testées et la variable prédite (distribution sous H_0) est obtenue à partir de plusieurs itérations du processus suivant :

1. Les valeurs de la variables à prédire Y sont permutées tout en gardant les autres variables explicatives constantes.
2. Une forêt aléatoire est entrainée sur les données d'entrainement.
3. L'importance de chaque variable est calculée à partir de la forêt.

Une fois que la distribution de l'importance des variables sous H_0 est estimée, les auteurs proposent deux façons de calculer la valeur-p associée à la pertinence de chaque variable.

La première façon consiste à calculer la valeur-p de manière non paramétrique à partir de tests de permutation (voir section 1.1.3 test de permutation).

La seconde façon de calculer la valeur-p consiste à supposer une distribution (Gaussienne, Log Normale, Gamma) pour les mesures d'importance des variables. Les paramètres de ces distributions sont trouvés à l'aide du maximum de vraisemblance à partir des scores d'importance calculés à partir de la distribution des mesures d'importance sous H_0. Il suffit alors de calculer la valeur-p comme la probabilité d'observer un score d'importance qui est plus grand que l'original sachant H_0. La librairie R `Rangercrée` par Wright and Ziegler (2015) permet d'implémenter la méthode de Altmann et al. (2010).

Méthode de Janitza et al.

Janitza et al. (2016) proposent une méthode permettant de sélectionner des variables dans le contexte d'un grand nombre de dimensions. Cette méthode est basée sur le concept de "*cross validated variable importance*", une alternative à la méthode de permutation classique de Léo Breiman qui permet d'économiser du temps de calcul en utilisant les observations out of bag. L'idée de ce concept est très similaire à la notion de validation croisée. Les données sont séparées en E échantillons et E forêts aléatoires sont construites. Chaque forêt aléatoire est entrainée sur E-1 échantillon. Pour chaque forêt, l'échantillon non utilisé lors de l'entrainement est noté S_L. Par la suite, seules les données S_L sont utilisées pour calculer l'importance des variables. Le calcul de l'importance des variables est alors le suivant (en utilisant les mêmes notations que précédemment) :

$$VI(X_k)^{CV(L)} = \frac{1}{M}\sum_{t=1}^{M}\frac{1}{|S_L|}\sum_{i\in S_L} I(y_i - \widehat{y}_{ti}) - I(y_i - \widehat{y}_{ti\pi_k}),$$

où $VI(X_k)^{CV(L)}$ est la *cross validated variable importance* estimée de la variable k à partir des données S_L.

L'importance de chaque variable est ensuite calculée à l'aide d'une moyenne de la *"cross validated importance"* des E échantillons :

$$VI(X_k) = \frac{1}{E}\sum_{l=1}^{E} VI(X_k)^{CV(l)},$$

Dans le cas où $E = 2$, on appelle cette mesure d'importance *Hold-out variable importance*.

L'algorithme proposé se décompose de la sorte :

1. L'importance de chaque variable est évaluée (utilisation de la méthode cross validated variable importance (Hold-out))
2. La distribution de l'hypothèse nulle de l'importance de chaque variable est approximée à partir des scores d'importance non positifs observés. Pour cela, 3 échantillons sont définis :

— $M_1 = \{VI(X_k)^{HO} | (VI(X_k)^{HO} < 0); k = 1...p\}$(c'est à dire tous les scores d'importance négatifs).

— $M_2 = \{VI(X_k)^{HO} | (VI(X_k)^{HO} = 0); k = 1...p\}$ (c'est à dire tous les scores d'importance égaux à 0).

— $M_3 = \{-VI(X_k)^{HO} | (VI(X_k)^{HO} < 0); k = 1...p\}$ (c'est à dire tous les scores d'importance négatif multipliés par -1).

La distribution cumulative empirique $\widehat{F_0}$ de $M = M_1 \cup M_2 \cup M_3$ est estimée.

3. La valeur-p correspondante à l'importance de chaque variable X_k est calculée comme étant égale à $1 - \widehat{F_0}(VI(X_k)^{HO})$

Le défaut majeur de cette méthode est qu'elle nécessite un nombre important de variables pour pouvoir estimer la distribution de l'hypothèse nulle. La librairie R `Ranger` crée par Wright and Ziegler (2015) permet d'implémenter la méthode de Janitza et al. (2016).

Méthode de « *Variable Selection Using Random Forests* » (VSURF)

Pour finir, nous présentons la méthode VSURF proposée par Genuer et al. (2015). L'algorithme est construit à partir de deux étapes :

1. Une élimination préliminaire des variables à partir d'un classement est effectuée. Cette étape consiste à trier les variables par ordre décroissant de VIM et d'éliminer les variables avec une faible importance. Le seuil de rejet est obtenu à partir de l'écart type des VIM.

2. La sélection des variables se divise en deux sous-étapes, la première concerne l'interprétation et la seconde concerne la prédiction.

   a. Interprétation

   Cette étape consiste à entraîner plusieurs forêts aléatoires en augmentant progressivement le nombre de variables sélectionnées à l'étape 1. L'ordre de l'ajout des variables est défini par le rang des VIM utilisées à l'étape 1. La

première variable ajoutée est donc la plus importante. Les variables sélectionnées sont celles contenues dans le modèle qui mènent à la plus petite erreur OOB.

b. Prédiction

Les variables utilisées dans cette étape sont celles sélectionnées à l'étape d'interprétation. Une séquence de forêts aléatoires est construite, toutes les variables sont testées à l'aide d'une procédure pas à pas ("*stepwise*"). Les variables du dernier modèle sont ensuite sélectionnées.

La librairie R `VSURF` crée par Genuer et al. (2015) permet d'implémenter cette méthode.

1.4 Résumé des librairies R présentées dans le chapitre

Le tableau présenté à cette section est un résumé des différentes librairies R présentées dans le chapitre 1. Ces librairies possèdent d'autre fonctions que celles présentées dans le tableau, pour plus d'informations, veuillez vous référer aux articles les concernant.

Librairie R	**Auteur(s)**	**Fonction**
`Rpart`	Therneau et al. (1997)	Permet d'implémenter des d'arbres CART.
`Random Forest`	Liaw and Wiener (2002)	Permet d'implémenter les différentes méthodes de VIM obtenues à partir des forêts aléatoires composées d'arbres CART.
`RandomForestSRC`	Ishwaran and Kogalur (2014)	Permet d'implémenter les différentes méthodes de VIM proposées par Ishwaran.
`party`	Hothorn et al. (2006a)	Permet d'implémenter les différentes méthodes de VIM obtenues à partir des forêts aléatoires composées d'arbres CTREE.
`partykit`	Hothorn et al. (2015)	Permet d'implémenter les différentes méthodes de VIM obtenues à partir des forêts aléatoires composées d'arbres CTREE.
`Ranger`	Wright and Ziegler (2015)	Permet d'implémenter les méthodes de sélection de variables de Altmann et al. (2010) et de Altmann et al. (2010).
`VSURF`	Genuer et al. (2015)	Permet d'implémenter la méthode de sélection de variables *VSURF* de Genuer et al. (2015).

TABLE 1.1 – Résumé des différentes librairie R présentées dans le chapitre 1.

Chapitre 2

Étude de simulations

Nous allons étudier dans ce chapitre le comportement des différentes méthodes de mesure d'importance des variables explicatives pour prédire la variable d'intérêt dans différents types d'environnements contrôlés. Nous avons pour cela évalué les 8 méthodes en fonction de leur proximité avec une définition commune et intuitive d'importance de variable que nous avons calculé de manière théorique. L'objectif de ces simulations est de pouvoir appliquer les deux meilleures méthodes de VIM à une application à la génétique. Les simulations ont été développées à partir des problématiques récurrentes associées à l'analyse de données génétiques. Faute de temps et de moyens, nous n'avons cependant pas simulé, la structure extrêmement complexe des données utilisées dans le cadre de l'application en génétique.

2.1 Introduction

Il nous a semblé indispensable d'évaluer le comportement des méthodes de VIM à partir de nos propres simulations pour plusieurs raisons. Tout d'abord, certains résultats d'expériences effectuées par divers auteurs se contredisent et aucun consensus n'a vraiment été trouvé. D'autre part certaines méthodes de VIM ont été évaluées dans des contextes où les données ont été simulées de manière trop parfaite, en totale inadéquation avec la réalité. Enfin, nous utilisons dans notre étude des librairies déjà conçues provenant

du langage de programmation R. Cependant, les articles expérimentant les différences de performance des méthodes de VIM n'ont pas nécessairement utilisé ce langage de programmation et ceci peut biaiser les résultats.

Nous avons étudié 8 méthodes de VIM différentes. Parmi les méthodes analysées, cinq sont calculées à partir de forêts aléatoires construites à partir d'arbres CART et trois proviennent de forêts aléatoires construites à partir d'arbres d'inférence conditionnelle (CTREE). Nous avons choisi ces méthodes en raison de leur popularité dans de nombreuses publications scientifiques et pour leur facilité de mise en application à partir de packages R. Les méthodes de VIM calculées à partir de forêts aléatoires composées d'arbres CART sont les suivantes :

— **VIM_Impureté** : méthode de VIM calculée à partir de l'impureté.

— **VIM_CART_Standardisée** : méthode de VIM calculée à partir de la permutation de Breiman standardisée.

— **VIM_CART_Non_Standardisée** : méthode de VIM calculée à partir de la permutation de Breiman non standardisée.

— **VIM_Ishwaran_Aléatoire** : méthode de VIM calculée à partir de la méthode d'ajout du bruit aléatoire de Ishwaran.

— **VIM_Ishwaran_Noeud_Opposé** : méthode de VIM calculée à partir de la méthode de sélection du nœud opposé d'Ishwaran.

Les méthodes de VIM construites à partir de forêts aléatoires composées d'arbres d'inférences conditionnels sont les suivantes :

— **VIM_CTREE** : méthode de VIM calculée à partir de la permutation de Breiman non standardisée.

— **VIM_CTREE_Strobl** : méthode de VIM calculée à partir de la permutation de Breiman non standardisée avec les mêmes hyper paramètres que les forêts aléatoires de l'article de Strobl et al. (2007). Seul le nombre de variables aléatoirement sélectionné à chaque nœud a été modifié en fonction du jeu de données utilisé.

— **VIM_CTREE_corrélation** : méthode de VIM calculée à partir de la permutation de Breiman non standardisée, corrigé pour la corrélation entre les variables explicatives.

Toutes ces méthodes ont été décrites dans le chapitre 1.

Notre étude de simulation sur la capacité des différentes méthodes de VIM à mesurer correctement l'importance des variables peut se résumer en cinq questions. Les cinq questions ont été développées à partir des problématiques récurrentes associées à l'analyse de données génétiques, qui a motivé notre recherche.

1. Est-ce que les méthodes de VIM sont robustes à une augmentation du nombre de variables non explicatives dans le modèle ?
2. Est-ce que les méthodes de VIM sont robustes à une augmentation du nombre de variables explicatives dans le modèle ?
3. Est-ce que les méthodes de VIM sont fiables quand les relations entre les variables explicatives et la variable d'intérêt sont linéaires ?
4. Est-ce que les méthodes de VIM sont fiables quand les relations entre les variables explicatives et la variable d'intérêt sont non linéaires ?
5. Est-ce que les méthodes de VIM sont robustes à la corrélation entre les variables explicatives ?

2.2 Méthodologie

2.2.1 Modèle de base et scénarios de simulations de base

Pour répondre aux questions proposées à la section précédente, un modèle de base a été défini à partir duquel les données ont été simulées. Ce modèle a les caractéristiques suivantes :

— Nombre d'individus pour l'entrainement : 1000.

— Nombre d'individus pour la validation : 200.

— Nombre de variables explicatives : 5.

— Nombre de variables non explicatives (bruit) dans le modèle : 15.

La variable dépendante Y est générée pour chaque individu selon le modèle suivant :

$$Y = -1.8X_1 + 1.6X_2 + 1.4X_3 - 1.7\sin(X_4) + 0.2\exp(X_5) + \varepsilon,$$

où $\varepsilon \sim N(0,2)$, $X_i \sim N(0,1)$, $\forall i = 1,...,20$.

Le principe de l'utilisation de ce modèle de base consiste à faire varier un seul paramètre du modèle à la fois pour chaque scénario de simulation et à garder les autres paramètres du modèle constants.

Illustrons ces propos à l'aide d'un exemple. Supposons que la simulation consiste à évaluer la robustesse des méthodes de VIM lorsque le nombre de variables non explicatives augmente à 45. Dans ce cas, un seul paramètre du modèle de base est modifié ; au lieu d'être dans une situation où le nombre de variables total est de 20, le modèle évalué contiendra 50 variables. Cette modification s'effectue en gardant les autres paramètres du modèle constants (tels que le nombre de variables explicatives ou les coefficients attribués aux variables). Les 21 scénarios de simulations étudiées ici sont présentés dans les 4 premières colonnes du tableau 2.2.

2.2.2 Comparaison des méthodes de VIM

Chaque méthode de VIM prédit des valeurs brutes d'ordre différent. Par exemple la méthode de VIM calculée à partir de l'impureté et la méthode de VIM calculée à partir de l'erreur quadratique moyenne auront toujours des valeurs brutes différentes même si l'ordre d'importance des variables et le modèle simulé sont identiques. Pour pouvoir comparer les méthodes de VIM entre elles, nous avons privilégié l'étude des importances relatives de chaque variable. Bien que cette approche rend les méthodes comparables entre elles, nous sommes conscients que l'utilisation de l'importance relative comparativement à l'importance brute comporte quelques inconvénients. Les deux problèmes majeurs identifiés sont les suivants :

Problème 1 : Une erreur d'évaluation d'une importance relative pour une variable peut décaler l'importance relative des autres variables.

Problème 2 : Il est possible que certaines importances brutes soient négatives et proches de 0. Cette situation peut potentiellement se produire lorsque la méthode de VIM est calculée à partir de la permutation de Breiman. Lorsque la variable n'a aucune importance pour prédire la variable d'intérêt, la prédiction à la suite de la permutation peut être légèrement meilleure que la prédiction originale. Notons que pour la permutation de Breiman standardisée, l'impact de ce phénomène est accentué par la division des mesures d'importances par leur l'écart-type.

La solution proposée pour résoudre ce problème est d'ajouter la valeur absolue de la valeur négative la plus élevée à toutes les mesures d'importance brutes avant de calculer l'importance relative de chaque variable. Ces propos sont illustrés à l'aide des exemples ci-dessous :

Example 1. Problème lié aux mesures d'importances brutes négatives

Soit VIM_k l'importance de la variable k.

Posons $VIM_1 = 50.5, VIM_2 = -0.5$.

Calcul de l'importance relative

$VIM_1 = 50.5/(50.5(-0.5)) = 1.01,$

$VIM_2 = -0.5/50.5(-0.5) = -0.009.$

Ce phénomène est problématique pour l'interprétation car l'importance relative d'une variable ne peut pas être négative ou plus importante que 100%.

Example 2. Problème lié aux corrections des mesures d'importances brutes négatives

Soient les importances non standardisées :

$VIM_1^{ND} = 50, VIM_2^{ND} = 10, VIM_3^{ND} = -0.5$ et les écarts type respectifs :

$SD_1 = 0.6, SD_2 = 0.4, SD_1 = 0.1.$

Calcul de l'importance standardisée :

$VIM_1^{SD} = VIM_1^{ND}/SD_1 = 50/0.6 = 83,33,$

$VIM_2^{SD} = VIM_2^{ND}/SD_2 = 10/0.4 = 25,$

$VIM_3^{SD} = VIM_3^{ND}/SD_3 = -0.5/0.1 = -5.$

Calcul de l'importance relative des mesures d'importances non standardisées :

$VIM_1^{ND} = (50+0.5)/((50+0.5)+(10+0.5)+(-0.5+0.5)) = 82.78\%,$

$VIM_2^{ND} = (10+0.5)/((50+0.5)+(10+0.5)+(-0.5+0.5)) = 17.21\%,$

$VIM_3^{ND} = (-0.5+0.5)/((50+0.5)+(10+0.5)+(-0.5+0.5)) = 0\%.$

Calcul de l'importance relative des mesures d'importances standardisées à la suite du décalage :

$VIM_1^{SD} = (83.33+5)/((83.33+5)+(25+5)+(-5+5)) = 74.64\%,$

$VIM_2^{SD} = (25+5)/((83.33+5)+(25+5)+(-5+5)) = 25.36\%,$

$VIM_3^{SD} = (-5+5)/((83.33+5)+(25+5)+(-5+5)) = 0\%.$

On observe effectivement que dans cet exemple, effectuer un décalage a plus d'impact sur la méthode de VIM calculée à partir de la permutation standardisée dans le cas où l'écart type est plus faible pour les variables négatives. Cette situation se présente souvent en pratique. Notons que la méthode de VIM calculée à partir de l'impureté est par définition toujours positive.

A ce stade, il manque une mesure d'importance de référence pour comparer les méthodes de VIM entre elles. A ce titre, l'importance théorique relative de chaque variable est aussi calculée. Elle permet entre autre d'obtenir une évaluation numérique de la performance de chaque méthode de VIM lors de chaque simulation. Le processus du calcul de l'importance théorique de chaque variable est le suivant :

Soit K l'ensemble des variables et β le vecteur des coefficients des variables explicatives dans le modèle permettant de générer les données. Pour simplifier l'explication, nous supposons ici qu'il y a 4 variables explicatives dans le modèle et 6 variables non explicatives.

1. La moyenne théorique de la variable dépendante ($Y_{theorique}$) est calculée telle que

$$Y_{theorique} = \beta_1 \times X_1 + \beta_2 \times X_2 + \beta_3 \times X_3 + \beta_4 \times X_4.$$

2. Pour chaque $k = 1,2,3,4$ à tour de rôle, on pose $\beta_k = 0$ (en gardant les coefficients des autres variables explicatives constants) et on recalcule la moyenne théorique de la variable dépendante notée Y_{X_k}, en supposant donc que X_k n'a pas d'effet sur Y

3. L'importance théorique de chaque variable k est calculée comme la différence quadratique moyenne entre $Y_{theorique}$ et Y_{X_k}.

$$VIMBRUT_k^{theorique} = \frac{1}{6+4} + (Y_{theorique} - Y_{X_k})^2.$$

4. L'importance relative théorique de chaque variable k est calculée.

$$VIM_k^{theorique} = \frac{VIMBRUT_k^{theorique}}{\sum_{k=1}^{10} VIMBRUT_k^{theorique}}.$$

Cette métrique nous permettra d'évaluer la qualité des méthodes de VIM d'après notre définition de qualité de mesure d'importance présentée à la section 1.3.

2.2.3 Procédure des simulation

Pour chaque simulation, le calcul des VIM à l'aide des différentes méthodes s'est effectué en deux étapes. La première étape a consisté à ajuster le modèle de forêt aléatoire aux données simulées et donc de déterminer quels hyper paramètres définir pour calculer les forêts aléatoires. La seconde étape a consisté à calculer les VIM à partir des forêt aléatoires entrainées. Pour ces deux étapes, le logiciel R a été utilisé. Le package `party` a été utilisé pour les forêts aléatoires construites à partir d'arbres d'inférences conditionnels. Le package `random Forest` et le package `randomForestSRC` ont été utilisés pour le calcul des Forêts aléatoires construites à partir d'arbres CART.

Déterminer les hyper-paramètres des forêts aléatoires

Pour commencer, définissons ce qu'est un hyper paramètre. Les hyper-paramètres sont des configurations faites à l'extérieur du modèle avant de commencer le processus d'entrainement des données. Par exemple dans le cas des forêts aléatoires, la taille minimale

des nœuds terminaux de chaque arbre est un hyper paramètre, car il est défini avant le processus d'entrainement des forêts aléatoires. Le choix des hyper paramètres d'un modèle peut avoir deux impacts principaux :

1. Il peut influencer la « capacité » du modèle et donc son aptitude à se « sur entrainer » ou à se « sous entrainer ».
2. Le choix des hyper paramètres peut également agir sur le temps d'apprentissage du modèle, c'est par exemple le cas pour le nombre d'arbres à utiliser dans la forêt.

Il existe plusieurs façons de trouver la meilleure combinaison d'hyper paramètres d'un modèle. Pour notre étude, nous avons eu recours à l'optimisation bayésiennes des hyper paramètres (Bergstra et al. (2011), Snoek et al. (2012), Brochu et al. (2010)). Plus précisément, nous avons utilisé un processus Gaussien pour modéliser la fonction d'erreur (Rasmussen (2004)). Nous avons utilisé l'"*Upper confidence bound criteria*" (kappa = 2.576) comme fonction d'acquisition. Nous avons initié le processus à l'aide de 50 essais aléatoires puis nous avons effectué 50 itérations. L'implantation en R a été possible à l'aide du package R `rBayesianOptimization`. Nous avons entrainé les modèles sur les 1000 individus simulés et choisi les hyper paramètres à partir d'un fichier de validation de 200 individus simulés. Pour des raisons de temps de calcul, nous avons optimisé les hyper paramètres les plus importants pour la qualité de prédiction des forêts aléatoires.

Dans le cas des forêts aléatoires CART entrainées à partir du package `randomForest`, nous avons fait varier le nombre de variables candidates sélectionnées aléatoirement lors de la séparation de chaque nœud (`mtry`), la taille minimale des nœuds terminaux de chaque arbre et nous avons fait varier le processus d'échantillonnage (utilisation du processus de boostrap (`Replace=TRUE`) ou utilisation des $\frac{2}{3}$ des observations sans remplacement (`Replace=FALSE`)). Pour les forêts aléatoires composées d'arbres d'inférences conditionnelles, nous avons fixé la statistique de test par défaut qui est le maximum de la valeur absolue de la statistique de test standardisé et ajusté la valeur p de chaque variable à l'aide d'un ajustement de Bonferroni. Nous avons fait varier le nombre de variables candidates sélectionnées aléatoirement lors de la séparation de chaque nœud

(`mtry`), le seuil $(1-p)$ qui doit être dépassé dans le but d'implémenter une séparation (`mincriterion`) et nous avons fait varier le processus d'échantillonnage (utilisation du processus de boostrap (`Replace=TRUE`) ou utilisation des $\frac{2}{3}$ des observations sans remplacement (`Replace=FALSE`)).

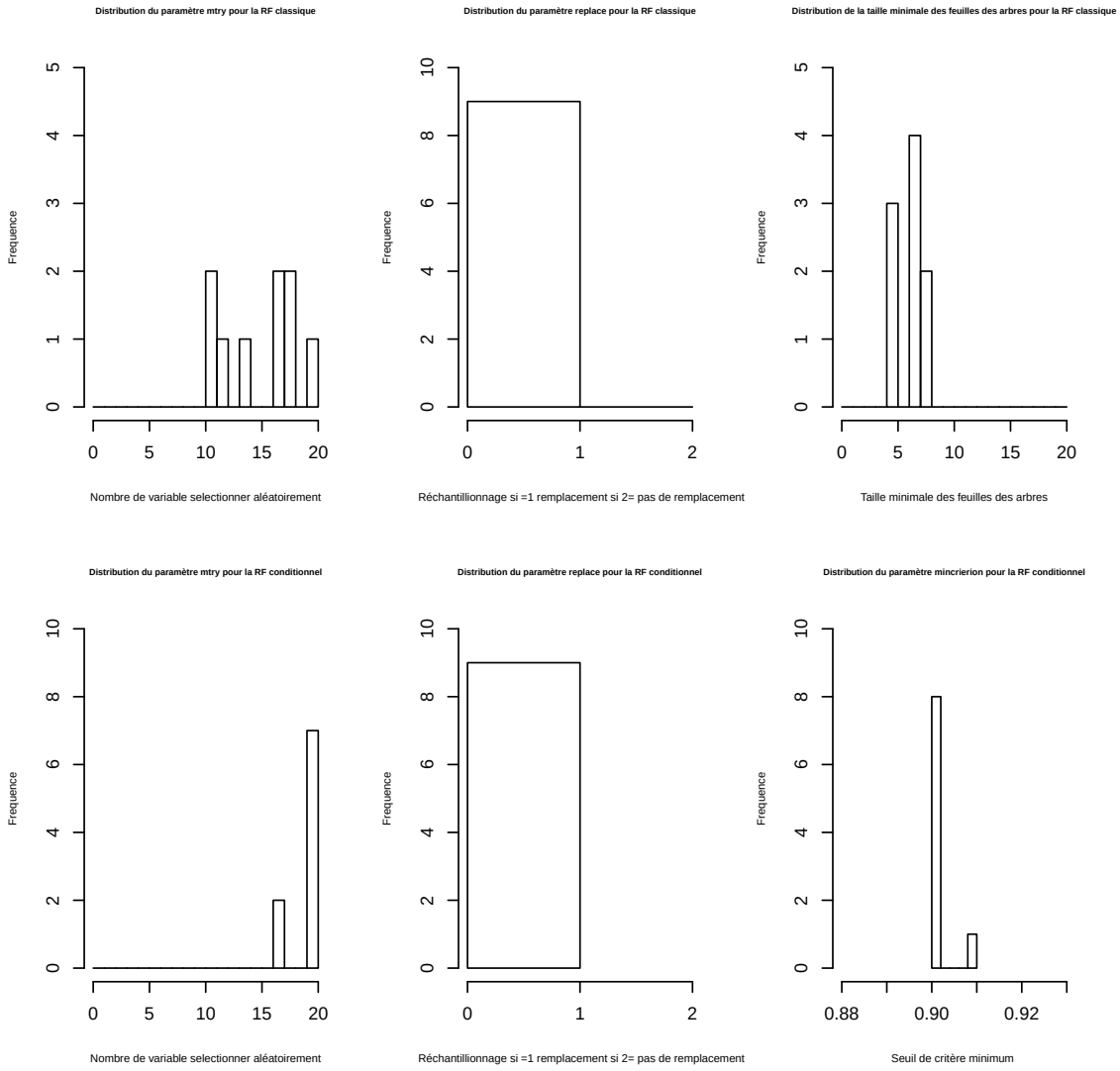

FIGURE 2.1 – Distribution des valeurs des hyper paramètres des modèles lors de 9 entrainements de Forêt aléatoire effectués à l'aide du processus d'optimisation bayésien.

Notons que pour les forêts composées d'arbres d'inférences conditionnelles utilisant les hyper paramètres de l'article de Strobl et al. (2007), nous avons utilisé les hyper pa-

ramètres par défaut (`mincriterion`=0, utilisation des $\frac{2}{3}$ des observations sans remplacement (`Replace=FALSE`)) et avons uniquement fait varier le nombre de variables candidates sélectionnées aléatoirement lors de la séparation de chaque nœud (`mtry`).

Pour les forêts aléatoires construites à partir du package `RandomForestSRC`, il n'est actuellement pas possible de faire varier le processus d'échantillonnage et de mesurer l'importance des variables, nous avons donc utilisé le processus de bootstrap par défaut. Nous avons donc fait varier le nombre de variables candidates sélectionnées aléatoirement lors de la séparation de chaque nœud (`mtry`) et la taille minimale des nœuds terminaux de chaque arbre.

Afin d'observer la cohérence de l'optimisation bayésienne des hyper paramètres dans le contexte de notre simulation, nous avons appliqué cet méthode à 9 échantillons aléatoires de modèles de base tels que décrits à la section 2.2.1. Les résultats de la figure 2.1 montrent que le choix des hyper paramètres est sensiblement le même pour les échantillons.

Nous avons fixé le nombre d'arbres par forêt à 500. Ce choix s'est effectué suite à l'observation visuelle de la convergence de la forêt aléatoire sur l'ensemble d'entrainement du modèle de base comme le montre la figure 2.2.

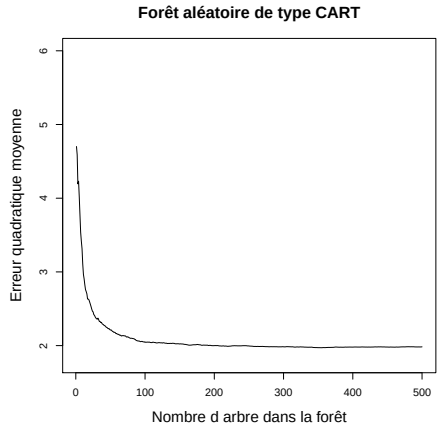

FIGURE 2.2 – Performance en fonction du nombre d'arbres. Obtenu à l'aide du package R "*RandomForest*".

Calcul des méthodes de VIM

Le temps de calcul a été un enjeu de taille pour l'évaluation des différentes méthodes de VIM.

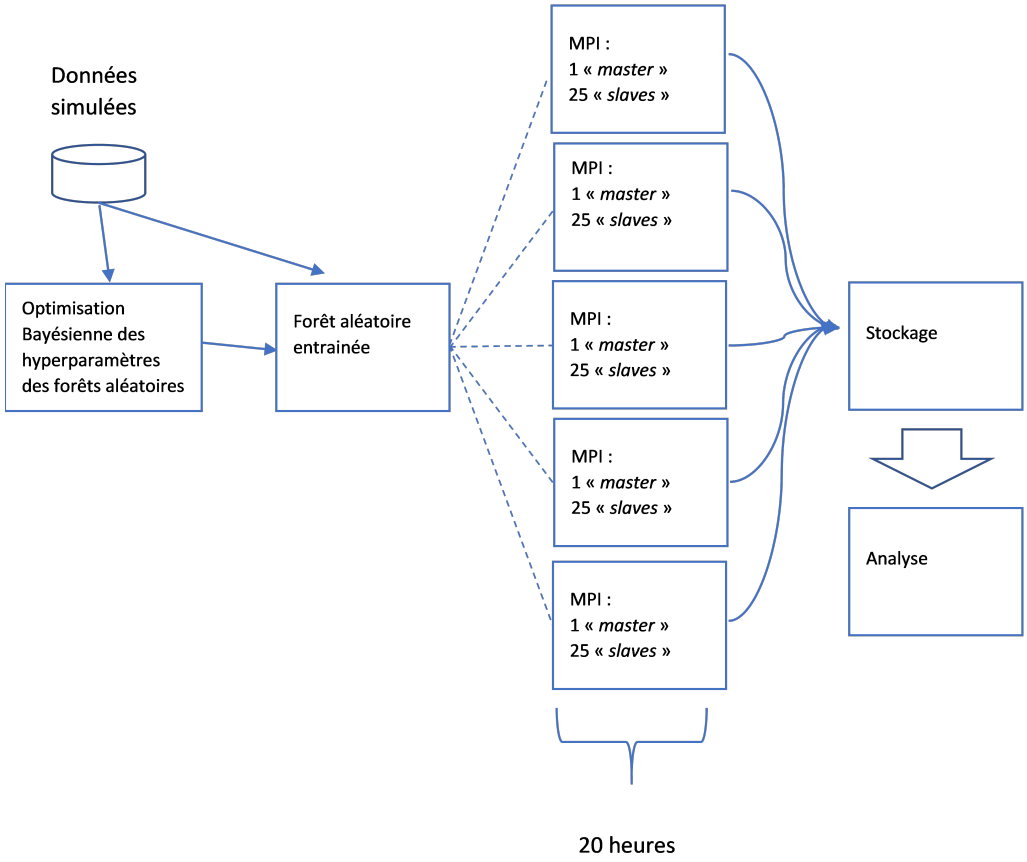

FIGURE 2.3 – Illustration du processus de calcul des simulations. La première étape a consisté à optimiser les hyper paramètres de la forêt aléatoire utilisés sur le jeu de données du scénario de simulation testé. La deuxième étape a consisté à calculer l'importance de chaque variable. Cette étape étant coûteuse en temps, nous avons distribué les taches sur 5 noeuds ou 10 noeuds (pour les simulations possédant 50 variables) sur le serveur *Calcul Canada*. Nous avons utilisé le package `Rmpi` (une interface de MPI (Message Passing Interface)) pour effectuer la parallélisation au sein des noeuds. Chaque noeud sont composés de 1 *"master"* et de 25 *"slave"*. Tous les nœuds utilisés ont des processeurs E5-2683 V4 d'Intel avec une fréquence de 2.1GHz et 95 Gio de mémoire ram. Le temps de calcul pour les 8 méthodes de VIM varie entre 15 et 20 heures pour les 500 répétitions des simulations, pour 1 scénario donné. Le résultat de la simulation est ensuite stocké dans un répertoire puis analysé.

Pour chaque scénario de simulation (voir tableau 2.3.1), nous avons répété le processus de simulation des données 500 fois en faisant varier le germe aléatoire du jeu de données initial.

Nous avons fixé le nombre de permutations à 500 pour les méthodes de VIM calculées à partir de la permutation de Breiman. Il n'est pas possible d'utiliser un ordinateur classique pour effectuer ces simulations, tous les calculs ont donc été effectués sur les serveurs de *Calcul Canada* à l'aide des grappes de calcul *Graham* et *Cedar*. Nous avons eu recours au package R `Rmpi` et au package R `dosnow` afin de paralléliser les calculs. Une partie des codes utilisés sont disponibles à l'annexe B. Le processus de calcul est détaillé à l'aide de la figure 2.3. Ce processus a été effectué pour les 21 scénarios de simulation.

Le calcul de la méthode VIM_CTREE_corrélation s'est révélé problématique. Cette méthode s'est avérée trop chronophage en termes de temps de calcul. A ce titre nous avons testé cette méthode uniquement dans le contexte des simulations testant la robustesse des corrélations. Lorsque cette méthode fut utilisé, nous avons limité le nombre de permutations à 20.

Afin de faciliter l'analyse des résultats, nous avons associé une mesure d'erreur à chaque mesure d'importance. Le calcul de cette erreur a été effectué de la manière suivante : Pour les K variables du modèle.

1. Calcul de l'importance relative $VIM_k^{relative}$ pour chaque variable $k = 1, ..., k$ à partir de l'importance brute VIM_k pour chaque itération i= 1,...,500.
$$VIM_{ki}^{relative} = \frac{VIM_{ki}}{\sum_{k=1}^{nbVariable} VIM_{ki}}.$$
2. Calcul de la différence au carré entre chaque importance de variable k théorique et l'importance relative pour chaque itération i= 1,...,500.
$$Diff_{ki} = (VIM_{ki}^{relative} - VIM_{ki}^{theorique})^2.$$
3. Calcul de l'erreur totale de chaque simulation en sommant la différence au carré de chaque variable k pour chaque itération i= 1,...,500.
$$ErreurTotal_i = \sum_{k=1}^{nbVariable} Diff_{ki}.$$

4. Calcul de la moyenne et de l'écart type de l'erreur totale au travers des 500 itérations.

$$Erreur_{final} = \frac{1}{500}\sum_{i=1}^{500} ErreurTotal_i.$$

2.3 Résultats

2.3.1 Résultats des scénarios de simulation

Dans cette partie, nous détaillons les résultats obtenus pour chaque scénario. Un résumé des résultats numériques peut être consulté au tableau 2.2. Il est nécessaire de rappeler que nous avons évalué les 8 méthodes en fonction de leur proximité avec une définition commune et intuitive d'importance de variable que nous avons calculé de manière théorique.

Scénario de simulation 1

Le premier scénario de simulation a consisté à comparer les différentes méthodes de VIM lorsque le nombre de variables non explicatives dans le modèle augmente. Pour ce faire nous avons utilisé le modèle de base (décrit à la sous section 2.2.1) et avons fait varier uniquement le nombre de variables non explicatives. Il y a donc en tous temps 5 variables explicatives et la relation entre la variable indépendante et les variables dépendantes est celle du modèle de base, soit $Y = -1.8x_1 + 1.6x_2 + 1.4x_3 - 1.7\sin(x_4) + 0.2\exp(x_5) + \varepsilon$ où $\varepsilon \sim N(0,2)$. Nous avons expérimenté les situations où le nombre de variables non explicatives est égal à 0, 5, 15 et 45. Les résultats sont présentés à la figure 2.4 pour les méthodes VIM_CART_Non_Standardisée et VIM_Impureté. Les résumés de tous les résultats se trouvent dans le tableau 2.2.

Nous constatons dans un premier temps dans le tableau 2.2 que l'ajout de variables non explicatives fait légèrement augmenter l'erreur des méthodes de VIM ; cette tendance générale est vraie pour toutes les mesures d'importances étudiées. Malgré ce phénomène, les importances des variables mesurées par des méthodes de VIM sont proches de la

valeur théorique ce qui est prometteur, de plus il est important de préciser que l'ordre d'importance moyen de chaque variable est correctement prédit pour chaque méthode de VIM pour tous les cas étudiés dans le scénario 1, ce qui est très encourageant.

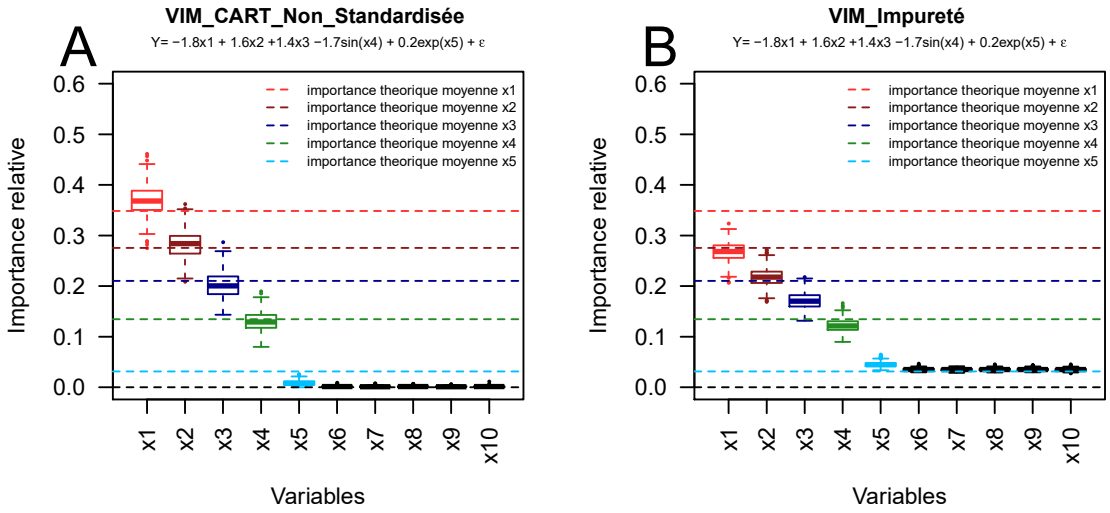

FIGURE 2.4 – Diagrammes en boite des importances relatives accordées aux variables explicatives (x1-x5) et non explicatives (x6-x10) pour les méthodes VIM_CART_Non_Standardisée et VIM_Impureté au travers des 500 simulations. Le graphiques A et B représentent les résultats de la situation 2 du scénario de simulation 1 pour les méthodes VIM_CART_Non_Standardisée et VIM_Impureté.

Dans un deuxième temps, nous observons que la méthode VIM_CART_Standardisée et la méthode VIM_Impureté ont tendance à donner plus d'importance aux variables non importantes comparativement aux autres méthodes testées. Un exemple de ce phénomène est illustré à l'aide de la figure 2.4. Il est en effet possible de constater dans cette illustration que la méthode VIM_Impureté (figure 2.4.B) concède plus d'importance aux

variables n'ayant pas d'importance (X_6, X_7, X_8, X_9, X_{10}) que la méthode VIM_CART Non_Standardisée (figure 2.4.A).

Enfin, nos résultats montrent que la méthode VIM_Ishwaran_Aléatoire et la méthode VIM_Ishwaran_Noeud_Opposé possèdent une erreur élevée par rapport aux autres méthodes étudiées. Cette erreur s'explique par le fait que ces méthodes de VIM ont une erreur élevée dans la prédiction de l'importance des variables non linéaires du modèle de base (X_4 et X_5). Ceci est potentiellement dû la construction des forêts aléatoires à partir de la librairie R `randomforestsrc`, des vérifications supplémentaires seraient pertinentes pour de prochains travaux.

L'annexe C montre la meilleure méthode de VIM pour chaque situation du scénario 1 d'après notre définition de mesure d'importance de variable.

Scénario de simulation 2

Le deuxième scénario de simulation a consisté à comparer les différentes méthodes de VIM lorsque le nombre de variables explicatives dans le modèle augmente. Nous avons testé les situations où le nombre de variables explicatives est égal à 1, 5, 10, 20 et 50 et avons fixé le nombre de variables total à 50 afin de complexifier les modèles. Pour ce scénario, nous avons utilisé des coefficients linéaires fixes pour chaque variable explicative. Ainsi la relation entre la variable indépendante et les variables dépendantes varie en fonction du sous scénario testé. Par exemple, si le nombre de variables explicatives est de 5 alors $Y = 50X_1 + 49X_2 + 48X_3 + 47X_4 + 46X_5 + \varepsilon$ où $\varepsilon \sim N(0,2)$.

Nous constatons dans le tableau 2.2 que lorsque le nombre de variables explicatives augmente, l'erreur de prédiction des mesures d'importance des variables augmente pour toutes les méthodes hormis pour les méthodes VIM_CART_Standardisée et VIM_Impureté. En effet, pour ces 2 méthodes, l'erreur de prédiction de l'importance des variables diminue. Nous expliquons ce phénomène par le fait que ces méthodes accordent plus d'importance aux variables non importantes. Lorsqu'il y a beaucoup de variables explicatives, plus de poids sont donnés aux variables moins importantes, ce qui permet de mieux répartir l'importance des variables. Ce phénomène est illustré en partie à l'aide de l'annexe C.

Malgré les disparités entre les méthodes étudiées, l'ordre d'importance moyen de chaque variable est correctement prédit pour chaque situation du scénario 2.

Scénario de simulation 3

Le troisième scénario de simulation a consisté à comparer les différentes méthodes de VIM lorsque les coefficients des variables explicatives avec effets linéaires, dans le modèle varient. Pour ce scénario, nous avons utilisé le modèle de base (décrit à la sous section 2.2.1) en faisant varier uniquement les coefficients des variables explicatives. Il y a donc en tout temps 5 variables explicatives et 15 variables non explicatives. Pour les 4 situations simulées, la variable dépendante Y est générée pour chaque individu selon les modèles suivant :

— Situations 1 : $Y = 10X_1 + 8X_2 + 6X_3 + 4X_4 + 2X_5 + \varepsilon$ où $\varepsilon \sim N(0,2)$.

— Situations 2 : $Y = 7.5X_1 + 7X_2 + 6.5X_3 + 6X_4 + 5.5X_5 + \varepsilon$ où $\varepsilon \sim N(0,2)$.

— Situations 3 : $Y = 9X_1 + 8X_2 + 4.5X_3 + 4X_4 + 3.5X_5 + \varepsilon$ où $\varepsilon \sim N(0,2)$.

— Situations 4 : $Y = X_1 + X_2 + X_3 + X_4 + X_5 + \varepsilon$ où $\varepsilon \sim N(0,2)$. Dans cette situation les variables indépendantes ne suivent pas une loi normale de moyenne 0 et d'écart type 1. Ici $X_1 \sim N(0,4)$, $X_2 \sim N(0,3)$, $X_3 \sim N(0,2)$, $X_4 \sim N(0,1)$, $X_5 \sim N(0,0.1)$.

En règle générale, l'importance des variables (selon notre définition) est bien prédite dans ce scénario. Ceci s'explique par le fait que le nombre de variables explicatives et le nombre de variables total sont faibles et que les relations entre la variable à prédire Y et les variables explicatives sont linéaires. Dans ce scénario, l'ordre d'importance moyen de chaque variable est respecté pour chaque méthode de VIM.

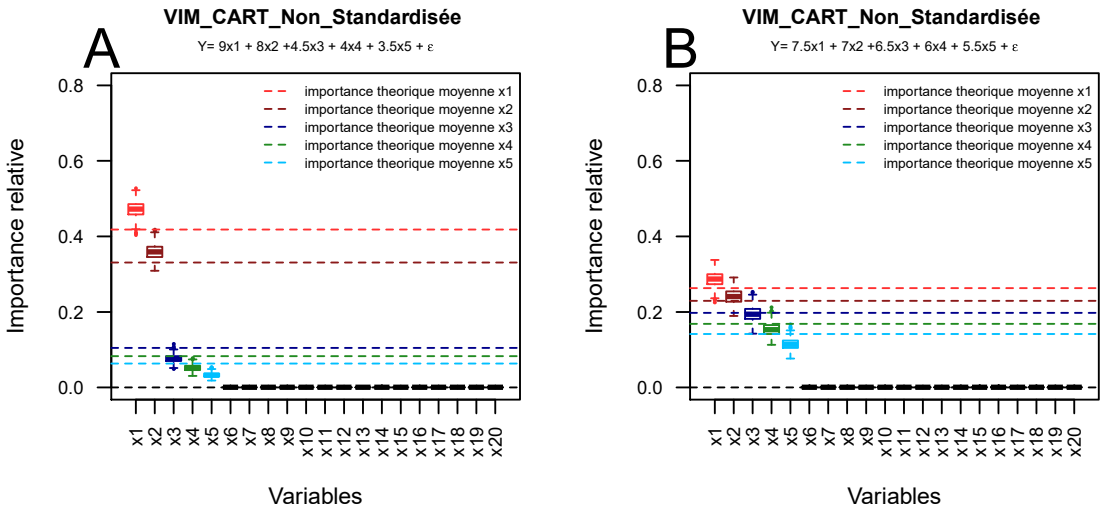

FIGURE 2.5 – Diagrammes en boite des importances relatives accordées aux variables explicatives (x1-x5) et non explicatives (x6-x20) pour la méthode VIM_CART_Non_Standardisée. Le graphique A représentent les résultats de la situation 3 du scénario 3 pour la méthode VIM_CART_Non_Standardisée. Le graphique B représentent les résultats de la situation 3 du scénario 2 pour la méthode VIM_CART_Non_Standardisée. Il est possible de constater qu'une erreur de prédiction de X_1 dans la situation 3 (graphique A) engendra plus de bruit qu'une variation de X_1 dans la situation 2 (graphique B).

Par la suite, nous avons remarqué que plus la différence entre les coefficients des variables explicatives est élevé, plus l'erreur est importante. Ceci s'explique par le fait que nous étudions des mesures d'importance relative. Si une importance relative de variable est sous-estimée ou sur-estimée cela peut se répercuter sur une autre importance relative de variable. L'effet du biais risque d'être plus élevé dans le cas où la différence entre les coefficients des variables explicatives est grande. La figure 2.5 illustre en partie ce phénomène. Le graphique A représente les résultats de la méthode VIM_Impureté pour la situa-

tion 3 et le graphique B représente les résultats de la méthode VIM_CART_Standardisée pour la situation 2. Il est possible de constater qu'une erreur de prédiction de l'importance de X_1 dans la situation 3 engendra plus de bruit qu'une erreur de prédiction de l'importance de X_1 dans la situation 2. Enfin, il est intéressant d'observer que lors de la situation 4, qui consistait à faire varier la variance des variables explicatives, seule la méthode VIM_CART_Standardisée a obtenu des résultats éloignés de leurs valeurs théoriques. Ce résultat est observable à l'annexe C.

Scénario de simulation 4

Le quatrième scénario de simulation a consisté à comparer les différentes méthodes de VIM lorsque les coefficients des variables explicatives dans le modèle varient. Pour ce scénario, nous avons utilisé le modèle de base (décrit à la sous section 2.2.1) en faisant varier uniquement les coefficients des variables explicatives avec effets linéaires et non linéaires. Il y a donc en tout temps 5 variables explicatives et 15 variables non explicatives. Pour les 4 situations, la variable dépendante Y est générée pour chaque individu selon les modèles suivant :

— Situation 1 : $2.5\sin(X_1) + 1.4X_2 + 1.1X_3 - 0.8X_4 + 0.2\exp(X_5) + \varepsilon$ où $\varepsilon \sim N(0,2)$.

— Situation 2 : $0.8\exp(X_1) + 1.6X_2 + 1.4X_3 - 1.7\sin(X_4) + X_5 + \varepsilon$ où $\varepsilon \sim N(0,2)$.

— Situation 3 : $3\sin(X_1) + 0.6\exp(X_2) + 1.4X_3 - X_4 + 0.9X_5 + \varepsilon$ où $\varepsilon \sim N(0,2)$.

— Situation 4 : $3\sin(X_1) + 0.6\exp(X_2) + 1.4X_3^2 - X_4 + 0.9X_5 + \varepsilon$ où $\varepsilon \sim N(0,2)$.

En règle générale, plus les coefficients non linéaires des variables explicatives pour prédire Y sont importants, plus la précision des différentes méthodes de VIM se détériore. Dans ce scénario de simulation, l'ordre d'importance moyen des variables n'a pas été en tout temps respecté. La figure 2.6.B illustre ce phénomène pour la méthode VIM_CTREE dans le cas de la situation 4 du scénario 4. On constate que l'importance de la variable X_3 est sous évaluée par rapport à son importance théorique.

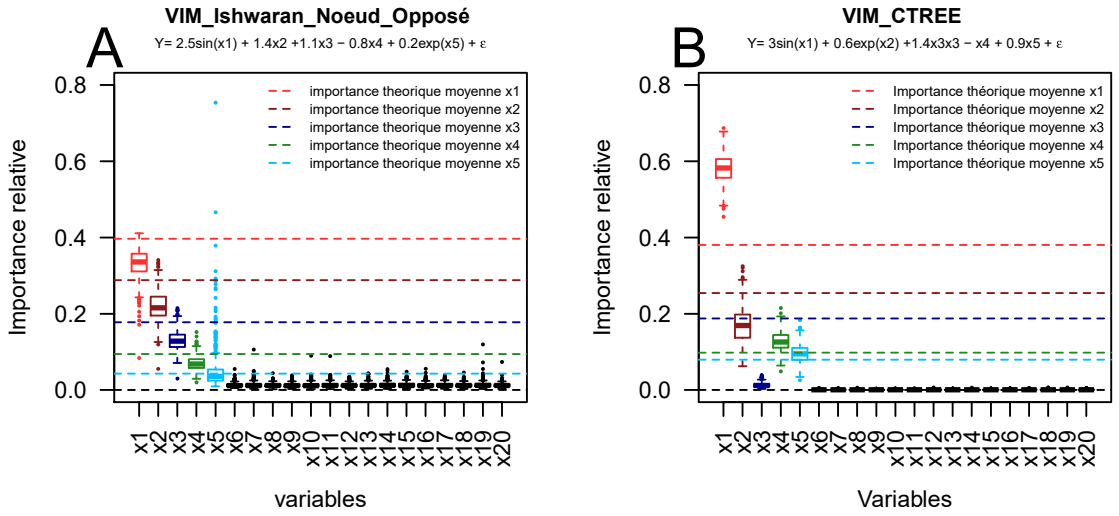

FIGURE 2.6 – Diagrammes en boite des importances relatives accordées aux variables explicatives (x1-x5) et non explicatives (x6-x20) pour les méthodes VIM_Ishwaran_Noeud_Opposé et VIM_CTREE. Le graphique A et B représentent réspectivement les résultats de la situation 1 du scénario 4 pour les méthodes VIM_Ishwaran_Noeud_Opposé et VIM_CTREE.

Encore une fois, les méthodes proposées par Ishwaran (méthode VIM_Ishwaran- -_Aléatoire et méthode VIM_Ishwaran_Noeud_Opposé) ont une erreur de prédiction de l'importance des variables très élevée par rapport aux autres méthodes. Cette observation vient corroborer le fait que ces méthodes ne sont pas nécessairement fiables dans le cas où les relations entre la variable dépendante et les variables explicatives ne sont pas linéaires. Ce phénomène est illustré à l'aide de la figure 2.6. A pour la méthode VIM_Ishwaran_Noeud_Opposé dans la situation 1 du scénario 4. On constate de plus que les variances associées à la prédiction de l'importance des variables non linéaires X_1 et X_5 sont très élevées par rapport aux variables explicatives linéaires.

Scénario de simulation 5

Le cinquième scénario de simulation a consisté à comparer les différentes méthodes de VIM lorsqu'il existe de la corrélation entre les variables explicatives. Pour ce scénario, nous avons utilisé le modèle de base (décrit à la sous section 2.2.1) en ajoutant de la corrélation entre les variables. Il est important de préciser que ce scénario a été effectué dans un but expérimental. Il est souvent délicat de tirer des conclusions fiables lorsqu'il existe de la corrélation entre les variables explicatives. Nous voulions cependant observer comment se comportaient les méthodes de VIM dans ces conditions. Il y a donc en tout temps 15 variables non explicatives et 5 variables explicatives. La relation entre la variable indépendante et les variables dépendantes est celle du modèle de base soit

$$Y = -1.8X_1 + 1.6X_2 + 1.4X_3 - 1.7\sin(X_4) + 0.2\exp(X_5) + \varepsilon \text{ où } \varepsilon \sim N(0,2).$$

Nous avons simulé 4 types de situations.

La première situation a consisté à introduire de la corrélation (70%) entre deux variables importantes, X_1 et X_2. Nous constatons, avec le tableau 2.2, que dans ce scénario toutes les méthodes de VIM (hormis la méthode VIM_CTREE_corrélation) ont prédit un ordre d'importance moyen des variables non correct. En effet toutes les méthodes ont eu tendance à surestimer l'importance de X_2 par rapport au reste des variables explicatives.

Il est intéressant d'observer que la variance associée à l'importance des variables mesurées par la méthode VIM_CTREE_corrélation s'est révélée très élevée et en moyenne l'erreur de mesure d'importance associée à cette méthode de VIM est plus importante que pour le reste des méthodes. Cette méthode est fiable pour déterminer l'ordre d'importance moyen des variables mais semble peu adaptée pour évaluer l'importance relative de chaque variable de manière individuelle. Ces propos sont illustrés à l'aide du graphique A et B de la figure 2.7.

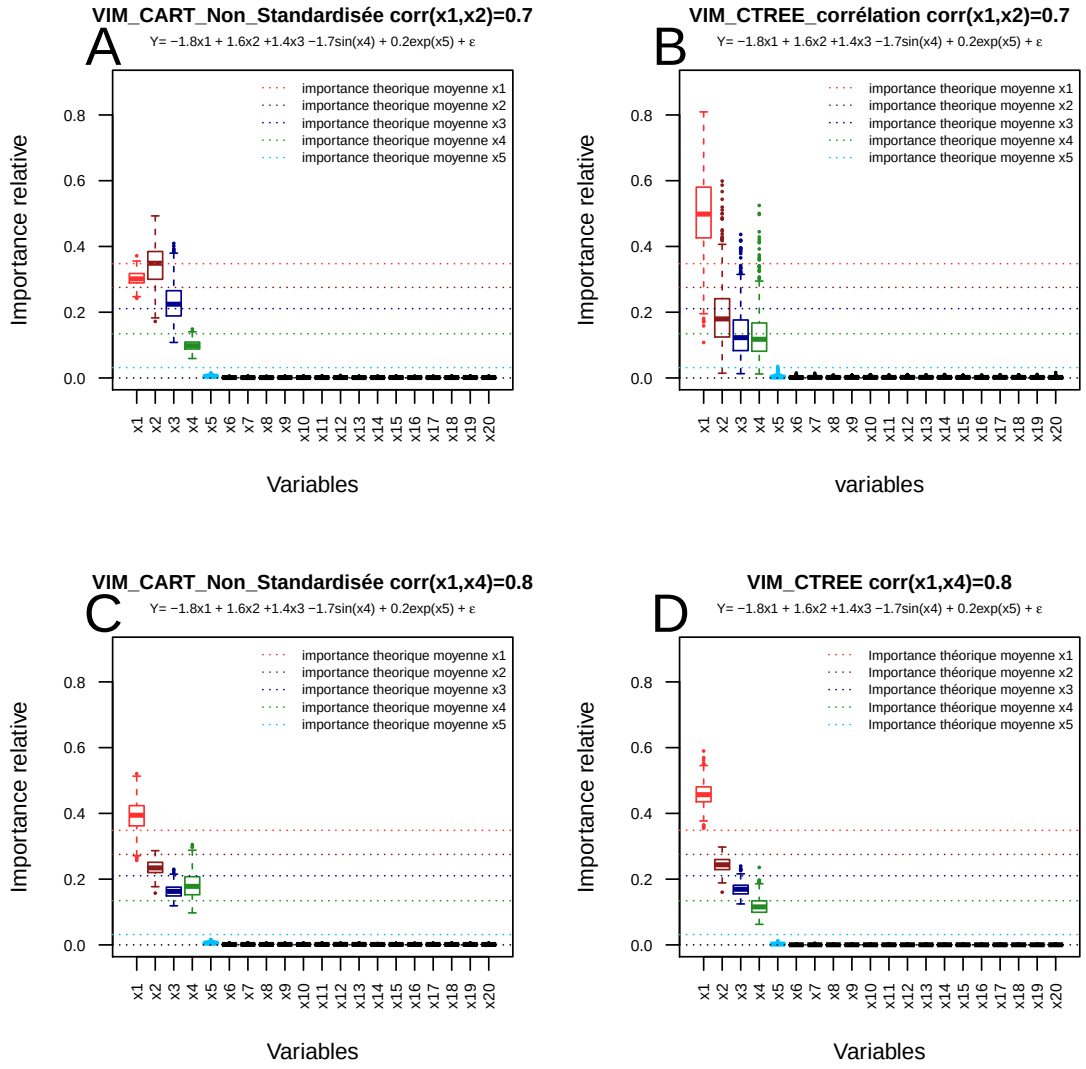

FIGURE 2.7 – Diagrammes en boite des importances relatives accordées aux variables explicatives (x1-x5) et non explicative (x6-x20) pour les méthodes VIM_CART_Non_Standardisée, VIM_CTREE et VIM_CTREE_Corrélation. Le graphiques A et B représentent respectivement les résultats de la situation 1 du scénario 5 pour les méthodes VIM_CART_Non_Standardisée et VIM_CTREE_Corrélation. Dans cette situation le modèle de base est utilisé en introduisant une corrélation entre x_1 et X_2 de 0.7. Le graphique C et D représentent réspectivement les résultats de la situation 2 du scénario 5 pour les méthodes VIM_CART_Non_Standardisée et VIM_CTREE. Dans cette situation le modèle de base est utilisé en introduisant une corrélation entre X_1 et X_4 de 0.8.

La deuxième situation a consisté à introduire de la corrélation (80%) entre une variable

importante X_1 et une variable de faible importance X_4. Dans cette situation, l'ordre moyen des méthodes de VIM est respecté uniquement pour les méthodes construites à partir de forêts aléatoires composées de CTREE. Les méthodes construites à partir de forêts aléatoires composées d'arbres CART ont tendance à surestimer l'impact de la variable de faible importance x_4. Ces propos sont illustrés à l'aide du graphique C et D de la figure 2.7.

La troisième situation à consisté à introduire de la corrélation (80%) entre une variable importante X_1 et une variable non explicative X_6. Dans cette situation, toute les méthodes ont prédit un ordre d'importance moyen incorrect en concédant trop d'importance à X_6. Enfin, la quatrième situation a consisté à introduire simultanément trois type de relation de corrélation entre les variables :

— Introduction de corrélation (80%) entre une variable importante X_1 et une variable non explicative X_6.

— Introduction de corrélation entre deux variables moyennement importantes (60%), X_2 et X_3.

— Introduction de corrélation entre deux variables moyennement importantes (70%), X_3 et X_4.

Dans cette situation, toute les méthodes ont prédit un ordre d'importance moyen faussé.

Simulation[1234]	Modèle	Variables total	Variables explicatives	VIM CART Non Standardisée	VIM CART Standardisée	VIM CTREE	VIM CTREE Strobl	VIM Impureté	VIM CTREE Corrélation	VIM Ishwaran Aléatoire	VIM Ishwaran Noeud Opposé
Variation du nombre de variable non explicative	$Y=-1.8x_1+1.6x_2+1.4x_3-1.7\sin(x_4)+0.2\exp(x_5)+\varepsilon$	5	5	**0.28[0.0079]**	0.40[0.0097]	0.46[0.0113]	0.42[0.0104]	0.46[0.0087]	NA	0.9[0.0912]	0.72[0.0917]
	$Y=-1.8x_1+1.6x_2+1.4x_3-1.7\sin(x_4)+0.2\exp(x_5)+\varepsilon$	10	5	**0.32[0.0097]**	0.34[0.0087]	0.50[0.0130]	0.44[0.0119]	1.88[0.0115]	NA	1.12[0.0765]	1.43[0.0648]
	$Y=-1.8x_1+1.6x_2+1.4x_3-1.7\sin(x_4)+0.2\exp(x_5)+\varepsilon$	20	5	0.48[0.0136]	**0.32[0.0081]**	0.63[0.0166]	0.52[0.0142]	1.87[0.0121]	NA	1.5[0.1346]	1.96[0.1315]
	$Y=-1.8x_1+1.6x_2+1.4x_3-1.7\sin(x_4)+0.2\exp(x_5)+\varepsilon$	50	5	0.59[0.0167]	0.68[0.0097]	0.69[0.0179]	**0.57[0.0155]**	2.66[0.0145]	NA	2.22[1.63]	2.09[1.39]
Variation du nombre de variable explicative (Linéaire)	$Y=50x_1+\varepsilon$	50	1	5.96e-04 [1.25e-04]	7.43 [1.02e-01]	1.67e-08 [3.18e-10]	**1.91e-09 [6.19e-11]**	2.93e-04 [1.32e-05]	NA	5.69e-04 [1.23e-04]	2.2e-04 [4.74e-05]
	$Y=50x_1+49x_2+48x_3+47x_4+46x_5+\varepsilon$	50	5	0.20[0.0062]	0.69[0.0087]	**0.15[0.0049]**	0.2[0.0061]	0.62[0.0066]	NA	0.34[0.0049]	0.77[0.0210]
	$Y=50x_1+49x_2+...+42x_9+41x_{10}+\varepsilon$	50	10	0.92[0.0180]	**0.63[0.0066]**	1.00[0.0187]	1.12[0.0197]	1.04[0.0067]	NA	0.77[0.0107]	1.1[0.0192]
	$Y=50x_1+49x_2+...+32x_{19}+31x_{20}+\varepsilon$	50	20	1.23[0.0223]	**0.38[0.0039]**	1.64[0.0280]	1.82[0.0299]	0.74[0.0053]	NA	0.77[0.0116]	0.99[0.0165]
	$Y=50x_1+49x_2+...+2x_{49}+1x_{50}+\varepsilon$	50	50	1.57[0.0295]	**0.31[0.0037]**	1.95[0.0333]	1.99[0.0313]	0.45[0.0049]	NA	0.67[0.0228]	0.88[0.0282]
Variation des coefficients des variables explicatives (Linéaire)	$Y=10x_1+8x_2+6x_3+4x_4+2x_5+\varepsilon$	20	5	0.43[0.0064]	1.16[0.0146]	0.48[0.0065]	0.53[0.0073]	**0.23[0.0032]**	NA	0.28[0.0085]	2.39[0.0317]
	$Y=7.5x_1+7x_2+6.5x_3+6x_4+5.5x_5+\varepsilon$	20	5	0.27[0.0065]	**0.15[0.0029]**	0.27[0.0064]	0.34[0.0076]	0.43[0.0054]	NA	0.27[0.0054]	1.03[0.0242]
	$Y=9x_1+8x_2+4.5x_3+4x_4+3.5x_5+\varepsilon$	20	5	0.68[0.0077]	0.72[0.0100]	0.79[0.0082]	0.82[0.0085]	**0.28[0.0040]**	NA	0.36[0.0089]	1.96[0.0371]
	$Y=x_1+x_2+x_3+x_4+x_5+\varepsilon$[5]	20	5	**0.28[0.0069]**	2.09[0.0237]	0.33[0.0074]	0.34[0.0076]	0.45[0.0034]	NA	0.29[0.0072]	2.49[0.0353]
Variation des coefficients des variables explicatives (Non linéaire)	$Y=2.5\sin(x_1)+1.4x_2+1.1x_3-0.8x_4+0.2\exp(x_5)+\varepsilon$	20	5	**0.60[0.0153]**	0.75[0.0134]	0.67[0.0159]	0.57[0.0143]	2.88[0.0195]	NA	4.24[0.1511]	2.11[0.1368]
	$Y=0.8\exp(x_1)+1.6x_2+1.4x_3-1.7\sin(x_4)+x_5+\varepsilon$	20	5	**2.70[0.0603]**	6.26[0.1343]	3.19[0.0631]	3.51[0.0650]	4.41[0.0330]	NA	9.89[0.3175]	13.23[0.3765]
	$Y=3\sin(x_1)+0.6\exp(x_2)+1.4x_3-x_4+0.9x_5+\varepsilon$	20	5	2.74[0.0498]	**2.02[0.0500]**	3.29[0.0507]	3.28[0.0528]	2.47[0.0189]	NA	10.75[0.3935]	8.23[0.4132]
	$Y=3\sin(x_1)+0.6\exp(x_2)+1.4x_3^2-x_4+0.9x_5+\varepsilon$	20	5	4.74[0.0629]	**2.39[0.0482]**	8.24[0.0805]	7.83[0.0703]	2.93[0.0188]	NA	11.54[0.3739]	9.1[0.4087]
Ajout de corrélation	$Y=-1.8x_1+1.6x_2+1.4x_3-1.7\sin(x_4)+0.2\exp(x_5)+\varepsilon$[6]	20	5	**1.65[0.0479]**	2.21[0.0272]	1.93[0.0611]	2.81[0.0882]	3.22[0.0385]	7.57[0.2665]	2.42[0.0876]	5.6[0.1377]
	$Y=-1.8x_1+1.6x_2+1.4x_3-1.7\sin(x_4)+0.2\exp(x_5)+\varepsilon$[7]	20	5	**0.79**[0.0279]	1.30[0.0361]	1.26[0.0319]	3.07[0.0457]	2.62[0.0350]	6.44[0.2529]	1.85[0.0429]	2.13[0.0393]
	$Y=-1.8x_1+1.6x_2+1.4x_3-1.7\sin(x_4)+0.2\exp(x_5)+\varepsilon$[8]	20	5	**0.48[0.0108]**	2.52[0.0256]	0.48[0.0122]	0.76[0.0137]	2.8[0.0141]	6.24[0.2866]	2.34[0.0563]	1.83[0.0478]
	$Y=-1.8x_1+1.6x_2+1.4x_3-1.7\sin(x_4)+0.2\exp(x_5)+\varepsilon$[9]	20	5	19.2[0.0822]	**9.2**[0.0604]	20.6[0.0802]	20.1[0.0771]	13.3[0.0568]	23.8[0.2229]	11.2[0.0930]	18.6[0.1176]

TABLE 2.2 – Résultat des scénarios de simulation : erreur des méthodes de VIM telles que décrites à la section 2.2.3.

1)Le score entre [..] correspond à l'écart type de l'erreur des méthodes de VIM des 500 itérations telles que décrites à la section 2.2.3.

2) L'erreur la plus faible de chaque mesure d'importance pour chaque scénario est représenté en rouge.

3) Sauf indication contraire, toutes les variables simulées sont distribuées selon une loi normale de moyenne 0 et d'écart type 1.

4) $\varepsilon \sim N(0,2)$

5) Dans cette situation $x_1 \sim N(0,4)$,$x_2 \sim N(0,3)$,$x_3 \sim N(0,2)$,$x_4 \sim N(0,1)$,$x_5 \sim N(0,0.1)$

6) Structure de corrélation cor(x1,x2)=0.7

7) Structure de corrélation cor(x1,x4)=0.8

8) Structure de corrélation cor(x1,x6)=0.8

9) Structure de corrélation cor(x1,x6)=0.8 cor(x2,x3)=0.6 cor(x3,x4)=0.6

Résumé des analyses des scénarios de simulation

Nous présentons dans cette sous-section le résumé des analyses des scénarios de simulation. Ces résultats ont été obtenus selon notre définition de la qualité des méthodes de VIM. Ainsi, d'après notre définition, une variable qui a beaucoup d'importance est une variable qui, si elle n'était pas utilisée pour prédire la variable cible, engendrerait une augmentation de l'erreur de prédiction et cela de manière proportionnelle à son importance. Notre définition ne prend pas en compte le rôle des variables dans le processus de construction de l'arbre.

1. Les méthodes de VIM étudiées sont fiables lorsque les relations entre les variables cibles et les variables explicatives sont linéaires.
2. Les méthodes de VIM analysées sont robustes à un nombre de variables non explicatives élevé.
3. La méthode VIM_CART_Non_Standardisée et la méthode VIM_CART_Standardisée sont les méthodes qui obtiennent, en général, la moyenne de l'erreur de mesure d'importance la plus faible.
4. La méthode VIM_CART_Standardisée et la méthode VIM_Impureté sont robustes à une augmentation du nombre de variables explicatives linéaires. Elles ont par ailleurs tendance à répartir l'importance des variables de manière plus équitables entre les différentes variables.
5. Nous n'avons pas observé de différence majeure entre la méthode VIM_CART_-Non_Standardisée (optimisée à l'aide du processus d'optimisation bayésien) et la méthode VIM_CTREE_Strobl.
6. D'après notre scénario de simulation expérimental, nous observons que la corrélation semble être problématique dans le contexte d'évaluation d'importance de variables. Aucune méthode ne s'est révélée meilleure dans toutes les situations. Nous avons déterminé que la méthode de VIM _CTREE_corrélation est utile pour déterminer l'ordre d'importance moyen des variables dans le cas où deux variables

importantes sont corrélées. Cette méthode s'est en revanche avérée peu fiable pour déterminer le niveau d'importance de chaque variable tel que mesuré d'après notre définition. De plus, lorsque plusieurs variables sont corrélées, cette méthode s'est révélée biaisée par rapport à notre définition .

2.3.2 Scénario de simulation à partir de données de la littérature

Pour cette sous section, nous nous sommes inspirés des jeux de données utilisés dans le mémoire de Isabelle Bernard dont le but était de comparer la prédiction de deux types de forêts aléatoires. Il nous semblait intéressant d'observer si la méthodologie appliquée dans notre mémoire pouvait être appliquée à des données ayant déjà été utilisées dans la littérature. Parmi les jeux de données utilisés par Isabelle Bernard, nous en avons sélectionné 3 qui avaient pour but de prédire une variable de type continue. Les 3 jeux de données sont disponibles dans le package R `mlbench`. Ces jeux de données furent utilisées par Friedman (1991) puis par Breiman (1996).

Premier jeu de données simulé de Friedman (Friedman1) :

Les caractéristiques du jeu de données dans le cadre de nos évaluations d'importance des variables sont les suivantes :

— Nombre d'individus pour l'entrainement : 1000

— Nombre d'individus pour la validation : 200

— Nombre de variables explicatives : 5

— Nombre de variables totales dans le modèle : 10

La variable dépendante Y est générée pour chaque individu selon le modèle suivant :

$$Y = 10\sin(\pi X_1 X_2) + 20(X_3 - 0.5)^2 + 10X_4 + 5X_5 + \varepsilon,$$

où $\varepsilon \sim N(0, sd)$.

Les variables indépendantes sont distribuées de manière uniforme entre 0 et 1. Nous avons effectué deux scénarios à partir de ces modèles. Dans le premier scénario, nous avons fixé l'écart type de l'erreur (sd) à 1 comme cela est suggéré dans la littérature. Le deuxième scénario a consisté à fixer l'écart type de l'erreur (sd) à 2.

Deuxième jeu de données simulé de Friedman (Friedman2) :

Nous avons effectué quelques modifications par rapport au modèle utilisé par les auteurs. Ces modifications ont été nécessaires pour calculer l'importance théorique de chaque variable. En effet, nous calculons l'importance théorique de chaque variable en posant successivement les variables testées à 0. Il n'est donc pas possible de calculer l'importance théorique dans certaines conditions comme par exemple pour la relation $\frac{1}{X}$. Les caractéristiques du jeu de données dans le cadre de nos évaluations d'importance des variables sont les suivantes :

— Nombre d'individus pour l'entrainement : 1000

— Nombre d'individus pour la validation : 200

— Nombre de variables explicatives : 4

— Nombre de variables totales dans le modèle : 4

La variable dépendante Y est générée pour chaque individu selon le modèle suivant :

$$Y = X_1^2 + (X_2X_3 - (X_2X_4)^2)^{0.5} + \varepsilon,$$

où $\varepsilon \sim N(0, sd)$.

Les variables indépendantes sont distribuées de manière uniforme définie par les rangs suivants : $0 \leq X1 \leq 100$, $40\pi \leq X2 \leq 560\pi$, $0 \leq X3 \leq 1$ et $1 \leq X4 \leq 11$.

Nous avons effectué deux scénarios à partir de ces modèles. Dans le premier scénario, nous avons fixé l'écart type de l'erreur (sd) à 125 comme cela est suggéré dans la littérature. Le deuxième scénario a consisté à fixer l'écart type de l'erreur (sd) à 250.

Troisième jeu de données simulé de Friedman (Friedman3) :

Nous avons effectué quelques modifications par rapport au modèle utilisé par les auteurs. Ces modifications ont été nécessaires pour calculer l'importance théorique de chaque variable. Les caractéristiques du jeu de données dans le cadre de nos évaluations d'importance des variables sont les suivantes :

— Nombre d'individus pour l'entrainement : 1000

— Nombre d'individus pour la validation : 200

— Nombre de variables explicatives : 4

— Nombre de variables totales dans le modèle : 4

La variable dépendante Y est générée pour chaque individu selon le modèle suivant :

$$Y = \arctan((X_2X_3(1+(X_2X_4))) - 1000X_1) + \varepsilon,$$

où $\varepsilon \sim N(0, sd)$.

Les variables indépendantes sont distribuées de manière uniforme définie par les rangs suivants : $0 \leq X1 \leq 100$, $40\pi \leq X2 \leq 560\pi$, $0 \leq X3 \leq 1$, $1 \leq X4 \leq 11$.

Nous avons effectué deux scénarios à partir de ces modèles. Dans le premier scénario, nous avons fixé l'écart type de l'erreur (sd) à 0.1 comme cela est suggéré dans la littérature. Le deuxième scénario a consisté à fixer l'écart type de l'erreur (sd) à 0.2.

Résultats des scénarios de simulation à partir de données de la littérature

Les résultats numériques sont présentés dans le tableau 2.4 et les graphiques des meilleures modèles pour chaque scénario sont disponibles en annexe D.

Pour les scénarios effectués à l'aide des données "Friedman1", la méthode VIM Ishwaran_Aléatoire est la méthode dont les prédictions obtiennent l'erreur de mesure d'importance la plus faible. Pour les scénarios effectués à l'aide des données "Friedman2" et "Friedman3", la méthode VIM_Impureté est la méthode dont les prédictions obtiennent l'erreur la plus faible. Il est difficile de tirer des conclusions de ces résultats puisque pour toutes les méthodes, l'ordre d'importance moyen des variables n'est pas respecté.

Nous pensons que ces scénarios sont certainement adaptés pour la prédiction mais sont trop complexes pour correctement mesurer l'importance de chaque variable à l'aide des méthodes proposées dans ce mémoire. Ceci peut notamment être causé par les relations d'interactions entre les variables. Il faudrait effectuer des simulations supplémen-

taires pour vérifier si les méthodes de VIM sont robustes dans les cas ou il y a présence d'interaction entre les variables.

Simulation[1][2]	Modèle	Variables total	Variables explicatives	Écart type de l'erreur[3]	VIM CART Non Standardisée	VIM CART Standardisée	VIM CTREE	VIM CTREE Strobl	VIM Impureté	VIM Ishwaran aléatoire	VIM Ishwaran Noeud opposé
Simulation Friedman1	$Y = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + \varepsilon$	10	5	1	2.71[0.0285]	1.66[0.018]	4.04[0.0369]	4.01[0.0367]	2.76[0.0277]	**1.23[0.0166]**	4.09[0.0374]
	$Y = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + \varepsilon$	10	5	2	2.7[0.0354]	1.54[0.0195]	4.09[0.0448]	3.94[0.0428]	2.79[0.0268]	**1.4[0.0204]**	4.5[0.0499]
Simulation Friedman2	$Y = (x_1^2 + (x_2 x_3 - (x_2 x_4)^2)^{0.5} + \varepsilon$	4	4	125	0.56[0.0124]	0.51[0.0088]	0.59[0.0136]	0.62[0.0147]	**0.48[0.0117]**	0.62[0.0131]	1.99[0.0529]
	$Y = (x_1^2 + (x_2 x_3 - (x_2 x_4)^2)^{0.5} + \varepsilon$	4	4	250	0.57[0.0126]	0.52[0.0092]	0.58[0.0138]	0.62[0.0147]	**0.45[0.0114]**	0.62[0.0131]	1.97[0.053]
Simulation Friedman3	$Y = \arctan((x_2 x_3 (1 + (x_2 x_4))) - 1000x_1) + \varepsilon$	4	4	0.1	8.29[0.103]	5.69[0.0894]	10.2[0.1305]	11.26[0.1434]	**5.23[0.071]**	8.52[0.1925]	10.73[0.228]
	$Y = \arctan((x_2 x_3 (1 + (x_2 x_4))) - 1000x_1) + \varepsilon$	4	4	0.2	8.06[0.1022]	5.7[0.0857]	10.31[0.1337]	11.16[0.1436]	**5[0.0496]**	8.6[0.1739]	10.48[0.193]

TABLE 2.4 – Résultat des scénarios de simulation des jeux de données de Friedman : erreur des méthode de VIM telles que décrites à la section 2.2.3.

1)Le score entre [..] correspond à l'écart type de l'erreur des méthodes de VIM des 500 itérations telles que décrites à la section 2.2.3.

2) L'erreur la plus faible de chaque mesure d'importance pour chaque scénario est représenté en rouge.

3) L'écart type de l'erreur correspond à l'écart type de la distribution de $\varepsilon \sim N(0,\sigma)$

Chapitre 3

Application à la génétique

L'objectif de ce chapitre est d'utiliser les deux meilleures méthodes de VIM déterminées au chapitre 2 afin de classer par ordre d'importance des variables génétiques dans la prédiction d'une mesure d'intelligence générale (le facteur G). À terme, le but de cette application est d'apporter une contribution supplémentaire dans la compréhension des troubles neuro-développementaux en élargissant le domaine d'analyse du laboratoire du docteur Jacquemont au centre de recherche de l'hôpital du CHUM de Saint-Justine.

3.1 Introduction aux concepts de base en génétique

La génétique est un des domaines de la biologie définit par l'étude de la transmission des caractères héréditaires. Il est admis que ce sont les travaux publiés par Gregor Mendel en 1886 qui sont à l'origine de cette discipline. C'est au cours de ses expériences sur la transmission des caractéristiques morphologiques (couleur, taille ...) chez les petits pois, qu'il a pu définir que celles-ci se transmettent d'une génération à une autre et non de manière aléatoire. De ces travaux sont nées les lois de Mendel et les premières bases de la génétique. C'est dans les années 50, soit un siècle plus tard, que le support de l'information héréditaire a été découvert. Ce sont les travaux de Rosaline Franklin qui ont permis aux Docteur Watson et Crick d'identifier la structure en double hélice de l'acide désoxyribonucléique (ADN) comme étant le support de l'information génétique. Cette découverte

annonça les prémices de la génétique moléculaire.

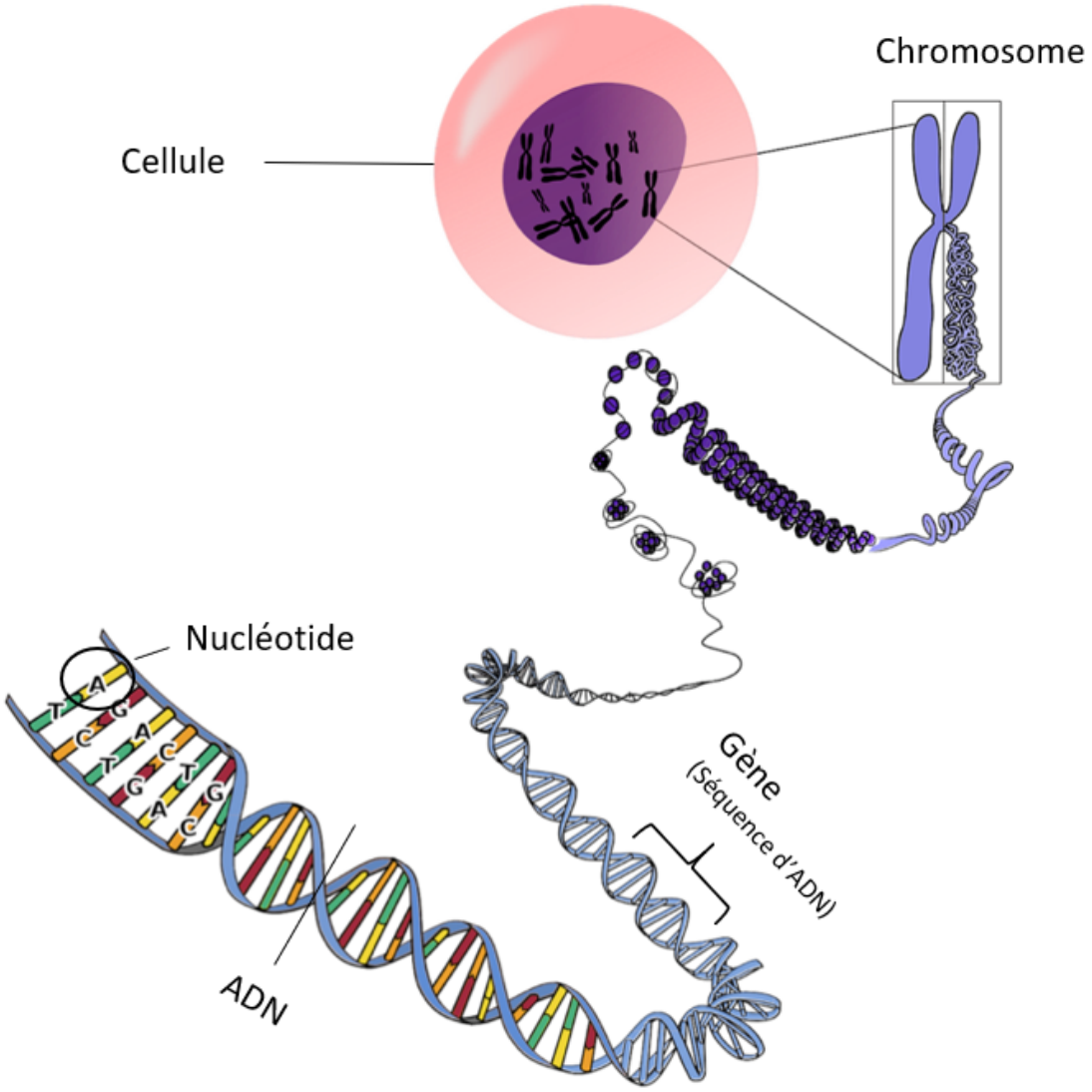

FIGURE 3.1 – Illustration de différentes composantes du matériel génétique. Chaque être humain est composé de cellules formées d'un noyau. Chaque noyau contient 23 paires de chromosomes qui sont héritées des parents de l'individu. Chaque chromosome est constitué d'une double hélice D'ADN (acide désoxyribonucléique). C'est cette structure qui code l'information nécessaire au fonctionnement des cellules. L'ADN est composé de 4 molécules organiques de base, les nucléotides. Les 4 nucléotides formant l'ADN sont l'adénine (A), la cytosine (C), la guanine (G) et la thymine (T). L'agencement de ces quatre nucléotides permet de coder l'information contenue dans l'ADN.
Source : `https://www.geneticsdigest.com/how-many-chromosomes-do-humans-have/` (traduit en Français).

Dans un premier temps, les nouvelles découvertes furent la compréhension de la séquence ADN avec ces 4 constituants : les nucléotides. Il existe 4 types de nucléotides, l'adénine (A), la thymine (T), la cytosine (C) et la guanine (G). L'ensemble des molécules ADN (matériel génétique) codant l'information nécessaire au fonctionnement de la cellule se nomme le génome. Généralement celui-ci est localisé dans le noyau de la cellule.

Il est à noter que les molécules d'ADN sont généralement condensées sous forme de chromosomes. Ces propos sont illustrés à l'aide de la figure 3.1

Les secrets de la structure des gènes ainsi que les mécanismes biologiques utilisés par les cellules pour décrypter l'ADN en protéines furent découverts chez des organismes simples, tels que les bactéries et les levures avant d'être décelés chez l'être humain. Ainsi un gène fut défini comme une séquence de nucléotide qui code une protéine. C'est donc l'agencement des quatre nucléotides qui permet de coder l'information contenue dans l'ADN. Il est possible d'imaginer ces propos en comparant ce phénomène avec le fonctionnement binaire des ordinateurs actuels. Les ordinateurs traduisent l'information à partir d'un langage binaire 0 et 1 et c'est l'agencement de diverses combinaisons de 0 et de 1 qui définit ce que l'ordinateur doit exécuter. Autrement dit, un gène est une séquence d'ADN qui conditionne la transmission et l'expression d'un caractère héréditaire déterminé d'un individu. Il peut avoir pour fonction de déterminer une ou plusieurs caractéristiques d'un individu.

Pendant de nombreuses années, la lenteur des avancées technologiques fut un frein à de nouvelles découvertes. A titre d'exemple, ce n'est qu'en 2003 que la première version du génome humain a été décrypté après un effort international et plusieurs milliards de dollars. Ces résultats ont permis de révolutionner la recherche en génétique humaine. Nous savons maintenant qu'il y a 3.2 milliards de nucléotides répartis en 23 paires de chromosomes dans le génome humain. Parmi ces milliards de nucléotides, seulement 2 % représentent les gènes.

Cette tendance s'est inversée depuis quelques années. En effet la quantité d'informations disponible augmente de manière exponentielle. Ceci est dû en partie à la baisse des

coûts lié aux nouvelles technologies. Par exemple il coûte aujourd'hui moins de 1500 dollars US pour séquencer un génome humain. Il faut savoir que plus de 99.9 % de l'information génétique d'un génome est identique entre individus. La plupart des études en génétique portent donc sur l'information qui diffère entre les individus. Les études ne s'intéressent alors qu'à une petite partie du génome. Considérant les quantités colossales de données disponibles, cela représente quand même 3 à 4 millions de nucléotides qui varient entre individus. Ainsi la tâche d'analyse est de plus en plus complexe et nécessite des nouveaux outils technologiques et statistiques. Dès lors, le problème ne vient plus de l'acquisition de l'information génétique, mais des outils informatiques qui ne sont pas suffisamment performants pour analyser les données. La capacité d'analyse de grandes bases de données génétiques constitue donc un enjeu majeur pour effectuer de nouvelles découvertes.

Fort des connaissances acquises, il est important de définir les notions de **phénotype** et de **génotype**. Le phénotype d'un individu est l'ensemble des traits observables de celui-ci, qu'ils soient physiques ou mentaux, tels que sa couleur de cheveux, son quotient intellectuel, son poids, etc. Le génotype d'un individu correspond à l'ensemble de ses caractères génétiques. Un même caractère est présent en deux copies par individu, chacun est transmis par l'un des deux parents. Ces différentes versions sont appelées allèles.

Les relations entre les caractères génétiques et phénotypiques peuvent être simple à définir, par exemple lorsqu'il n'y a qu'un facteur génétique qui influence le phénotype. Ceci est le cas pour la mucoviscidose qui est le phénotype associé aux mutations du gène *CFTR*, et la Myopathie de Duchene qui est le phénotype associé aux mutations du gène *DMD*. Néanmoins, pour les phénotypes plus complexes, c'est la combinaison de plusieurs facteurs génétiques qui sont à identifier.

Enfin, l'environnement est un acteur à ne pas sous estimer. Celui-ci joue un rôle important dans l'expression du génotype en phénotype chez les individus. Son effet est par contre difficile à évaluer.

3.1.1 Modification de l'ADN

Le génome peut être altéré pendant la phase de développement pré-natal mais aussi après. Dans certains cas, une modification de l'ADN peut engendrer des problèmes comme des cancers mais aussi des maladies dites génétique comme la trisomie 21. Le syndrome de la trisomie 21 est causé par une copie supplémentaire du chromosome 21. La plupart du temps, ces modifications de l'ADN sont plus localisées et n'impactent qu'un ou une dizaine de gènes. Les raisons des altérations génétiques que l'on appellera mutations, peuvent être de causes naturelles ou environnementales.

Les mutations naturelles peuvent se produire pendant les phases de réparation ou de réplication du génome lorsque les cellules se divisent.

Les mutations génétiques liées à l'environnement sont causées par des raisons extérieures tels que l'exposition à des rayonnement (radioactifs, des rayons X, les UV) mais aussi à certain produits chimiques (amiante, la fumée de cigarette (à cause du benzopyrène qu'elles contiennent), etc...).

Dans l'univers de la science-fiction, les mutations génétiques sont souvent présentées comme les éléments déclencheurs permettant d'accroître les capacités cognitives ou physiques des individus. Elles permettraient par exemple d'avoir accès à des « superpouvoirs ». Dans les faits, les mutations génétiques n'ont souvent aucun impact direct sur l'organisme (elles sont neutres). Et lorsque celles-ci ont un impact, celui-ci est généralement négatif. Il est toutefois possible dans de très rares cas, qu'une mutation génétique soit avantageuse ; de telles mutations obéissent en général à la théorie de l'évolution.

Il existe de nombreux types de mutations génétiques. Dans le cadre de notre application, nous en étudierons deux. Elles portent sur les variations de la structure du génome :

— **Les Duplications** : Une duplication est une variation du nombre de copies d'une séquence du génome. Autrement dit, au lieu d'avoir deux copies d'une même séquence dans le génome (une copie du père et une copie de la mère) comme c'est le cas normalement, l'individu aura 3 copies ou plus.

— **Les Délétions** : Une délétion est une variation du nombre de copies d'une séquence

du génome. Autrement dit, au lieu d'avoir deux copies d'une même séquence dans le génome (une copie du père et une copie de la mère) comme c'est le cas normalement, l'individu aura 1 ou aucune copie.

Dans les deux cas, on parle de **CNV** (*"copy number variant"*). Il est possible d'illustrer ce concept à l'aide de la métaphore suivante. Supposons que l'ADN représente les pages d'un livre. Une délétion correspond à l'acte d'enlever une lettre, un mot ou même une page et une duplication consiste à copier à l'identique une lettre, un mot ou même une page. Dans les deux cas, cela peut créer de la confusion pour comprendre le propos du livre. Il se peut également que ces deux phénomènes n'entrainent pas de problème de compréhension si la partie arrachée ou la partie copiée ne sont pas très importantes dans le livre.

Généralement, il est admis par la communauté scientifique que l'impact des délétions est plus négatif que celle des duplications, d'autant plus lorsque celui-ci touche des gènes qui ont des scores d'intolérance aux mutations élevés (Huguet et al., 2018).

Il est nécessaire de comprendre à quoi correspond le score d'intolérance pour la compréhension de l'application de ce mémoire.

Un gène est dit intolérant aux mutations génétiques si, lorsqu'il est muté, son porteur a une faible probabilité de transmettre son patrimoine génétique à la génération suivante. Ceci peut être causé par le fait que la mutation impacte la survie de l'individu ou qu'elle entraine une incapacité à la reproduction (individu stérile, individu avec une déficience musculaire etc . . .).

L'intolérance aux mutations génétiques de chaque gène est évalué à partir de scores d'intolérance aux mutations délétères (mutation endommageant la protéine). Les étapes du calcul des scores d'intolérance de chaque gène sont les suivantes :

1. Les différentes mutations considérées comme délétères (mutation endommageant la protéine) de chaque gène sont recensées dans les génomes d'une population générale.

2. La moyenne des différentes mutations considérées comme délétères de chaque gène

présent dans le génome est ensuite calculée. Notons que cette moyenne globale est ajustée en fonction de variables biologiques (taille des gènes, localisation dans le génome etc...).

3. Le score d'intolérance de chaque gène est défini en comparant le nombre des différentes mutations observées pour le gène évalué à la moyenne globale calculée. Si le gène d'intérêt possède moins de mutations, son score d'intolérance sera élevé. Dans le cas contraire son score d'intolérance sera faible. Le postulat fait lors de la création des scores consiste à dire que s'il on observe peu de mutations différentes pour un gène, c'est parce que la présence d'une mutation entraine la mort de l'individu ou une incapacité à se reproduire. Dès lors les mutations de ce gène ne se transmettent pas de génération en génération. Ce gène est donc intolérant aux mutations génétiques.

Les travaux réalisés par l'équipe du Dr Jacquemont ont montré qu'il était possible d'étudier les impacts des duplications et délétions du génome à partir de **scores d'intolérance aux mutations génétiques** sur des traits cognitifs comme le quotient intellectuel.

Si on reprend la métaphore précédente sur le génome, un score d'intolérance aux mutations permet de déterminer la probabilité que l'ensemble des pages altérées impacte la compréhension du livre. De plus il permet d'aller plus loin dans la compréhension des impacts. Par exemple, il est possible que l'impact d'une seule page manquante soit trop faible pour ne plus comprendre le récit du livre, en revanche l'accumulation de plusieurs pages manquantes (modèle additif) peut avoir un effet. Dans cette métaphore, nous avons utilisé l'exemple de page manquante (délétion). Il convient de stipuler que l'idée est similaire dans le cas où il s'agit de page dupliquée (duplication).

Enfin, il est important de préciser qu'il existe de nombreux type de scores d'intolérance aux mutations dans la littérature. Les scores diffèrent selon les points suivants :

— Ils dépendent des méthodologies de calcul et des variables biologiques prises en compte.

— Ils dépendent des cohortes d'individus utilisées en tant que référence pour faire ces

calculs.

Nous ne présenterons pas le détail des différents scores d'intolérance aux mutations provenant de la littérature, car les connaissances nécessaires en génétique dépassent le cadre de ce mémoire. Pour plus de détails veuillez consulter les travaux de Huguet et al. (2018).

3.1.2 Problématiques du laboratoire

Ce mémoire a été effectué en collaboration avec le laboratoire du Docteur Jacquemont au centre de recherche du Centre Hospitalier Universitaire de Saint-Justine. Les projets de recherches effectués au laboratoire portent sur les troubles neuro-psychiatriques d'origine génétique. Le laboratoire s'intéresse particulièrement aux CNV et à leur impact sur le quotient intellectuel. Les mutations génétiques ont été étudiées dans des travaux antérieurs (Huguet et al. (2018)) sur des individus provenant de deux populations différentes :

— 2090 adolescents européens provenant du projet *« IMAGEN »*.

— 1983 enfants et parents provenant de l'étude *« Saguenay Youth Study »*.

Les CNV ont été étudiés par l'entremise des scores présentés à la section précédente.

Le lien entre les mutations génétiques (CNV) et le Quotient intellectuel (QI) a ensuite été déterminé à l'aide d'un modèle de régression linéaire. Notons que plusieurs scores d'intolérance génétiques ont été testés et que lien fut statistiquement démontré pour plusieurs d'entre eux. Le score « *PLi pour les délétion* » fut le score d'intolérance génétique ayant obtenu la plus faible valeur-p parmi tous les autres scores testés.

Nous pensons qu'il est peu probable que toutes les relations entre les scores d'intolérance aux mutations génétiques étudiées et le QI soient linéaires. Ainsi, il est possible que le lien entre certains scores et l'intelligence générale ne soient pas correctement modélisés par les modèles linéaires.

L'objectif de notre application est donc d'utiliser l'algorithme des forêts aléatoires pour déterminer l'importance de différents scores d'intolérance aux mutations génétiques pour prédire une mesure d'intelligence générale. Les mesures d'importance calculées à

partir de forêts aléatoires permettront de classer les scores génétiques. Ce classement pourra permettre d'étendre l'analyse effectuée à l'aide de la régression linéaire. Ces analyses seront effectuées à l'aide d'un nouveau jeu de donnée : **Generation of Scotland**.

3.2 Données

Generation of Scotland est le jeu de données utilisé pour notre analyse. Après le contrôle qualité, ce jeu de données comporte 12,743 adultes volontaires provenant d'Écosse. Il est important de préciser que ces individus n'ont pas été recrutés suite à des troubles neuro-développementaux mais de façon non sélective, ils sont ainsi représentatifs de la population générale écossaise.

Nous présenterons dans un premier temps la mesure d'intelligence générale mesurée chez les individus de cette cohorte, puis nous verrons les différentes variables explicatives utilisées.

3.2.1 G facteur

Dans notre application, nous utilisons le facteur G (Spearman, 1904) comme mesure d'intelligence générale. Nous utilisons cette mesure d'intelligence générale comme alternative au quotient intellectuel, non mesuré dans le jeu de données. Le facteur G est défini à partir de la première composante principale de plusieurs tests de cognition présents dans le jeu de données. Ces tests portent sur le vocabulaire, le langage, la logique et la mémoire. Les 4 tests utilisés sont les suivants :

— **Test d'association de symboles à des chiffres** (DigitSymbole) :

  Le but de ce test de cognition est de remplir dans un temps prédéterminé une grille comprenant des chiffres (de 1 à 9) en associant chaque chiffre à un symbole déterminé en début d'expérience. Il y a 133 associations maximum.

— **Test de fluidité verbale** (VerbalFluency) :

Le but de ce test de cognition est de dire le plus de mots possibles en une minute à partir de thèmes définis par l'examinateur.

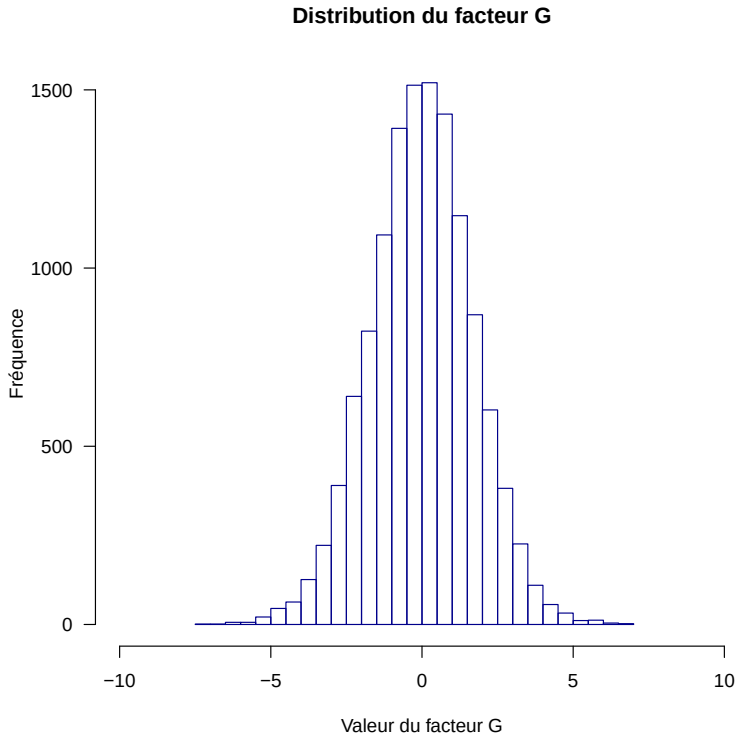

FIGURE 3.2 – Distribution du facteur G des 12743 individus étudiés. Les valeurs minimales et maximales du facteur G sont respectivement de -7.3 et de 6.6.

— **Test de mémoire logique immédiate et avec délai** (LogicalMem) :

Le but de ce test de cognition est de répéter le plus de détails possibles d'une histoire racontée par l'évaluateur. Chaque mot clef est noté, il y en a jusqu'à 25. L'évaluation est effectuée en deux étapes, une première évaluation est effectuée juste après la lecture et une autre évaluation est effectuée après un délai de 30 minutes ou d'une heure après la lecture.

— **Test de vocabulaire (Vocabulary)** :

Le test Mill Hill consiste à choisir les 2 mots de vocabulaire les plus semblables parmi une liste de cinq mots. Ce procédé est effectué 44 fois.

Tous les tests de cognitions ont préalablement été standardisés, c'est pourquoi la moyenne du facteur G est de 0 comme le montre la figure 3.2. Le choix des tests de cognition utilisés a été inspiré par d'autres études utilisant le jeu de données **Génération of Scotland** (Clarke et al., 2016), (Marioni et al., 2014a), (Marioni et al., 2014b). Il est important de préciser qu'il existe certaines discordances sur les tests de cognition utilisés parmi les articles présentés. Le facteur G est une bonne approximation du QI, il est couramment utilisé dans la littérature et des études montrent qu'il est corrélé entre 45% et 50% avec le QI (Clarke et al., 2016), (Marioni et al., 2014a), (Marioni et al., 2014b).

Pour conclure sur cette mesure, il est important de relativiser la définition d'intelligence générale. L'intelligence est un concept relatif à la culture et au libre arbitre de chaque individu. De plus il faut savoir que le quotient intellectuel et le facteur G sont des mesures fortement corrélées au niveau scolaire et au statut socio-économique de chaque individu et de leur famille, ces mesures sont en ce sens critiquables. L'utilisation de ce type de mesure d'intelligence dans le cadre de notre étude se justifie par le fait qu'en général les troubles neuraux développementaux ont un impact négatif sur ce type de mesures.

3.2.2 Variables explicatives

Dans cette sous-section, nous présentons les différents types de variables explicatives utilisées dans notre analyse. Comme précisé dans la section précédente, le facteur G dépend en partie de l'environnement. C'est pourquoi nous incluons aussi dans notre modèle des variables environnementales.

Dans un premier temps, nous présenterons les variables génétiques utilisées puis nous nous intéresserons aux variables environnementales et phénotypiques.

Variables génétiques

Comme mentionné lors de la section 3.1.2, nous étudions des scores génétiques permettant d'évaluer l'impact de CNV sur la survie et la capacité à se reproduire des indivi-

dus. L'idée de la conception de ces scores est présentée à la section 3.1.1. La notation des variables scores est représentée de la manière suivante :

— La première partie du nom de la variable correspond au type de CNV (DEL pour les délétion et DUP pour les duplications). Lorsqu'il y a la présence de "nom_Gene--_complete" après le type de CNV, cela signifie simplement que le score génétique a été calculé uniquement à partir de CNV provenant de région d'ADN comprenant des gènes complets.

— La deuxième partie correspond au nom du score génétique calculé.

Par exemple la variable *"DEL.nom_Gene_complete_pLI"* correspond au score *"pLI"* calculé à partir de la sommes des pLI des gènes complets présent dans les délétions de l'individu. Nous avons également introduit la taille des CNV dans notre analyse. Le système de notation est le même à la différence près que le nom du score est remplacé par *"SIZE"*.

Variables environnementales et phénotypiques

Les variables environnementales et phénotypiques utilisées dans le cadre de notre étude sont les suivantes :

— L'âge des individus.

— Le sexe des individus.

— La nationalité des individus.

— Le rang associé au classement de la zone géographique dans laquelle l'individu testé habite. Les zones géographiques sont classées des plus pauvres au plus aisées. Ces zones proviennent de "*The Scottish Index of Multiple Deprivation*" (SIMD) calculé en 2009. Le SIMD est déterminé à partir du revenu, du taux de chômage, de l'éducation, des soins de santé disponibles, du taux de criminalité et du prix de l'immobilier.

3.3 Résultats de l'application

3.3.1 Méthodologie

Suite aux expérimentations effectuées dans le chapitre 2, nous avons comparé uniquement des variables scores peu corrélées entre elles. Les relations de corrélation entre les différentes variables explicatives sont illustrées dans l'annexe E.

Dans cette application, nous ne présentons pas les résultats de toutes les méthodes de VIM testées au cours de notre étude de simulation effectuée au chapitre 2. Nous avons sélectionné 2 méthodes de VIM parmi les 8 méthodes de VIM qui se sont révélées les plus fiables dans notre étude de simulation selon notre définition de qualité de méthode de VIM présenté à la section 1.3. A ce titre, nous avons appliqué la VIM_CART_Non_Standardisée et la VIM_CART_Standardisée.

Nous avons par la suite séparé le jeu de données en deux sections :

— 10,000 individus ont été utilisés pour entraîner le modèle (ensemble d'entraînement (78%)).

— 2,743 individus ont été utilisés pour tester le modèle (ensemble de tests (22%)).

Nous avons effectué le même processus que pour l'étude des simulations présentée au chapitre 2.

Nous avons donc dans un premier temps déterminés les hyper-paramètres permettant d'optimiser la prédiction de la forêt aléatoire à partir de l'ensemble d'entraînement à l'aide du processus d'optimisation Bayésien. Nous avons fait varier les hyper paramètres suivants :

— Le nombre de variables candidates sélectionnées aléatoirement lors de la séparation de chaque nœud.

— La taille minimale des nœuds terminaux de chaque arbre.

— Le processus d'échantillonnage (utilisation du processus de boostrap ou utilisation des $\frac{2}{3}$ des observations sans remplacement).

Dans un second temps, nous avons calculé les deux méthodes de VIM utilisées dans notre application sur l'ensemble d'entraînement à partir des hyper paramètres optimisés. Nous avons réutilisé les hyper paramètres optimisés à partir de l'ensemble d'entraînement pour construire une forêt aléatoire sur l'ensemble test. Nous avons par la suite déterminé l'importance de chaque variable sur l'ensemble test.

3.3.2 Résultats

Plusieurs points ressortent de notre analyse dont les résultats sont présentés à la figure 3.3 pour la méthode VIM CART_Non_Standardisée. Tout d'abord on constate que le rang associé au classement de la zone géographique dans laquelle les individus testés habitent et l'âge des individus sont les variables les plus importantes parmi celles testées pour prédire le facteur G. Ce résultat montre que l'environnement a un impact non négligeable dans la prédiction du facteur G. Ce phénomène est en total adéquation avec les études antérieures effectuées par le laboratoire.

De plus, Il est intéressant de constater que certains scores ont une importance très similaire dans la prédiction du facteur G. On constate par exemple à l'aide de la figure 3.3 que 10 variables ont une importance relative entre 1.3% et 3%. Ce phénomène induit des légers changements d'ordre d'importance des variables entre l'ensemble d'entraînement et l'ensemble test. Le code couleur mis en place permet cependant d'illustrer une certaine concordance entre l'importance relative mesurée dans les deux jeux de données. Cette observation a été possible en associant une couleur à chaque variable en fonction de son importance relative dans l'ensemble d'entraînement. La couleur associée à chaque variables est fixe et ne change pas, même si l'importance relative change dans l'ensemble test.

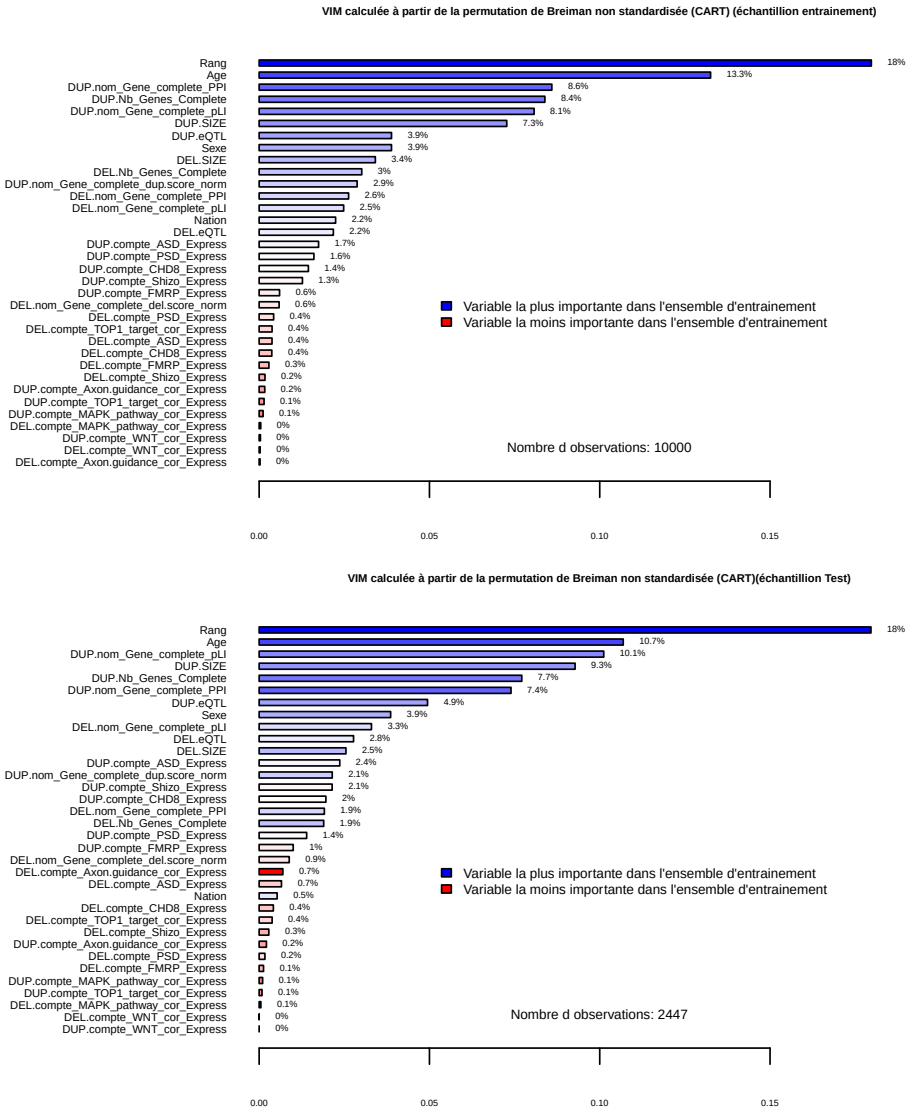

FIGURE 3.3 – Histogramme des importances des différentes variables utilisées dans l'application en génétique obtenues à l'aide la méthode VIM CART_Non_Standardisée. Le graphique du haut représente les résultats des observations dans l'échantillon d'entraînement et le graphique du bas représente les résultats des observations dans l'échantillon de test. Une couleur a été associée à chaque variable en fonction de l'ordre d'importance définie par le modèle dans l'échantillon d'entraînement. Les variables les plus importantes ont une couleur bleu foncé tandis que les variables les moins pertinentes ont une couleur rouge. L'ajout de ces couleurs est utile pour la comparaison de classement entre l'ensemble d'entraînement et l'ensemble de test.

Le spectre des couleurs varie du bleu au rouge, le bleu étant associé aux variables obtenant les importances relatives les plus élevées dans l'ensemble d'entrainement. On

constate que les couleurs associées aux variables importantes dans l'ensemble test ont une couleur proche du bleu ce qui est concordant avec les résultats observés dans l'ensemble d'entrainement. La figure 3.4 vient renforcer ce constat. En effet, l'illustration montre que parmi les 10 variables considérées comme les plus importantes par la VIM_CART_Non-_Standardisée, 9 sont communes entre la méthode calculée à partir de l'ensemble d'entraînement et celle calculée à partir de l'ensemble test.

Nous observons également que le score "*PLi pour les délétion*" ne fait pas partie des variables les plus importantes déterminées par les deux méthodes de VIM. Pour rappel, le score "*PLi pour les délétion*" est le score génétique permettant d'obtenir le meilleur modèle linéaire dans l'article de (Huguet et al., 2018). Ces résultats sont intéressants car nous constatons que les méthodes de VIM utilisés accordent plus d'importance aux scores associés aux duplications qu'aux scores associés aux délétions. Ce constat est vrai pour les deux méthode de VIM appliqués (ce phénomène est observable pour la VIM_CART_Standardisée dans l'annexe F). Or, dans la littérature, il est admis que les délétions ont plus d'effet sur l'intelligence générale que les duplications. Plusieurs hypothèse peuvent expliquer nos résultats :

— Il est possible que le nombre élevé (4084) de duplications comparativement au nombre de délétions (1765) ait un impact sur les mesures d'importances de variables. Illustrons ce phénomène à l'aide d'un exemple en supposant qu'un événement, modélisé par une variable nommée $V1$, soit très important pour prédire le G facteur et qu'un autre événement modélisé par une variable nommée $V2$ soit moyennement important pour prédire le facteur G. On suppose que l'événement important est présent pour 10 individus et l'événement moyennement important est présent pour 100 individus. Dans le cas ou les événements ne sont pas présent, la valeur 0 est attribué aux variables les modélisant. La forêt aléatoire est construite de façon à minimiser l'erreur moyenne des prédictions. Si l'événement modélisé par $V1$ n'est pas suffisamment important, il est possible que la variable $V2$, modélisant l'événement moyennement important plus souvent présente, soit d'avantage utile

pour améliorer la prédiction générale du modèle.

— Il est possible que les duplications aient plus d'impact que les délétions sur les fonctions verbales de l'intelligence générale. En effet, les tests de cognition entrant dans la composition du facteur G valorisent d'avantage les capacités verbales des individus comparativement aux tests utilisés pour le calcul du quotient intellectuel qui font l'objet de l'étude de Huguet et al. (2018).

Même s'il existe quelques différences entre la VIM_CART_Non_Standardisée et la VIM CART_Standardisée, les résultats obtenus par les deux méthodes de VIM sont assez similaires. La figure 3.4 montre que parmi les 10 variables considérées comme les plus importante par la VIM_CART_Non_Standardisée et par la VIM_CART_Standardisée, 7 sont communes entre les deux méthodes.

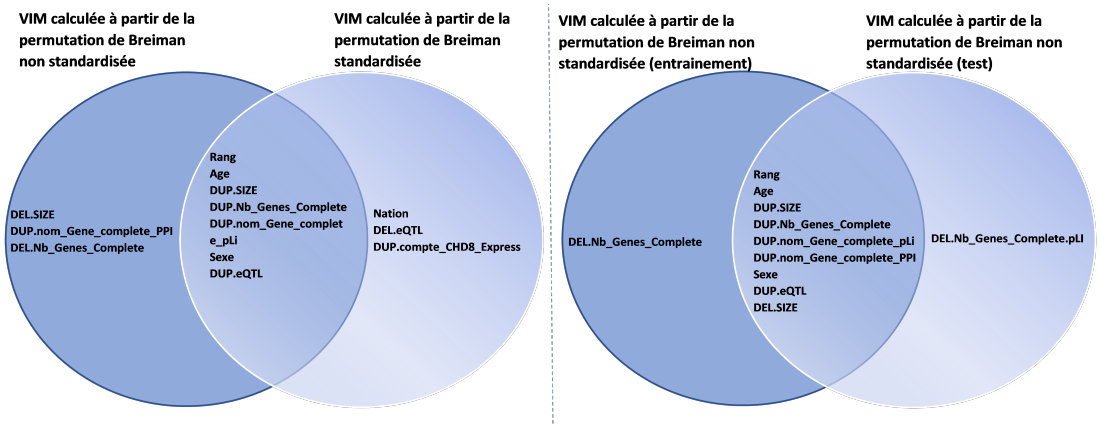

FIGURE 3.4 – Recoupement des 10 variables utilisées dans l'application en génétique les plus importantes en fonction de la méthode de VIM utilisée et de l'ensemble de données utilisées. Le diagramme de Venne à gauche illustre l'ensemble des 10 variables les plus importante pour la VIM_CART_Non_Standardisée et pour la VIM_CART_Standardisée. Le diagramme de Venne à droite illustre l'ensemble des 10 variables les plus importante pour la VIM_CART_Non_Standardisée sur l'ensemble d'entraînement et sur l'ensemble test.

Pour conclure, cette application a permis d'élargir le domaine d'analyse du laboratoire dans la compréhension des troubles neuro-développementaux. Il est cependant important

de garder un esprit critique par rapport aux résultats obtenus. Mesurer l'importance des variables à l'aide de forêts aléatoires ne permet pas de quantifier l'effet de chaque variable explicative sur la variable dépendante. Ces méthodes permettent de classer l'importance des variables utilisées pour la prédiction d'une variable d'intérêt. Lorsque les relations entre les variables explicatives et la variable dépendante sont strictement linéaires, il est important de considérer ces méthodes comme des outils complémentaires et non comme une alternative aux analyses effectuées par des modèles linéaires. Les relations entre les variables explicatives et la variable dépendante ne sont pas linéaires dans le jeu de données étudié, il est donc pertinent d'appliquer les méthodes de VIM pour effectuer une analyse. La prochaine étape de notre analyse consistera à valider les observations effectuées sur la cohorte génération Scotland aux deux jeux de données actuellement utilisés par le laboratoire et présentés à la section 3.1.2.

Conclusion

Au cours de ce mémoire, nous avons étudié les avantages et les limites de 8 différentes méthodes de mesure d'importance de variable (VIM) obtenues à partir de forêts aléatoires. Nous avons évalué les 8 méthodes en fonction de leur proximité avec une définition commune et intuitive d'importance de variable que nous avons calculé de manière théorique. Ainsi, d'après notre définition, une variable qui a beaucoup d'importance est une variable qui, si elle n'était pas utilisée pour prédire la variable cible, engendrerait une augmentation de l'erreur de prédiction et cela de manière proportionnelle à son importance. Notre définition ne prend pas en compte le rôle de certaines variables dans le processus de construction de l'arbre. Nous avons par la suite appliqué les deux meilleures méthodes de VIM parmi les 8 analysées à un cas réel en génétique.

Notre étude de simulation montre que les méthodes de VIM peuvent être appliquées en toute confiance lorsqu'il y a un nombre élevé de variables (explicatives et non explicatives) et que les relations entre les variables explicatives et la variable d'intérêt sont linéaires. En revanche, nous observons que lorsqu'il y a une augmentation du nombre de relations non linéaires entre les variables explicatives et la variable d'intérêt, la qualité de prédiction des 8 méthodes de VIM se détériore. D'après nos résultats, nous déconseillons l'utilisation de la méthode VIM_Ishwaran_Noeud_Opposé et la méthode VIM_Ishwaran--_Aléatoire dans ce type de situation.
De plus, notre travail montre que les méthodes de VIM calculées à partir de forêts aléatoires composées d'arbres d'inférence conditionnelle sont meilleures que les méthodes de VIM calculées à partir de forêts aléatoires composées d'arbres CART lorsqu'il y a de

la corrélation entre deux variables explicatives peu importantes. Ces observations avaient déjà été observées par Strobl et al. (2007). Ce constat n'est cependant plus vrai lorsqu'il existe de la corrélation entre deux variables explicatives importantes. La solution proposée par Strobl et al. (2008) qui consiste à utiliser la méthode VIM_CTREE_Corrélation s'est montrée fiable pour déterminer l'ordre d'importance des variables explicatives lors de la présence de corrélation entre deux variables importantes. En revanche, la variance associée à l'importance de chaque variable au cours des 500 itérations de ré-échantillonnage s'est révélée très importante par rapport aux autres méthodes. Lorsqu'il existe de la corrélation entre plus d'une variable explicative, toutes les méthodes se sont révélées biaisées. Au vu de ces résultats, nous déconseillons d'utiliser les méthodes de VIM lorsque plusieurs variables explicatives sont corrélées.
Les modèles provenant des jeux de données utilisés dans la littérature par Friedman (1991) et Breiman (1996) se sont révélés trop complexes pour étudier des caractéristiques supplémentaires pour les 8 méthodes de VIM. Cela est certainement dû aux relations d'interactions entre les variables présentes dans ces jeux de données. Le procédé proposé dans ce mémoire qui permet de comparer les méthodes de VIM entre elles n'est pas encore adapté pour analyser ce type de relation, de futurs travaux en ce sens sont à considérer.
Les résultats de ce mémoire sont valables lorsque toutes les variables explicatives sont de type continu. Ils sont donc valables dans le cadre de notre application en génétique. De futurs travaux sont nécessaires pour généraliser ces résultats lorsque les variables explicatives sont catégorielles ou binaires.

Le but de l'application en génétique consistait à classer différentes variables génétiques (scores d'intolérance aux mutations génétiques) afin de prédire une mesure d'intelligence générale (le facteur G). Pour répondre à cette tâche nous avons appliqué la méthode VIM_CART_Non_Standardisée et la méthode VIM_CART_Standardisée qui se sont révélées les deux méthodes les plus fiables pour évaluer l'importance de chaque variable parmi les 8 méthodes de VIM étudiées dans notre scénario de simulation.
Les deux méthodes de VIM ont permis de montrer que le rang associé au classement de la

zone géographique dans laquelle les individus testés habitent et l'âge utilisé pour l'ajustement des modèles sont les variables les plus importantes dans la prédiction du facteur G. Ce résultat montre que l'environnement a un impact non négligeable sur la mesures "d'intelligence" générale. D'après notre analyse, il est difficile de définir le score d'intolérance aux mutations génétiques le plus pertinent pour prédire le facteur G. En effet, même s'il existe des similitudes entre les résultats des deux méthodes de VIM appliquées, l'ordre d'importance des variables n'est pas le même pour les deux méthodes. On constate cependant que les scores d'intolérance calculés à partir des duplications ont des importances relatives plus importantes que les scores d'intolérance calculés à partir des délétions. Il est possible que ce résultat soit imputable au nombre de duplications plus important dans le jeu de données que le nombre de délétions. Des analyses supplémentaires prenant en compte un déséquilibre de quantité d'information dans les jeux de données seraient nécessaires afin de tirer une conclusion fiable. Il est également important de préciser que les scores génétiques utilisés expliquent très peu le facteur G. A titre d'exemple, la régression linéaire simple composée du score génétique *"DEL.nom_Gene_complete_pLI"* ayant pour but d'expliquer le facteur G possède un R^2 inférieur à 1%. Nous pensons que se phénomène, qui est attendu en génétique des traits complexes, est susceptible d'impacter les méthodes de VIM. Il serait donc pertinent dans de futures études d'évaluer le comportement des VIM dans le contexte où les variables explicatives expliquent très peu la variable d'intérêt.

Dans le cadre de notre application, nous avons entraîné la forêt aléatoire à partir d'un ensemble d'entraînement et d'un ensemble test. Les résultats se sont révélés concordants entre les deux ensembles de données. L'utilisation de ce procédé reste cependant critiquable. D'après nos connaissances, aucune étude ne traite de manière spécifique cet aspect des méthodes de VIM. Nous pensons que ce type de choix est inévitable lorsque l'on essaie d'effectuer de l'inférence statistique à l'aide d'algorithme exposé au surapprentissage.

En définitive, nous pensons que ce travail à permis de présenter le potentiel d'action des

méthodes de VIM obtenues à partir des forêts aléatoires. Ces méthodes nous permettent par exemple de répondre à d'autres problématiques du laboratoire. Nous sommes en cours de développement d'un outil permettant d'accélérer la phase de contrôle de qualité des CNV détectés (voir annexe A). À l'heure actuelle, l'algorithme des forêts aléatoires est le meilleur candidat pour prédire si les CNV détectés sont corrects. Mesurer l'importance de chaque variable explicative permet d'analyser quelles variables utilisés dans le contrôle qualité sont les plus importantes. A terme cet atout permettra certainement d'améliorer le contrôle qualité des CNV.

Bibliographie

Altmann, A., L. Toloşi, O. Sander, and T. Lengauer

2010. Permutation importance : a corrected feature importance measure. *Bioinformatics*, 26(10) :1340–1347.

Auret, L. and C. Aldrich

2011. Empirical comparison of tree ensemble variable importance measures. *Chemometrics and Intelligent Laboratory Systems*, 105(2) :157–170.

Bergstra, J. S., R. Bardenet, Y. Bengio, and B. Kégl

2011. Algorithms for hyper-parameter optimization. In *Advances in neural information processing systems*, Pp. 2546–2554.

Biau, G. and E. Scornet

2016. A random forest guided tour. *Test*, 25(2) :197–227.

Boulesteix, A.-L., A. Bender, J. Lorenzo Bermejo, and C. Strobl

2011. Random forest gini importance favours snps with large minor allele frequency : impact, sources and recommendations. *Briefings in Bioinformatics*, 13(3) :292–304.

Breiman, L.

1996. Bagging predictors. *Machine learning*, 24(2) :123–140.

Breiman, L.

2001. Random forests. *Machine learning*, 45(1) :5–32.

Breiman, L., J. Friedman, R. Olshen, and C. Stone

1984. *Classification and regression trees*.

Brochu, E., V. M. Cora, and N. De Freitas

2010. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv :1012.2599*.

Chaudhuri, P., M.-C. Huang, W.-Y. Loh, and R. Yao

1994. Piecewise-polynomial regression trees. *Statistica Sinica*, Pp. 143–167.

Clarke, T., M. Lupton, A. Fernandez-Pujals, J. Starr, G. Davies, S. Cox, A. Pattie, D. Liewald, L. Hall, D. MacIntyre, et al.

2016. Common polygenic risk for autism spectrum disorder (asd) is associated with cognitive ability in the general population. *Molecular psychiatry*, 21(3) :419.

Cutler, D. R., T. C. Edwards, K. H. Beard, A. Cutler, K. T. Hess, J. Gibson, and J. J. Lawler

2007. Random forests for classification in ecology. *Ecology*, 88(11) :2783–2792.

Díaz-Uriarte, R. and S. A. De Andres

2006. Gene selection and classification of microarray data using random forest. *BMC bioinformatics*, 7(1) :3.

Efron, B. and R. J. Tibshirani

1994. *An introduction to the bootstrap*. CRC press.

Friedman, J. H.

1991. Multivariate adaptive regression splines. *The annals of statistics*, Pp. 1–67.

Friedman, J. H., T. Hastie, and R. Tibshirani

2001. *The elements of statistical learning*, volume 1. Springer series in statistics New York.

Genuer, R., J.-M. Poggi, and C. Tuleau-Malot

2010. Variable selection using random forests. *Pattern Recognition Letters*, 31(14) :2225–2236.

Genuer, R., J.-M. Poggi, and C. Tuleau-Malot

2015. Vsurf : an r package for variable selection using random forests. *The R Journal*, 7(2) :19–33.

Good, P.

2013. *Permutation tests : a practical guide to resampling methods for testing hypotheses*. Springer Science Business Media.

Gregorutti, B., B. Michel, and P. Saint-Pierre

2017. Correlation and variable importance in random forests. *Statistics and Computing*, 27(3) :659–678.

Hothorn, T., K. Hornik, and A. Zeileis

2006a. Party : A laboratory for recursive part (y) itioning. r package version 0.9-11.

Hothorn, T., K. Hornik, and A. Zeileis

2006b. Unbiased recursive partitioning : A conditional inference framework. *Journal of Computational and Graphical statistics*, 15(3) :651–674.

Hothorn, T., K. Hornik, and A. Zeileis

2015. ctree : Conditional inference trees. *The Comprehensive R Archive Network*.

Huguet, G., C. Schramm, E. Douard, L. Jiang, A. Labbe, F. Tihy, G. Mathonnet, S. Nizard, E. Lemyre, A. Mathieu, et al.

2018. Measuring and estimating the effect sizes of copy number variants on general intelligence in community-based samples. *JAMA psychiatry*.

Ishwaran, H. et al.

2007. Variable importance in binary regression trees and forests. *Electronic Journal of Statistics*, 1 :519–537.

Ishwaran, H. and U. Kogalur

2014. randomforestsrc : Random forests for survival, regression and classification (rf-src). r package version 1.4.

Ishwaran, H., U. B. Kogalur, E. H. Blackstone, and M. S. Lauer

2008. Random survival forests. *The annals of applied statistics*, Pp. 841–860.

James, G., D. Witten, T. Hastie, and R. Tibshirani

2013. *An introduction to statistical learning*, volume 112. Springer.

Janitza, S., E. Celik, and A.-L. Boulesteix

2016. A computationally fast variable importance test for random forests for high-dimensional data. *Advances in Data Analysis and Classification*, Pp. 1–31.

Kass, G. V.

1980. An exploratory technique for investigating large quantities of categorical data. *Applied statistics*, Pp. 119–127.

Kim, H. and W.-Y. Loh

2001. Classification trees with unbiased multiway splits. *Journal of the American Statistical Association*, 96(454) :589–604.

LaFleur, B. J. and R. A. Greevy

2009. Introduction to permutation and resampling-based hypothesis tests. *Journal of Clinical Child Adolescent Psychology*, 38(2) :286–294.

Larivière, B. and D. Van den Poel

2005. Predicting customer retention and profitability by using random forests and regression forests techniques. *Expert Systems with Applications*, 29(2) :472–484.

Legendre, P. and L. F. Legendre

2012. *Numerical ecology*, volume 24. Elsevier.

Liaw, A. and M. Wiener

2002. Classification and regression by randomforest. *R news*, 2(3) :18–22.

Loh, W.-Y.

2008. Classification and regression tree methods. *Encyclopedia of statistics in quality and reliability*.

Loh, W.-Y.

2009. Improving the precision of classification trees. *The Annals of Applied Statistics*, Pp. 1710–1737.

Loh, W.-Y.

2011. Classification and regression trees. *Wiley Interdisciplinary Reviews : Data Mining and Knowledge Discovery*, 1(1) :14–23.

Loh, W.-Y.

2014. Fifty years of classification and regression trees. *International Statistical Review*, 82(3) :329–348.

Loh, W.-Y. and Y.-S. Shih

1997. Split selection methods for classification trees. *Statistica sinica*, Pp. 815–840.

Loh, W.-Y. and N. Vanichsetakul

1988. Tree-structured classification via generalized discriminant analysis. *Journal of the American Statistical Association*, 83(403) :715–725.

Louppe, G., L. Wehenkel, A. Sutera, and P. Geurts

2013. Understanding variable importances in forests of randomized trees. In *Advances in neural information processing systems*, Pp. 431–439.

Marioni, R. E., G. D. Batty, C. Hayward, S. M. Kerr, A. Campbell, L. J. Hocking, G. Scotland, D. J. Porteous, P. M. Visscher, and I. J. Deary

2014a. Common genetic variants explain the majority of the correlation between height and intelligence : the generation scotland study. *Behavior genetics*, 44(2) :91–96.

Marioni, R. E., G. Davies, C. Hayward, D. Liewald, S. M. Kerr, A. Campbell, M. Luciano, B. H. Smith, S. Padmanabhan, L. J. Hocking, et al.

2014b. Molecular genetic contributions to socioeconomic status and intelligence. *Intelligence*, 44 :26–32.

Meng, Y. A., Y. Yu, L. A. Cupples, L. A. Farrer, and K. L. Lunetta

2009. Performance of random forest when snps are in linkage disequilibrium. *BMC bioinformatics*, 10(1) :78.

Morgan, J. N. and J. A. Sonquist

1963. Problems in the analysis of survey data, and a proposal. *Journal of the American statistical association*, 58(302) :415–434.

Nicodemus, K. K.

2011. Letter to the editor : On the stability and ranking of predictors from random forest variable importance measures. *Briefings in bioinformatics*, 12(4) :369–373.

Nicodemus, K. K. and J. D. Malley

2009. Predictor correlation impacts machine learning algorithms : implications for genomic studies. *Bioinformatics*, 25(15) :1884–1890.

Nicodemus, K. K., J. D. Malley, C. Strobl, and A. Ziegler

2010. The behaviour of random forest permutation-based variable importance measures under predictor correlation. *BMC bioinformatics*, 11(1) :110.

Quinlan, J. R.

1986. Induction of decision trees. *Machine learning*, 1(1) :81–106.

Quinlan, J. R.

1993. C4. 5 : Programming for machine learning. *Morgan Kauffmann*, 38 :48.

Quinlan, J. R.

2014. *C4. 5 : programs for machine learning*. Elsevier.

Rasmussen, C. E.

2004. Gaussian processes in machine learning. In *Advanced lectures on machine learning*, Pp. 63–71. Springer.

Rodenburg, W., A. G. Heidema, J. M. Boer, I. M. Bovee-Oudenhoven, E. J. Feskens, E. C. Mariman, and J. Keijer

2008. A framework to identify physiological responses in microarray-based gene expression studies : selection and interpretation of biologically relevant genes. *Physiological genomics*, 33(1) :78–90.

Shotton, J., A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake

2011. Real-time human pose recognition in parts from single depth images. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, Pp. 1297–1304. Ieee.

Snoek, J., H. Larochelle, and R. P. Adams

2012. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, Pp. 2951–2959.

Spearman, C.

1904. " general intelligence," objectively determined and measured. *The American Journal of Psychology*, 15(2) :201–292.

Strasser, H. and C. Weber

1999. On the asymptotic theory of permutation statistics.

Strobl, C., A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis

2008. Conditional variable importance for random forests. *BMC bioinformatics*, 9(1) :307.

Strobl, C. and A. Zeileis

2008. Danger : High power !–exploring the statistical properties of a test for random forest variable importance.

Strobl, l., A.-L. Boulesteix, A. Zeileis, and T. Hothorn

2007. Bias in random forest variable importance measures : Illustrations, sources and a solution. *BMC bioinformatics*, 8(1) :25.

Svetnik, V., A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan, and B. P. Feuston

2003. Random forest : a classification and regression tool for compound classification and qsar modeling. *Journal of chemical information and computer sciences*, 43(6) :1947–1958.

Therneau, T. M., E. J. Atkinson, et al.

1997. An introduction to recursive partitioning using the rpart routines.

Wright, M. N. and A. Ziegler

2015. ranger : A fast implementation of random forests for high dimensional data in c++ and r. *arXiv preprint arXiv :1508.04409*.

Ziegler, A. and I. R. König

2014. Mining data with random forests : current options for real-world applications. *Wiley Interdisciplinary Reviews : Data Mining and Knowledge Discovery*, 4(1) :55–63.

Annexe

Table des matières

Annexe A – Détection des CNV et contrôle qualité

Cette section est présente à titre informatif uniquement. Elle a pour but de comprendre comment il est possible de détecter la présence de délétion ou de duplication dans le génome. Pour comprendre cette manœuvre, quelques termes supplémentaires sont nécessaires.

Le décryptage du génome humain a permis de recenser des différences entre chaque individu dans la suite de nucléotides (A, T, G ou C) composant l'ADN. Par exemple, pour un site spécifique du génome, certains individus auront un nucléotide T au lieu d'un A. Lorsqu'une variation de nucléotide est observée chez plus de 1% de la population, cette mutation est appelée *"single nucleotide polymorphism"* (SNP).

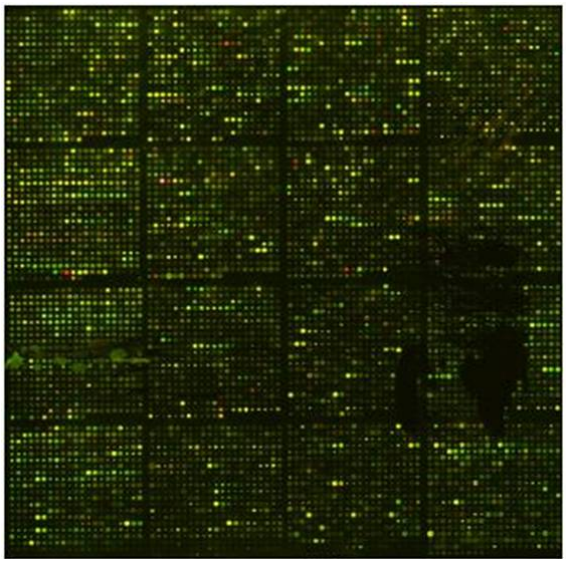

FIGURE 1 – Illustration d'une puce a ADN.
Source : http ://www.edu.upmc.fr/sdv/masselot_05001/polymorphisme/images/pic049.jpg.

Les puces à ADN permettent de connaître les allèles des SNP par individu. Elles sont constituées d'un support rigide en verre sur lequel de courtes séquences d'ADN ont été déposées. Chacune des courtes séquences (sondes) a la particularité de se lier à une région du génome de l'individu testé et cela de façon spécifique. Dans notre cas, elles ciblent les SNP déjà identifiés dans le génome humain.

Les sondes sont marquées via une fluorescence verte ou rouge selon les allèles détectés. Ainsi, les allèles G et C sont marqués en vert et A et T en rouge. Lorsque les capteurs n'ont qu'une couleur c'est qu'ils n'ont fixé qu'un seul type allèle. Dans cette situation l'individu a dans son génome deux fois le même allèle. Dans le cas contraire, le spot apparaîtra jaune et cela indiquera que l'individu possède deux allèles différents.

La fluorescence totale indique la réussite de liaison de la sonde de la séquence du génome testé. Cela permet d'établir la quantité de copie d'allèle présente pour chaque sonde. La fluorescence totale sera plus élevé dans le cas d'un allèle dupliqué que pour un allèle manquant.

Dans la suite de l'annexe nous présentons en détail le processus de détection des mutations génétiques qui se résume en trois étapes :

1. Détection de la présence de différents allèles pour chaque SNP connu.
2. Détection de duplication et de délétion.
3. Détection des CNV.

Détection de la présence de différents allèles pour chaque SNP connu.

Pour acquérir les données issues de la puce pour chaque SNP via les sondes, il faut faire appel à des plateformes de laboratoire spécialisées dans cette procédure. Il faut également un échantillon d'ADN de bonne qualité des personnes que l'on veut étudier. L'ADN peut être extrait du sang ou de la salive. Les 6 principales étapes dans la manipulation des puces sont résumées à l'aide de la figure 2 :

1 Acquisition de l'ADN de l'individu à étudier.

2 L'ADN de l'individu étudié est répliqué pour en avoir suffisamment pour l'expérience, puis il est coupé en morceaux.

3 L'ADN de l'individu est mis en contact avec la puce pour que les sondes se fixent sur les morceaux de celui-ci.

4 Marquage par fluorescence des sondes selon l'allèle de la séquence fixée

5 La puce est scannée pour avoir l'information des fluorescences de chaque sonde.

6 Analyses du scan pour définir le génotype de l'individu via le ratio des fluorescences et la quantité de copies pour chaque SNPs.

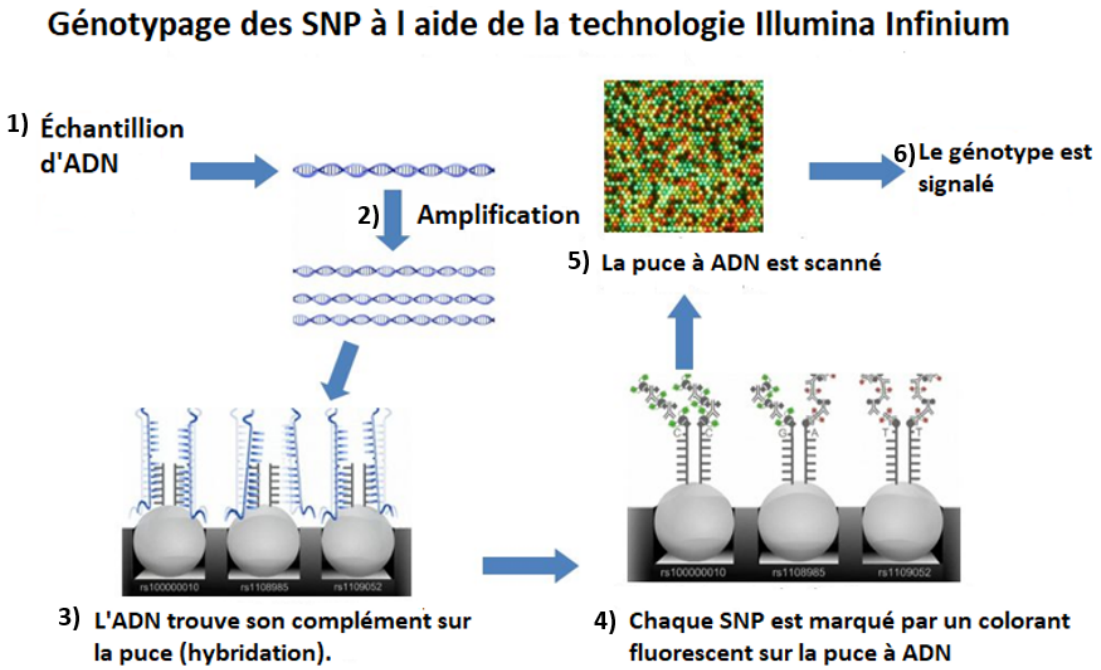

FIGURE 2 – Génotypage des SNP à l'aide de la technologie Illumina Infinium.
Source : http ://slideplayer.com/slide/3373386/ (traduit en francais).

La détection de duplication ou de délétion

À partir du résultat obtenu suite aux étapes décrites ci-dessus, il est possible de détecter la présence de duplication ou de délétion. Pour cela, deux nouveaux concepts complémentaires sont introduits, le *log R Ratio* et la *B allèle frequency* (BAF).

Le *log R ratio* : le log R ratio mesure la luminosité de chaque signal de fluorescence renvoyé par les spots de la puce. Lorsqu'il y a la présence de duplication, le log R ratio du SNP en question est plus élevé que la moyenne des autres spots de la puce, car plus d'ADN a été en contact avec les sondes en question. Inversement, lorsqu'il y a la présence d'une

délétion, le log R ratio est plus bas que la moyenne, car moins d'ADN a été en contact avec les sondes en question.

La *B allèle frequency* (BAF) : la BAF est égal au ratio des fluorescences rouges et vertes. Il s'agit dans ce cas de calculer la fréquence de l'allèle B. L'allèle A et B sont définis au début de la manipulation et la fréquence de A est égale à 1 moins la fréquence de l'allèle B. Pour comprendre correctement son interprétation, il est nécessaire de se rappeler que chaque individu hérite normalement d'un allèle de chacun de ces parents.

Dans cette situation, la BAF peut prendre 3 valeurs pour chaque SNP de la puce ADN :

— BAF=0/2, dans le cas où l'individu a hérité des deux parents de l'allèle A

— BAF=1/2, dans le cas où l'individu à hérité d'un parent l'allèle A et d'un autre parent l'allèle B

— BAF=2/2=1, dans le cas ou l'individu a hérité des deux parents l'allèle B

La BAF est utile pour déterminer les délétions et les duplications. Dans ce cas, quatre autres situations sont possibles :

1. Délétion totale : dans ce cas de figure l'individu ne possède pas d'allèle A et d'allèle B et la BAF n'existe pas, il ne s'agit que de bruit. Il est possible de visualiser ce type de mutation à l'aide de la figure 3.
2. Délétion d'une copie : dans ce cas de figure, l'individu possède 1 allèle au lieu de deux. La BAF peut prendre deux valeurs :
   — BAF=0/1=0 si l'allèle restant se trouve être l'allèle A
   — BAF=1/1=1 si l'allèle restant se trouve être l'allèle B

   Il est possible de visualiser ce type de mutation à l'aide de la figure 4.
3. Duplication d'une copie : dans ce cas de figure, l'individu possède une duplication d'un allèle. La BAF peut prendre 4 valeurs
   — BAF=0 si l'individu a 3 allèles A et 0 allèle B
   — BAF=1/3 si l'individu possède 2 allèles A et 1 allèle B
   — BAF= 2/3 si l'individu possède 1 allèle A et 2 allèles B

— BAF=3/3= 1 si l'individu possède 0 allèle A et 3 allèles B

Il est possible de visualiser ce type de mutation à l'aide de la figure 5

4. Duplication de deux copie : dans ce cas de figure, l'individu possède deux duplications d'un allèle. La BAF peut prendre 4 valeurs :

   — BAF=0 si l'individu a 4 allèles A et 0 allèle B

   — BAF=1/4 si l'individu possède 3 allèles A et 1 allèle B

   — BAF= 2/4=0.5 si l'individu possède 2 allèles A et 2 allèles B

   — BAF= 3/4 si l'individu possède 1 allèle A et 3 allèles B

   — BAF=4/4= 1 si l'individu possède 0 allèle A et 4 allèles B

Détection de CNV

Les CNV peuvent être vus comme l'agrégation de plusieurs SNP (délété ou dupliqué). Il est possible de détecter des CNV après avoir ordonné les SNP selon leur position dans le génome. Pour cela, un modèle de Markov caché est utilisé pour prédire les CNV en fonction des SNP disponibles. Des logiciels tels que PennCNV ou QuantiSNP sont utilisés pour effectuer cette étape. Les éléments nécessaires à la détection des CNV sont les suivants :

— Un modèle de Markov caché entrainé.

— Log R ratio des SNP.

— La BAF des SNP.

— La fréquence de l'allèle B dans la population générale.

— Les coordonnées des SNP dans le génome.

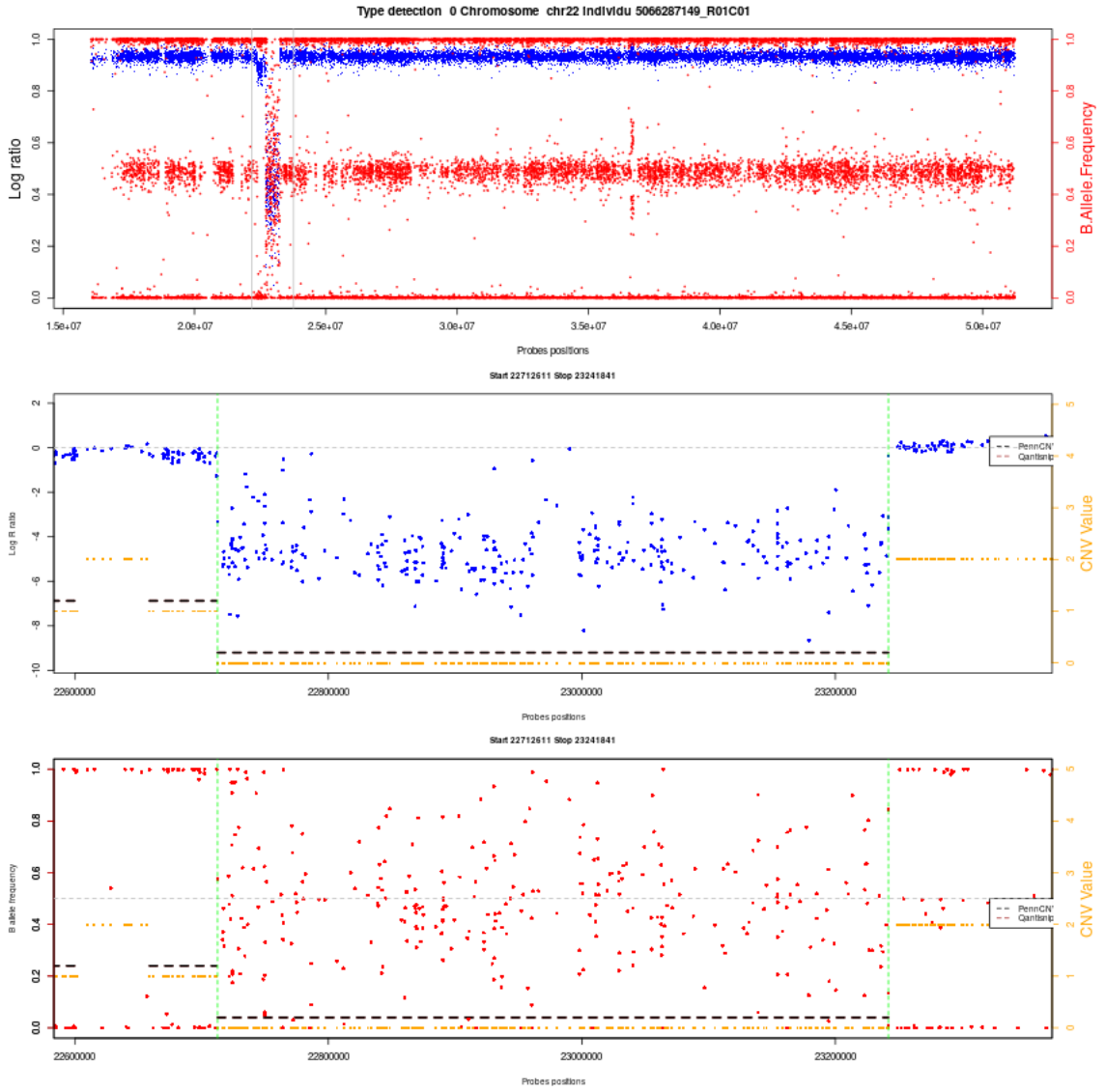

FIGURE 3 – Détection d'une délétion de type 0.

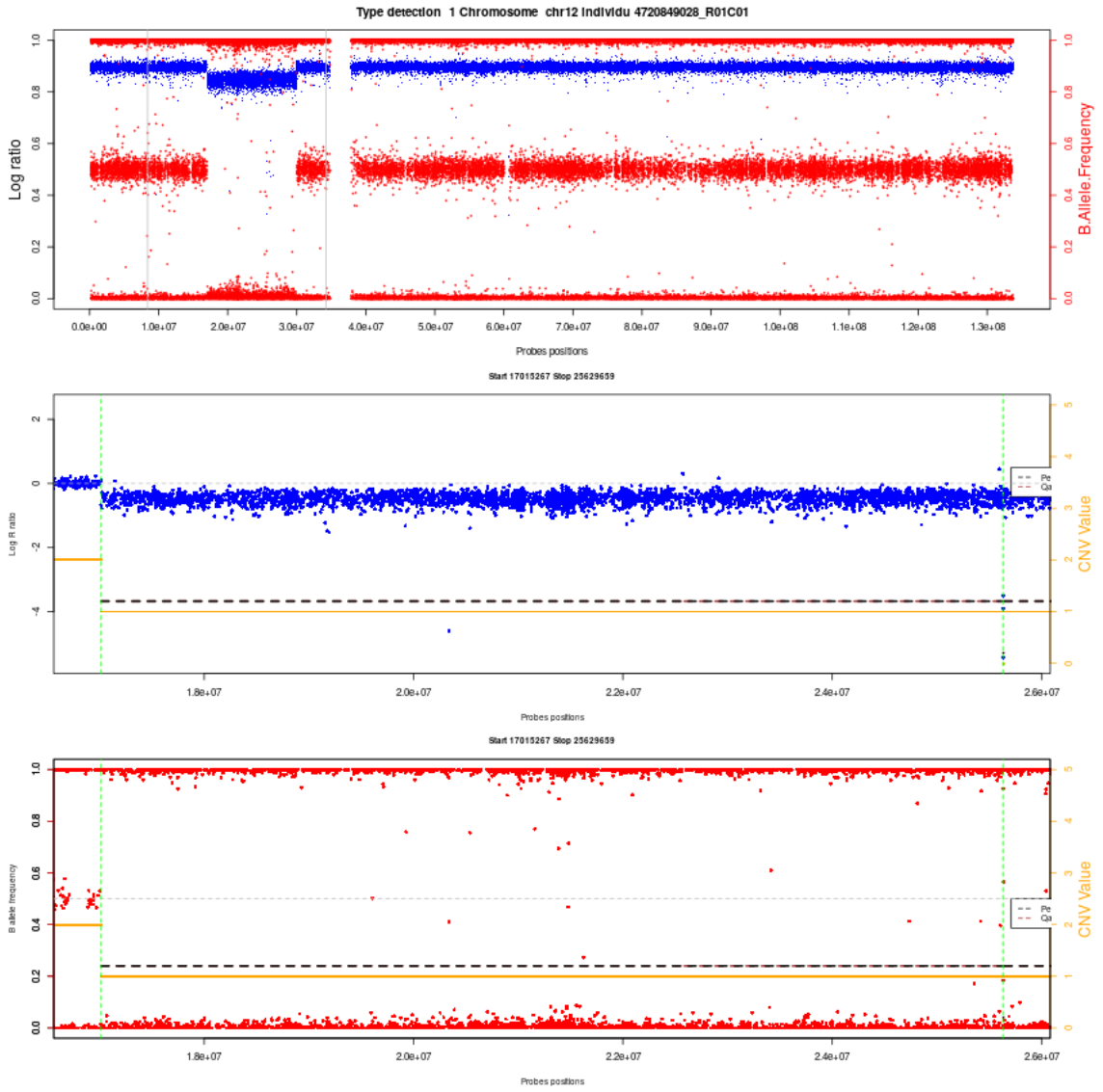

FIGURE 4 – Détection d'une délétion de type 1.

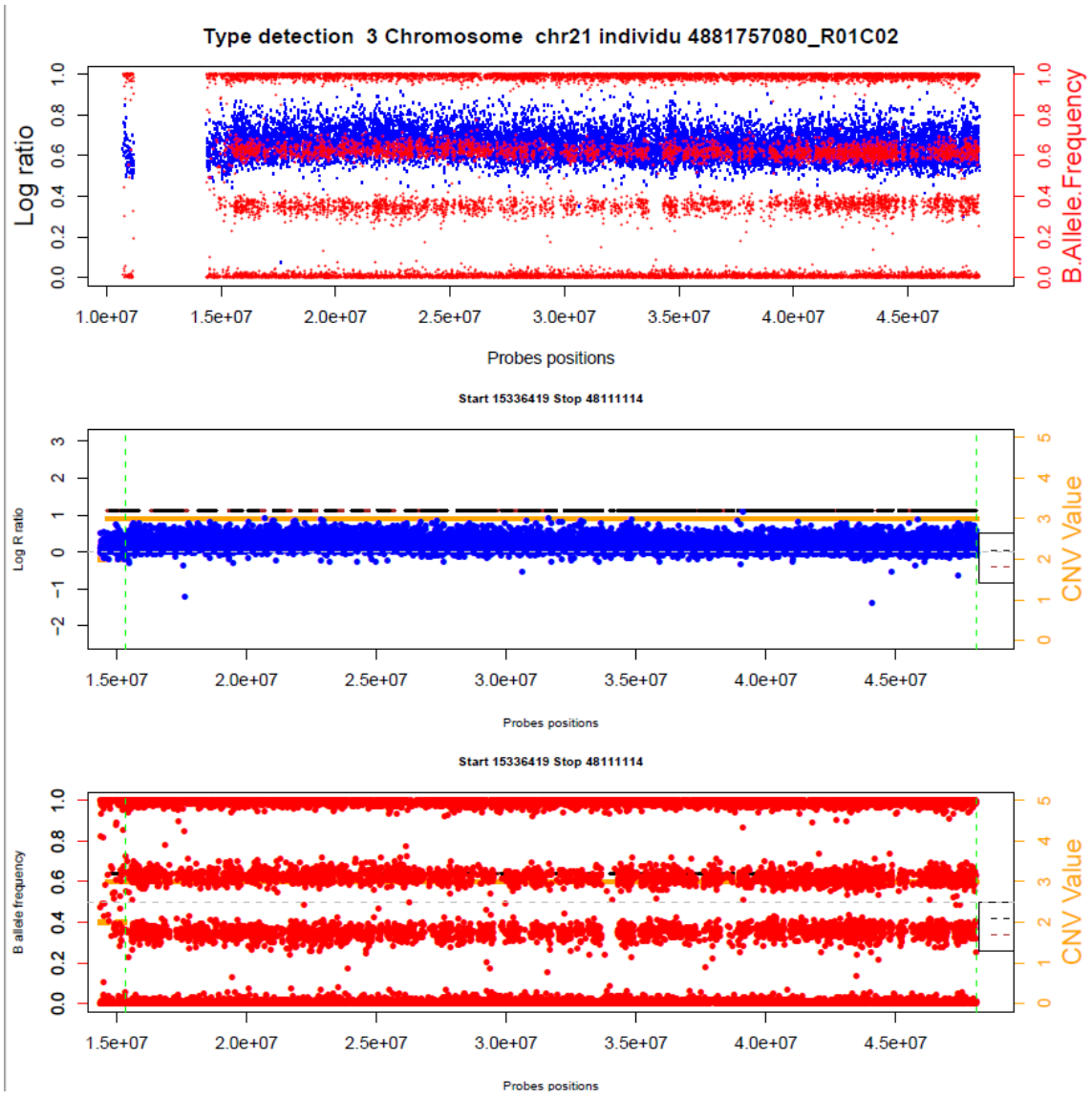

FIGURE 5 – Détection d'une duplication de type 3.

Annexe B – Exemple de codes utilisés dans le cadre de la méthodologie

Nous présentons dans cette annexe un échantillon des codes R et bash utilisés pour effectuer les simulations. Les codes présentés ont été utilisés uniquement pour calculer l'importance des variables dans le contexte du modèle de base. Ainsi, les codes utilisés peuvent varier en fonction du scénario de simulation appliqué.

Création du jeu de données

Ce code R a été utilisé pour la création du jeu de données pour les scénarios consistant à faire varier le nombre de variables non explicatives en conservant les autres paramètres du modèle de base constant.

```
# Fonction pour creer un jeu de donnees
# pour le cas des scenarios de simulation 1.
# Dans ce cas le nombre de variables explicatives
# est de 1 par default.
# Au cours des scenarios de simulations, nous faisons varier,
# la graine aleatoire d'echantillionnage, le nombre de variables non
    explicatives,
# et l'ecart type du bruit associe aux simulations
creation_jeu_donne<-function(i,nb_variable_non_explicative,ecart_type){
  # Implentation de la graine aleatoire
  set.seed(i)
  # Creation des variables explicatives.
  # par default l'ecart type des variables
  # explicatives est egal a 1
  x1<-rnorm(n=1400,mean=0,sd=1)
  jeux_donne<-data.frame(x1=x1)

  for (k in 1:(nb_variable_non_explicative+5)){
    nomvariable<-paste("x",k,sep="")
```

```
19      jeux_donne[nomvariable]<-rnorm(n=1400,mean=0,sd=1)}
20    nom_variable<-variable.names(jeux_donne)
21
22    #Relation entre Y et les variables explicatives.
23    y<-(-1.8*jeux_donne$x1+1.6*(jeux_donne$x2)+1.4*jeux_donne$x3
24          -1.7*sin(jeux_donne$x4)+0.2*exp(jeux_donne$x5))
25
26    #Ajout du bruit. L'ecart_type est defini par l'utilisateur
27    noise<-rnorm(n=1400,mean=0,sd=ecart_type)
28
29    # Ajout du bruit
30    jeux_donne["y_theorique"]<-y
31    jeux_donne["y"]<-y+noise
32
33    #On ordonne le jeu de donnees (pour des raisons de simplication dans
          les plots)
34    ordre_nom<-c("y_theorique","y",nom_variable)
35    jeux_donne<-jeux_donne[,ordre_nom]
36
37    #Separation des donnees:
38    #       entrainenement
39    #       validation
40    #       test
41    echantion_validation_test<-sample.int(1400,400,replace = FALSE)
42
43    ################################
44    #Ensemble d entrainement
45    ################################
46    entrainement<-jeux_donne[-echantion_validation_test,]
47    dim(entrainement)
48
49    ################################
50    #Echantillionnage validation et test
51    ################################
```

```
validation_test<-jeux_donne[echantion_validation_test,]
echantion_validation<-sample.int(400,200,replace = FALSE)

################################
#Ensemble de validation
################################
validation<-validation_test[-echantion_validation,]
dim(validation)

################################
#Ensemble de test
################################
test<-validation_test[echantion_validation,]
dim(test)

#Verification sommaire
print('test de verification doit etre egal a 0')
print(sum(test$x1 %in% validation$x1))
print(sum(entrainement$x1 %in% validation$x1))
print(sum(entrainement$x1 %in% test$x1))

#On enleve y theorique pour entrainement
entrainement2<-entrainement[,-1]

y_train<-entrainement2[,1]
x_train<-entrainement2[,-1]

#On enleve y theorique pour validation
validation1<-validation[,-1]
y_val<-validation1[,1]
x_val<-validation1[,-1]
```

```
86  ########################################
87  #              Calcul importance
88  #              des variables theoriques
89  ########################################
90
91  #On enleve Y
92  entrainement1<-entrainement[,-2]
93  resultat<-data.frame(Y=1)
94  coef<-c(-1.8,1.6,1.4,-1.7,0.2)
95
96  #Pour chaque variable on pose successivement
97  #le coeficient de i = a 0
98  for (j in 1:5){
99    coef1<-coef
100   nomvariable<-paste("x",(j),sep="")
101   entrainementj<-entrainement1
102   coef1[j]<-0
103   y_x<-(coef1[1]*entrainementj$x1+coef1[2]*(entrainementj$x2)+coef1
          [3]*entrainementj$x3
104           +coef1[4]*sin(entrainementj$x4)+coef1[5]*exp(entrainementj$x5
                ))
105   resultat[nomvariable]<-(1/length(entrainement1$y_theorique))*
106     (sum((entrainement1$y_theorique-y_x)**2))}
107
108 # On sait qu'en theorie,
109 #les variables non explicatives ont une importance = a 0.
110 if(nb_variable_non_explicative!=0){
111   for (r in 6:(nb_variable_non_explicative+5)){
112     nomvariable<-paste("x",r,sep="")
113     resultat[nomvariable]<-0}}
114
115 #On calcule ensuite l'importance relative de chaque variable
116 resultat<-resultat[,-1]
117 sum_resultat<-sum(resultat)
```

```
118   resultat_relatif<-resultat/sum_resultat
119
120   return(list(jeux_donne=jeux_donne,entrainement=entrainement2,
121               y_train=y_train,
122               x_train=x_train,
123               x_val=x_val,
124               y_val=y_val,
125               resultat_relatif=resultat_relatif))
126 }
```

donne.R

Calcul des VIM

Ce code R a été effectué pour calculer les VIM obtenues à l'aide des packages `RandomForest` et `Party`.

```
# Pour la fonction les parametres sont :
# "nombre_permutation". (Pour les  m thodes  de VIM basee sur la
    permutation)
# "rf_meileure_param"= meilleures hyperparametres de la RF
#"controle_c.optimal"= meilleures hyperparametres de la RF (ctree)
#"nombre_devariable_non_explicative1"= nombre de variable non
    explicatives
#"lien_export1"= lien dans lequel les resultats seront exportes
#"ecart_type" = ecart type du bruit dans le jeu de donnees utilise.
simulation1<-function(i,nombre_permutation,
                      rf_meileure_param=OPT_RF$Best_Par,
                      controle_c.optimal=controle_c.optimal,
                      nombre_devariable_non_explicative1,
                      lien_export1,
                      ecart_type){

    #Source du jeu de donnees
    source("/home/antostj/scratch/antoine/simulation2/simulation1/donne
        .R")

    # Librairie foret  a l atoire
    library(randomForest)
    library(party)

    #Recuperation du jeu de donnees (cree dans donne.R)
    donne<-creation_jeu_donne(i=i,nb_variable_non_explicative=nombre_
        devariable_non_explicative1,ecart_type=ecart_type)
    x_train<-donne$x_train
    y_train<-donne$y_train
    y_val<-donne$y_val
```

```
27      x_val<-donne$x_val
28      entrainement<-donne$entrainement
29
30      # Meilleurs hyperparametres pour la foret aleatoire.
31      # (avec le package randomForest).
32      mtry_best<-rf_meileure_param[1]
33      replace_best<-rf_meileure_param[2]
34      nodesize_best<-rf_meileure_param[3]
35      ifelse(replace_best==1,replace_best<-TRUE,replace_best<-FALSE)
36      #Calcul de la foret aleatoire optimisee
37      rf_opt<-randomForest(x=x_train,y=y_train,mtry=mtry_best,ntree=500
38                           ,replace=replace_best
39                           ,nodesize=nodesize_best,importance = TRUE,nPerm=
                                 nombre_permutation)
40
41      #Calcul de l'importance des methodes CART_non_standardisee et CART_
            standardisee.
42      importance_rf_NO_SCALE_f<-importance(rf_opt,type=NULL, scale=FALSE)
43      importance_rf_scale_f<-importance(rf_opt,type=NULL, scale=TRUE)
44
45      # Meilleurs hyperparametres pour les forets aleatoires.
46      # (avec le package party).
47      #Calcul de la fort aleatoire optimisee
48      Forest_importance_opt_C=cforest(y~., data=entrainement,controls =
            controle_c.optimal)
49
50      #Calcul de l'importance de la methodes CTREE.
51      importance_conditionnel_f=varimp(Forest_importance_opt_C,
            conditional = FALSE, nperm = nombre_permutation)
52
53      #Creation d'un Data frame permettant de regrouper les VIM des 500
            iterations
54      importance_rf_NO_SCALE_N<-paste("importance_rf_NO_SCALE",i,sep="_")
55      importance_RF_Inpurity_N<-paste("importance_RF_Inpurity",i,sep="_")
```

```
56      importance_rf_scale_N<-paste("importance_rf_scale",i,sep="_")
57      importance_conditionnel_N<-paste("importance_conditionnel",i,sep="_
            ")
58
59      nom_variable<-colnames(x_train)
60
61      importance_rf_NO_SCALE_f1<-data.frame(nom_variable=nom_variable,
            importance_rf_NO_SCALE_f=importance_rf_NO_SCALE_f[,1])
62      importance_RF_Inpurity_f2<-data.frame(nom_variable=nom_variable,
            importance_rf_NO_SCALE_f=importance_rf_NO_SCALE_f[,2])
63      importance_rf_scale_f<-data.frame(nom_variable=nom_variable,
            importance_rf_scale_f=importance_rf_scale_f[,1])
64      importance_conditionnel_f<-data.frame(nom_variable=nom_variable,
            importance_conditionnel_f=importance_conditionnel_f)
65
66      # Decalage effectue pour regler le probleme des importances
            negatives
67      for( k in importance_rf_NO_SCALE_f1$importance_rf_NO_SCALE_f){
68        if(k<0){
69          importance_rf_NO_SCALE_f1$importance_rf_NO_SCALE_f<-importance_
                rf_NO_SCALE_f1$importance_rf_NO_SCALE_f-min(importance_rf_
                NO_SCALE_f1$importance_rf_NO_SCALE_f)
70          print('decalage perm no scale')
71          break}}
72      for( k in importance_RF_Inpurity_f2$importance_rf_NO_SCALE_f){
73        if(k<0){
74          importance_RF_Inpurity_f2$importance_rf_NO_SCALE_f<-importance_
                RF_Inpurity_f2$importance_rf_NO_SCALE_f-min(importance_RF_
                Inpurity_f2$importance_rf_NO_SCALE_f)
75          print('decalage perm inpurity')
76          break}}
77      for( k in importance_rf_scale_f$importance_rf_scale_f){
78        if(k<0){
79          importance_rf_scale_f$importance_rf_scale_f<-importance_rf_
```

```
            scale_f$importance_rf_scale_f-min(importance_rf_scale_f$
            importance_rf_scale_f)
80          print('decalage perm scale')
81          break}}
82      for( k in importance_conditionnel_f$importance_conditionnel_f){
83        if(k<0){
84          importance_conditionnel_f$importance_conditionnel_f<-importance
              _conditionnel_f$importance_conditionnel_f-min(importance_
              conditionnel_f$importance_conditionnel_f)
85          print('decalage perm scale')
86          break}}
87
88      #Calcul des importances relatives
89      importance_rf_NO_SCALE_f1$importance_rf_NO_SCALE_f<-importance_rf_
          NO_SCALE_f1$importance_rf_NO_SCALE_f/sum(importance_rf_NO_SCALE
          _f1$importance_rf_NO_SCALE_f)
90      importance_RF_Inpurity_f2$importance_rf_NO_SCALE_f<-importance_RF_
          Inpurity_f2$importance_rf_NO_SCALE_f/sum(importance_RF_Inpurity
          _f2$importance_rf_NO_SCALE_f)
91      importance_rf_scale_f$importance_rf_scale_f<-importance_rf_scale_f$
          importance_rf_scale_f/sum(importance_rf_scale_f$importance_rf_
          scale_f)
92      importance_conditionnel_f$importance_conditionnel_f<-importance_
          conditionnel_f$importance_conditionnel_f/sum(importance_
          conditionnel_f$importance_conditionnel_f)
93
94      colnames(importance_rf_NO_SCALE_f1)[2]<-importance_rf_NO_SCALE_N
95      colnames(importance_RF_Inpurity_f2)[2]<-importance_RF_Inpurity_N
96      colnames(importance_rf_scale_f)[2]<-importance_rf_scale_N
97      colnames(importance_conditionnel_f)[2]<-importance_conditionnel_N
98
99
100     lien_export<-lien_export1
101     #Ecriture des differentes VIM pour les 500 iterations.
```

```
102     no_scale1<-paste(lien_export,importance_rf_NO_SCALE_N,sep="")
103     no_scale1<-paste(no_scale1,"txt",sep=".")
104     write.table(importance_rf_NO_SCALE_f1,no_scale1,quote = FALSE,sep="
            \t")
105
106     inpurity<-paste(lien_export,importance_RF_Inpurity_N,sep="")
107     inpurity<-paste(inpurity,"txt",sep=".")
108     write.table(importance_RF_Inpurity_f2,inpurity,quote = FALSE,sep="\
            t")
109
110     scale1<-paste(lien_export,importance_rf_scale_N,sep="")
111     scale1<-paste(scale1,"txt",sep=".")
112     write.table(importance_rf_scale_f,scale1,quote = FALSE,sep="\t")
113
114     cond<-paste(lien_export,importance_conditionnel_N,sep="")
115     cond<-paste(cond,"txt",sep=".")
116     write.table(importance_conditionnel_f,cond,quote = FALSE,sep="\t")
117
118     # Ecriture de l'importance theorique
119     theorique<-paste(lien_export)
120     nomfichier<-paste('theorique',i,sep='_')
121     nomfichier<-paste(nomfichier,'txt',sep='.')
122     nomfichier<-paste(theorique,nomfichier,sep='/')
123     importance_theorique<-donne$resultat_relatif
124     write.table(importance_theorique,nomfichier,quote = FALSE,sep="\t")
            }
```

simulation_aggreger.R

Calcul des VIM sur le serveur Calcul Canada

Ces 3 codes ont été effectués pour calculer les simulations à partir du serveur calcul Canada.

```
#!/bin/bash
#Chargement du module R sur le serveur Calcul Canada.
module load r/3.3.3
#Envoi des 500 iterations en lot de 50.
for fichier_debut in 1 51 101 151 201 251 301 351 401 451;
do
  fichier_fin=$(($fichier_debut+49));
  cd /home/antostj/scratch/antoine/simulation2/simulation1/p_20;
  dos2unix simulation.sh;
  chmod +x simulation.sh;
  #Envoi du code bash simulation.sh sur le serveur a l'aide de sbatch.
  #Sbatch permet de lancer les calculs sur le serveur Calcul Canada.
  #sbatch permet d'envoyer des scripts batch a Slurm
  #SLURM (Simple Linux Utility for Resource Management)
  #est une solution open source d'ordonnancement de taches
      informatiques.
  sbatch --export=debut=$fichier_debut,fin=$fichier_fin simulation.sh
      ;
  done
```

envoie.sh

```
#!/bin/bash
#SBATCH --nodes=1          # Nombre de noeud demande sur le serveur.
#SBATCH --ntasks-per-node=26    # Tache par noeud demandee sur le
    serveur.
#SBATCH --mem=95G          # Memoire de calcul demandee sur le serveur.
#SBATCH --time=0-20:00           # Temps demandE sur le serveur de
    calcul CEDAR (DD-HH:MM).

#Variables transmisent par le code bash precedant
```

```
#Ces variables seront transmises au fichier R permettant de calculer les VIM.
fichier_debut=$debut;
fichier_fin=$fin;


#Chargement des modules R et Openmpi
module load r/3.4.0
module load openmpi/1.10.7
export R_LIBS=~/local/R_libs/
#Utilisation de mpirun pour effectuer la parallelisation.
mpirun -np 1 Rscript simulation.R $fichier_debut $fichier_fin --save > "$fichier_debut.test.out" 2>  "$fichier_debut.test.er";
```

simulation.sh

```
# Lance le code creant le jeu de donnees
source("/home/antostj/scratch/antoine/simulation2/simulation1/donne.R")
# Lance le code calculant l'importance des variables. On recupere la fonction simulation1
source("/home/antostj/scratch/antoine/simulation2/simulation1/simulation_aggreger.R")
# Telecharge les hyperparametres optimises calcules a l aide du package RandomForest
load(file="/home/antostj/scratch/antoine/simulation2/simulation1/p_20/model_entrainer/OPT_RF_p20")
# Telecharge les hyperparametres optimises calcules a l aide du package party
load(file="/home/antostj/scratch/antoine/simulation2/simulation1/p_20/model_entrainer/controle_c.optimal_p20")

# Librairie utilisees pour effectuer de la parralelisation
library("Rmpi")
library(doSNOW)

```

```
# Recuperation des parametres fournis par le fichier simulation.sh
args <- commandArgs(TRUE)
print(args)
debut<-as.numeric(args[1])
fin<-as.numeric(args[2])

# Enregistre le nombre de task demandees par le serveur -1
# car il faut une task pour distribuer les ordres aux autres taches.
ns <- mpi.universe.size() - 1
# Enregistrement du cluster
cl<- makeMPIcluster(ns, type="MPI")
registerDoSNOW(cl)
# Parallelisation distribuee sur toutes les (tasks-1) du noeud demande
# Les 500 iterations sont calculees en parallele a l'aide de la
    fonction simulation1
foreach(x = debut:fin,
        .combine = c)  %dopar% simulation1(i=x,
                                           nombre_permutation=1000,
                                           rf_meileure_param=OPT_RF$
                                               Best_Par,
                                           controle_c.optimal=controle_
                                               c.optimal,
                                           nombre_devariable_non_
                                               explicative1=15,
                                           ecart_type=2,
                                           lien_export1="/home/antostj/
                                               scratch/antoine/
                                               simulation2/simulation1/
                                               p_5/resulat_sim1/")

#Ferme le cluster
stopCluster(cl)
mpi.quit()
```

simulation.R

Annexe C – Résultats graphiques complémentaire à l'étude des scénarios de simulations.

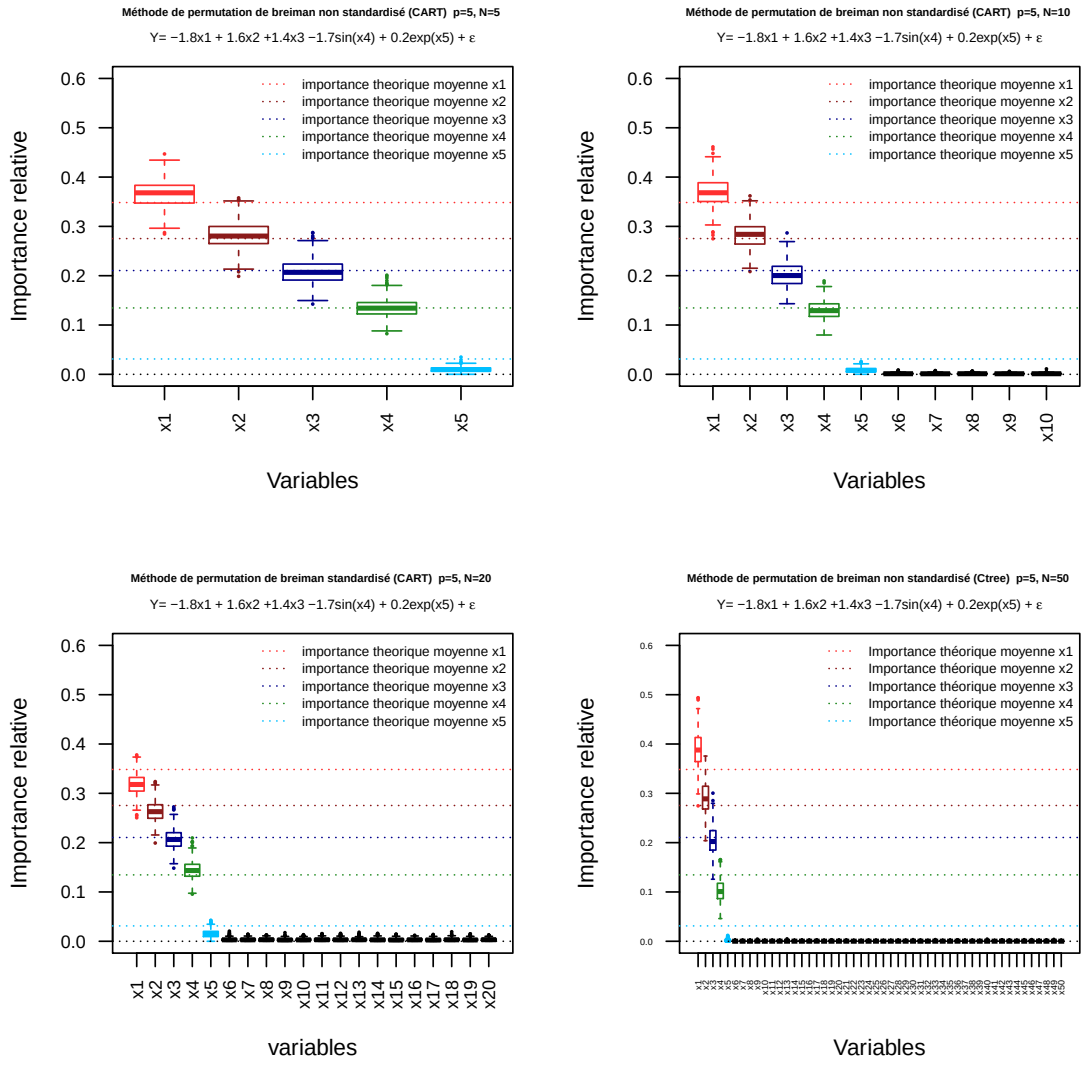

FIGURE 6 – Diagrammes en boite des importances relatives accordées aux variables explicatives pour les méthodes VIM CART_Non_Standardisée et VIM_CART- -_Standardisée (situation 1,2,3 et 4 du scénario de simulation 1). Ce graphique montre la meilleure méthode de VIM pour chaque situation du scénario 1 d'après notre définition de mesure d'importance de variable.

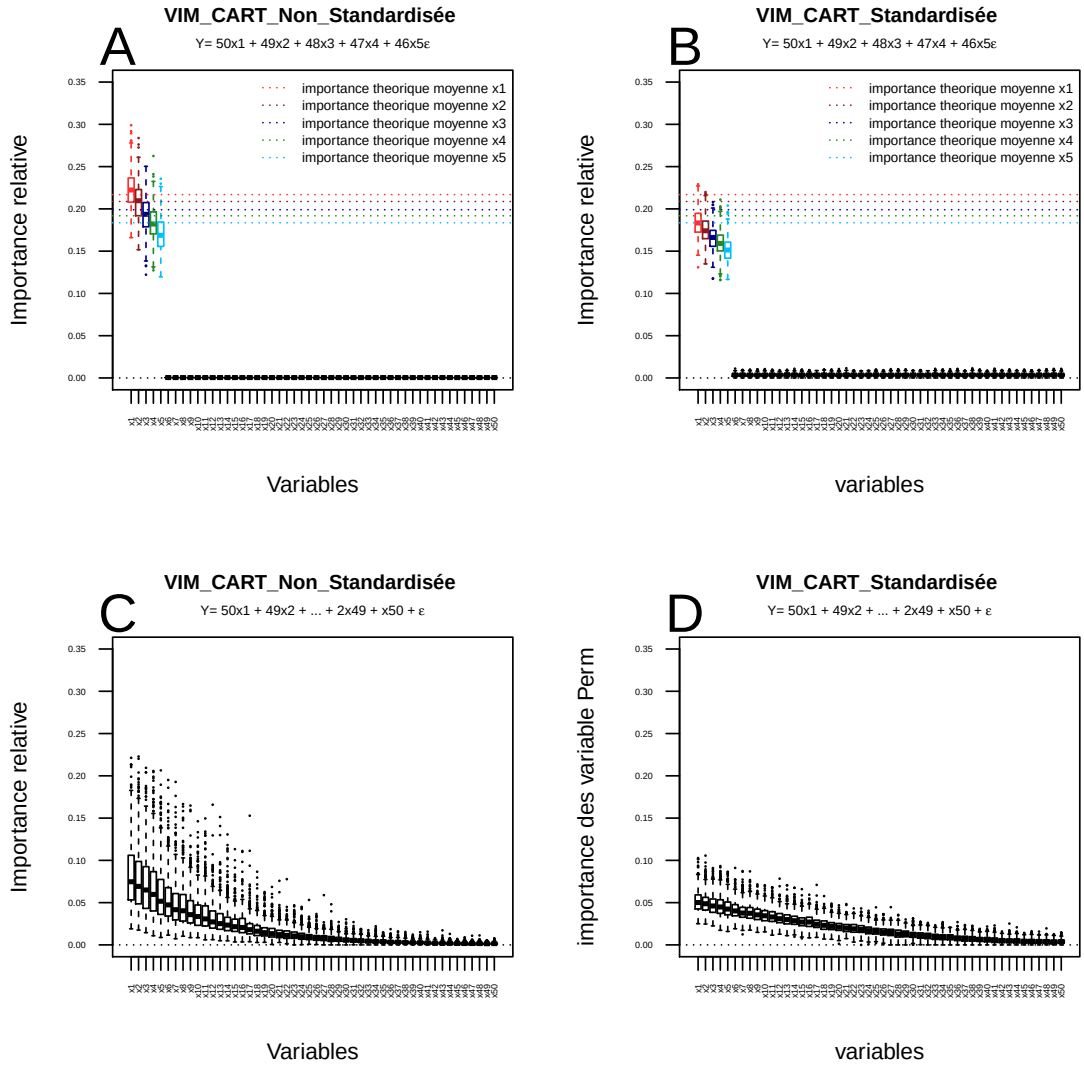

FIGURE 7 – Diagrammes en boite des importances relatives accordées aux variables explicatives pour les méthodes VIM_CART_Non_Standardisée et VIM_CART_Standardisée. Les graphiques A et B représentent réspectivement les résultats de la situation 2 (5 variables explicatives et 45 variables non explicatives) du scénario de simulation 2 pour la méthode VIM_CART_Non_Standardisée et pour la méthode VIM_CART_Standardisée. Les graphiques C et D représentent réspectivement les résultats de la situation 5 (50 variables explicatives) du scénario de simulation 2 pour les méthodes VIM_CART_Non_Standardisée et VIM_CART_Standardisée.
On constate qu'une importance relative moins élevée est accordée aux variables plus importantes dans le cas de la VIM_CART_Standardisée.

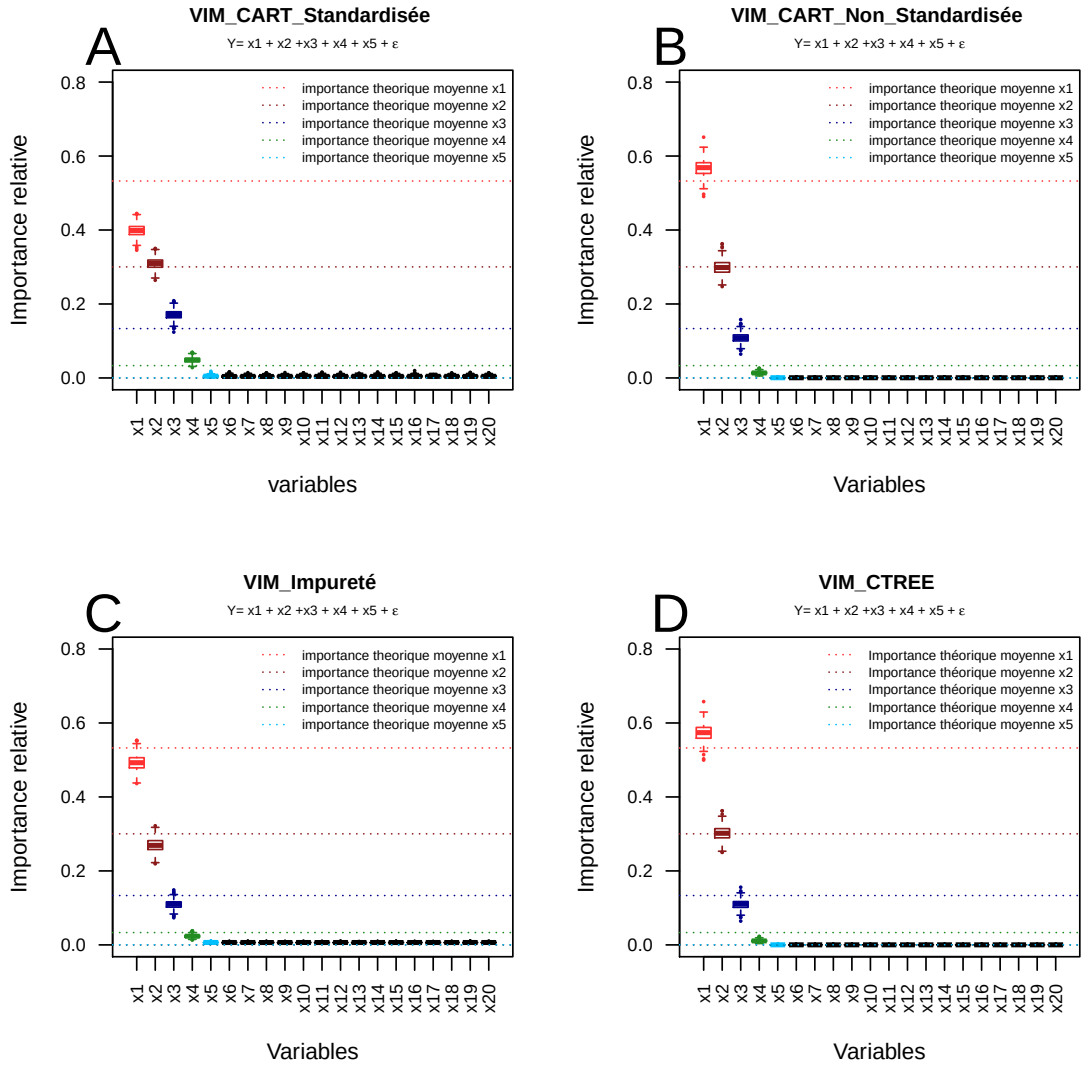

FIGURE 8 – Diagrammes en boite des importances relatives accordées aux variables explicatives (x1-x50) et non explicatives(x6-x20) pour la méthode VIM_CART_Non_Standardisée, la méthode VIM_CART_Standardisée, la méthode VIM_CTREE et la méthode VIM_Impureté.
Dans ce scénario de simulation,$x_1 \sim N(0,4)$,$x_2 \sim N(0,3)$,$x_3 \sim N(0,2)$,$x_4 \sim N(0,1)$ et $x_5 \sim N(0,0.1)$. Les graphiques A, B, C, et D représentent réspectivement les résultats pour la situation 4 du scénario de simulation3 pour la la méthode VIM_CART_Standardisée, la méthode VIM_CART_Non_Standardisée, la méthode VIM_Impureté et la méthode VIM_CTREE. On constate que seule la méthode VIM_CART_Standardisée est biaisée dans cette situation.

Annexe D – Résultats graphiques des meilleurs modèles obtenus avec les jeux de données de Freidman.

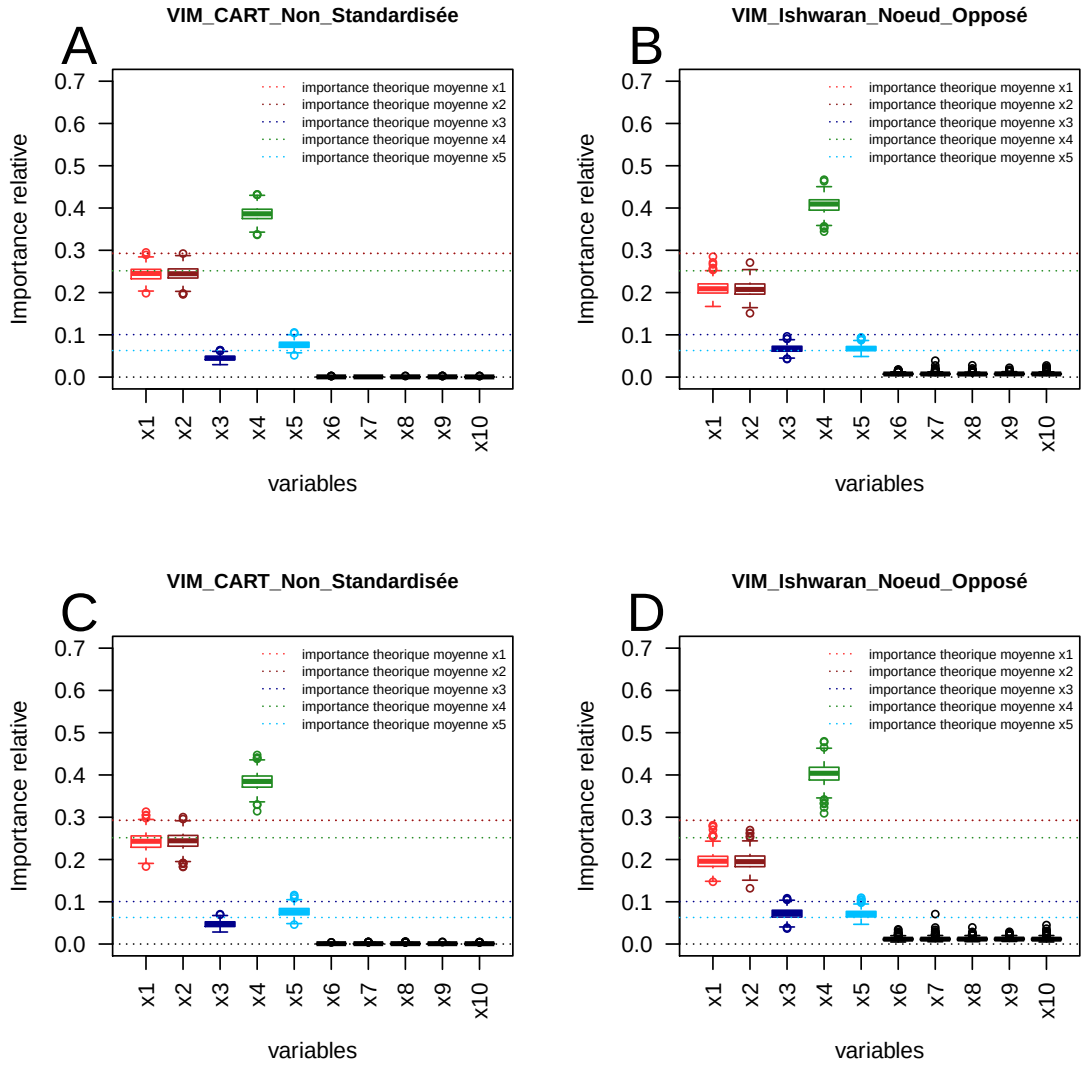

FIGURE 9 – Diagrammes en boite des importances relatives accordées aux variables explicatives (x1-x4) et non explicative (x5-x10) pour les méthodes VIM_CART_Non_Standardisée et VIM_Ishwaran_Noeud_Opposé. Les graphiques A et B représentent les résultats du scénario Freidman1 avec un écart-type de 1 pour la méthode VIM_CART_Non_Standardisée et la méthode VIM_Ishwaran_Noeud_Opposé. Les graphiques C et D représentent les résultats du scénario Freidman1 avec un écart-type de 2 pour la méthode VIM_CART_Non_Standardisée et la méthode VIM_Ishwaran_Noeud_Opposé.

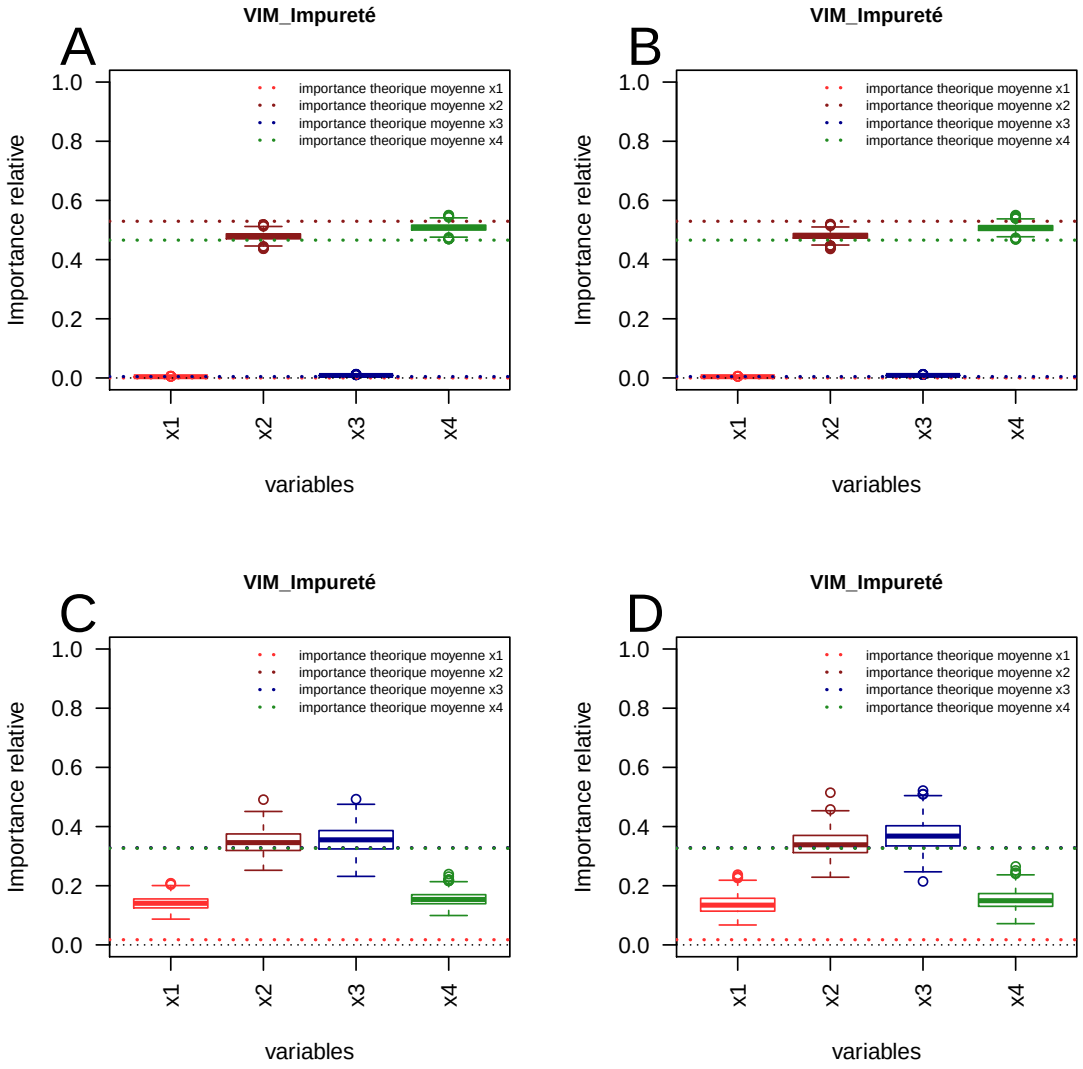

FIGURE 10 – Diagrammes en boite des importances relatives accordées aux variables explicatives (x1-x4) pour la méthode VIM_Impureté. Les graphiques A et B représentent les résultats du scénario Freidman2 avec un écart-type de 125 et 250 pour les méthodes de VIM_Impureté. Les graphiques C et D représentent les résultats du scénario Freidman3 avec un écart-type de 0.1 et 0.2 pour les méthodes VIM_Impureté.

Annexe E – Corrélation entre les variables utilisées dans l'application en génétique.

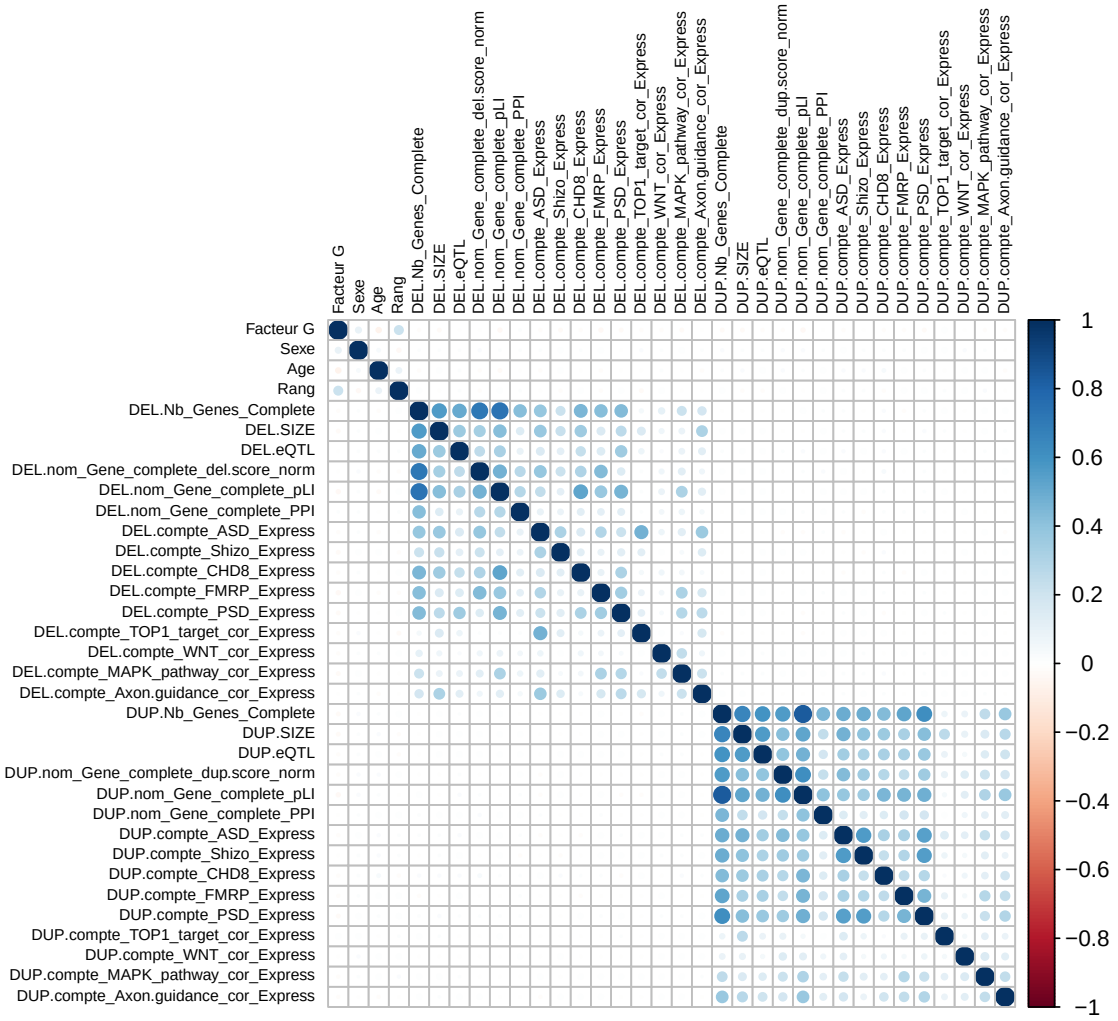

FIGURE 11 – Corrélation entre les différentes variables utilisées dans l'application en génétique.

Annexe F – Résultat supplémentaire de l'application en génétique

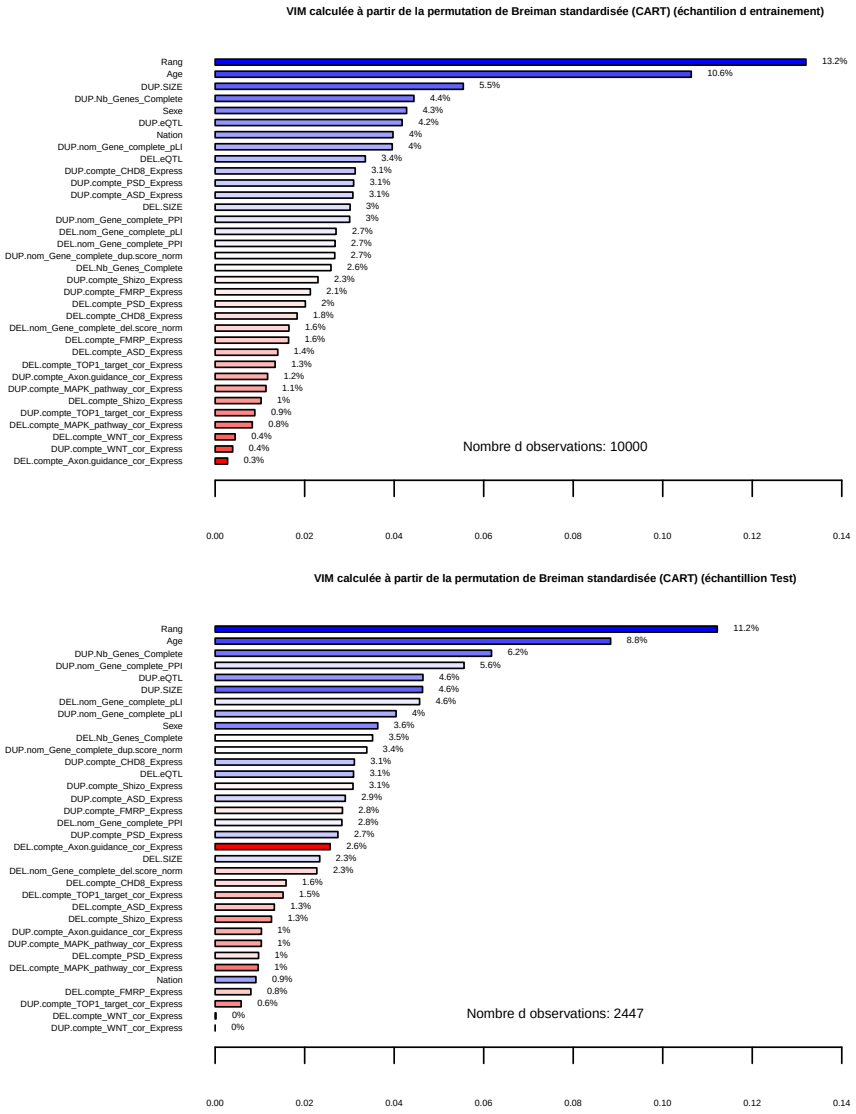

FIGURE 12 – Importance des différentes variables utilisées dans l'application en génétique obtenues à l'aide de la méthode VIM_CART_Standardisée. Le graphique du haut représente les résultats des observations dans l'échantillon d'entraînement et le graphique du bas représente les résultats des observations dans l'échantillon de test. Une couleur a été associée à chaque variable en fonction de l'ordre d'importance définie par le modèle dans l'échantillon d'entraînement. Les variables les plus importantes ont une couleur bleu foncé tandis que les variables les moins pertinentes ont une couleur rouge. L'ajout de ces couleurs est utile pour la comparaison de classement entre l'ensemble d'entraînement et l'ensemble de test.