AgroParisTech

Institut Pasteur

Détection de mutations de pharmaco-résistance dans des sequences génétiques de VIH

Luc Blassel

Mémoire de 3ème année cursus ingénieur.
Dominante IODAA (de l'Information à la Décision par l'Analyse et l'apprentissage)

13 octobre 2021

MÉMOIRE

Présenté par : BLASSEL Luc.....

Dans le cadre de la **dominante d'approfondissement** :
IODAA.....
ou
Dans le cadre du **Master recherche :**
.....
.....

Stage effectué du (jj/mm/aa) : 30/04/2018 au 30/09/2018
À :

Institut Pasteur.....

28 Rue du docteur Roux.....

75015, PARIS.....

Sur le **thème** :
Utilisation de techniques de machine learning pour la détection de mutations de résistance dans les séquences génétiques de VIH.....

.....

Eventuellement : rapport confidentiel : ☐ Date d'expiration de confidentialité : ….. /……/…...

Pour l'obtention du :
DIPLÔME D'INGÉNIEUR AGROPARISTECH

Enseignant/e-tuteur responsable de stage : Christine MARTIN.....

Maître de stage : Olivier GASCUEL .....

Soutenu le (jj/mm/aa) **:** ……/……/……..

Abstract : Drug resistance mutations (DRM) are still a major obstacle in treating HIV infections. A list of surveillance DRMs has been established and is maintained by experts, but there are cases where treatment failure is seen without presence of any SDRMs. The goal of this project is to use machine learning methods to detect new DRMs.

Random forests and Logistic regressions are trained on HIV-1 Reverse transcriptase (RT) sequences from an African dataset (3990 sequences) and a European dataset (55544 sequences). There are several HIV subtypes in each dataset. Sequences are One-hot encoded and the classifiers are trained to differentiate sequences that are treatment-experienced and treatment-naive. Feature importances and feature contributions (Random Forest), as well as feature weights (Logistic regression) are used, subtype-wise, to select a sub-group of important features for discrimination of the two classes. Difference of proportions of mutations corresponding to each selected feature between naive and experienced samples are then tested with a z-score, corrected for multiple testing with the Bonferroni correction.

Classifiers have an accuracy between 55% and 75% depending on subtype, there is signal. A sub-group of 19 mutations has been identified, 12 of which have a significant difference of proportion between the experienced and naive classes in both datasets. One of these mutations : $I31L$ has been identified in at least two other studies of resistance in HIV-1.

Further *ex-silico* studies are necessary to confirm or not the DRM status of the identified mutations. However results are encouraging in regards to this method's validity.

Résumé : Les mutations de résistance (DRM) aux traitements antirétroviraux restent un obstacle majeur dans la thérapie contre le VIH. Des listes de mutations de résistances de surveillance (SDRM) sont établie et maintenues par des experts mais dans certains cas il y a échec thérapeutique sans détecter de SDRMs. En utilisant des technique d'apprentissage automatique nous essayons de détecter de nouvelles DRMs.

Des modèles de forêts aléatoires et de régression logistique ont été entraînés sur des séquences de Reverse Transcriptase (RT) de VIH-1 de plusieurs sous-types issues d'un jeu de données Africain (3990 séquences) et d'un jeu de données Européen (55544 séquences).Les séquences sont encodées avec un encodage One-Hot et les variables correspondant aux positions de DRMs connues sont supprimées. Les classificateurs ont été entraînés sur une tâche de classification binaire supervisée, pour discriminer les séquences traitées des non traitées. Les importances et contributions de variables (forêts aléatoires) et les coefficients (régression logistique) sont observés sous-type par sous-type pour sélectionner un ensemble de variables les plus discriminantes. Les différences de proportions des mutations correspondant aux variables sélectionnées entre les séquences traitées et non traitées sont ensuite testées avec des z-tests corrigés avec une correction de Bonferroni.

Les classificateurs ont une précision globale de classification entre 55% et 75% en fonction des sous-types, indiquant l'existence d'un signal dans les données. Un sous-ensemble de 19 mutations a été identifié. Parmi celles-ci 12 ont une différence significative de proportion entre les séquences traitées et non traitées pour les deux jeux de données. L'une de ces mutations : $I31L$ a été identifiés dans au moins deux autre études de phénomènes de résistance du VIH-1.

D'autre études *ex silico* seront nécessaires pour confirmer le statut de DRM ou non des mutations identifiées, mais les résultats obtenus sont encourageant quant à la validité de la méthode.

Remerciements

Je tiens à remercier toutes les personnes qui ont contribué à la réalisation de ce projet.

Dans un premier temps je voudrais remercier Olivier Gascuel pour son encadrement, ses conseils et sa supervision.

Dans un second temps, Anna Zhukova pour ses conseils, son aide et son temps consacré à la correction de ce mémoire,

Ma responsable de stage, Christine Martin pour son encadrement.

L'équipe pédagogique d'AgroParisTech et plus particulièrement de la dominante IODAA pour leur enseignements et leurs conseils,

Et finalement tous les membres de l'équipe de bio-informatique évolutive du C3BI de l'institut Pasteur, pour leur accueil chaleureux.

Table des matières

Remerciements **i**

Table des figures **v**

Liste des tableaux **vi**

Liste des Abréviations **vii**

Introduction **1**

1 Contexte et prérequis **2**
1.1 Quelques notions de génétique . . . 2
1.1.1 Qu'est ce que l'ADN ? . . . 2
1.1.2 L'expression de l'information génétique . . . 3
1.1.2.1 La Transcription . . . 3
1.1.2.2 La Traduction . . . 3
1.1.3 Les protéines . . . 3
1.2 Le VIH . . . 4
1.2.1 Présentation du VIH . . . 4
1.2.2 Génome du VIH . . . 4
1.2.3 Le cycle du VIH . . . 4
1.3 Les mutations de résistance . . . 5
1.3.1 Comment apparaissent les mutations ? . . . 5
1.3.2 Les mutations de surveillance . . . 5
1.3.3 Nommage des mutations . . . 6
1.4 État de l'art . . . 6
1.4.1 Détection de DRMs . . . 6
1.4.2 Problèmes Statistiques . . . 6

2 Préparation des données **7**
2.1 Données utilisées . . . 7
2.1.1 Données Africaines . . . 7
2.1.2 Données Européennes . . . 7
2.2 Création du jeu de données . . . 8
2.2.1 Traduction et alignement . . . 8
2.3 Encodage des séquences . . . 8
2.3.1 Simples encodages numériques . . . 9
2.3.1.1 L'étiquetage . . . 9
2.3.1.2 L'encodage "Bitwise" . . . 9
2.3.1.3 L'encodage "OneHot" . . . 9

2.3.2 Encodages physico-chimiques . . . 9
2.3.2.1 Encodage basé sur AAIndex . . . 10
2.3.2.2 Encodage d'appartenance de groupes . . . 10

3 Méthodes employées 11
3.1 Classification des données . . . 11
3.1.1 Forêts Aléatoires . . . 12
3.1.1.1 Choix de l'algorithme . . . 12
3.1.1.2 Importance des variables . . . 12
3.1.2 Regression logistique . . . 13
3.2 Contribution des variables . . . 13
3.2.1 Calcul des contributions . . . 13
3.2.2 Contributions jointes . . . 14
3.3 Réduction du jeu de données . . . 14
3.3.1 Identification de DRMs . . . 14
3.3.2 Élimination des DRMs . . . 14
3.4 Tests statistiques . . . 15
3.4.1 Comparaisons de proportions . . . 15
3.4.2 Problèmes de tests multiples . . . 16
3.5 Autres outils . . . 16
3.5.1 Librairies Python . . . 16
3.5.2 Outil de work-flow . . . 16
3.5.3 Utilisation d'un cluster de calcul haute performance . . . 16

4 Résultats 17
4.1 Choix de l'encodage . . . 17
4.1.1 Performance des encodages . . . 17
4.1.2 Interprétabilité . . . 17
4.2 Performance des classificateurs . . . 17
4.2.1 Avec DRMs . . . 17
4.2.2 Sans DRMs . . . 18
4.3 Mutations identifiées . . . 19
4.3.1 Validation de la méthode . . . 19
4.3.1.1 Forêts aléatoires . . . 19
4.3.1.2 Régression logistique . . . 21
4.3.1.3 Contributions de variables . . . 22
4.3.2 Nouvelles mutations . . . 23
4.3.2.1 Identification des mutations . . . 23

5 Discussion 25
5.1 Comparaison à l'état de l'art . . . 25
5.2 Pistes à suivre . . . 25
5.2.1 Étude de l'épistasie . . . 25
5.2.2 Étude phylogénétique . . . 25
5.2.3 Prise en compte du traitement . . . 26
5.2.4 Étude d'autres protéines . . . 26
5.2.5 Étude *ex silico* . . . 26
5.3 Autres approches . . . 27

Conclusion 28

Bibliographie **29**

Annexes **33**

A Tableaux de DRM **33**

B Exemple de calcul de contribution de variables **35**
B.1 Données 35
B.2 Contribution simple 35
B.3 Contributions jointes 37

C Les acides aminés **38**

D Résultats complets **39**
D.1 Données Africaines 39
D.1.1 Avec DRMs 39
D.1.1.1 Forêts aléatoires 39
D.1.1.2 Régression Logistique 41
D.1.1.3 Contributions de variables 43
D.1.2 Sans DRMs 45
D.1.2.1 Forêts aléatoires 45
D.1.2.2 Régression Logistique 47
D.1.2.3 Contributions de variables 49
D.2 Données Européennes 51
D.2.1 Avec DRMs 51
D.2.1.1 Forêts aléatoires 51
D.2.1.2 Régression Logistique 53
D.2.1.3 Contributions de variables 55
D.2.2 Sans DRMs 57
D.2.2.1 Forêts aléatoires 57
D.2.2.2 Régression Logistique 59
D.2.2.3 Contributions de variables 61

E code **63**

Table des figures

1.1 Structure de l'ADN en double hélice 2
1.2 Expression de l'information génétique 3
1.3 Le cycle de réplication du VIH schématisé 5

4.1 Importances des variables sur 100 itérations de stabilité. Données Africaines, tous sous-types 20
4.2 Importances des variables sur 100 itérations de stabilité. Données Européennes, tous sous-types 20
4.3 Pondérations des variables, moyenne sur 20 itérations. Données Africaines. Tous sous-types 21
4.4 Pondérations des variables, moyenne sur 20 itérations. Données Européennes. Tous sous-types 22
4.5 Hautes contributions, moyennées sur tous les individus. Tous sous-types 22

5.1 Exemple d'arbre phylogénétique 26

Liste des tableaux

2.1 Fréquences des sous-types les plus prévalents, Données Africaines 7
2.2 Fréquence des sous-types, données Européennes 7
2.3 Exemple d'encodage "OneHot" 9

4.1 performance d'une Forêt aléatoire pour diffŕents encodages 17
4.2 Performance des classificateurs (avec DRMs) 18
4.3 Performance des classificateurs (sans DRMs) 18
4.4 Mutations identifiées par sous-type par les importances de variables (clair), les poids de régressions logistiques (moyen) et les contributions de variables (foncé) . 23
4.5 p-valeurs des tests de différence de proportions 24

Liste des Abréviations

AA Acide aminé
ADN Acide Désoxyribonucléique
ARN Acide Ribonucléique
ART Thérapie Anti-rétrovirale
DRM Mutation de résistance *(Drug resistance mutation)*
HPC Cluster de calcul haute performance
IN Integrase
NNRTI Non-Nucleoside Reverse Transcriptase Inhibitor
NRTI Nucleoside Reverse Transcriptase Inhibitor
PR Protease
RT Reverse Transcriptase
SDRM Mutation de résistance de surveillance
SIDA Syndrome de l'Immunodéficience Acquise
VIH Virus de l'Immunodéficience Humaine

Introduction

Les Mutations de résistance (DRMs) qui apparaissent dans le VIH sous pression sélective médicamenteuse sont dangereuses, particulièrement lorsqu'elles sont transmises. Le receveur se trouve alors infecté par une souche qui n'est plus susceptible à certaines actions thérapeutiques. De plus les individus infectés par des souches résistantes mais non exposés à des traitements peuvent faciliter encore plus facilement la dispersion de souches résistantes (Mourad et al., 2015), rendant alors le nombre de traitements possibles limité au niveau d'une population. La détection de ces DRMs au sein des différentes souches virales est donc une étape primordiale lors du choix de la stratégie thérapeutique la plus adaptée.

Une liste des Mutations de résistance de surveillance (SDRMs), établie par des experts et mise à jour au cours des années, recense les DRMs les plus importantes. Les SDRMs de cette liste sont identifiées selon des critères variés : expériences *in vitro*, tests de susceptibilité sur des isolats cliniques ou de laboratoire ou encore larges études statistiques d'association entre traitement et le comportement viral (Liu and Shafer, 2006; Shafer, 2006; Wensing et al., 2017). Cependant cette liste reste incomplète, et des cas où un échec thérapeutiques est constaté ont été observés même en l'absence de SDRMs.

Le but de cette étude est d'appréhender le problème de détection de DRMs en utilisant une approche nouvelle basée sur les outils de l'apprentissage automatique. Ces outils seront utilisés sur des données génomiques de VIH.

Chapitre 1

Contexte et prérequis

1.1 Quelques notions de génétique

Des notions de base des mécanismes biomoléculaires de la génétique sont nécessaires pour bien comprendre certains choix et approches choisies au cours de cette étude.

1.1.1 Qu'est ce que l'ADN ?

L'Acide Désoxyribonucléique (ADN), est une des molécules du vivant les plus importante. Elle contient toute l'information génétique d'un être vivant que ce soit un arbre, une giraffe ou une bactérie. Elle se présente sous forme d'une double hélice, dont chaque brin est composé d'un squelette de sucre et de phosphate sur lequel vient se fixer une séquence de 4 bases azotées : A (adénine), T (thymine), G (guanine) et C (cytosine). Les bases azotées se lient entre elles pour unifier les deux brins : A avec T, et G avec C. Cette spécificité des liaisons induit donc une complémentarité des brins et la séquence de l'un peut être reconstruite à partir de l'autre. Un seul brin est alors nécessaire, on l'appelle le brin codant. La Figure 1.1 représente cette structure.

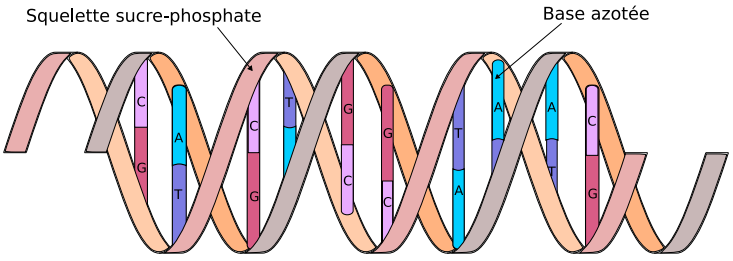


FIGURE 1.1 – Structure de l'ADN en double hélice

Toute la séquence de 4 bases, contenue sur la molécule d'ADN, constitue un génome. Le fait de parcourir la molécule d'ADN sur toute sa longueur et de noter la succession de bases est appelé séquençage.

Bien que la molécule d'ADN soit indispensable au fonctionnement de la vie, elle n'est pas directement fonctionnelle. L'information contenue dans la séquence de bases azotées doit être lue et ensuite exprimée.

1.1.2 L'expression de l'information génétique

L'expression de l'information génétique s'effectue en deux étapes : la transcription et la traduction.

1.1.2.1 La Transcription

Au cours de cette étape, la séquence d'un fragment du brin d'ADN codant est transcrite sur un brin d'Acide Ribonucléique (ARN). L'ARN à une structure presque identique à celle d'un simple brin d'ADN, et est aussi une séquence de 4 bases azotées. Il y a cependant une différence au niveau des bases utilisées : dans l'ARN la thymine de l'ADN est remplacée par de l'uracile (U), mais cela ne change pas la séquence. La structure en simple brin de l'ARN facilite l'étape suivante de traduction, ainsi que le déplacement de celui ci au sein de la cellule.

1.1.2.2 La Traduction

Cette seconde étape permet, à partir de la séquence d'ARN, de synthétiser des protéines. Les protéines sont les molécules effectrices du vivant, elles sont constituées d'une séquence d'Acide aminé (AA). La séquence d'AA découle de celle des bases azotées de l'ARN, grâce à une correspondance entre un enchaînement de 3 bases, appelé codon, et un AA. C'est ce qui est appelé code génétique (Annexe C). Il est important de noter que cette correspondance c'est pas bijective, plusieurs codons différents peuvent coder le même AA. De plus il existe des codons STOP, indiquant aux enzymes responsables de la traduction où s'arrêter. La redondance du code génétique permet notamment de limiter les conséquences de mutations de substitution l'ADN, puisque l'échange d'une base azotée par une autre change le codon mais pas forcement l'AA. Le code génétique est disponible en annexe C.

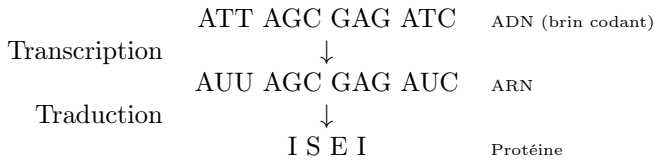


Figure 1.2 – Expression de l'information génétique

L'expression de l'information génétique est représentée en figure 1.2. Pour faciliter la lecture des séquences d'ADN et d'ARN, ces dernières ont été découpées selon leur codons mais bien entendu, dans le vivant, ces molécules se présentent sous forme d'une seule longue séquence.

1.1.3 Les protéines

Les protéines sont les molécules effectrices du vivant et ont des rôles divers et variés. Elles peuvent être enzymatiques et catalyser des réactions du vivant, structurelles au sein de tissus, régulatrices, etc... Ces protéines, comme vu plus haut, sont des séquences d'AA (structure primaire), mais des repliements au sein de la chaîne peptidique et dans l'espace (structures secondaires et tertiaires) donnent aux protéines une structure tridimensionnelle. Certaines protéines sont même composées de plusieurs chaînes liées entre elles (structure quaternaire). C'est cette forme complexe dans l'espace qui donne aux protéines leur fonction en regroupant, sur un même

site, des groupes fonctionnels d'AAs pouvant être très éloignés les uns des autres dans la séquence brute d'AAs codée par l'ADN.

Dans beaucoup de cas, et plus particulièrement dans le cas du VIH, les traitements contre des agents pathogènes sont basés sur la perturbation de l'action des protéines du pathogène. Le traitement peut dégrader la protéine ou plus souvent se fixer sur les sites effecteurs de la protéine cible pour inhiber son action. On parle alors d'inhibiteur.

1.2 Le VIH

1.2.1 Présentation du VIH

Le Virus de l'Immunodéficience Humaine (VIH) est un rétrovirus affectant les cellules du système immunitaire humain. Il est responsable du Syndrome de l'Immunodéficience Acquise (SIDA) rendant les individus affectés très vulnérables à des infection secondaires opportunistes. Le VIH est transmissible sexuellement ou lors de contact avec ou transfert de sang. Il existe deux espèces de VIH, VIH-1 et VIH-2 mais cette étude se limitera au VIH-1 qui est beaucoup plus infectieux et prévalent que le VIH-2 (Gilbert et al., 2003). Le VIH est classifié en plusieurs sous-types dénotés par une lettre (A, B, C, ...) constitués sur une base génétique. Il existe également des formes recombinantes notées CRF qui ont un génome constitué d'un assemblage de génomes de plusieurs sous-types. Par exemple la forme recombinante CRF02_AG associes des génomes des sous-types A et G.

1.2.2 Génome du VIH

Le VIH a un génome composé de 9 gènes, dont trois qui sont particulièrement importants (Fisher et al., 2007, p.295) : *gag* qui code des protéines de la matrice virale, *env* qui code les glycoprotéines de l'enveloppe virale et le gène *pol* qui code la Reverse Transcriptase (RT), Protease (PR) et Integrase (IN). Ce dernier gène est le seul étudié lors de cette étude puisque les traitements anti-rétroviraux se concentrent sur le blocage de l'action de RT, PR et IN. C'est donc sur ce gène que sont situées les mutations de pharmaco-résistance les plus importantes. Un intérêt particulier sera porté à la Reverse Transcriptase.

1.2.3 Le cycle du VIH

Le VIH infecte les cellules du système immunitaire humain. Son cycle est représenté sous une forme simplifiée, dans la figure 1.3. Dans un premier temps le virus se fixe sur la cellule hôte et fusionne avec pour faire y faire rentrer l'ARN et des protéines virales (PR, RT et IN). L'ARN viral est reversement transcrit en ADN viral par la RT. Cet ADN rentre alors dans le noyau de la cellule hôte. Au sein du noyau l'IN insère l'ADN viral dans l'ADN de l'hôte, celui-ci est exprimé via les processus de transcription et de traduction décrits plus haut, pour produire l'ARN viral et des précurseurs des protéines du VIH. La PR agit sur ces précurseurs pour en faire les protéines virales matures. Ces protéines, ainsi que l'ARN migrent vers la membrane de la cellule hôte, et via bourgeonnement forment un nouveau virus.

Une cellule hôte produit une multitude de VIH et ce cycle de réplication entraîne la mort de celle-ci. La destruction des cellules du système immunitaire lors de la réplication du VIH est la cause du SIDA, et ce n'est pas le VIH qui est directement responsable en cas de mort de l'individu infecté, mais plutôt un agent pathogène opportuniste qui aura pu se développer en l'absence de réponse immunitaire.

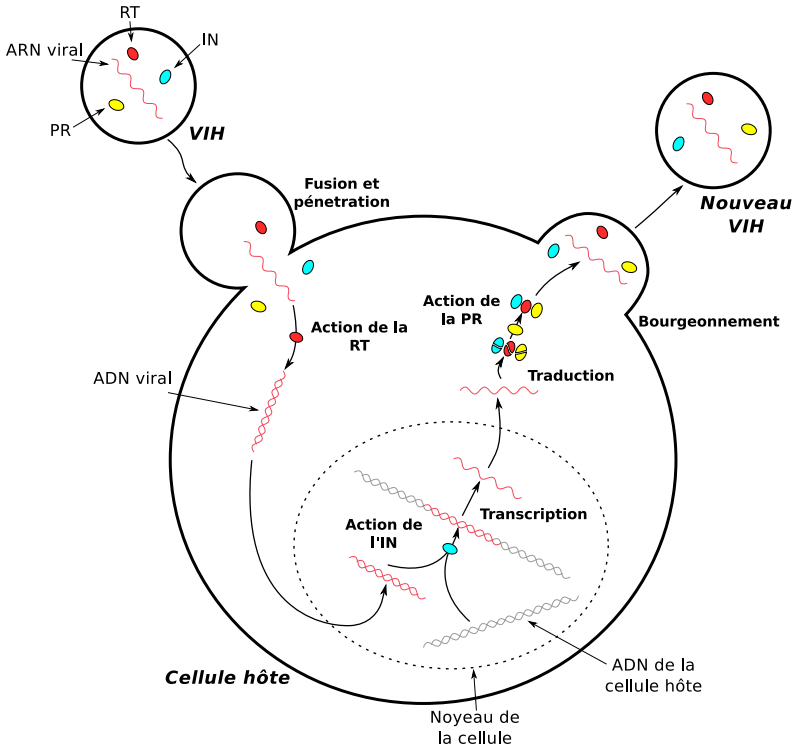


Figure 1.3 – Le cycle de réplication du VIH schématisé

1.3 Les mutations de résistance

1.3.1 Comment apparaissent les mutations ?

Lors des Thérapie Anti-rétrovirale (ART), les actions de RT, PR et IN sont bloquées par les composés médicamenteux. Ces protéines étant indispensable à la réplication du VIH la multiplication des virus est stoppée. Cette pression thérapeutique entraîne la sélection de mutations permettant de résister à un ou plusieurs des composés médicamenteux. Ces DRMs acquises apparaissent donc chez les individus infectés suivant un traitement, c'est la transmission verticale. Cependant s'il y a transmission d'un virus muté, d'un individu ayant été exposé à une ART vers un autre individu, ou entre deux individus non exposés (Mourad et al., 2015), il peut y avoir des cas d'échec thérapeutique chez des patients qui n'ont jamais été traités auparavant mais qui sont quand même infectés par des virus resistants. Il est cependant intéressant de noter qu'il est admis que le VIH sauvage *(i.e. sans DRMs)* a une meilleure capacité de survie que ses contreparties mutées, il en découle qu'en l'absence de pression thérapeutique (donc chez les individus non exposés aux traitements), il y a réversion des DRMs après un certain temps (Castro et al., 2013).

1.3.2 Les mutations de surveillance

Certaines DRMs sont plus prévalentes que d'autres, ou encore on un effet plus grave. C'est pourquoi il existe une liste de SDRMs établie par des chercheurs de l'université de Stanford (Bennett et al., 2009; Rhee et al., 2003; Shafer, 2006). D'autres listes qui contiennent des DRMs de plusieurs niveaux d'importance existent aussi (Clark et al., 2007; Wensing et al., 2017). Cependant ces listes ne sont pas exhaustives, dans certains cas il peut y avoir un échec thérapeutique sans pour autant constater la présence de SDRMs dans les séquences du virus. De plus comme le virus est transmissible, un individu qui n'a jamais reçu de traitement peut être infecté par une

souche résistante de VIH.

1.3.3 Nommage des mutations

La convention de nommage des DRMs est la suivante :

$$Nom \ = \ AA \ originel + position + AA \ muté$$

Par exemple une mutation au $184^{\text{ème}}$ AA de la Méthionine (M) vers la Valine (V) se nomme $M184V$. C'est avec cette nomenclature que les DRMs et SDRMs seront désignées au cours de cette étude.

1.4 État de l'art

1.4.1 Détection de DRMs

Il existe plusieurs manières d'étudier les DRMS et d'en identifier. Toutes passent par l'identification de corrélation entre la variabilité génotypique des virus et d'autres facteurs tels que le phénotype, le traitement ou la réponse virologique à un traitement (Liu and Shafer, 2006; Shafer, 2006; Wensing et al., 2017). Ce genre d'études peut être fait *in-vitro* ou *in-vivo*.

Dans les associations génotype-réponse on observe la réponse virologique à un traitement donné. Par exemple on mesure le changement de charge virale en réponse à un composé médicamenteux spécifique (Masquelier et al., 2004; Molina et al., 2005).

Dans les associations génotype-phénotype on corrèle le génotype du VIH avec sa susceptibilité à un traitement donné. Ce genre d'études n'est effectué que *in-vitro* puisqu'il nécessite des faire tests de susceptibilité sur des isolats de VIH, permettant de mesurer cette dernière pour un ensemble de traitements. On quantifie la résistance en mesurant la différence de concentration du composé médicamenteux nécessaire à l'inhibition de la réplication du VIH entre l'isolat du patient infecté et une souche sauvage de référence (Pattery et al., 2012; Villacian, 2009).

Dans les associations génotype-traitement on corrèle le génotype avec le traitement reçu. Pour faire cela on compare la prévalence de chaque mutation entre les patients qui suivent un traitement en particulier et ceux qui ne le suivent pas. Ce genre d'étude peut être fait *in-vivo* ou *in-vitro*. Cependant les mutations sélectionnés lors des études *in-vitro* ne sont en général qu'un sous ensemble de celles sélectionnées *in-vivo* (Liu and Shafer, 2006).Dans cette étude une approche de ce type a été utilisée.

1.4.2 Problèmes Statistiques

Les associations génotype-traitement reposent sur de grandes études statistiques, impliquant des tests multiples puisque la différence de prévalence de chaque mutation doit être testée entre les échantillons traités et non-traités. Ces tests multiples conduisent à une diminution du pouvoir statistique et une augmentation du risque d'erreur de type II (Sham and Purcell, 2014). Dans cette étude des techniques d'apprentissage automatique sont utilisées pour identifier un sous-groupe de mutations "intéressantes" pour lesquelles des tests seront ensuite effectués. Cela permet de cibler les tests et augmenter leur pouvoir statistique.

Chapitre 2

Préparation des données

2.1 Données utilisées

Dans cette étude deux ensembles de séquences du gène *pol* issus de VIH-1 ont été utilisés. Les virus ont été isolés dans des patients ayant suivi un traitement ou non. On parlera respectivement de séquences *treated* et *naive*.

2.1.1 Données Africaines

Le premier de ces jeux de données contient 3990 séquences de RT de VIH-1, collectées durant l'étude de Villabona & al en 2016 (Villabona-Arenas et al., 2016) dans 10 pays d'Afrique occidentale et centrale. Ces séquences sont séparées en séquences *naive* et *treated*, avec $n_{naive} = 2316$ (58%) et $n_{treated} = 1674$ (42%). Parmi celles-ci 72 (3.1%) séquences *naive* et 1361 (81%) *treated* présentent des SDRMs. Ces SDRMs ainsi que le sous-types (Table 2.1) sont obtenus grâce au web service sierra de l'université de Stanford (Tang et al., 2012) et sa librairie Python Sierrapy.

Table 2.1 – Fréquences des sous-types les plus prévalents, Données Africaines

Naive		Treated	
CRF02_AG	**28.6%**	CRF02_AG	**48.8%**
C	**25.7%**	A	**11.2%**
A	**12.3%**	CRF06_cpx	**7.6%**
CRF01_AE	**8.2%**	G	**6.4%**
CRF06_cpx	**6.5%**	C	**6.3%**
G	**4.4%**	CRF01_AE	**5.0%**

Table 2.2 – Fréquence des sous-types, données Européennes

Naive		Treated	
B	**69%**	B	**64%**
C	**31%**	C	**36%**

2.1.2 Données Européennes

Le second jeu de données a été élaboré en Europe. Ce jeu de données contient 55544 séquences de VIH-1 *pol* contenant chacun le gène de RT et de PR. Ces séquences sont aussi séparées en *naive*

et *treated*, avec $n_{naive} = 41925$ (75%) et $n_{treated} = 13619$ (25%). Parmi ces séquences 1472 (3.5%) séquences *naive* et 870 (6.4%) *treated* sequences présentent des SDRMs. Les séquences de ce jeu de données n'appartiennent qu'aux sous-types B et C (voir Table 2.2). Les sous-types de ces séquences sont déterminés par la méthode REGA (Pineda-Peña et al., 2013).

Mis à part leur taille, ces jeux de données présentent des différences clés :
- Dans les données Africaines, il est admis que tous les échantillons *treated* on suivi le même régime thérapeutique détaillé dans l'étude de Villabonna (Villabona-Arenas et al., 2016). Dans les données Européennes la stratégie thérapeutique est inconnue pour la plupart des séquences.
- Comme on peut l'observer en regardant les métriques précédentes, la composition phylogénétique est très différente dans les deux jeux de données.
- Proportionnellement, les séquences *treated* du jeu Africain ont un beaucoup plus grand taux de SDRMs que leur homologues Européennes.

2.2 Création du jeu de données

Les séquences disponibles sont des séquences d'ADN, or dans le VIH (et dans tous les autres organismes) ce n'est pas l'ADN qui est directement responsable des mécanismes du vivant mais les protéines. Il est donc nécessaire d'obtenir les séquences protéiques à partir des séquences génétiques disponibles.

2.2.1 Traduction et alignement

Les séquences d'ADN sur lesquelles cette étude est effectuée sont alignées entre elles pour chaque jeu de données. Le web service *Sierra* de l'Université de Stanford (Tang et al., 2012) permet dans un premier temps de traduire les séquences d'ADN en séquences d'AAs, dans un second temps ces séquences protéiques sont alignées à une séquence consensus (une séquence de référence pour laquelle les experts sont d'accord qu'il n'y a pas de mutations) et d'obtenir les AAs de début et de fin des gènes qui nous intéressent (RT principalement).

2.3 Encodage des séquences

Par la suite de ce projet la librairie Python de machine-learning `scikit-learn` est utilisée. les algorithmes de cette librairie, de part leur implémentation ou de part leur nature, nécessitent des variables numériques. La régression logistique par exemple ne fonctionne que sur des variables réelles, et l'implémentation des Random Forest de `scikit-learn` nécessite également des variables numériques, malgré le fait que l'algorithme original de Leo Breiman fonctionne avec des variables catégoriques quelque soit leur nature (Breiman, 2001).

Il est donc nécessaire de trouver des encodages qui permettent d'obtenir des jeux de données numériques à partir des alignements de séquences. Chaque variable (donc position dans la séquence protéique) à potentiellement 21 valeurs possibles : les 20 AAs protéinogènes et un symbole de gap pour les mutations de délétion. Plusieurs types d'encodages différents ont été comparés lors de cette étude.

2.3.1 Simples encodages numériques

Les premiers types d'encodage ayant été utilisés sont de simples encodages numériques qui ne font que différencier les valeurs possibles des AAs. Ils sont faciles à mettre en place et ne nécessitent aucune information supplémentaire au jeu de données.

2.3.1.1 L'étiquetage

Le premier encodage, et le plus simple, est un simple étiquetage numérique. À chaque valeur possible des AAs est attribué un nombre entier. Les valeurs catégoriques sont ensuite remplacées par leur entier correspondant. Cet encodage présente un inconvénient majeur, il induit une notion d'ordre et de proximité des AAs qui n'a aucune base biologique.

2.3.1.2 L'encodage "Bitwise"

Ce second encodage est une extension de l'étiquetage. Le principe d'attribution d'un entier à chaque niveau de la variable catégorique est le même, mais au lieu de le laisser en base décimale il est converti en base binaire sur 5 bits. Chaque position de la séquence représente alors 5 colonnes du jeu de données encodées.

2.3.1.3 L'encodage "OneHot"

L'encodage "OneHot" (aussi appelé dummy encoding (Carey, 2013, p.234)) est souvent utilisé lorsque les variables utilisées sont catégoriques. Chaque variable est transformée en un vecteur ayant pour longueur le nombre de valeurs possibles pour cette variable. Pour chaque observation de cette variable un 1 est placé dans la colonne correspondant à la valeur de la variable pour cette observation et un 0 pour les autres valeurs possibles. Il est possible d'encoder une variable catégorique à K niveaux avec un vecteur de taille $K-1$ avec une des valeurs représentées par un vecteur rempli de 0, mais pour être explicite les vecteurs retournés par cet encodage dans cette étude ont pour taille le nombre de niveaux de chaque variable catégorique.
Dans l'exemple de la Table 2.3, on a 3 observations d'un variable catégorique à 3 valeurs possibles : A, K ou R. L'encodage retourne alors 3 vecteurs de longueur 3.

Table 2.3 – Exemple d'encodage "OneHot"

	sans encodage	encodage "OneHot"		
observation	Variable	V_A	V_K	V_R
1	A	1	0	0
2	R	0	0	1
3	K	0	1	0

Le plus grand avantage de cet encodage est que les variables résultantes traduisent la valeur de la variable originale. Donc lors de sélection de variables on peut identifier non seulement les variables importantes mais aussi quelles valeurs de ces dernières sont les plus importantes.

2.3.2 Encodages physico-chimiques

D'autres types d'encodages, basés sur les propriétés des AAs ont été essayés. Ces encodages ont pour but d'essayer de transmettre des informations en plus aux algorithmes qui seront exécutés sur le jeu de données.

2.3.2.1 Encodage basé sur AAIndex

Cet encodage a été élaboré en utilisant des valeurs de la base de données AAIndex (Kawashima et al., 2007) qui donne des valeurs pour une multitude de propriétés physico-chimiques (sous forme de nombres réels) pour chaque AA. Ces valeurs, plutôt que d'être des mesures des propriétés sont des valeurs relatives entre les AAs. Un sous-ensemble de ces propriétés a été sélectionné dans l'étude de Li & al. (Li et al., 2009) et a été utilisé pour récupérer les valeurs pertinentes de la base de données. Ces valeurs ont été additionnées pour chaque AA de manière à créer une valeur unique pour chaque niveau de la variable catégorique.

2.3.2.2 Encodage d'appartenance de groupes

Ce second encodage à été mentionné dans une conférence IEEE en 2011 (Zamani and Kremer, 2011) et est basé sur des sous-groupes d'AAs définis par rapport à une propriété. Chaque AA appartient ou pas à un 9 sous ensembles définis par Taylor en 1986 (Maetschke et al., 2005; Taylor, 1986) tels que les groupes d'AAs polaires, hydrophobes, petits, ... Un ensemble de 5 sous-groupes supplémentaires a été défini par Kremer et Hao (Kremer and Lac, 2009). Ces 14 propriétés permettent alors d'associer à chaque AA un vecteur binaire de dimension 14, avec des valeurs de 1 lorsque l'AA appartient au sous-groupe associé à la colonne considérée ou de 0 lorsque l'AA n'y appartient pas.

Le choix de l'encodage approprié se fera sur plusieurs critères. D'une part il faudra que cet encodage permette aux classificateurs de bien différencier les AAs à chaque position, cela devrait se traduire par une meilleure performance des classificateurs. D'autre part l'interprétabilité des résultats est primordiale, il sera donc tenu compte de cet aspect lors du choix final.

Chapitre 3

Méthodes employées

Pour détecter de nouvelles DRMs, plusieurs méthodes ont été essayées. Cependant elles reposent toutes sur le même principe de base : l'extraction de variable. En effet chaque position des séquences d'AAs correspond à une variable (ou plusieurs suivant l'encodage). De ce fait identifier des variables importantes revient à identifier des positions d'intérêt. Celles-ci permettent de cibler et de réduire le nombre de tests statistiques.

La méthode de sélection de ces variables importantes dépend de l'algorithme qui est implémenté, mais le principe de base est commun, quelque soit la méthode utilisée. Le point de départ est l'entraînement d'un classificateur binaire qui discrimine les séquences *naive* et *treated*. Les séquences *treated* sont issues de patients qui ont suivi un ART, donc qui sont susceptibles de développer des DRMs. Cependant si le séquences de VIH ont put être extraites de ces patients *treated*, c'est que la charge virale est suffisamment haute pour le séquençage et donc que le traitement a échoué. Cet échec pet être du à des DRMs ou à une mauvaise prise des traitements, mais malgré cette possibilité, les séquences *treated* sont considérées comme des séquences résistantes. Les classificateurs apprenent donc à classifier des séquences résistantes *versus* des séquences susceptibles aux traitements.

Lors de l'apprentissage de ce classificateur les variables importantes pour la tâche de discrimination sont identifiée. Ce classificateur binaire permet aussi l'identification d'un sous ensemble de séquences *treated*, celles qui auront été correctement classifiées, notées *vrai positifs*. Ces séquences sont intéressantes car elles présentent vraisemblablement des caractéristiques qui permettent facilement de les distinguer des séquences *naives* (*i.e* des DRMs). Des tests statistiques sont ensuite effectués pour comparer et étudier les différences entre les séquences *naives* et les *treated*. Dans le cas où les classes ont des effectifs trop différents (comme dans le cas des données Européennes), les proportions entre les classes sont rééquilibrées par sous-échantillonnage de la classe majoritaire.

3.1 Classification des données

Différents algorithmes sont utilisés mais ils sont appliqués de la même manière au jeu de données. Le jeu de données divisé en 20 plis de validation croisée. L'algorithme est alors entraîné sur le sous-jeu d'entraînement et sa performance est mesurée avec les scores de précision, de rappel et de précision totale sur les jeux de validation.

$$précision = \frac{vrais\ positifs}{vrais\ positifs + faux\ positifs}$$

$$rappel = \frac{vrais\ positifs}{vrais\ positifs + faux\ négatifs}$$

$$précision\ totale = \frac{vrais\ positifs + vrais\ négatifs}{nombre\ total\ de\ cas}$$

Étant donné que l'intérêt de la tâche de classification est d'extraire des variables importantes et que l'on ne veut pas classifier des séquences dont on ne connaît pas le statut, il n'y a pas de séparation en jeu d'entraînement et en jeu de test. Lors de l'entraînement le classificateur prédit la classe de chaque séquence de l'ensemble de validation. Cela permet d'avoir une prédiction pour chaque séquence. Cette validation croisée est effectuée de multiples fois et les séquences correctement classifiées comme *treated* sont gardées en mémoire. Ceci permet de vérifier s'il existe des séquences "robustes" qui sont facilement classifiables et qui peuvent alors présenter des caractéristiques intéressantes.

Une étape de sélection de variables est aussi utilisée. Pour réduire les effets aléatoires inhérents à certains algorithmes de classification, le classificateur est entraîné 100 fois. À chaque itération, les 20 variables avec les plus hautes importances sont enregistrées. La fréquence de sélection de chaque variable dans les plus importantes est calculée ainsi que l'importance relative moyenne sur toutes les itérations.

3.1.1 Forêts Aléatoires

Les forêts aléatoires ont été choisies au commencement de ce projet comme un algorithme approprié à la tâche de classification à résoudre.

3.1.1.1 Choix de l'algorithme

Les Forêts Aléatoires ont été choisies pour quelques raisons principales. D'une part elles sont simples à entraîner et fonctionnent bien sur de grands jeux de données tels que ceux utilisés dans cette étude sans pour autant faire de sur-apprentissage (Qi, 2012). D'autre part, elles permettent d'extraire des importances de variables. Finalement, bien que le modèle complet soit relativement difficile à interpréter, de nombreux travaux sont fait pour faciliter l'interprétation de ces modèles (Cui et al., 2015; Deng, 2014; Hara and Hayashi, 2016; Tan et al., 2016).

Lors de l'entraînement du classificateur, les hyper-paramètres sont choisis par optimisation aléatoire (Bergstra and Bengio, 2012) plutôt que de l'optimisation par "grid search", qui nécessite un temps plus long pour arriver à des résultats similaires.

3.1.1.2 Importance des variables

C'est là une des caractéristiques les plus utiles des Forêts Aléatoires dans le cadre de cette étude. Dans l'implémentation Python `scikit-learn` des Forêts Aléatoires, l'importance relative de chaque variable est calculée en regardant la profondeur des nœuds dans lesquels la variable est utilisée pour séparer le jeu de données. Les variables donnant les meilleures segmentations, donc diminuant l'impureté de Gini dans chaque sous-nœud (Trevor et al., 2009, p.309), sont situées

en haut de l'arbre donc en regardant le rang moyen des variables dans tous les arbres de décision de la forêt, une importance relative peut être attribuée aux variables explicatives.

3.1.2 Regression logistique

Dans les Forêts Aléatoires, lors du calcul des importances, la classe des séquences n'est pas prise en compte. On obtient donc des variables importantes lors de l'attribution de toutes les classes possibles (ici *naive* et *treated*). C'est pourquoi la régression logistique a aussi été utilisée. Durant l'entraînement de la régression, des coefficients sont attribués à chaque variable, ils sont positifs si une valeur élevée contribue à la classification dans la classe *treated* et négatifs s'ils contribuent à l'attribution de la classe *naive*. Cela permet d'extraire les variables qui permettent de séparer la classe *treated*.

La régression logistique ne peut utiliser que des variables numériques, il est donc important de bien choisir l'encodage des séquences d'AA pour pouvoir extraire le maximum d'information du jeu de données.

3.2 Contribution des variables

Les importances des variables extraites des Forêts Aléatoires ne prennent pas en compte les différentes classes. Pour pallier à ce problème le classifications effectuées par le modèle peuvent être interprétées en calculant les contributions de chaque variable à la classification d'une séquence. Cette méthode est issue de l'étude de Palczewska et al. (Palczewska et al., 2014).

3.2.1 Calcul des contributions

La première étape du calcul des contributions de variables est de calculer les incréments de contributions locales LI. Dans un problème de classification binaire, pour une variable f et un nœud c d'un arbre, l'incrément de contribution locale LI_f^c est égale à la différence de proportion de la classe "positive" (donc la classe *treated*) entre le nœud c et son parent si la séparation à ce nœud est fait sur la variable f et 0 sinon. On obtient donc les contributions de toutes les variables à tous les nœuds pour la classe "positive".

On peut ensuite calculer la contribution d'une variable f à la classification d'un séquence i en sommant les incréments de contributions locales LI_f sur tous les nœuds du chemin de décision que parcourt la séquence dans l'arbre considéré. On a donc pour un arbre t :

$$FC_{t,i}^f = \sum_{c \in path} LI_f^c$$

Finalement pour obtenir la contribution de la variable à la classification dans la classe "positive" d'un séquence i on fait la moyenne des contributions de la variable considérée pour tous les arbres de la forêt de taille T :

$$FC_i^f = \frac{1}{T} \sum_t FC_{t,i}^f$$

Ces contributions permettent de savoir pourquoi les séquences ont été classées dans une certaine classe, donnant une information plus précise que les importances de variables. En pratique, ces contributions sont calculées avec la librairie Python `treeinterpreter`. Un exemple de calcul de contributions détaillé est joint en annexe B.

3.2.2 Contributions jointes

La librairie `treeinterpreter` permet aussi d'obtenir des contributions jointes. Ces contributions jointes sont calculées de la même manière que pour les contributions variable par variable, à l'exception que la contribution calculée à un nœud d'un arbre n'est pas attribuée à la seule variable utilisée dans le nœud mais à une combinaison de toutes les variables des utilisées dans le chemin décisionnel parcouru par la séquence avant le nœud considéré. De ce fait la seule instance où l'on peut avoir une variable qui apporte sa contribution seule est lorsque le nœud considéré est la racine de l'arbre. Un exemple de calcul est aussi donné en annexe B.

3.3 Réduction du jeu de données

Les méthodes ci dessus sont utilisées pour détecter des différences entre les séquences *naive* et *treated*, en l'absence de facteurs confondants, ces différences sont considérées dues au traitement et donc potentiellement des DRMs. Cependant le but de cette étude est de trouver de nouvelles DRMs, et si les données contiennent des séquences présentant des DRMs, les méthodes mises en place plus haut ne détecterons que les DRMs connues et peu de nouvelles mutations seront trouvées.

Deux applications des algorithmes de classification ont été effectuées pour pallier à ce problème :

- Les classificateurs sont appliqués une première fois sur le jeu de données contenant les séquences simplement encodées. Les DRMs présentes dans les séquences y restent. Si l'algorithme permet de détecter les DRMs présentes dans le jeu de données, la méthode est validée.
- Les classificateurs sont appliqués une deuxième fois sur un jeu de données où les DRMs ont été éliminées. De ce fait les mutations détectées seront nouvelles.

3.3.1 Identification de DRMs

Pour pouvoir éliminer les DRMs des séquences du jeu de données, il faut d'abord les identifier. Cette étape est effectuée lors de la préparation des données, au moment de l'alignement avec la séquence consensus. L'utilisation du web-service *Sierra* est abordée en 2.2.1, et les DRMs des séquences sont identifiées grâce à cet outil. Cependant, il existe des DRMs identifiées sur d'autres listes d'experts que celle de Stanford. Pour les éliminer un tableau de DRMs complémentaires a été établi et sert également à identifier les séquences contenant ou non ces dernières.

3.3.2 Élimination des DRMs

Plusieurs méthodes ont été essayées pour éliminer les DRMs du jeu de données.

- La première était d'éliminer les séquences présentant des DRMs. Cette approche provoque 2 problèmes : d'une part en éliminant des séquences, la taille du jeu de données diminue et de ce fait la performance des classificateurs aussi. D'autre part, dans le cas des données Africaines, la proportion de séquences contenant des DRMs est très différente entre les classes *naive* et *treated* (cf. 2.1.1). Éliminer les séquences cause alors un gros problème de déséquilibre entre les deux classes qui, pour être résolu, nécessite un sous-échantillonnage des séquences conduisant à une autre diminution du jeu de données. Cette approche à donc été écartée.
- La seconde approche était de remplacer les DRMs par une autre valeur : soit un *valeur manquante* soit par l'AA le plus fréquent à la position considérée. Cependant cela implique

encore des problèmes d'un point de vue biologique. En effet si la DRM est remplacée par un autre AA, ou alors par *valeur manquante* qui sera imputée par le classificateur, des séquences qui n'ont pas de fondement biologique sont crées et de ce fait le modèle apprend sur des données erronées. Cette approche a aussi été écartée.

— La troisième et dernière approche consiste à retirer les positions présentant des DRMs du jeu de données. C'est l'approche inverse de la première : au lieu d'éliminer les lignes du jeu de données en retirant les séquences, ce sont les colonnes qui sont retirées. La taille du jeu de données reste constante mais la dimension des séquences diminue. Par exemple, si des séquences présentent les DRMs *M184V* ou *M184I* on élimine la position 184 de toutes les séquences du jeu de données. Ceci permet de garder tous les exemples du jeu de données sans pour autant créer de fausses valeurs. C'est donc cette méthode qui a été choisie, pour éliminer les DRMs du jeu de données.

3.4 Tests statistiques

Malgré le but de cet étude de diminuer le nombre de tests effectués, il est quand même nécessaire d'en faire. Les tests sont utilisés dans plusieurs buts mais ce sont tous des tests de comparaison de proportions entre deux populations.

3.4.1 Comparaisons de proportions

la comparaison est basé sur une statistique de test Z :

$$Z = \frac{(p_{naive} - p_{treated})}{\sqrt{p(1-p)(\frac{1}{n_{naive}} + \frac{1}{n_{treated}})}} \tag{3.1}$$

où p_{naive} est la proportion à tester dans la population *naive*, $p_{treated}$ celle de la population *treated*, p la proportion globale dans tout le jeu de données et n_{naive} et $n_{treated}$ le nombre de séquences *naive* et*treated* respectivement. Ce score z est ensuite utilisé dans les tests de proportions de sous-types et de mutations.

Dans le test des proportions de sous-types, effectué pour l'étude d'effets confondants, le test est celui de l'équation 3.2. Dans ce cas, les test est bilatéral car il n'y a pas d'*a priori* sur la distribution des sous-types, il n'est pas attendu d'avoir un certain sous-types plus présent dans une certaine classe. Si la différence de proportion sur un sous-type entre les séquences *naive* et *treated* est significative, cela veut dire qu'il y a une chance que le sous-type soit un facteur confondant et que le classificateur ait appris à discriminer selon le sous-type plutôt que selon le statut traité non traité.

$$\begin{cases} H0 & : p_{naive} = p_{treated} \\ H1 & : p_{naive} \neq p_{treated} \end{cases} \tag{3.2}$$

Dans le cas des tests sur la proportion des mutations identifiées, le test utilisé suit les hypothèses de l'équation 3.3. Dans ce cas ci le test est unilatéral. En effet si la mutation testée est bien une DRM, en raison de la pression de sélection induite par le traitement, il devrait y avoir une proportion plus élevée de cette mutation dans les séquences *treated* que dans les séquences *naive*.

$$\begin{cases} H0 & : p_{naive} \geq p_{treated} \\ H1 & : p_{naive} < p_{treated} \end{cases} \tag{3.3}$$

3.4.2 Problèmes de tests multiples

Malgré le fait que le but de cette approche soit de réduire le nombre de tests statistiques et ainsi d'augmenter la puissance statistique de ces derniers, des tests multiples sont quand même utilisés puisque ces derniers sont tous effectués sur le même jeu de données. Il faut donc apporter une correction pour pallier à l'augmentation de risque d'erreur de type I engendrée par les tests multiples. Par conséquent, lors de cette étude, la correction de Bonferroni est utilisée (Weisstein, 2004). Pour avoir un risque α global à la valeur que l'on veut, le risque individuel de chaque test α_i est réduit. Le test ne sera alors considéré comme significatif que si $p_{value} \leq \alpha_i$. Cette correction donne alors, pour m tests :

$$\alpha_i = \frac{\alpha}{m}$$

3.5 Autres outils

3.5.1 Librairies Python

Un certain nombre de librairies Python sont utilisées dans cette étude. Les données sont structurées avec al libraire Pandas (McKinney et al., 2010), et analysées avec la librairie SciPy (Jones et al., 01). Les algorithmes d'apprentissage automatique sont issus de la librairie scikit-learn (Pedregosa et al., 2011). Les tests statistiques ont été réalisés avec la librairie statsmodels (Seabold and Perktold, 2010). Les figures ont été réalisées avec la librairie seaborn (Waskom et al.), basée sur la librairie matplotlib (Hunter, 2007).

3.5.2 Outil de work-flow

Pour assurer la répétabilité des expériences et simplifier les executions, l'outil de gestion de work-flow Snakemake est utilisé (Köster and Rahmann, 2012). En écrivant des règles pour chaque étape de l'exécution, en indiquant les fichiers d'entrée et de sortie de chaque étape, un graphe acyclique de dépendance est généré et l'ordre d'exécution de chaque règle est inféré à partir de ce graphe. Ceci permet de simplifier considérablement l'exécution de tout le programme en évitant la ré-exécution superflue (notamment de préparation des données) à chaque nouvel essai.

3.5.3 Utilisation d'un cluster de calcul haute performance

L'institut Pasteur dispose d'un Cluster de calcul haute performance (HPC), TARS, composé de 380 serveurs et de plus de 4800 cœurs. Ce HPC est basé sur le système d'ordonnancement des tâches SLURM. Ce cluster permet de grandement diminuer les temps de calcul en offrant une très grosse capacité de parallélisation. L'outil de work-flow Snakemake permet également de paralléliser automatiquement les tâches parallélisables.

Chapitre 4

Résultats

4.1 Choix de l'encodage

Comme indiqué en 2.3.2.2, le choix de l'encodage se base sur plusieurs critères. D'une part sur la performance des classificateurs lorsque cet encodage est utilisé, et d'autre part sur des critères d'interprétabilité des résultats.

4.1.1 Performance des encodages

Les performances des encodages ont été testées dans des conditions "faciles" pour le classificateur, c'est à dire en laissant les DRMs dans le jeu de données. Les résultats présentés dans la table 4.1 indiquent un schéma d'encodage plus performant que les autres : l'encodage OneHot.

TABLE 4.1 – performance d'une Forêt aléatoire pour diffŕents encodages

	Etiquetage		Bitwise		OneHot		AAIndex		Groupes	
	Afrique	Europe	Afrique	Europe	Afrique	Europe	Afrique	Europe	Afrique	Europe
rappel	0.7191	*0.4739*	0.7193	*0.4940*	**0.8148**	*0.4368*	0.7312	***0.5183***	0.7354	*0.5084*
précision	**0.9914**	*0.5406*	0.9900	*0.5454*	0.9520	***0.8294***	0.9860	*0.5357*	0.9834	*0.5371*
précision totale	0.8796	*0.5355*	0.8792	*0.5411*	**0.9050**	***0.6733***	0.8827	*0.5344*	0.8837	*0.5351*

4.1.2 Interprétabilité

Sur le plan de la facilité d'interprétation des résultats, le meilleur encodage est l'encodage "OneHot". En effet, en créant une variable par AA présent à chaque position, les résultats des processus de sélection de variables qui sont le centre d'intérêt de cette étude, retourneront pas uniquement des positions intéressantes mais également quel AA spécifique est utilisé pour la discrimination. On obtient donc les mutations exactes pour lesquelles il faut tester les différences de proportions, ce qui permet de cibler encore plus les tests statistiques.

C'est pour cela que l'encodage OneHot a été sélectionné pour la suite de cette étude.

4.2 Performance des classificateurs

4.2.1 Avec DRMs

Tout d'abord les classificateurs ont été entraînés sur les jeux de données complets (donc sans éliminer les SDRMs) pour avoir une idée de la performance atteignable dans des conditions

idéales. Cette étape permettra également de valider la méthode de détection des DRMs.

Table 4.2 – Performance des classificateurs (avec DRMs)

		Afrique				Europe		
		Tous sous-Types	CRF02_AG	C	A	Tous sous-Types	B	C
Forêt aléatoire	rappel	0.8148	0.7517	0.6183	0.8250	0.4368	0.4334	0.4423
	précision	0.9520	0.9928	0.9650	0.9363	0.8294	0.8506	0.7637
	précision totale	0.9050	0.8592	0.7991	0.9071	0.6733	0.6784	0.6527
Régression Logistique	rappel	0.8208	0.8497	0.5842	0.8071	0.3321	0.3562	0.3345
	précision	0.9562	0.9562	0.9220	0.9534	0.8879	0.8770	0.8347
	précision totale	0.9090	0.8951	0.9288	0.9067	0.8259	0.8404	0.7963

Lorsque les DRMs sont présentes, les classificateurs arrivent, de manière générale, à bien discriminer les deux classes. Il est cependant possible de voir que les classificateurs ont de meilleures performances sur les données Africaines que sur les données Européennes, cela peut être expliqué par le fait que le pourcentage de séquence *treated* présentant des SDRMs est beaucoup plus important en Afrique qu'en Europe. Il est également intéressant de noter qu'au sein des séquences Européennes (Table 4.2), les sous-types C sont plus durs à classifier que les sous-types B. Cela pourrait être en partie expliqué par le fait que le sous-type C est le sous-type majoritaire en Afrique. De ce fait les séquences de ce sous-type en Europe sont largement issues de l'immigration et n'ont pas forcément reçu les mêmes traitements, ou n'ont pas la même couverture médicale que les séquences de sous-type B, qui est le sous-type majoritaire du monde occidental.

4.2.2 Sans DRMs

Lorsque les DRMs sont retirées du jeu de données les performances des classificateurs chutent, quelque soit le sous-type ou la provenance des données.

Table 4.3 – Performance des classificateurs (sans DRMs)

		Afrique				Europe		
		Tous sous-Types	CRF02_AG	C	A	Tous sous-Types	B	C
Forêt aléatoire	rappel	0.1681	0.6013	0.5617	0.1850	0.6025	0.5308	0.4658
	précision	0.8159	0.7108	0.6789	0.7167	0.5845	0.5967	0.5562
	précision totale	0.6379	0.6434	0.6408	0.6709	0.5871	0.5858	0.5469
Régression Logistique	rappel	0.6057	0.6651	0.4933	0.5767	0.5610	0.5884	0.5397
	précision	0.7584	0.7467	0.5139	0.6985	0.5771	0.5850	0.5706
	précision totale	0.7529	0.6888	0.5242	0.7289	0.5749	0.5856	0.5667

Comme on peut le voir dans la table 4.3, les classificateurs gardent quand même un certain pouvoir discriminatoire. Par exemple pour la forme recombinante 2 AG des données Africaines, les classificateurs classifient correctement entre 60 et 66% des séquences *treated* en gardant un taux de faux positifs de moins de 30%. Les classificateurs ont donc réussi à trouver des caractéristiques permettant en partie de discriminer les séquences *treated*. Il est important de noter que parmi les séquences *naive* il peut y avoir des DRMs, donc une partie des faux positifs pourrait être des séquences *naives* résistantes et donc en fait étant "mal étiquetées" comme *naive*.

En l'absence de DRMs, il est possible que les mutations identifiées grâce aux caractéristiques des modèles de classification, soient des DRMs non connues. Il est également important de noter

que, vu qu'il y a un classificateur entraîné par sous-type, l'effet confondant dû à ce dernier est éliminé.

4.3 Mutations identifiées

Bien qu'il soit important que les classificateurs performent bien lors de la tâche de classification sur laquelle ils sont entraînés, le point principal de cette recherche est d'identifier des potentielles DRMs.

4.3.1 Validation de la méthode

4.3.1.1 Forêts aléatoires

Les caractéristiques des modèles entraînés sur les données contenant des DRMs permettent d'identifier les variables (*i.e* les mutations) sur lesquelles le modèle se base pour effectuer sa classification. Ces mutations sont identifiées avec les importances de variables dans le cadre des forêts aléatoires, et avec les pondérations de variables dans le cas de la régression logistique.

Les résultats sur les données Africaines, présentés dans la Figure 4.1 sont très prometteurs. Il est possible de voir (Figure 4.1a) d'une part que les variables sélectionnées lors de l'entraînement répété du modèle, le sont de manière assez stable : la fréquence de sélection de la variable dans les 20 variables les plus importantes est élevée. Certaines variables : 184_V, 184_M, 103_K et 103_N ont des fréquences égales à 1, elles sont donc tout le temps sélectionnées en tant que variables importantes. Ce résultat est vérifié dans la plupart des cas : lorsque l'on entraîne les classificateurs sur chaque sous-type séparément, les fréquences de sélection de variables sont assez hautes avec au moins un sous-groupe de quelques variables qui ont des fréquences de sélection comprises entre 1 et 0.8.

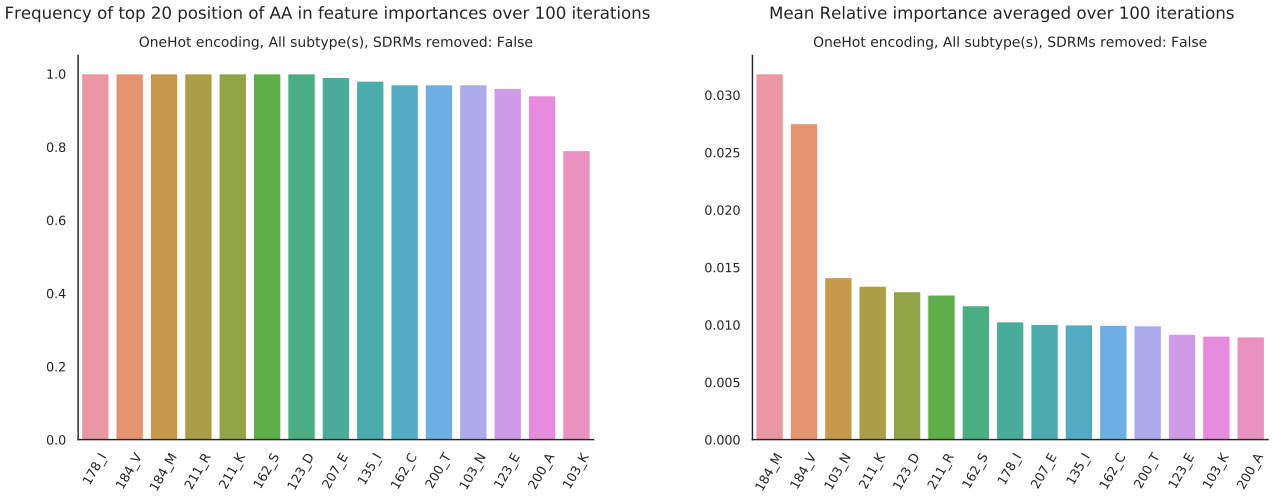
Lorsque l'on regarde les importances moyennes (Figure 4.1b), il est possible d'identifier deux couples de variables qui ont les plus grandes importances : 184_V et 184_M, ainsi que 103_K et 103_N : les variables ayant les plus hautes fréquences de sélection. Les positions 184 et 103 sont le siège de deux des SDRMs les plus prévalentes pour la protéine RT : $M184V$ et $K103N$. Les variables les plus importantes sont donc l'AA d'origine et l'AA muté pour ces deux SDRMs. De plus dans les autres variables sélectionnées, seules 249_$-$ (une mutation de délétion) et 211_Y ne sont pas connues puisque $Y181C$, $T215Y$, $G190A$ et $K70R$ sont des SDRMs pour RT (cf. annexe A), et la position 101 est le siège de plusieurs DRMs avec comme AA d'origine K. On observe, lorsque l'on entraîne les classificateurs sous-type par sous-type, similaires a ceux-ci. Les résultats sous-type par sous-type sont inclus dans l'annexe D.

(a) Fréquence de sélection des variables dans les 20 plus importantes

(b) Importance moyenne des variables

Figure 4.1 – Importances des variables sur 100 itérations de stabilité. Données Africaines, tous sous-types

Pour le cas des données Européennes, les résultats sont moins tranchants. En effet, dans la Figure 4.2b il est possible de voir que les variables 184_M et et184_V sont bien identifiées et se détachent clairement du lot. Les variables 103_K et 103_N sont aussi présentes mais elles sont au même niveau que les autres variables restantes. Aucune de ces dernières n'est un site impliqué dans des DRMs identifiées. Il en est de même pour les résultats sous-type par sous-type (cf. annexe D), où les mutations $M184V$ et $K103N$ se dégagent mais les autres positions identifiées ne sont pas le siège de DRMs.



(a) Fréquence de sélection des variables dans les 20 plus importantes

(b) Importance moyenne des variables

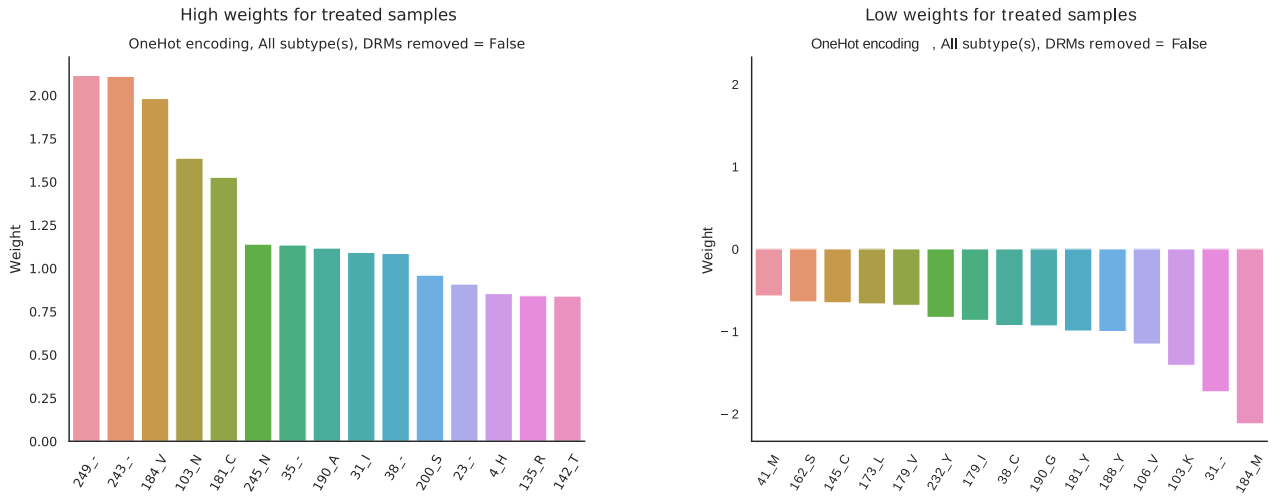
Figure 4.2 – Importances des variables sur 100 itérations de stabilité. Données Européennes, tous sous-types

4.3.1.2 Régression logistique

Les variables qui ont des poids élevés, dans les modèles de régression logistiques, sont celles qui favorisent la classification de la séquence dans la classe *treated*. Les variables ayant de pondérations négatives favorisent la classification dans la classe *naive*. Il est possible de voir en Figure 4.3a, pour les données Africaines, que les variables 184_V et 103_N sont de nouveau dans les pondérations les plus hautes. Il est aussi intéressant de remarquer que pour ces positions seul l'AA muté est sélectionné, leur pendants originaux sont présents dans les pondérations les plus basses, comme il est possible de le voir en Figure 4.3b.

Dans les poids les plus hauts on retrouve également les AA mutés des SDRMs $Y181C$ et $G190A$. Lorsque les classificateurs sont entraînés sous-type par sous-type les mêmes SDRMs qui sont détectées, avec détection additionnelle des mutations $V106A$ et $V179E$, ainsi que l'AA originel E à la position 203 impliquée dans plusieurs DRMs pour la forme recombinante CRF02_AG (Figure D.6).

On peut finalement remarquer que beaucoup des positions et variables sélectionnées comme importantes ne sont pas impliquées dans des SDRMs ni même des DRMs. En particulier les variables 249− et 243−, qui indiquent des mutations de délétion en fin de chaîne. Ceci pourrait être de nouvelles DRMs, ou alors c'est un signe de problèmes lors du séquençage des virus exposés à un traitement.

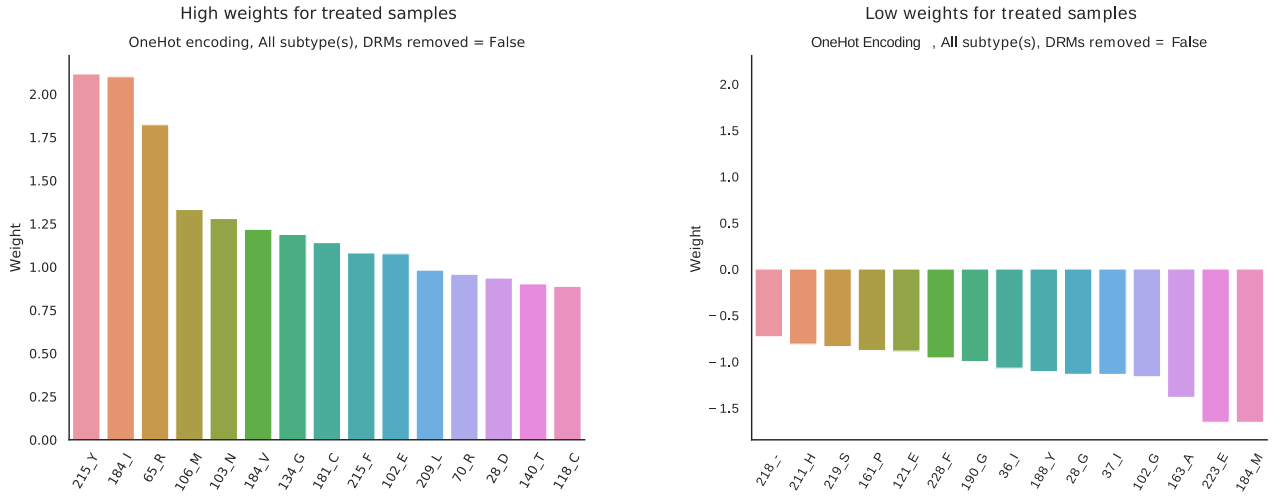


(a) Pondérations hautes

(b) Pondérations basses

Figure 4.3 – Pondérations des variables, moyenne sur 20 itérations. Données Africaines. Tous sous-types

Dans le cas des données Européennes, les résultats sont similaires. De nombreuses DRMs sont détectées dans les pondérations hautes : $T215Y$, $M184I$, $M184V$, $K65R$, $K103N$, $V106M$, $T215F$ et $K70R$, Figure 4.4a. Avec une fois de plus les variables correspondants aux AAs originels sont pondérées négativement (Figure 4.4b). Les résultats sous-type par sous-type sont similaire, avec les mêmes mutations étant détectées dans l'ensemble. Dans le cas du sous-type B, la SDRM $T215Y$ domine toutes les autres (Figure D.29)



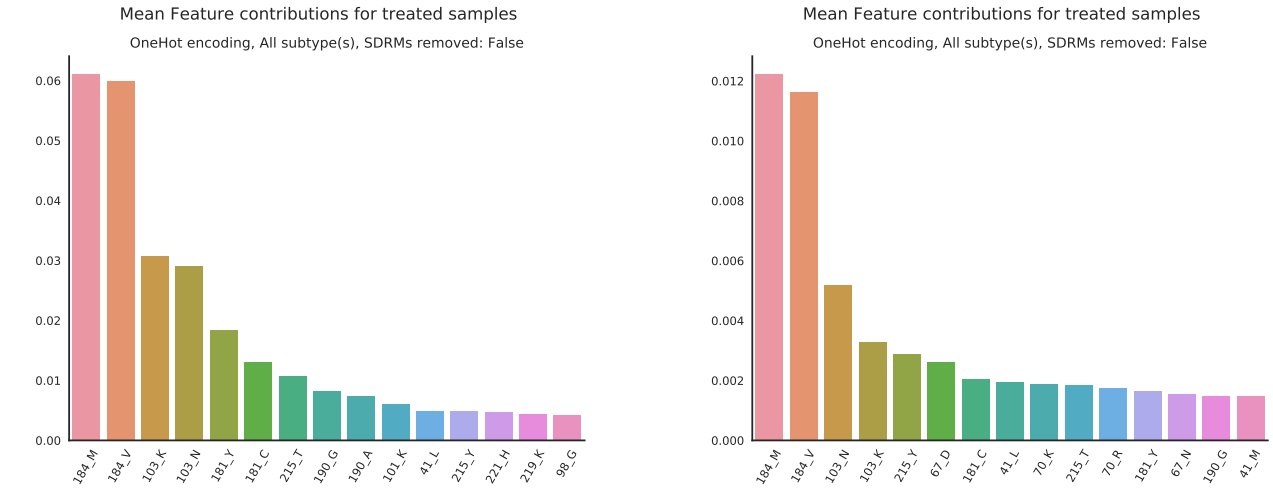
(a) Pondérations hautes

(b) Pondérations basses

Figure 4.4 – Pondérations des variables, moyenne sur 20 itérations. Données Européennes. Tous sous-types

4.3.1.3 Contributions de variables

Les résultats des contributions de variables sont très similaires à ceux des forêts aléatoires. Encore une fois, les positions impliquées dans des SDRMs importantes ont les contributions les plus hautes (Figure 4.5). On retrouve encore les 2 AAs des SDRMs suivantes : $M184V$, $K103N$, $Y181C$, $T215Y$, $G190A$, ainsi que les variables 101_K, 41_L, 221_H, 219_K et 98_G toutes impliquées dans des DRMs ou SDRMs.



(a) Données Africaines

(b) Données Européennes

Figure 4.5 – Hautes contributions, moyennées sur tous les individus. Tous sous-types

Avec toutes ces méthodes, les SDRMs les plus prévalentes sont présentes parmi les variables identifiées et dans la plupart des cas avec l'importance la plus haute. Ceci est encourageant quant à la validité de la méthode.

Table 4.4 – Mutations identifiées par sous-type par les importances de variables (clair), les poids de régressions logistiques (moyen) et les contributions de variables (foncé)

	Afrique									Europe				
	CRF02_AG			A			C			B			C	
K11T				✓	✓	✓								
K20R							✓	✓	✓					
E28K		✓			✓						✓			
I31L		✓	✓	✓	✓	✓					✓		✓	
E36A							✓		✓			✓		✓
T39E														
S48T							✓		✓					✓
D123G							✓		✓			✓		
I135L								✓	✓				✓	
I135T				✓					✓	✓				
S162C										✓	✓			✓
D177E										✓	✓	✓		✓
T200A										✓	✓			
Q207E							✓	✓		✓		✓		
R211K	✓						✓			✓	✓	✓		
P243−		✓	✓		✓	✓								✓
P247−			✓	✓		✓					✓		✓	
E248−	✓		✓	✓		✓								
K249−	✓	✓	✓		✓									

4.3.2 Nouvelles mutations

L'identification de SDRMs dans les jeux de données est un bon départ, mais n'est pas l'objectif principal de cette étude.

4.3.2.1 Identification des mutations

Dans cette partie, pour éliminer l'effet confondant du sous-type, les classificateurs entraînés le sont sous-type par sous-type. L'apparition de DRMs ne devrait pas dépendre de ce dernier, et les mutations véritablement sélectionnées par le traitement devraient être présentes quelque soit le sous-type. Dans le cas de la régression logistique, seules les pondérations hautes sont étudiées.

Il est possible de voir dans la Table 4.4 les mutations détectées par les différentes méthodes sur les deux jeux de données. Ce tableau a été construit en rassemblant les variables sélectionnées dans le plus importantes pour au moins deux triplets *(origine ; sous-type ; méthode)* différents. Ces variables les plus importantes sont disponibles par origine et par sous-type dans l'Annexe D. Ont ensuite été éliminés les variables correspondant aux AAs d'origine pour la position considérée.

Ces mutations seront celles pour lesquelles la différence de proportions sera testée. Parmi celles-ci les plus intéressantes ont été sélectionnées par les deux types de classificateurs dans plusieurs sous-types différents comme $I31L$. Les mutations $P243-$, $P247-$, $E248-$ et $K249-$ sont toutes des délétions en fin de chaîne, il se peut qu'elles soient dues au séquençage qui aurait pu produire des séquences de différentes tailles ce qui aurait fait apparaître ces délétions lors de l'alignement des séquences, ou alors l'apparition d'un codon STOP avant dans la chaîne. Toutes ces mutations sont testées sur le jeu de données, cela fait 14 mutations à tester.On applique la correction de Bonferroni pour garder un risque α global de 5%, donc pour chaque test individuel : $\alpha_i = \frac{0.05}{19} \simeq 2.6316 \cdot 10^{-3}$.

Table 4.5 – p-valeurs des tests de différence de proportions

	Afrique		Europe	
$K11T$	0.0322		0.4063	
$K20R$	$4.2525 \cdot 10^{-12}$	✓	0.0135	
$E28K$	$5.8985 \cdot 10^{-23}$	✓	$2.5518 \cdot 10^{-08}$	✓
$I31L$	$1.6123 \cdot 10^{-23}$	✓	$2.9600 \cdot 10^{-25}$	✓
$E36A$	$1.0677 \cdot 10^{-34}$	✓	0.0108	
$T39E$	$2.1876 \cdot 10^{-43}$	✓	$3.5114 \cdot 10^{-21}$	✓
$S48T$	$5.1240 \cdot 10^{-37}$	✓	$2.2353 \cdot 10^{-10}$	✓
$D123G$	$5.1108 \cdot 10^{-08}$	✓	0.2189	
$I135L$	$4.2273 \cdot 10^{-08}$	✓	$2.1722 \cdot 10^{-11}$	✓
$I135T$	$2.7424 \cdot 10^{-11}$	✓	$1.0867 \cdot 10^{-11}$	✓
$S162C$	0.0116		$6.3833 \cdot 10^{-17}$	✓
$D177E$	0.2145		0.2263	
$T200A$	$1.8570 \cdot 10^{-07}$	✓	$7.4076 \cdot 10^{-10}$	✓
$Q207E$	$8.6886 \cdot 10^{-06}$	✓	$6.7892 \cdot 10^{-19}$	✓
$R211K$	$1.7329 \cdot 10^{-06}$	✓	0.1315	
$P243-$	$1.3645 \cdot 10^{-30}$	✓	$1.6052 \cdot 10^{-05}$	✓
$P247-$	$6.5374 \cdot 10^{-44}$	✓	$9.9623 \cdot 10^{-11}$	✓
$E248-$	$8.8136 \cdot 10^{-61}$	✓	$9.2259 \cdot 10^{-09}$	✓
$K249-$	$7.4597 \cdot 10^{-70}$	✓	$9.2222 \cdot 10^{-09}$	✓

On peut voir dans la Table 4.5 la plupart des tests de différence de proportions entre les séquences *naive* et *treated* sont significatifs. De plus les p-valeurs sont extrêmement petites, avec souvent plusieurs ordres de grandeur de moins que le risque corrigé α_i.

Il est intéressant de noter que la mutation $I31L$ a été identifiés comme ayant une corrélation avec des phénomènes de résistance aux inhibiteurs de RT dans au moins deux études (Parkin et al., 2006; Shahriar et al., 2009) mais ne figure sur aucune liste d'experts. Une étude de cas sur une patiente Italienne infectée par une souche multi-résistante identifie les mutations $E28K$, $I135T$, $R211K$ et $Q207E$ (Pinnetti et al., 2010) comme ayant un lien avec des phénomènes de résistance aux traitements.

Chapitre 5

Discussion

5.1 Comparaison à l'état de l'art

Grâce à cette méthode, le nombre de tests multiples est réduit au nombre de 19. Dans le cadre d'un étude statistique utilisées dans l'état de l'art, toutes les mutations présentes dans les séquences exposées à un traitement ayant une prévalence supérieure à 0.5% (Shafer, 2006). Dans le cas présent, si on considère le jeu de données Européennes il y a 488 mutations avec une prévalence supérieure à 0.5%, et 404 avec une prévalence supérieure à 1%. Si chacune de ces mutations est testée, avec la correction de Bonferroni, la puissance statistique est inférieure d'un ordre de grandeur à celle des tests ciblés par apprentissage automatique.

5.2 Pistes à suivre

À la suite de cette étude, il y a plusieurs pistes de continuation possibles.

5.2.1 Étude de l'épistasie

L'épistasie, dans le cadre de cette étude, désigne l'interaction entre deux DRMs. En effet une seule mutation ne suffit pas forcement, à elle seule, à induire une résistance. Mais une action conjointe de plusieurs mutations peut engendrer des résistances. Ces effets d'interaction sont très difficile à étudier avec les méthodes décrites en 1.4.

Les Forêts aléatoires, ainsi que les contributions jointes pourraient faciliter la tâche. En effet en augmentant progressivement la profondeur des arbres de décision, il est possible d'observer le gain de performance à chaque augmentation de profondeur. S'il y a un gain de performance lorsqu'une variable de plus est utilisée pour la décision, c'est que l'interaction entre la variable rajoutée et les précédentes est informative. En calculant les contributions jointes, il devient possible de dire quelles variables sont les plus importantes, conjointement, à la décision du classificateur.

5.2.2 Étude phylogénétique

La phylogénie est la science qui étudie l'histoire évolutive et les relations génétiques entre les êtres vivants. Grâce à des méthodes d'inférence phylogénétiques il est possible, à partir de séquences d'ADN de plusieurs individus, de reconstruire un arbre phylogénétique retraçant l'histoire évolutive de ces séquences : lesquelles sont les plus proches génétiquement parlant ? Quelles séquences ont un ancêtre commun ? etc... Au sein de ces arbres les longueurs de branches désignent la distance génétique entre les deux nœuds.

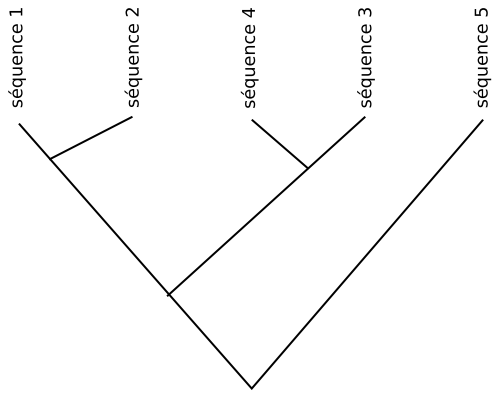


Figure 5.1 – Exemple d'arbre phylogénétique

Dans l'exemple de la figure 5.1 on peut voir l'ancêtre commun à toutes les séquences : c'est le nœud à la racine de l'arbre. La séquence 5 est la plus différente des autres, alors que les séquences 1 et 2 sont proches génétiquement et descende d'un ancêtre commun récent.

Dans le cadre de cette étude, les séquences peuvent être utilisées pour construire un arbre phylogénétique. De cette manière il sera possible de voir la distribution dans l'arbre des séquences correctement classées comme *treated* de manière stable par les classificateurs. Si toutes ces séquences sont très proches dans l'arbre, il est alors possible que les classificateurs aient appris à reconnaître les séquences issues d'un ancêtre en particulier plutôt que les séquences *treated* en général. Cette étude complémentaire permettrais alors d'étudier l'influence de l'évolution comme facteur confondant.

5.2.3 Prise en compte du traitement

Les DRMs apparaissent lorsque le virus est soumis à une pression thérapeutique sélective. De ce fait, le type de DRMs sélectionné dans un virus est intrinsèquement lié au type de traitement administré. Il serait donc logique d'étudier les résultats de cette étude en fonction des différents traitements.

Ceci a en partie été fait lors de l'étude du jeu de données Africain puisque tous les virus dont les séquences constituent ce jeu de données ont été exposés au même traitement(Villabona-Arenas et al., 2016). Cependant, n'ayant d'information sur le type de traitement suivi que pour une petite fraction des séquences Européennes, il n'a pas été possible d'identifier des mutations en fonction du traitement pour ces séquences. Il faudrait donc trouver d'autres jeux de données où le traitement suivi est connu pour pouvoir identifier des mutations en fonction des composés médicamenteux auxquels les virus ont été exposés.

5.2.4 Étude d'autres protéines

Dans le cadre de ce projet, l'étude a été restreinte à la protéine RT. Cependant il existe des DRMs et SDRMs sur les autres protéines codées par le gène *pol* : PR et IN. Il serait donc nécessaire de trouver des séquences pour ces protéines ainsi que leur statut de thérapeutique.

5.2.5 Étude *ex silico*

Cette étude permet d'identifier des DRMs potentielles. De ce fait les mutations détectées par cette étude ne peuvent pas être considérées comme conférant une résistance à un ou plusieurs

composés médicamenteux sur la seule base de cette étude. Des études plus concrètes *in vitro* et potentiellement *in vivo* seront nécessaires pour valider ou non le statut de DRM. Ces études devront étudier la susceptibilité aux traitements de virus possédant une ou plusieurs des mutations identifiées, ou encore les mécanismes d'action moléculaires dans lesquels sont impliqués ces mutations.

5.3 Autres approches

Dans un dernier temps il est possible d'approcher la question de la détection de DRMs avec d'autres méthodes que l'apprentissage automatique. Au sein de l'unité de bio-informatique évolutive, dans laquelle cette étude a été réalisée, un autre projet de détection de DRMs a été mis en place. Cette méthode alternative utilise des phylogénies pour essayer de détecter des convergences évolutives. En comparant l'évolution réelle de chaque position dans un grand alignement de séquences au cours du temps, à un modèle d'évolution neutre, il est possible de détecter des mutations convergentes. Cette convergence se traduit par un taux d'apparition indépendant pour chaque mutation supérieur à un taux d'apparition simulé sous un modèle d'évolution neutre. Les SDRMs sont relativement bien détectées avec cette méthode. Les mutations $D121H$, $K122E$, $I135T$, $Q207E$ et $R211K$ sont également détectées avec cette méthode.

Conclusion

L'utilisation de techniques d'apprentissage automatique, a permis de cibler un nombre extrêmement faible de mutations d'intérêt pour une étude statistique. Cette diminution du nombre de tests multiples permet d'augmenter considérablement la puissance statistique de chacun de ces tests par rapport aux techniques utilisées habituellement dans les problèmes de détection de DRMs. Cette réduction du nombre de tests permet de détecter de nouvelles mutations, qui on été auparavant faiblement corrélées à des phénomènes de résistance, avec de fortes significativités.

Ces nouvelles mutations nécessiteront une étude plus minutieuse, et des étapes de validation avant de pouvoir être considérées comme des DRMs, mais ces résultats sont prometteurs, surtout au vu des mutations détectées lorsque les jeux de données contiennent les DRMs. La convergence des résultats obtenus lors de cette étude avec ceux de méthodes alternatives au sein de l'unité de recherche est rassurante et encourage la continuation de ce projet.

Ce projet sera d'ailleurs continué au sien de l'équipe de bio-informatique évolutive, en explorant plusieurs des pistes mentionnées comme perspectives prometteuses.

Bibliographie

Bennett, D. E., R. J. Camacho, D. Otelea, D. R. Kuritzkes, H. Fleury, M. Kiuchi, W. Heneine, R. Kantor, M. R. Jordan, J. M. Schapiro, et al.
2009. Drug resistance mutations for surveillance of transmitted hiv-1 drug-resistance : 2009 update. *PloS one*, 4(3) :e4724.

Bergstra, J. and Y. Bengio
2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb) :281–305.

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

Carey, G.
2013. Quantitative methods in neuroscience. draft at http://psych.colorado.edu/~carey/qmin/QMIN_2013_03_17.pdf.

Castro, H., D. Pillay, P. Cane, D. Asboe, V. Cambiano, A. Phillips, D. T. Dunn, U. C. G. on HIV Drug Resistance, C. Aitken, D. Asboe, et al.
2013. Persistence of hiv-1 transmitted drug resistance mutations. *The Journal of infectious diseases*, 208(9) :1459–1463.

Clark, S., C. Calef, and J. W. Mellors
2007. Mutations in retroviral genes associated with drug resistance. *HIV sequence compendium*, Pp. 58–158.

Cui, Z., W. Chen, Y. He, and Y. Chen
2015. Optimal action extraction for random forests and boosted trees. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Pp. 179–188. ACM.

Deng, H.
2014. Interpreting tree ensembles with intrees. *arXiv preprint arXiv :1408.5456*.

Fisher, B. D., R. A. Harvey, and P. C. Champe
2007. *Lippincott's Illustrated Reviews : Microbiology (Lippincott's Illustrated Reviews Series)*, 2 edition. Lippincott Williams & Wilkins.

Gilbert, P. B., I. W. McKeague, G. Eisen, C. Mullins, A. Guéye-NDiaye, S. Mboup, and P. J. Kanki
2003. Comparison of hiv-1 and hiv-2 infectivity from a prospective cohort study in senegal. *Statistics in medicine*, 22(4) :573–593.

Hara, S. and K. Hayashi
2016. Making tree ensembles interpretable. *arXiv preprint arXiv :1606.05390*.

Hunter, J. D.
2007. Matplotlib : A 2d graphics environment. *Computing In Science & Engineering*, 9(3) :90–95.

Jones, E., T. Oliphant, P. Peterson, et al.
2001–. SciPy : Open source scientific tools for Python.

Kawashima, S., P. Pokarowski, M. Pokarowska, A. Kolinski, T. Katayama, and M. Kanehisa
2007. Aaindex : amino acid index database, progress report 2008. *Nucleic acids research*, 36(suppl_1) :D202–D205.

Köster, J. and S. Rahmann
2012. Snakemake—a scalable bioinformatics workflow engine. *Bioinformatics*, 28(19) :2520–2522.

Kremer, S. and H. Lac
2009. Method, system and computer program product for levinthal process induction from known structure using machine learning.

Li, Z.-C., X.-B. Zhou, Z. Dai, and X.-Y. Zou
2009. Prediction of protein structural classes by chou's pseudo amino acid composition : approached using continuous wavelet transform and principal component analysis. *Amino acids*, 37(2) :415.

Liu, T. F. and R. W. Shafer
2006. Web resources for hiv type 1 genotypic-resistance test interpretation. *Clinical infectious diseases*, 42(11) :1608–1618.

Maetschke, S., M. Towsey, and M. Boden
2005. Blomap : an encoding of amino acids which improves signal peptide cleavage site prediction. In *Proceedings of the 3rd Asia-Pacific bioinformatics conference*, Pp. 141–150. World Scientific.

Masquelier, B., C. Tamalet, B. Montès, D. Descamps, G. Peytavin, L. Bocket, M. Wirden, J. Izopet, V. Schneider, V. Ferré, et al.
2004. Genotypic determinants of the virological response to tenofovir disoproxil fumarate in nucleoside reverse transcriptase inhibitor-experienced patients. *Antiviral therapy*, 9(3) :315–324.

McKinney, W. et al.
2010. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, Pp. 51–56. Austin, TX.

Molina, J.-M., A.-G. Marcelin, J. Pavie, L. Heripret, C. M. De Boever, M. Troccaz, G. Leleu, and A.-. J. S. Team
2005. Didanosine in hiv-1–infected patients experiencing failure of antiretroviral therapy : a randomized placebo-controlled trial. *The Journal of infectious diseases*, 191(6) :840–847.

Mourad, R., F. Chevennet, D. T. Dunn, E. Fearnhill, V. Delpech, D. Asboe, O. Gascuel, and S. Hue
2015. A phylotype-based analysis highlights the role of drug-naive hiv-positive individuals in the transmission of antiretroviral resistance in the uk. *Aids*, 29(15) :1917–1925.

Palczewska, A., J. Palczewski, R. M. Robinson, and D. Neagu
2014. Interpreting random forest classification models using a feature contribution method. In *Integration of reusable systems*, Pp. 193–218. Springer.

Parkin, N. T., S. Gupta, C. Chappey, and C. J. Petropoulos
2006. The k101p and k103r/v179d mutations in human immunodeficiency virus type 1 reverse transcriptase confer resistance to nonnucleoside reverse transcriptase inhibitors. *Antimicrobial agents and chemotherapy*, 50(1) :351–354.

Pattery, T., Y. Verlinden, H. De Wolf, D. Nauwelaers, K. Van Baelen, M. Van Houtte, P. Mc Kenna, and J. Villacian
2012. Development and performance of conventional hiv-1 phenotyping (antivirogram®) and genotype-based calculated phenotyping assay (virco® type hiv-1) on protease and reverse transcriptase genes to evaluate drug resistance. *Intervirology*, 55(2) :138–146.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al.
2011. Scikit-learn : Machine learning in python. *Journal of machine learning research*, 12(Oct) :2825–2830.

Pineda-Peña, A.-C., N. R. Faria, S. Imbrechts, P. Libin, A. B. Abecasis, K. Deforche, A. Gómez-López, R. J. Camacho, T. de Oliveira, and A.-M. Vandamme
2013. Automated subtyping of hiv-1 genetic sequences for clinical and surveillance purposes : performance evaluation of the new rega version 3 and seven other tools. *Infection, Genetics and Evolution*, 19 :337–348.

Pinnetti, C., E. Tamburrini, E. Ragazzoni, A. De Luca, and P. Navarra
2010. Case report decreased plasma levels of darunavir/ritonavir in a vertically infected pregnant woman carrying multiclass-resistant hiv type-1. *Antiviral therapy*, 15 :127–129.

Qi, Y.
2012. Random forest for bioinformatics. In *Ensemble machine learning*, Pp. 307–323. Springer.

Rhee, S.-Y., M. J. Gonzales, R. Kantor, B. J. Betts, J. Ravela, and R. W. Shafer
2003. Human immunodeficiency virus reverse transcriptase and protease sequence database. *Nucleic acids research*, 31(1) :298–303.

Seabold, S. and J. Perktold
2010. Statsmodels : Econometric and statistical modeling with python. In *9th Python in Science Conference*.

Shafer, R. W.
2006. Rationale and uses of a public hiv drug-resistance database. *The Journal of infectious diseases*, 194(Supplement_1) :S51–S58.

Shahriar, R., S.-Y. Rhee, T. F. Liu, W. J. Fessel, A. Scarsella, W. Towner, S. P. Holmes, A. R. Zolopa, and R. W. Shafer
2009. Nonpolymorphic human immunodeficiency virus type 1 protease and reverse transcriptase treatment-selected mutations. *Antimicrobial agents and chemotherapy*, 53(11) :4869–4878.

Sham, P. C. and S. M. Purcell
2014. Statistical power and significance testing in large-scale genetic studies. *Nature Reviews Genetics*, 15(5) :335.

Tan, H. F., G. Hooker, and M. T. Wells
2016. Tree space prototypes : Another look at making tree ensembles interpretable. *arXiv preprint arXiv :1611.07115*.

Tang, M. W., T. F. Liu, and R. W. Shafer
2012. The hivdb system for hiv-1 genotypic resistance interpretation. *Intervirology*, 55(2) :98–101.

Taylor, W. R.
1986. The classification of amino acid conservation. *Journal of theoretical Biology*, 119(2) :205–218.

Trevor, H., T. Robert, and F. Jerome
2009. *The elements of statistical learning : data mining, inference, and prediction*, 2 edition. New York, NY : Springer.

Villabona-Arenas, C. J., N. Vidal, E. Guichet, L. Serrano, E. Delaporte, O. Gascuel, and M. Peeters
2016. In-depth analysis of hiv-1 drug resistance mutations in hiv-infected individuals failing first-line regimens in west and central africa. *Aids*, 30(17) :2577–2589.

Villacian, J.
2009. Basics of hiv resistance basics of hiv resistance and testing and testing. TREAT Asia TREAT Asia Annual Network meeting Network meeting. available at https://www.amfar.org/uploadedFiles/Around_the_World/TreatAsia/Meetings/Villacian.pdf.

Waskom, M., O. Botvinnik, P. Hobson, J. B. Cole, Y. Halchenko, S. Hoyer, A. Miles, T. Augspurger, T. Yarkoni, T. Megies, L. P. Coelho, D. Wehner, cynddl, E. Ziegler, diego0020, Y. V. Zaytsev, T. Hoppe, S. Seabold, P. Cloud, M. Koskinen, K. Meyer, A. Qalieh, and D. Allan
. seaborn : v0.5.0 (november 2014).

Weisstein, E. W.
2004. Bonferroni correction. Wolfram Research, Inc. http://mathworld.wolfram.com/BonferroniCorrection.html.

Wensing, A. M., V. Calvez, H. F. Günthard, V. A. Johnson, R. Paredes, D. Pillay, R. W. Shafer, and D. D. Richman
2017. 2017 update of the drug resistance mutations in hiv-1. *Topics in antiviral medicine*, 24(4) :132–133.

Zamani, M. and S. C. Kremer
2011. Amino acid encoding schemes for machine learning methods. In *Bioinformatics and Biomedicine Workshops (BIBMW), 2011 IEEE International Conference on*, Pp. 327–333. IEEE.

Annexe A

Tableaux de DRM

Ci-dessous sont les listes des DRMs et SDRMs maintenues par HIVdb à l'université de Stanford, accessibles au lien suivant : https://hivdb.stanford.edu/pages/surveillance.html

La colonne *Pos* désigne la position de l'acide aminé concerné dans la séquence. Dans le cas des Nucleoside Reverse Transcriptase Inhibitor (NRTI) et des Non-Nucleoside Reverse Transcriptase Inhibitor (NNRTI) le gène concerné est la RT. La colonne *WT (Wild Type)* désigne l'AA de la séquence consensus pour la position donnée. La colonne *Mut* donne la mutation de substitution : on donne l'AA remplaçant le *WT*, d'insertion : *i*, ou de délétion : *d*. Finalement la colonne SDRM indique si la mutation est considérée comme SDRM ou non.

Table A.1 – Liste des DRMs affectant les NRTI

Pos	WT	Mut	SDRM	Pos	WT	Mut	SDRM
41	M	L	✓	75	V	I	
62	A	V		75	V	M	✓
65	K	E		75	V	S	✓
65	K	N		75	V	T	✓
65	K	R	✓	77	F	L	✓
67	D	E	✓	115	Y	F	✓
67	D	G	✓	116	F	Y	✓
67	D	H		151	Q	L	
67	D	N	✓	151	Q	M	✓
67	D	S		184	M	I	✓
67	D	T		184	M	V	✓
67	D	d		210	L	W	✓
68	S	d		215	T	A	
69	T	D	✓	215	T	C	✓
69	T	G		215	T	D	✓
69	T	d		215	T	E	✓
69	T	i	✓	215	T	F	✓
70	K	E	✓	215	T	I	✓
70	K	G		215	T	L	
70	K	N		215	T	N	
70	K	Q		215	T	S	✓
70	K	R	✓	215	T	V	✓
70	K	S		215	T	Y	✓
70	K	T		219	K	E	✓
70	K	d		219	K	N	✓
74	L	I	✓	219	K	Q	✓
74	L	V	✓	219	K	R	✓
75	V	A	✓	219	K	W	

Table A.2 – Liste des DRMs affectant les NNRTI

Pos	WT	Mut	SDRM	Pos	WT	Mut	SDRM
98	A	G		181	Y	I	✓
100	L	I	✓	181	Y	S	
100	L	V		181	Y	V	✓
101	K	E	✓	188	Y	C	✓
101	K	H		188	Y	F	
101	K	P	✓	188	Y	H	✓
103	K	H		188	Y	L	✓
103	K	N	✓	190	G	A	✓
103	K	S	✓	190	G	C	
103	K	T		190	G	E	✓
106	V	A	✓	190	G	Q	
106	V	M	✓	190	G	S	✓
108	V	I		190	G	T	
138	E	A		190	G	V	
138	E	G		221	H	Y	
138	E	K		225	P	H	✓
138	E	Q		227	F	C	
138	E	R		227	F	L	
179	V	D		230	M	I	
179	V	E		230	M	L	✓
179	V	F	✓	238	K	N	
179	V	L		238	K	T	
181	Y	C	✓	318	Y	F	
181	Y	F		348	N	I	
181	Y	G					

Annexe B

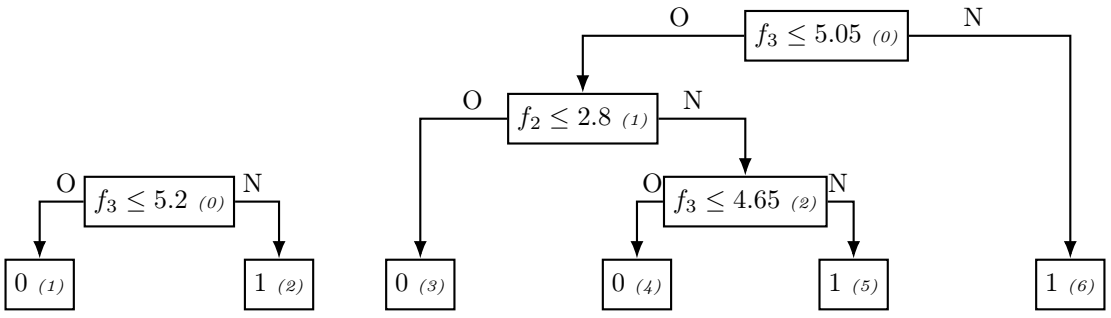
Exemple de calcul de contribution de variables

B.1 Données

Cet exemple est tiré du papier de Palczewska et al. (Palczewska et al., 2014), il détaillera le calcul de contributions seules et jointes pour un petit jeu de données. Ce jeu est un sous ensemble de 10 exemples du jeu de données Iris de UCI. On ne gardera en variable que *Sepal Length*, *Sepal Width* et *Petal Length*, f_1, f_2 et f_3 respectivement. Les classes 0 et 1 correspondent à *iris versicolor* et *virginica*. La forêt aléatoire est composée de deux arbres de décisions, représentés à la figure B.1. Les bootstraps ayant servi à élaborer l'arbre de décision sont indiqués sous chaque arbre.

TABLE B.1 – Jeu de données exemple

	f_1	f_2	f_3	class
x_1	6.4	3.2	4.5	0
x_2	6.3	2.5	4.9	0
x_3	6.4	2.9	4.3	0
x_4	5.5	2.5	4.0	0
x_5	5.5	2.6	4.4	0
x_6	7.7	3.0	6.1	1
x_7	6.4	3.1	5.5	1
x_8	6.0	3.0	4.8	1
x_9	6.7	3.3	5.7	1
x_{10}	6.5	3.0	5.2	1



(a) 1er arbre de décision
bootsrap $\{x_1, x_2, x_3, x_4, x_6, x_7, x_9\}$

=

(b) 2ème arbre de décision
$bootsrap = \{x_1, x_2, x_5, x_7, x_8, x_9, x_{10}\}$

FIGURE B.1 – Les 2 arbres de décision utilisés

B.2 Contribution simple

Dans un nœud c on appelle Y^c_{mean} La fréquence d'instances de classe 1. Pour un nœud c et une variable f on peut calculer l'incrément de contribution local LI^c_f. Avec ces incréments locaux on

Table B.2 – Incréments de contribution dans les deux arbres

	Arbre 1		**Arbre 2**	
node	Y_{mean}	LI	Y_{mean}	LI
0	$\frac{3}{7}$		$\frac{4}{7}$	
1	0	$LI_{f_3} = \frac{-3}{7}$	$\frac{1}{4}$	$LI_{f_3} = \frac{-9}{28}$
2	1	$LI_{f_3} = \frac{4}{7}$	$\frac{1}{2}$	$LI_{f_2} = \frac{1}{4}$
3			0	$LI_{f_2} = \frac{-1}{4}$
4			0	$LI_{f_3} = \frac{-1}{2}$
5			1	$LI_{f_3} = \frac{1}{2}$
6			1	$LI_{f_3} = \frac{3}{7}$

peut calculer la contribution de la variable f pour une instance i dans l'arbre t, $FC^f_{t,i}$. Au final on peut calculer la contribution de f à la classification de i dans la forêt T arbres FC^f_i On a donc :

$$LI_f^{enfant} = \begin{cases} Y_{mean}^{enfant} - Y_{mean}^{parent}, & \text{Si la séparation chez le parent se fait sur } f, \\ 0, & \text{Sinon.} \end{cases} \tag{B.1}$$

$$FC^f_{t,i} = \sum_{c \in chemin} LI_f^c \tag{B.2}$$

$$FC^f_i = \frac{1}{T} \sum_t FC^f_{t,i} \tag{B.3}$$

On commence, sur l'arbre n°1, par calculer les Y_{mean}. À a racine de l'arbre tous les instances du bootsrap sont présents, donc $Y^0_{mean} = \frac{3}{7}$. Dans le nœud 1, on a les instances $\{x_1, x_2, x_3, x_4\}$ donc $Y^1_{mean} = 0$ et $Y^2_{mean} = 1$. De cette manière on peut en déduire les incréments locaux de contribution pour la variable f_3 aux nœuds 1 et 2 : $LI^1_{f_3} = -\frac{3}{7}$ et $LI^2_{f_3} = \frac{4}{7}$. Les incréments pour les autres variables sont égaux à 0 puisque la seule séparation se fait sur la variable f_3. On obtient au final les valeurs de la table B.2.

Pour chaque instance on peut alors calculer les contributions de chaque variable à sa classification. Prenons par exemple l'instance x_1. On a les chemins de décision suivants :

arbre 1 : $(0) \to (1) \Rightarrow$ classe 0.

arbre 2 : $(0) \to (1) \to (2) \to (4) \Rightarrow$ classe 0.

On peut d'abord avoir la classe prédite de x_1 : $\hat{Y} = \frac{1}{2}(0 + 0) = 0$. On peut également calculer les contribution suivantes pour x_1 dans les 2 arbres :

$$FC^{f_2}_{1,1} = 0$$
$$FC^{f_2}_{2,1} = \frac{1}{4}$$
$$FC^{f_3}_{1,1} = \frac{-3}{7}$$
$$FC^{f_3}_{2,1} = \frac{-9}{28} + \frac{-1}{2} = \frac{-23}{28}$$

Il ne reste plus qu'a moyenner les contributions par arbre pour obtenir la contribution totale de la variable à la prédiction pour x_1

$$FC_1^{f_2} = \frac{1}{2}(\frac{1}{4} + 0) = 0.125$$
$$FC_1^{f_3} = \frac{1}{2}(\frac{-3}{7} + \frac{-23}{28}) = -0.625$$

Au final on calcule les contributions pour tous les exemples du jeu de données et la prédiction du modèle de Forêt aléatoire dans la table B.3.

Table B.3 – Contributions de variables pour tous les exemples

	FC^{f_1}	FC^{f_2}	FC^{f_3}	$\hat{Y}$	classe
x_1	0	0.125	-0.625	*0.0*	**0**
x_2	0	-0.125	-0.375	*0.0*	**0**
x_3	0	0.125	-0.625	*0.0*	**0**
x_4	0	-0.125	-0.375	*0.0*	**0**
x_5	0	-0.125	-0.375	*0.0*	**0**
x_6	0	0	0.5	*1.0*	**1**
x_7	0	0	0.5	*1.0*	**1**
x_8	0	0.125	-0.125	*0.5*	**1**
x_9	0	0	0.5	*1.0*	**1**
x_{10}	0	0	0	*0.5*	**1**

On peut remarquer que, dans les cas où les contributions s'annulent, ou alors qu'elles sont égales à 0, le modèle ne peut pas prédire la classe de l'exemple considéré.

B.3 Contributions jointes

Le calcul se passe exactement de le même manière que pour les contributions simples mais au lieu d'attribuer l'incrément local d'un nœud à la variable sur laquelle est la séparation au niveau du nœud parent, on l'attribue à toutes les variables du chemin de décision jusqu'à ce nœud.
On calcule les contributions jointes pour l'instance x_1.

Dans le 1$^{\text{er}}$ arbre, la seule séparation se fait sur f_3 donc $FC^{\{f_3\}}$ est la seule contribution jointe possible. Dans le 2$^{\text{nd}}$ arbre, les séparations se font sur f_2 et f_3 pour x_1, donc on calculera les 2 contributions jointes : $FC^{\{f_3\}}$ et $FC^{\{f_2,f_3\}}$. Il ne peut pas y avoir la contribution de f_2 uniquement car la séparation à la racine du 2$^{\text{nd}}$ arbre se fait sur f_3 qui va donc apparaître dans tous les chemins de décision et donc dans toutes les contributions jointes . On obtient alors :

$$FC_{1,1}^{\{f_3\}} = \frac{-3}{7}$$
$$FC_{2,1}^{\{f_3\}} = \frac{-9}{28}$$
$$FC_{2,1}^{\{f_2,f_3\}} = \frac{1}{4} - \frac{1}{2} = \frac{-1}{2}$$

Annexe C

Les acides aminés

TABLE C.1 – abréviations de Acide aminés protéinogènes

Acide Aminé	Abréviation	Acide Aminé	Abréviation
Alanine	A	Lysine	K
Arginine	R	Méthionine	M
Asparagine	N	Phénylalanine	F
Aspartate	D	Proline	P
Cystéine	C	Pyrrolysine	O
Glutamate	E	Sélénocystéine	U
Glutamine	Q	Sérine	S
Glycine	G	Thréonine	T
Histidine	H	Tryptophane	W
Isoleucine	I	Tyrosine	Y
Leucine	L	Valine	V

TABLE C.2 – Table du code génétique. (* = codon STOP)

1^{ere} base	2^{eme} base				3^{eme} base
	U	C	A	G	
U	(UUU) **F**	(UCU) **S**	(UAU) **Y**	(UGU) **C**	U
	(UUC) **F**	(UCC) **S**	(UAC) **Y**	(UGC) **C**	C
	(UUA) **L**	(UCA) **S**	(UAA) *****	(UGA) *****	A
	(UUG) **L**	(UCG) **S**	(UAG) *****	(UGG) **W**	G
C	(CUU) **L**	(CCU) **P**	(CAU) **H**	(CGU) **R**	U
	(CUC) **L**	(CCC) **P**	(CAC) **H**	(CGC) **R**	C
	(CUA) **L**	(CCA) **P**	(CAA) **Q**	(CGA) **R**	A
	(CUG) **L**	(CCG) **P**	(CAG) **Q**	(CGG) **R**	G
A	(AUU) **I**	(ACU) **T**	(AAU) **N**	(AGU) **S**	U
	(AUC) **I**	(ACC) **T**	(AAC) **N**	(AGC) **S**	C
	(AUA) **I**	(ACA) **T**	(AAA) **K**	(AGA) **R**	A
	(AUG) **M**	(ACG) **T**	(AAG) **K**	(AGG) **R**	G
G	(GUU) **V**	(GCU) **A**	(GAU) **D**	(GGU) **G**	U
	(GUC) **V**	(GCC) **A**	(GAC) **D**	(GGC) **G**	C
	(GUA) **V**	(GCA) **A**	(GAA) **E**	(GGA) **G**	A
	(GUG) **V**	(GCG) **A**	(GAG) **E**	(GGG) **G**	G

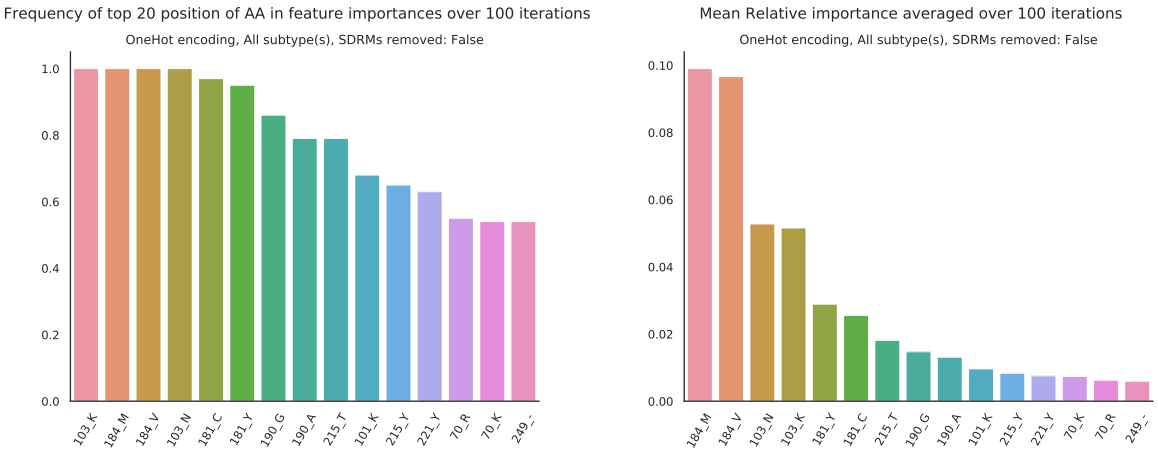
Annexe D

Résultats complets

D.1 Données Africaines

D.1.1 Avec DRMs

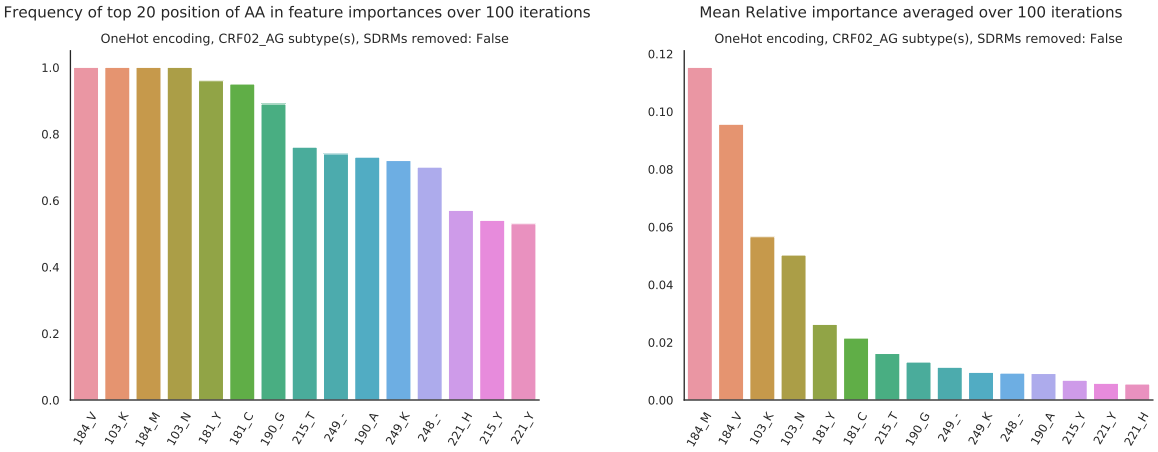
D.1.1.1 Forêts aléatoires



(a) Fréquence de sélection de la variable dans les 20 plus importantes

(b) Importance moyenne de la variable

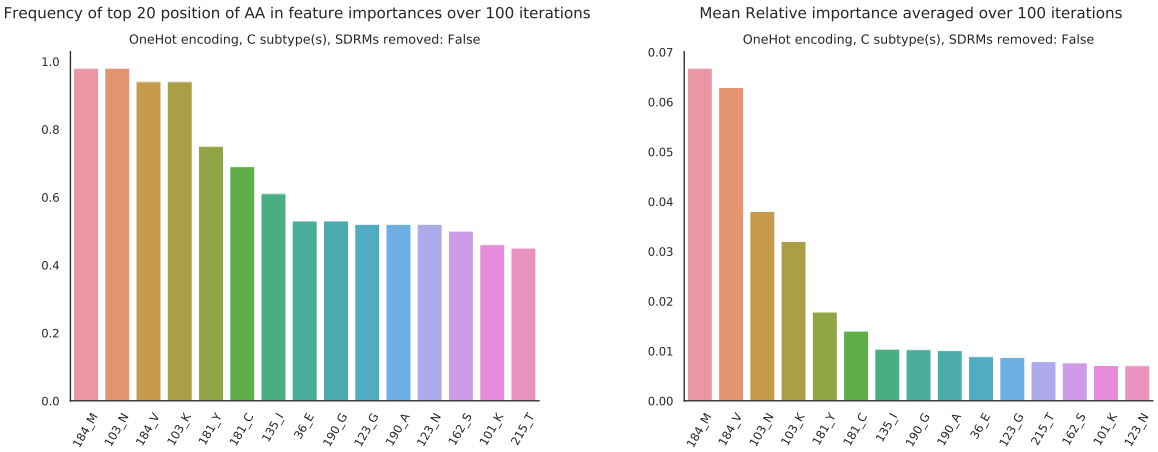
FIGURE D.1 – Importances des variables sur 100 itérations de stabilité. Données Africaines, tous sous-types



(a) Fréquence de sélection de la variable dans les 20 plus importantes

(b) Importance moyenne de la variable

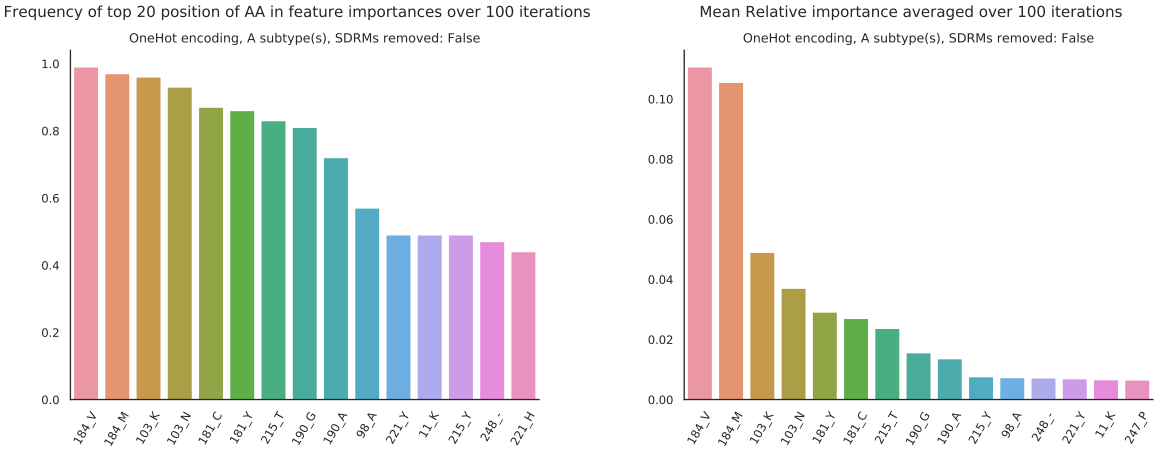
FIGURE D.2 – Importances des variables sur 100 itérations de stabilité. Données Africaines, sous-type CRF02_AG



(a) Fréquence de sélection de la variable dans les 20 plus importantes

(b) Importance moyenne de la variable

Figure D.3 – Importances des variables sur 100 itérations de stabilité. Données Africaines, sous-type C



(a) Fréquence de sélection de la variable dans les 20 plus importantes

(b) Importance moyenne de la variable

Figure D.4 – Importances des variables sur 100 itérations de stabilité. Données Africaines, sous-type A

D.1.1.2 Régression Logistique

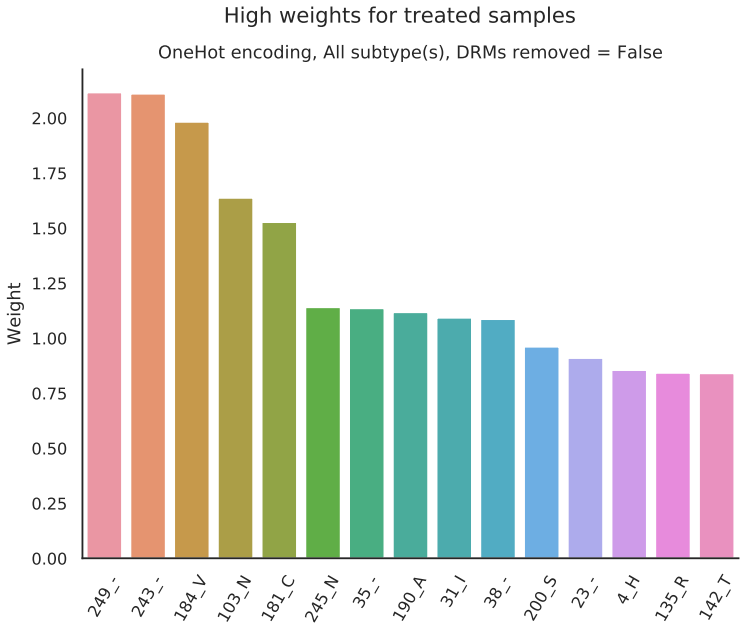


Figure D.5 – Hauts Coefficients de régression logistique. Données Africaines, tous sous-types

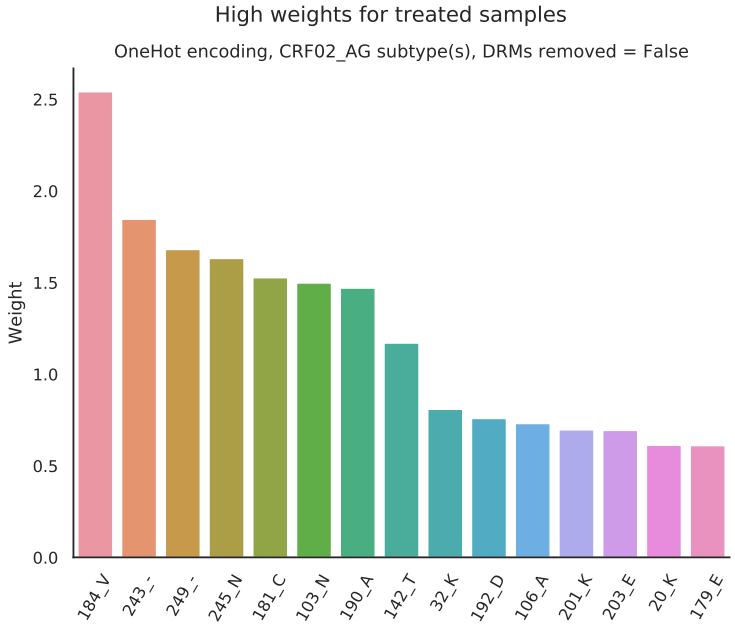


Figure D.6 – Hauts Coefficients de régression logistique. Données Africaines, sous-type CRF02_AG

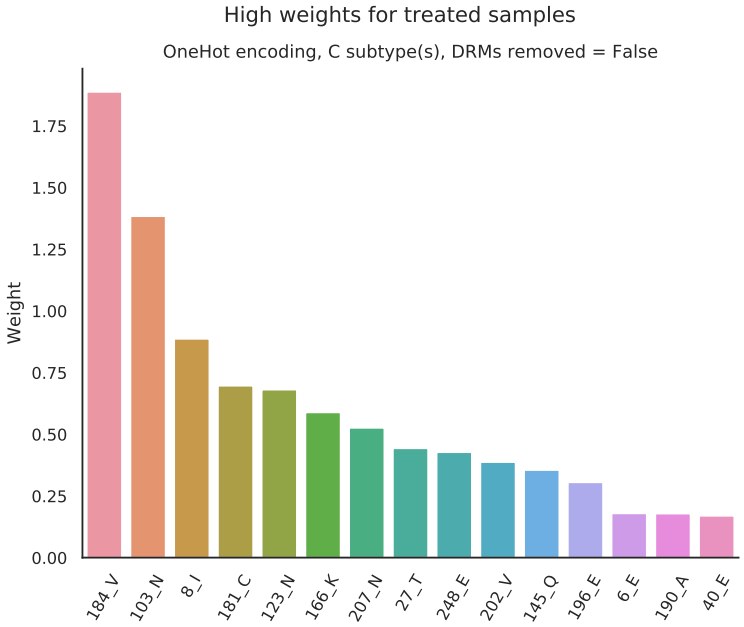


Figure D.7 – Hauts Coefficients de régression logistique. Données Africaines, sous-type C

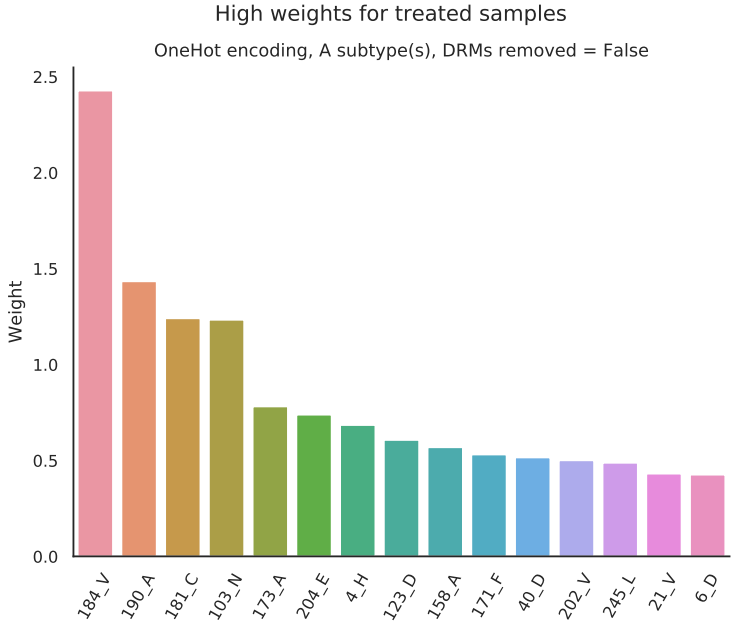


Figure D.8 – Hauts Coefficients de régression logistique. Données Africaines, sous-type A

D.1.1.3 Contributions de variables

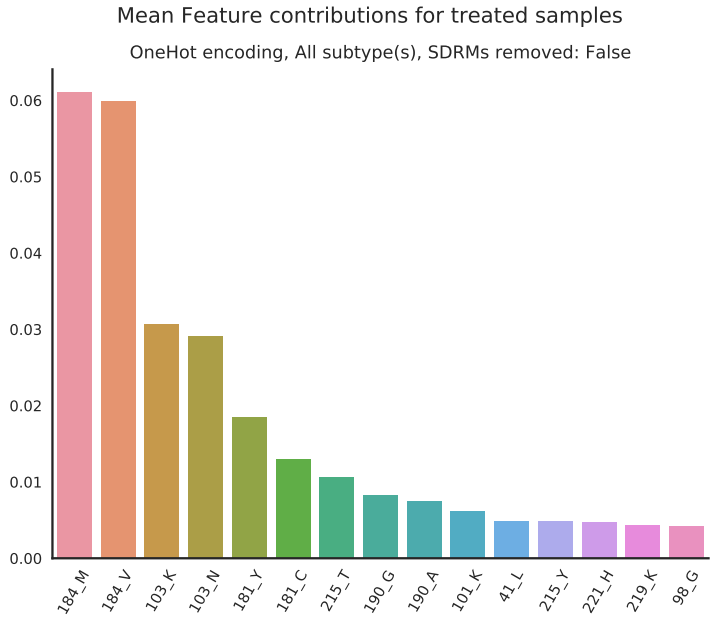


Figure D.9 – Hautes contributions de variables moyennées. Données Africaines, tous sous-types

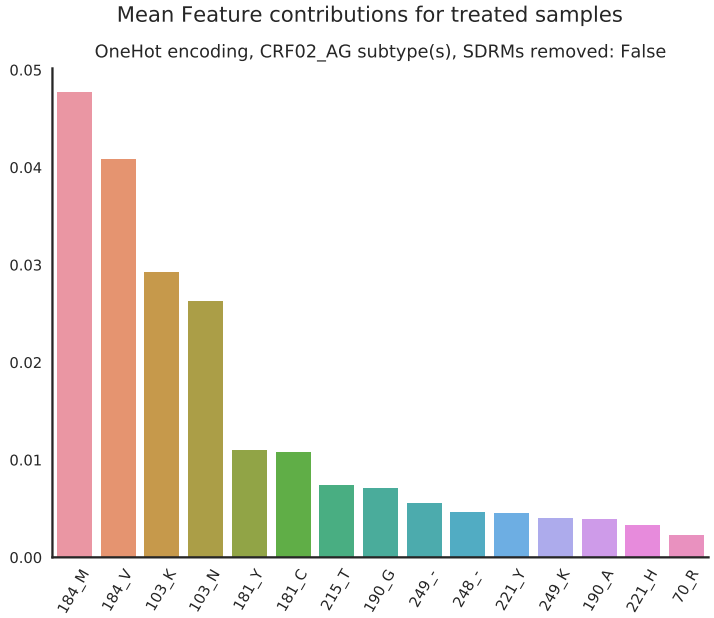


Figure D.10 – Hautes contributions de variables moyennées. Données Africaines, sous-type CRF02_AG

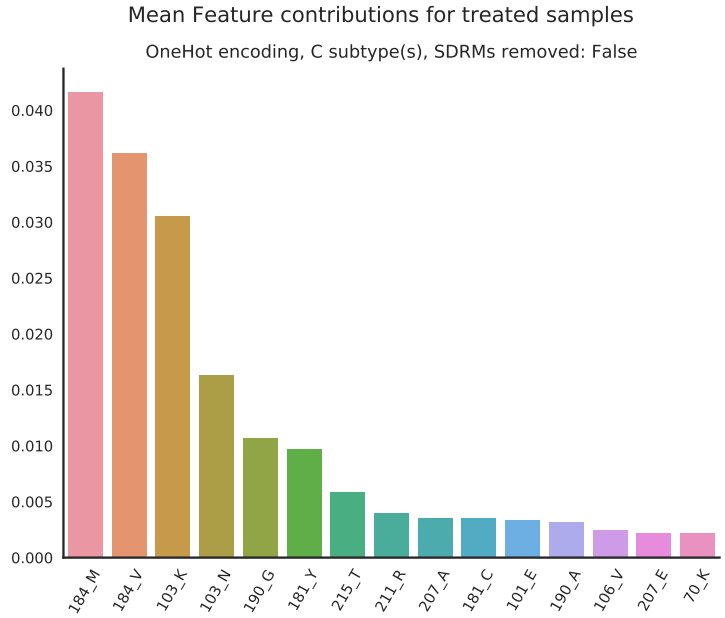


Figure D.11 – Hautes contributions de variables moyennées. Données Africaines, sous-type C

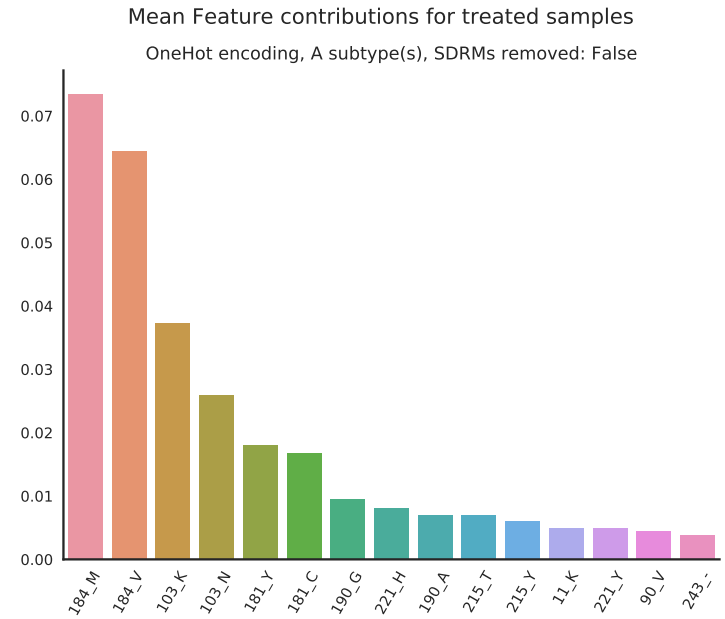
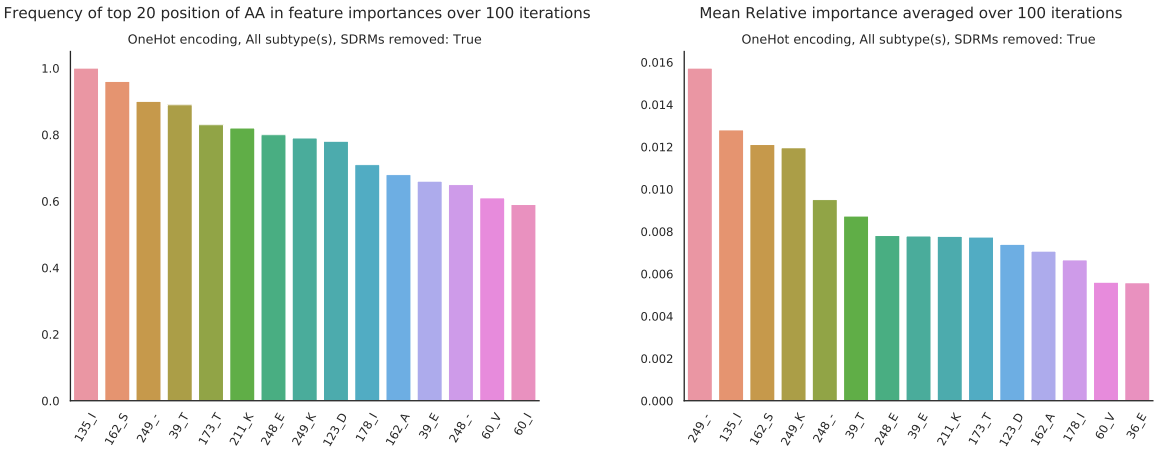


Figure D.12 – Hautes contributions de variables moyennées. Données Africaines, sous-type A

D.1.2 Sans DRMs

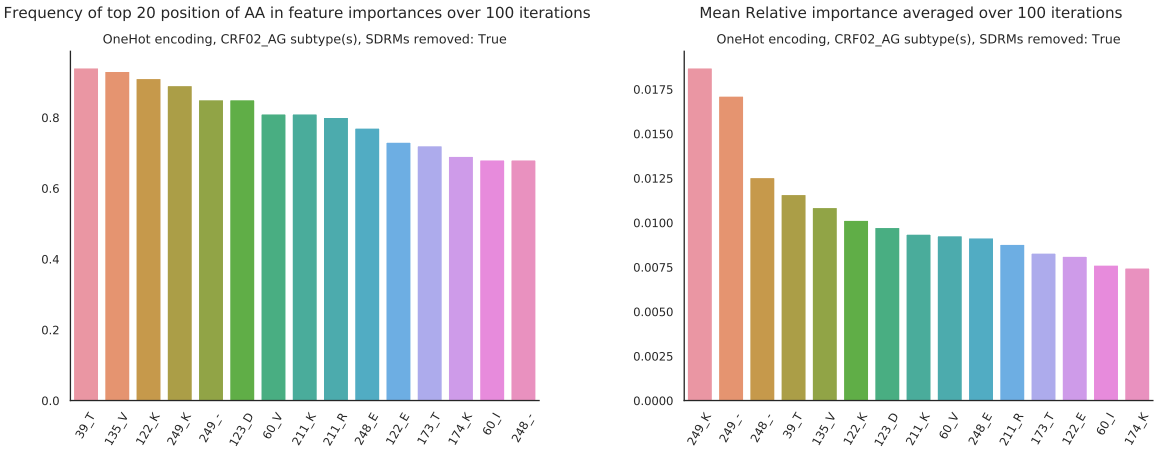
D.1.2.1 Forêts aléatoires



(a) Fréquence de sélection de la variable dans les 20 plus importantes

(b) Importance moyenne de la variable

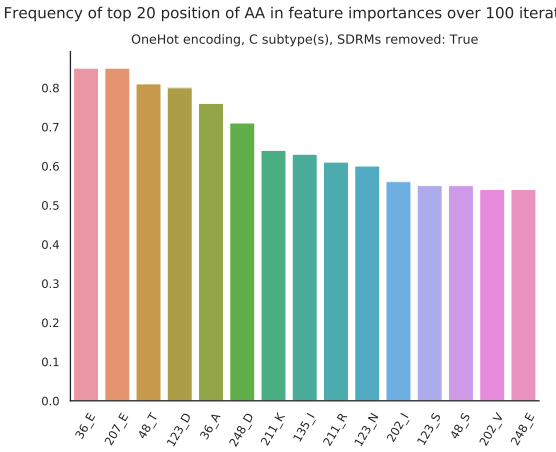
Figure D.13 – Importances des variables sur 100 itérations de stabilité. Données Africaines, tous sous-types



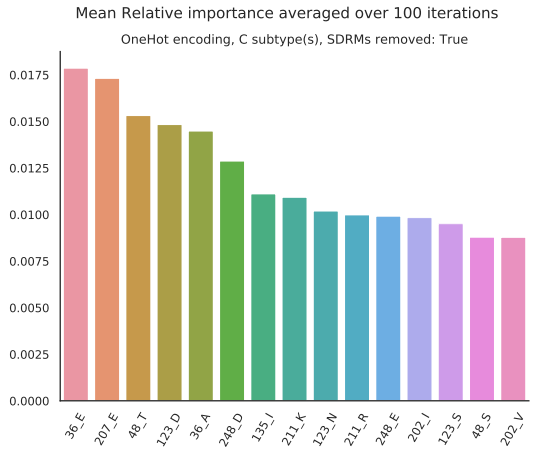
(a) Fréquence de sélection de la variable dans les 20 plus importantes

(b) Importance moyenne de la variable

Figure D.14 – Importances des variables sur 100 itérations de stabilité. Données Africaines, sous-type CRF02_AG

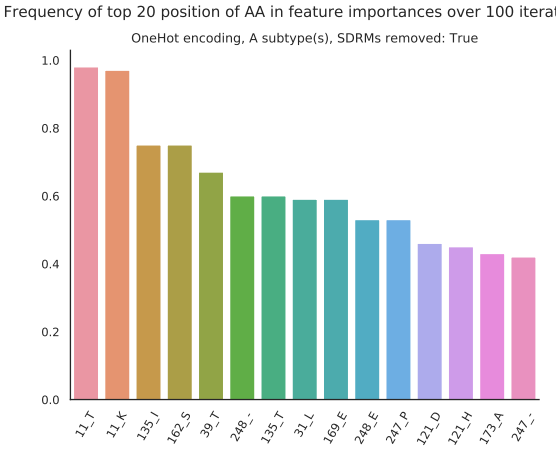


(a) Fréquence de sélection de la variable dans les 20 plus importantes

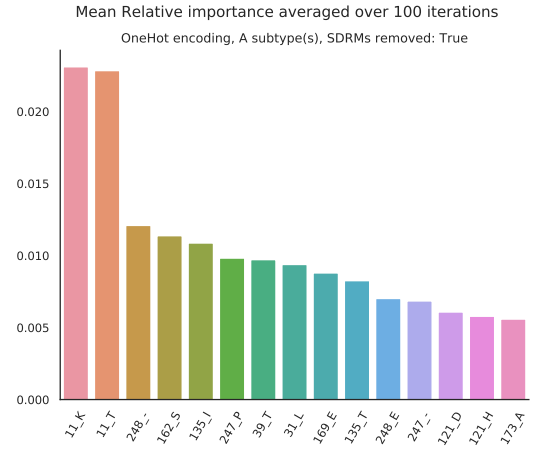


(b) Importance moyenne de la variable

Figure D.15 – Importances des variables sur 100 itérations de stabilité. Données Africaines, sous-type C



(a) Fréquence de sélection de la variable dans les 20 plus importantes



(b) Importance moyenne de la variable

Figure D.16 – Importances des variables sur 100 itérations de stabilité. Données Africaines, sous-type A

D.1.2.2 Régression Logistique

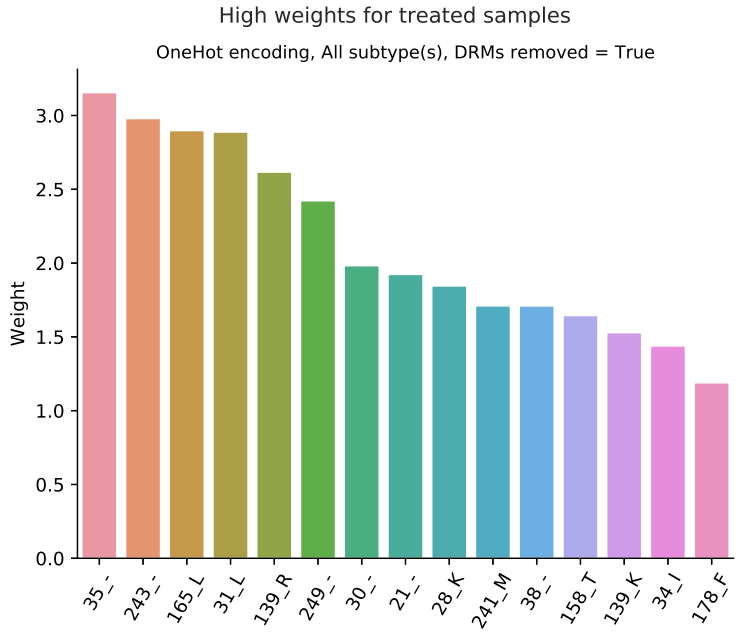


Figure D.17 – Hauts Coefficients de régression logistique. Données Africaines, tous sous-types

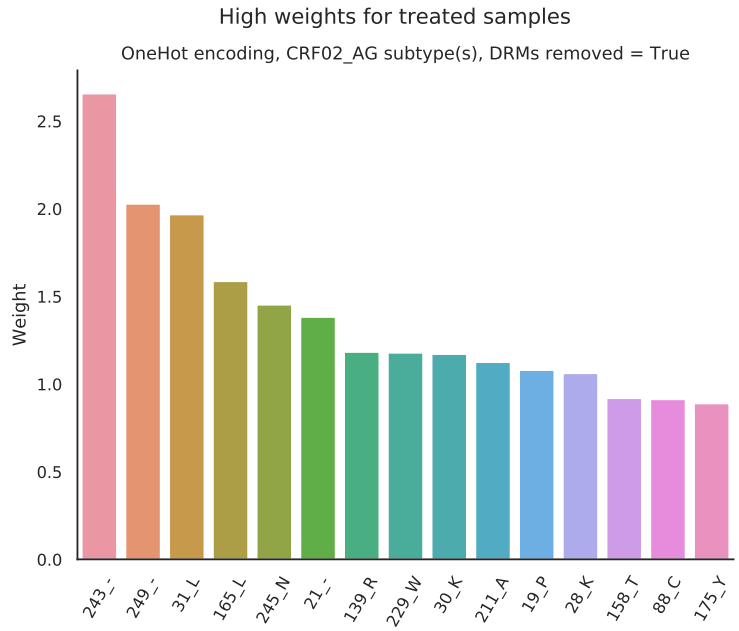


Figure D.18 – Hauts Coefficients de régression logistique. Données Africaines, sous-type CRF02_AG

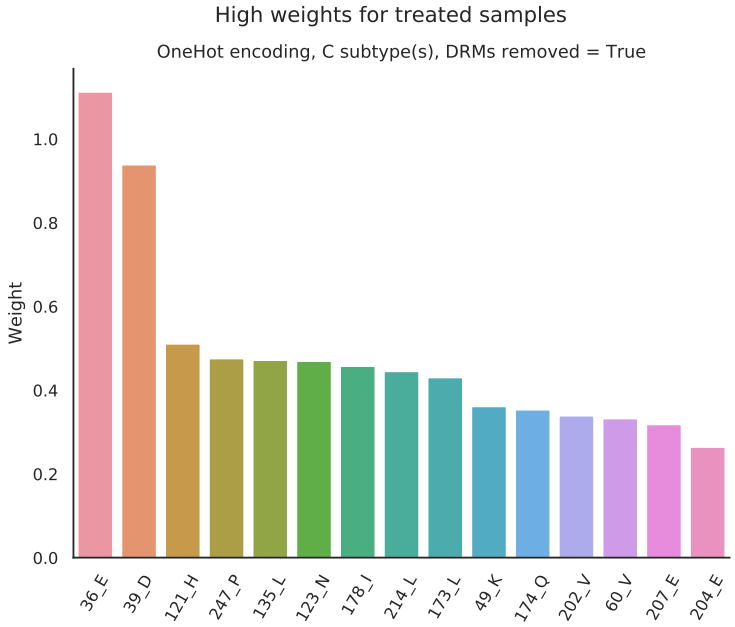


Figure D.19 – Hauts Coefficients de régression logistique. Données Africaines, sous-type C

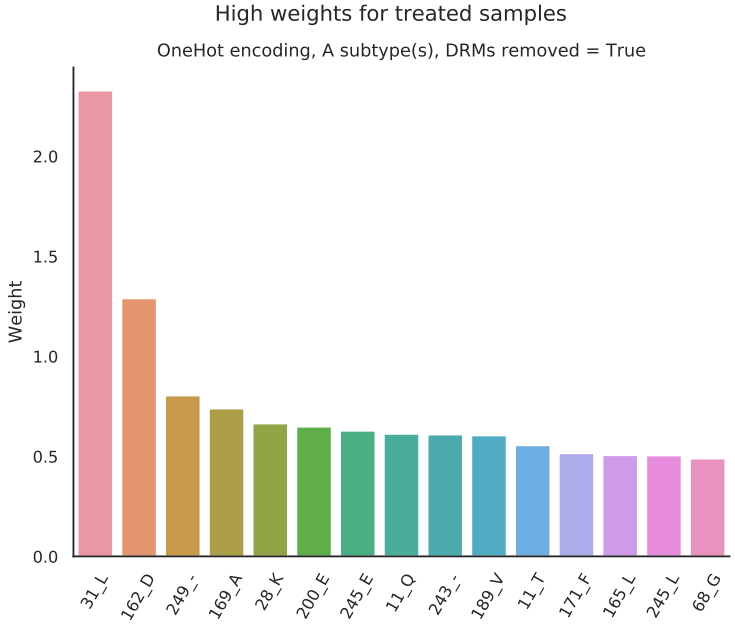


Figure D.20 – Hauts Coefficients de régression logistique. Données Africaines, sous-type A

D.1.2.3 Contributions de variables

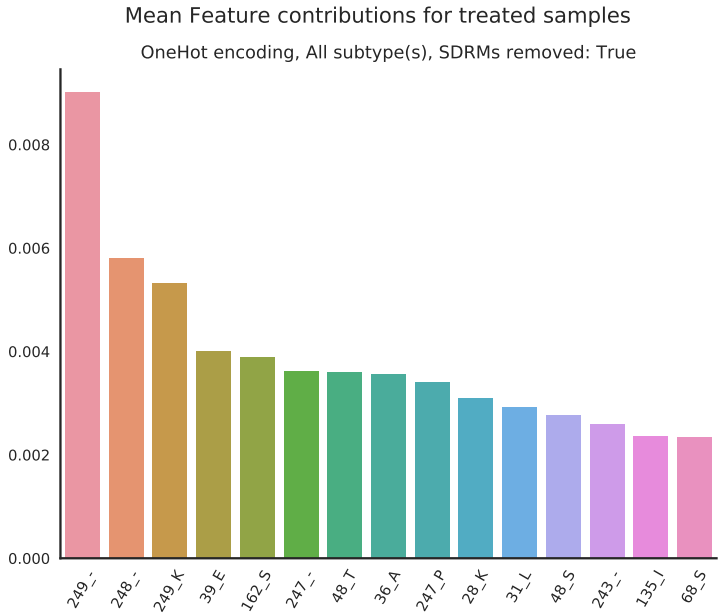


Figure D.21 – Hautes contributions de variables moyennées. Données Africaines, tous sous-types

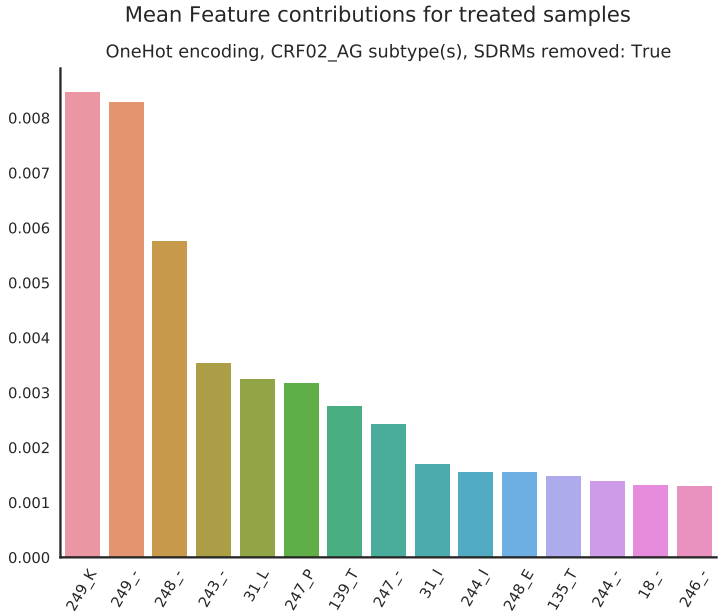


Figure D.22 – Hautes contributions de variables moyennées. Données Africaines, sous-type CRF02_AG

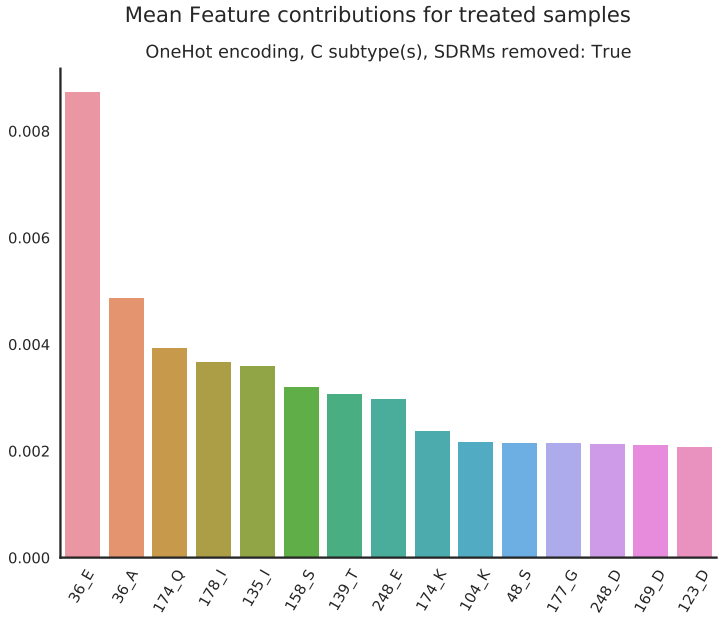


Figure D.23 – Hautes contributions de variables moyennées. Données Africaines, sous-type C

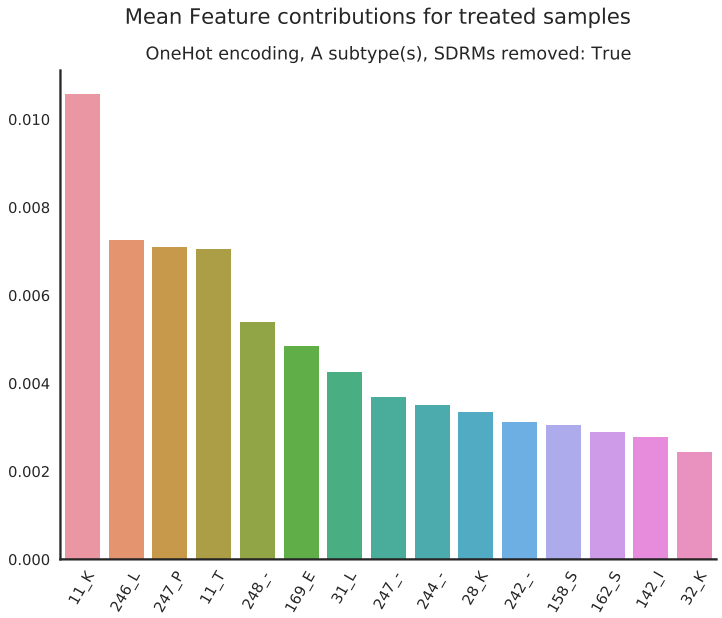
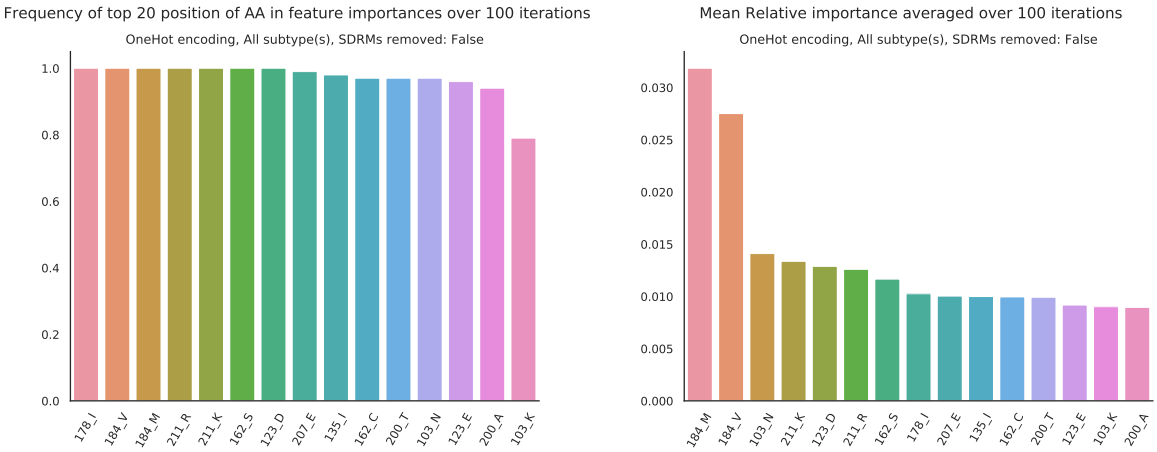


Figure D.24 – Hautes contributions de variables moyennées. Données Africaines, sous-type A

D.2 Données Européennes

D.2.1 Avec DRMs

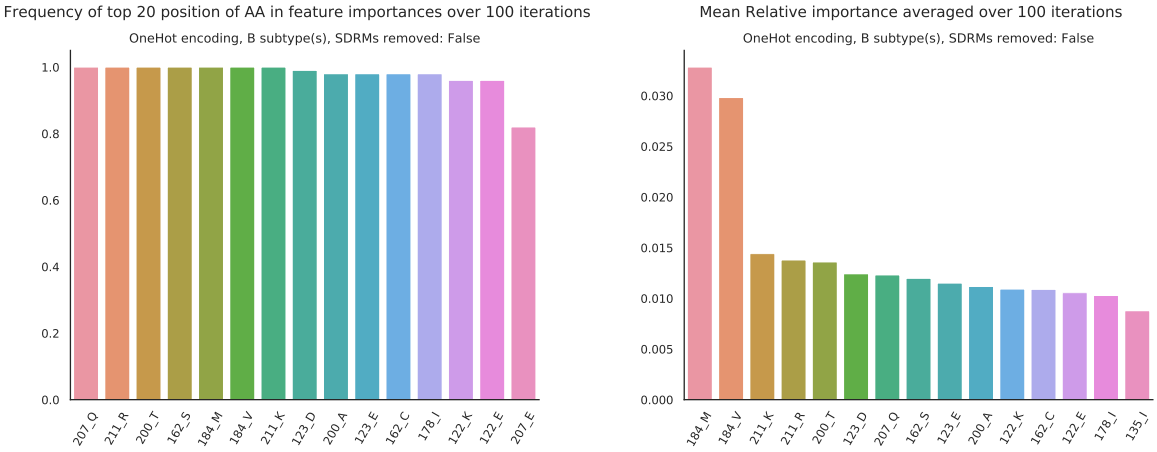
D.2.1.1 Forêts aléatoires



(a) Fréquence de sélection de la variable dans les 20 plus importantes

(b) Importance moyenne de la variable

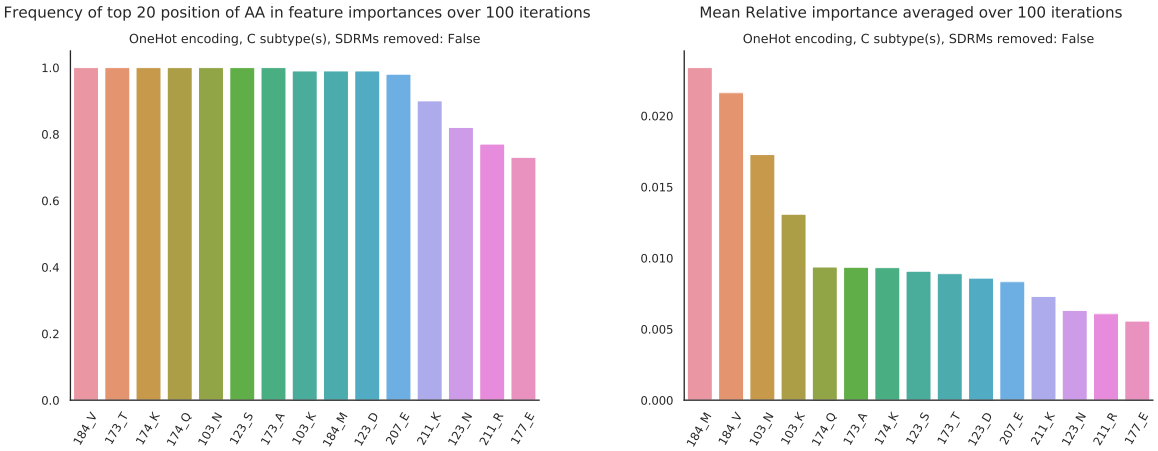
Figure D.25 – Importances des variables sur 100 itérations de stabilité. Données Européennes, tous sous-types



(a) Fréquence de sélection de la variable dans les 20 plus importantes

(b) Importance moyenne de la variable

Figure D.26 – Importances des variables sur 100 itérations de stabilité. Données Européennes, sous-type B



(a) Fréquence de sélection de la variable dans les 20 plus importantes

(b) Importance moyenne de la variable

Figure D.27 – Importances des variables sur 100 itérations de stabilité. Données Européennes, sous-type C

D.2.1.2 Régression Logistique

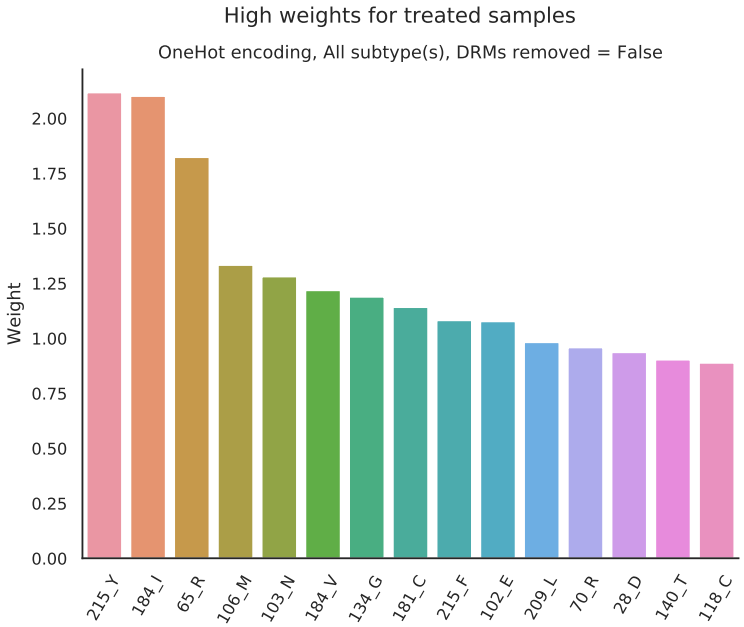


Figure D.28 – Hauts Coefficients de régression logistique. Données Européennes, tous sous-types

Figure D.29 – Hauts Coefficients de régression logistique. Données Européennes, sous-type B

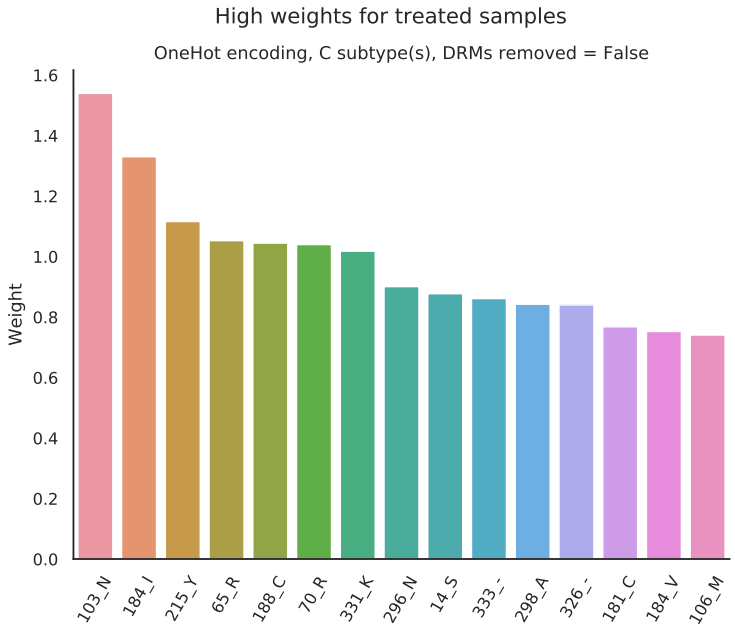


Figure D.30 – Hauts Coefficients de régression logistique. Données Européennes, sous-type C

D.2.1.3 Contributions de variables

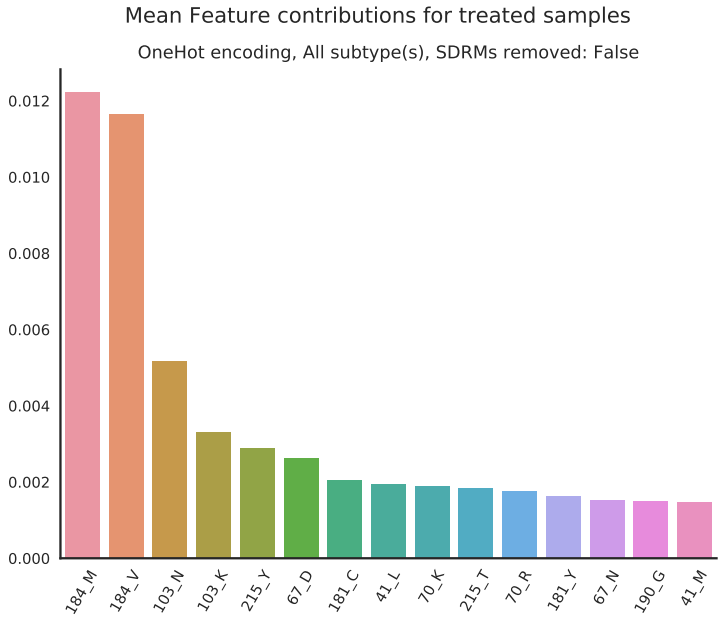


Figure D.31 – Hautes contributions de variables moyennées. Données Européennes, tous sous-types

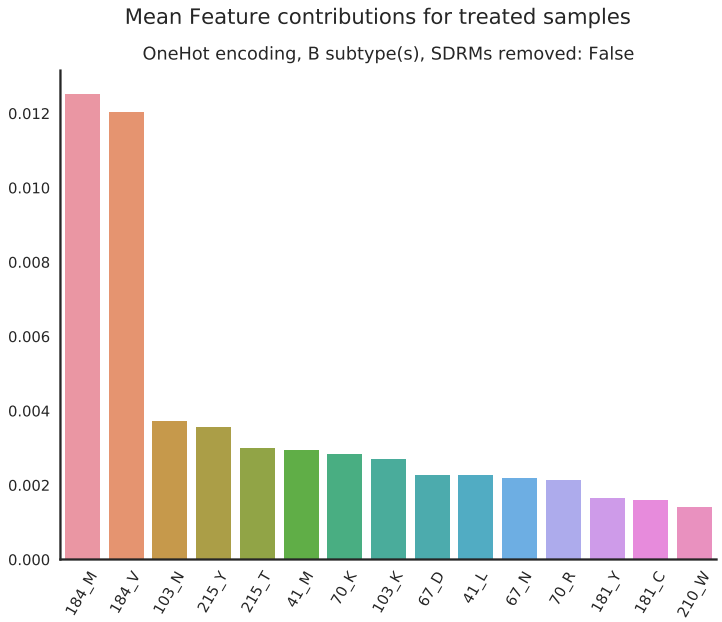


Figure D.32 – Hautes contributions de variables moyennées. Données Européennes, sous-type V

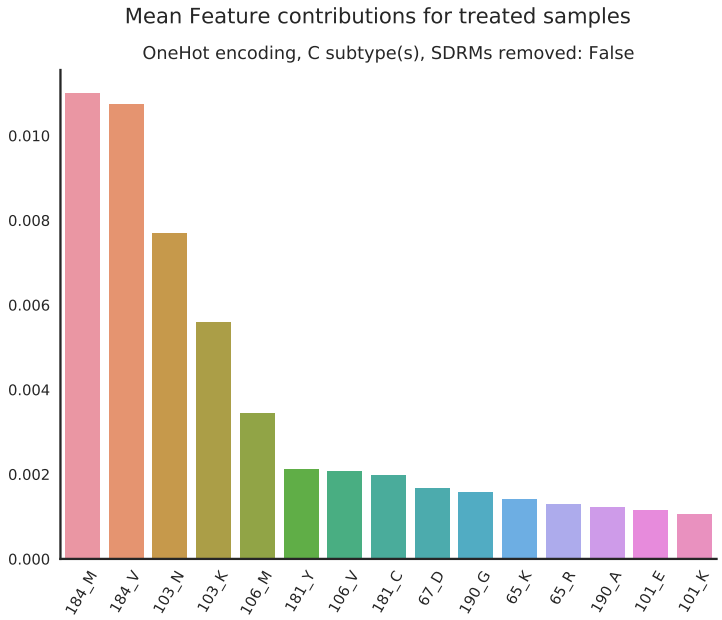
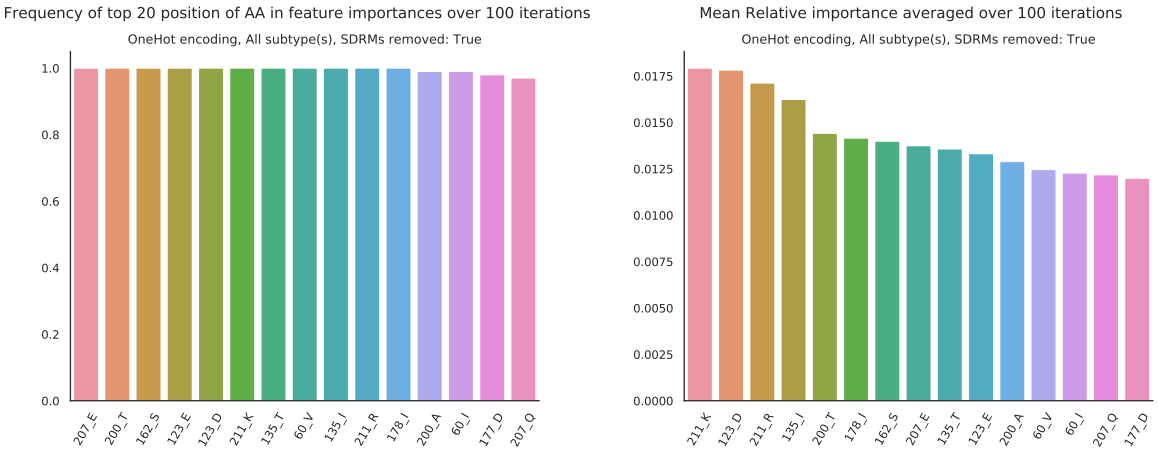


Figure D.33 – Hautes contributions de variables moyennées. Données Européennes, sous-type C

D.2.2 Sans DRMs

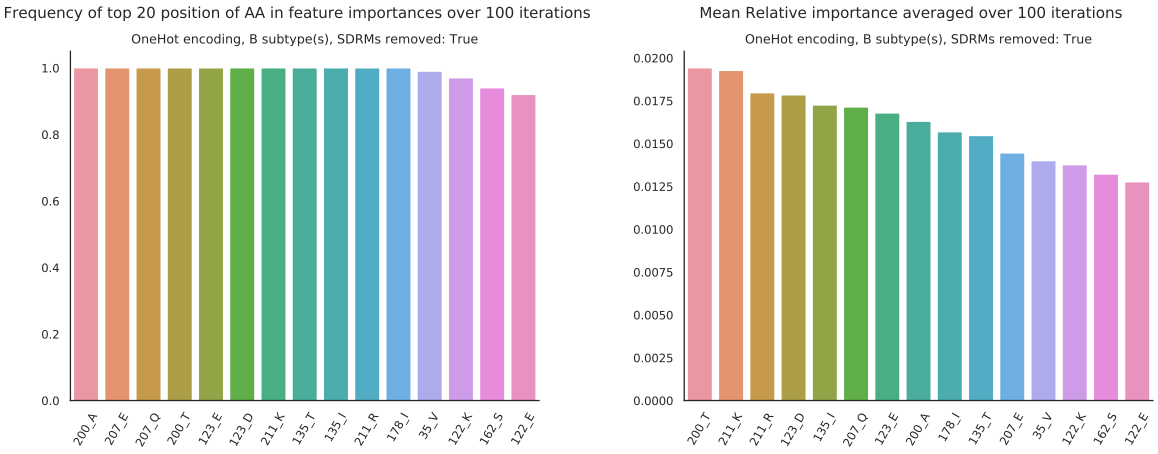
D.2.2.1 Forêts aléatoires



(a) Fréquence de sélection de la variable dans les 20 plus importantes

(b) Importance moyenne de la variable

Figure D.34 – Importances des variables sur 100 itérations de stabilité. Données Européennes, tous sous-types



(a) Fréquence de sélection de la variable dans les 20 plus importantes

(b) Importance moyenne de la variable

Figure D.35 – Importances des variables sur 100 itérations de stabilité. Données Européennes, sous-type B

(a) Fréquence de sélection de la variable dans les 20 plus importantes

(b) Importance moyenne de la variable

Figure D.36 – Importances des variables sur 100 itérations de stabilité. Données Européennes, sous-type C

D.2.2.2 Régression Logistique

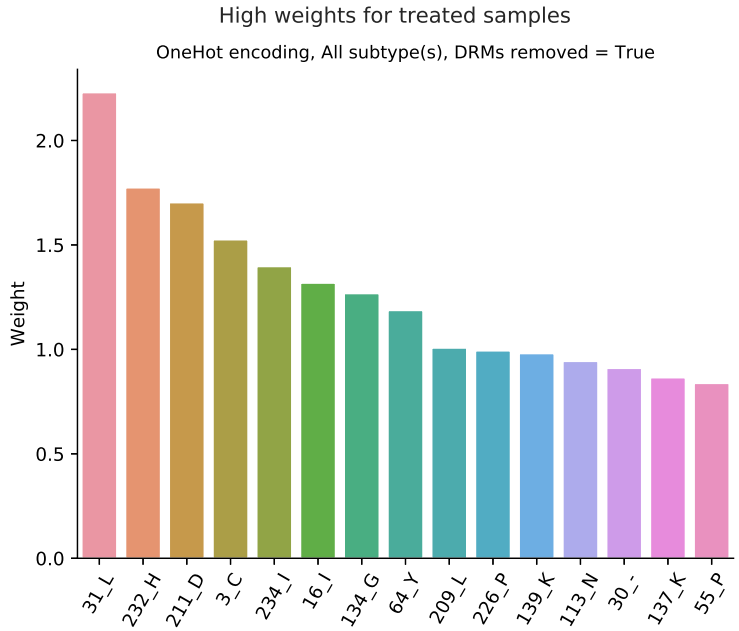


Figure D.37 – Hauts Coefficients de régression logistique. Données Européennes, tous sous-types

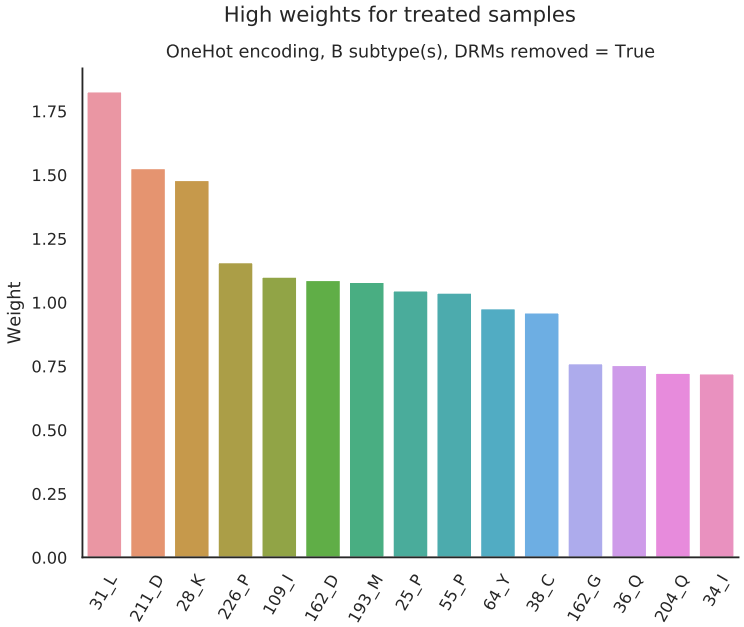


Figure D.38 – Hauts Coefficients de régression logistique. Données Européennes, sous-type B

Figure D.39 – Hauts Coefficients de régression logistique. Données Européennes, sous-type C

D.2.2.3 Contributions de variables

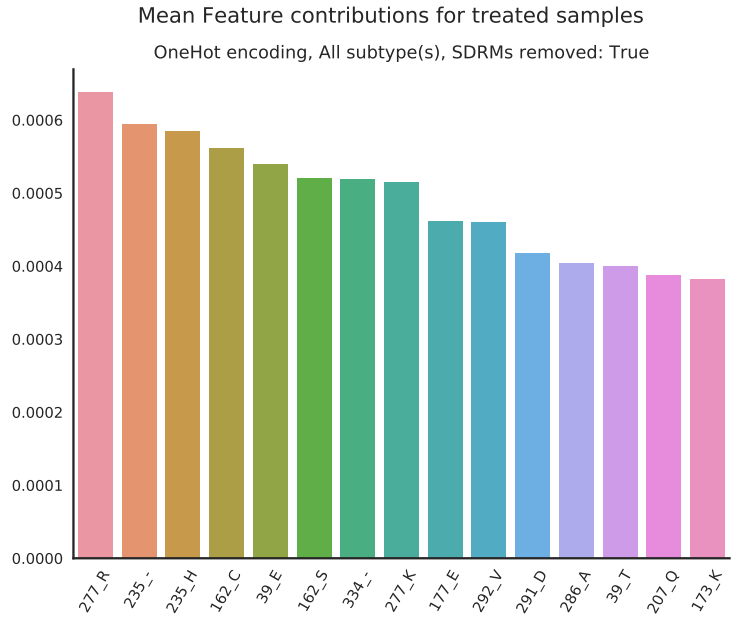


Figure D.40 – Hautes contributions de variables moyennées. Données Européennes, tous sous-types

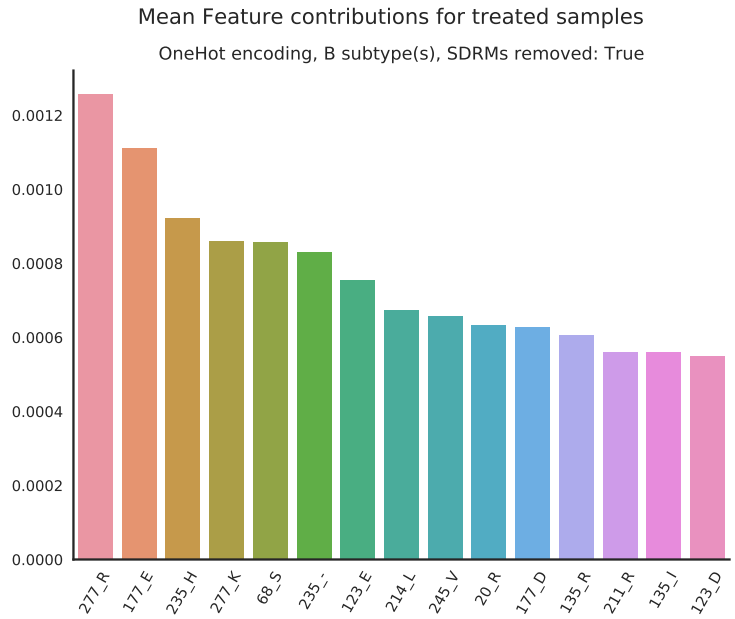


Figure D.41 – Hautes contributions de variables moyennées. Données Européennes, sous-type V

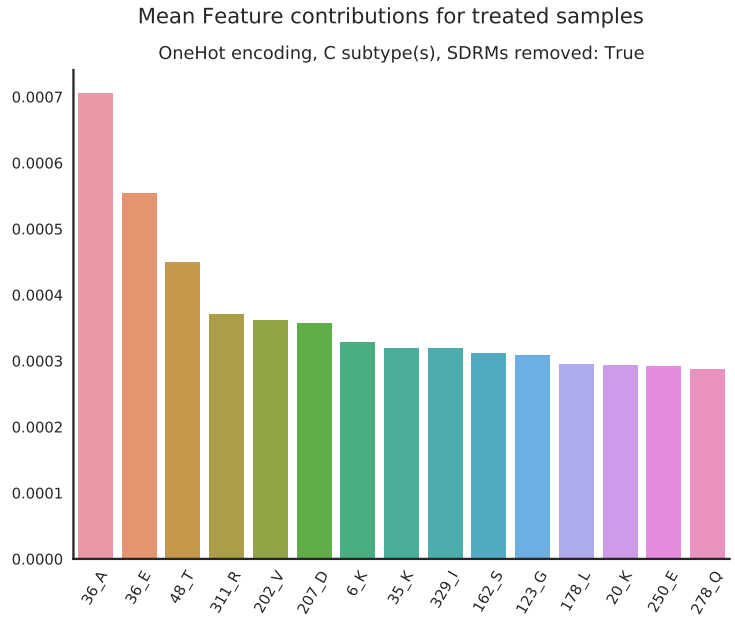


Figure D.42 – Hautes contributions de variables moyennées. Données Européennes, sous-type C

Annexe E

code

Le code est disponible au lien suivant :
https://github.com/lblassel/HIV-RF