Optimisation combinatoire pour la sélection de variables en régression en grande dimension : Application en génétique animale

Julie Hamon

Université Lille 1
École Doctorale Sciences Pour l'Ingénieur
Centre de recherche Inria Lille - Nord Europe
Laboratoire d'Informatique Fondamentale de Lille (UMR CNRS 8022)

Thèse CIFRE présentée pour obtenir le titre de **Docteur en Informatique** de l'Université Lille 1

Julie HAMON

Optimisation combinatoire pour la sélection de variables en régression en grande dimension : *Application en génétique animale*

Soutenance prévue le 26 novembre 2013

Jury :

Rapporteurs :	Charles BOUVEYRON	- Professeur des Universités, *Université Paris Descartes*
	Frédéric LARDEUX	- Maître de conférences HDR, *Université d'Angers*
Examinateurs :	Laurence DUCHIEN	- Professeur des Universités, *Université Lille 1*
	Stéphane CHRÉTIEN	- Maître de conférences, *Université de Franche-Comté*
Invité :	Claude GRENIER	- Directeur du développement, *Gènes Diffusion*
Directeurs :	Clarisse DHAENENS	- Professeur des Universités, *Université Lille 1*
	Julien JACQUES	- Maître de conférences HDR, *Université Lille 1*

Numéro d'ordre : 41253 | Année : 2013

Résumé

Le développement des technologies de séquençage et de génotypage haut-débit permet de mesurer, pour un individu, une grande quantité d'information génomique.

L'objectif de ce travail est, dans le cadre de la sélection génomique animale, de sélectionner un sous-ensemble de marqueurs génétiques pertinents permettant de prédire un caractère quantitatif, dans un contexte où le nombre d'animaux génotypés est largement inférieur au nombre de marqueurs étudiées.

Ce manuscrit présente un état de l'art des méthodes actuelles permettant de répondre à la problématique. Nous proposons ensuite de répondre à notre problématique de sélection de variables en régression en grande dimension en combinant approches d'optimisation combinatoire et modèles statistiques. Nous commençons par paramétrer expérimentalement deux méthodes d'optimisation combinatoire, la recherche locale itérée et l'algorithme génétique, combinées avec une régression linéaire multiple et nous évaluons leur pertinence. Dans le contexte de la génomique animale les relations familiales entre animaux sont connues et peuvent constituer une information importante. Notre approche étant flexible, nous proposons une adaptation permettant de prendre en considération ces relations familiales via l'utilisation d'un modèle mixte. Le problème du sur-apprentissage étant particulièrement présent sur nos données dû au déséquilibre important entre le nombre de variables étudiées et le nombre d'animaux disponibles, nous proposons également une amélioration de notre approche permettant de diminuer ce sur-apprentissage.

Les différentes approches proposées sont validées sur des données de la littérature ainsi que sur des données réelles de *Gènes Diffusion*.

Mots clés

Sélection de variables, régression, optimisation combinatoire, modèle mixte, grande dimension, génomique animale.

Combinatorial optimization for variable selection in high dimensional regression : *Application in animal genetic*

Abstract

Advances in high-throughput sequencing and genotyping technologies allow to measure large amounts of genomic information.
The aim of this work is dedicated to the animal genomic selection is to select a subset of relevant genetic markers to predict a quantitative trait, in a context where the number of genotyped animals is widely lower than the number of markers studied.
This thesis introduces a state-of-the-art of existing methods to address the problem. We then suggest to deal with the variable selection in high dimensional regression problem combining combinatorial optimization methods and statistical models. We start by experimentally set two combinatorial optimization methods, the iterated local search and the genetic algorithm, combined with a linear multiple regression and we evaluate their relevance. In the context of animal genomic, family relationships between animals are known and can be an important information. As our approach is flexible we suggest an adaptation to consider these familial relationships through the use of a mixed model. Moreover, the problem of over-fitting is particularly present in such data due to the large imbalance between the number of variables studied and the number of animals available, so we suggest an improvement of our approach in order to reduce this over-fitting.
The different suggested approaches are validated on data from the literature as well as on real data of *Gènes Diffusion*.

Keywords

Variable selection, regression, combinatorial optimization, mixed model, high dimension, animal genomic.

Table des matières

Définitions et abréviations

- ***ADN*** : Acide DésoxyriboNucléique
- ***AG*** : Algorithme Génétique
- ***AIC*** : *Akaike Information Criterion*
- ***Allèle*** : version d'un gène ou d'un locus
- ***BIC*** : *Bayesian Information Criterion*
- ***EBV*** : *Estimated Breeding Value*
- ***EN*** : Elastic-Net
- ***DRP*** : *De-Regressed Proof*
- ***Génotype*** : information portée par le génome d'un individu
- ***Haplotype*** : groupe de marqueurs en fort déséquilibre de liaison
- ***Héritabilité*** : part de génétique dans l'apparition d'un caractère au sein d'une population
- ***ILS*** : *Iterated Local Search*
- ***LD*** : *Linkage Disequilibrium* : association non aléatoire d'allèles
- ***Locus*** : position sur le génome
- ***LOO*** : *Leave-One-Out*
- ***Marqueur*** : variation observée dans l'ADN
- ***Phénotype*** : caractère observé
- ***PLS*** : *Partial Least Square*
- ***QTL*** : *Quantitative Trait Locus* : région du génome connue pour avoir un effet sur un caractère
- ***RLM*** : Régression Linéaire Multiple
- ***RMSEP*** : *Root Mean Square Error of Prediction*
- ***SNP*** : *Single Nucleotide Polymorphism* : variation d'une séquence d'ADN où un nucléotide a deux formes possibles.

Introduction

Le travail présenté dans cette thèse est issu d'une problématique posée par la société *Gènes Diffusion*, qui finance cette thèse CIFRE. *Gènes Diffusion* est une entreprise leader en sélection animale qui mène, en partenariat avec l'institut Pasteur de Lille, des expérimentations en génétique animale grâce à leur plate-forme Génomique. Cette thèse s'inscrit dans le cadre d'une collaboration avec Inria (centre Inria Lille - Nord Europe) et plus spécifiquement les équipes-projet DOLPHIN, travaillant sur des méthodes d'optimisation combinatoire avec notamment des applications en extraction de connaissances pour des données génomiques, et MODAL, spécialisée dans le domaine de l'apprentissage statistique, dont le traitement des données de grande dimension. Le thème principal de la thèse traite de l'amélioration génétique.

L'amélioration génétique est un problème au cœur des intérêts de tout éleveur, qui cherche à améliorer les performances de ses espèces, à savoir la résistance aux maladies ou la production de lait des bovins par exemple. Elle est basée sur deux méthodes : le croisement et la sélection. L'enjeu de la sélection animale est d'être capable de prédire, pour un animal, un caractère d'intérêt donné afin de sélectionner les meilleurs animaux, qui seront conservés pour les générations suivantes. Les méthodes de sélection jouent un rôle majeur dans l'amélioration génétique et ont bien évolué.

En effet, l'apparition récente des technologies de séquençage et génotypage haut-débit est une révolution dans le domaine de la sélection animale qui permet maintenant de prédire la valeur génétique d'un animal à partir de marqueurs répartis sur l'ensemble du génome. Cependant, outre les contraintes biologiques et informatiques liées à l'ampleur des données disponibles (jusqu'à 800 000 marqueurs pour une puce à ADN en espèce bovine), les méthodologies statistiques doivent être adaptées. En effet, le nombre important de marqueurs génétiques (bien plus important que le nombre d'individus étudiés) ne permet plus d'utiliser les techniques statistiques multivariées classiques. De nouvelles méthodes ont donc été proposées, principalement basées sur de la réduction de dimension en sélectionnant un sous-ensemble de variables (marqueurs).

Ainsi, l'analyse de données génomiques de grande dimension et spécifiquement le cas où le nombre n d'individus disponibles est très faible comparé au nombre p de variables étudiées ($n \ll p$), est une problématique largement étudiée. En effet, différentes communautés (statistique, optimisation combinatoire, data mining) mènent une recherche active sur le sujet proposant un large spectre de méthodes. Pour exemple, en 2012, le $16^{\text{ème}}$ challenge QTLMAS proposait aux participants de confronter leurs approches d'analyse de données génomiques. La mise en place

annuelle de ce challenge montre l'importance de la problématique qui est largement étudiée.

À travers cette thèse, l'objectif de *Gènes Diffusion* est de faire un état des lieux des méthodes existantes et de palier leurs limites en proposant, si nécessaire, une nouvelle approche. Pour cela, nous étudions et comparons dans un premier temps les approches proposées dans la littérature notamment celles considérant des caractères quantitatifs. Puis, nous proposons une nouvelle approche basée sur le constat que les problématiques d'analyse de données en grande dimension telles que les données génomiques, peuvent également être vues, dans la plupart des cas, comme des problèmes d'optimisation combinatoire. En effet, les problèmes d'extraction de connaissances générés par ce type de données recherchent des associations de variables ayant une influence sur le caractère étudié. Nous sommes donc dans le cas classique d'un problème d'optimisation combinatoire dans lequel la solution recherchée est un sous-ensemble de variables optimisant l'influence sur le caractère étudié. L'intérêt de l'utilisation de méthodes d'optimisation combinatoire est qu'elles permettent une exploration efficace de l'espace de recherche constitué des sous-ensembles de variables.

Ainsi, l'objectif de cette thèse consiste à évaluer la pertinence de combiner une approche d'optimisation combinatoire et une méthode statistique afin de sélectionner un sous ensemble de marqueurs intéressants permettant d'élaborer un modèle prédictif pour un trait quantitatif. Il est à noter que les animaux étudiés en sélection génomique sont généralement issus de populations au sein desquelles ont eu lieu des croisements, sur plusieurs générations, impliquant une non indépendance entre ces animaux. En effet, contrairement aux études menées en humain, en sélection animale il existe des relations familiales généralement connues entre les animaux étudiés. Elles peuvent être une source d'information importante à considérer. Nous proposerons donc une approche permettant de les intégrer.

Ce mémoire se décompose en 5 chapitres :

Le **Chapitre 1** décrit tout d'abord le contexte dans lequel s'inscrit cette thèse et présente l'entreprise *Gènes Diffusion*. Nous présentons également les données simulées et pseudo-réelles (QTLMAS) que nous utilisons tout au long de cette thèse afin de paramétrer les différents algorithmes d'optimisation combinatoire utilisés et d'évaluer leurs performances. Nous détaillons la problématique de sélection de variables en régression en grande dimension pour des données de génomique ainsi que la modélisation que nous proposons.

Le **Chapitre 2** présente dans un premier temps les méthodes classiques de la littérature statistique traitant du problème de régression en grande dimension. Nous exposons ensuite les méthodes spécifiques à la génomique qui ont été proposées. Le problème de sélection d'un sous-ensemble de variables parmi un

grand ensemble pouvant être vu comme un problème d'optimisation combinatoire, nous présentons également dans ce chapitre les méthodes classiques d'optimisation combinatoire pour la sélection de variables ainsi que leurs applications en génomique.

Le **Chapitre 3** présente la première approche que nous proposons basée sur une métaheuristique à solution unique, la recherche locale itérée, couplée à un modèle de régression linéaire multiple. Nous étudions dans ce chapitre différentes configurations de l'algorithme d'optimisation combinatoire afin de définir la configuration la plus adaptée à notre problématique. Enfin nous comparons les performances de notre approche à des méthodes classiques de la littérature sur données simulées et pseudo-réelles.

Suite aux analyses menées dans le Chapitre 3, nous proposons dans le **Chapitre 4** une seconde approche basée sur une métaheuristique plus sophistiquée, l'algorithme génétique, pour une meilleure exploration de l'espace de recherche. Une fois l'algorithme adapté à notre étude, nous comparons cette approche à celle que nous avons proposée dans le chapitre précédent, ainsi qu'à des méthodes classiques de la littérature, sur données simulées et pseudo-réelles.

Les approches proposées dans les chapitres 3 et 4 donnant de bons résultats sur données simulées et pseudo-réelles, nous évaluons, dans le **Chapitre 5**, leurs performances sur données réelles. Du fait de la spécificité des données réelles étudiées, dont un déséquilibre entre le nombre de variables et le nombre d'individus beaucoup plus important que dans les données utilisées dans les chapitres précédents, ainsi que des relations familiales entre animaux particulièrement importantes, nous proposons deux améliorations de notre approche combinant un algorithme génétique et une régression linéaire multiple. La première traite le problème du sur-apprentissage, particulièrement présent dans cette application. La seconde s'intéresse à la présence de relations familiales entre animaux. En effet, la régression linéaire multiple utilisée dans les chapitres 3 et 4, ainsi que la plupart des approches classiques, supposent une indépendance entre les individus. Or, nous savons qu'il existe des relations familiales entre les animaux étudiés et ces relations étant généralement connues, nous proposons une amélioration de notre approche en remplaçant la régression multiple par un modèle mixte. Nous évaluons donc les performances de ces deux nouvelles approches comparées aux autres méthodes que nous avons proposées précédemment ainsi qu'aux méthodes classiques de la littérature.

Ce mémoire se termine par une conclusion qui reprend nos contributions et ouvre sur différentes perspectives.

Chapitre 1

Contexte et objectifs

Sommaire

Dans ce chapitre, nous exposons la problématique de la société qui est à l'origine de ce travail, *Gènes Diffusion*, coopérative agricole spécialisée en génétique et reproduction animale sur les espèces bovines, équines, porcines et lapines. L'objectif final est de sélectionner au plus tôt les meilleurs animaux à partir de leur ADN (sélection génomique) afin d'obtenir des populations de plus en plus performantes. Le problème consiste à identifier un sous-ensemble de marqueurs génétiques significatifs pour un caractère quantitatif (phénotype) donné et à déterminer un modèle prédictif pour ce caractère.

Nous introduisons dans un premier temps l'évolution des méthodes de sélection animale jusqu'à la sélection génomique ainsi que les données qui y sont associées. Nous exposons ensuite l'objectif de *Gènes Diffusion*, et pour finir nous présentons

le format des données utilisées dans cette thèse ainsi que la modélisation proposée du problème.

1.1 Du testage sur descendance à la sélection génomique

Grâce aux avancées technologiques, les méthodes de sélection animale ont grandement évolué. Basées initialement uniquement sur la génétique, elles utilisent maintenant des informations biologiques. Cette partie introduit le testage sur descendance puis présente les évolutions qui ont mené à la sélection génomique.

1.1.1 Le testage sur descendance

La *génétique* permet d'étudier la transmission de caractères des parents à leurs enfants. C'est en se basant sur ce précepte que, depuis la domestication des espèces, deux grandes méthodes d'amélioration génétique sont mises en œuvre que ce soit dans le domaine animal ou végétal : le croisement et la sélection. L'objectif est d'obtenir des populations de plus en plus performantes pour un caractère donné. Nous nous intéresserons ici uniquement à la partie sélection. Le *progrès génétique* (ΔG) [Rendel 1950], généré par un programme de sélection mis en place, peut être mesuré suivant la formule $\Delta G = \frac{i*r}{t}$ où i est l'intensité de sélection, liée au pourcentage d'animaux sélectionnés, r la précision de la sélection et t l'intervalle de temps entre générations ou l'âge moyen des parents à la naissance de leurs descendants. En sélection génétique, le progrès obtenu pour un animal est définitif ; on ne peut pas lui retirer. Ce progrès est également cumulable ce qui permet d'obtenir des individus de plus en plus performants. En effet, le progrès génétique obtenu sur une génération est transmis à la génération suivante et s'ajoute donc au progrès génétique de cette dernière.

Chaque pays ou région a ses propres conditions d'élevage, les schémas de sélection (c'est à dire le choix des animaux considérés, les méthodes de sélection des meilleurs) sont donc différents selon l'environnement ou encore la race. Les animaux sont par exemple sélectionnés en fonction de caractères comme les performances sportives pour les chevaux de sport ou la quantité de viande pour les porcins. Le choix des critères est basé sur les besoins économiques actuels et à venir. La sélection se faisant sur plusieurs générations, les descendants devront répondre aux besoins de la période où ils naissent. Dans ce manuscrit, nous nous intéressons plus particulièrement à l'espèce bovine et aux caractères associés. Par exemple, la race Holstein étant sélectionnée en fonction de ses aptitudes laitières, les caractères principalement abordés sont la production de lait ou l'aptitude au vêlage (c'est-à-dire la facilité à mettre bas, mesurée sur une échelle de 1 à 5). Dans le cas de la race Charolaise, qui est une race dite «allaitante», c'est-à-dire vouée à la production de viande, le caractère d'intérêt sera généralement la qualité de la viande (qui regroupe la tendreté, la jutosité ou encore la cuisson) ou le rendement de carcasse (poids de

carcasse/poids total de l'animal). Ces caractères ne sont pas toujours mesurables directement sur l'animal. Plusieurs types de mesures de performance sont utilisés :

- la *performance propre* de l'animal lorsque celle-ci est mesurable, comme par exemple le poids de naissance. En revanche, il n'est par exemple évidemment pas possible de mesurer la performance propre d'un taureau en terme de production laitière ou la qualité de viande pour un animal encore en vie.
- les *performances de l'ascendance* de l'animal. Par exemple, la performance laitière d'un taureau est calculée à partir des performances de sa génitrice, ainsi que de celles de son géniteur (elles-mêmes calculées à partir de celles de sa progéniture et de son ascendance). Cette mesure ne sera fiable qu'à condition que l'héritabilité du caractère soit élevée. En effet, l'*héritabilité* d'un caractère permet d'évaluer la part de génétique dans la mesure de ce caractère au sein d'une population. Une héritabilité élevée signifie donc que les performances d'un animal ont de fortes chances d'être de l'ordre de celles de ses géniteurs.
- les *performances de la descendance* de l'animal. Les performances laitières d'un taureau sont mesurées sur sa progéniture. Cette méthode, appelée le *testage sur descendance* permet également d'évaluer la transmission d'un caractère. Cependant, la procédure est coûteuse et longue puisqu'il faut attendre 3 ans avant de pouvoir mesurer une production de lait sur les génisses d'un taureau.

Lorsqu'elles sont mesurables, une combinaison de ces performances est utilisée pour obtenir la performance globale de l'animal.

1.1.2 L'utilisation de l'ADN

Dans les années 90, de nouvelles techniques apparaissent, basées sur la biologie moléculaire, et permettant d'identifier des variations dans l'ADN (Acide Désoxyribo-Nucléique). La méthode principale est la PCR (*Polymerase Chain Reaction*, [Innis 1990]) qui permet de dupliquer en grande quantité les zones de l'ADN souhaitées. L'*ADN*, support de l'information génétique, est une molécule

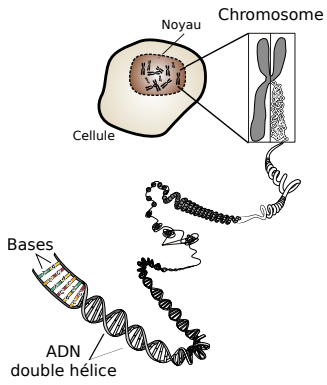

FIGURE 1.1 – De la cellule à l'ADN
(Source : fr.academic.ru)

contenant les informations nécessaires au développement et au fonctionnement d'un organisme (Figure 1.1). L'ADN est présent dans toutes les cellules et se présente sous la forme d'une double hélice composée de deux brins. Chaque brin est composé d'une séquence de quatre *nucléotides* (bases) différent(e)s : A (Adenine), C (Cytosine), G (Guanine) et T (Thymine). Les deux brins d'ADN formant l'hélice sont complémentaires : un nucléotide A sur un brin est associé à un nucléotide T sur l'autre brin, tandis que C est associé à G (Figure 1.2). La connaissance de la séquence d'un brin permet donc de déduire la séquence du deuxième.

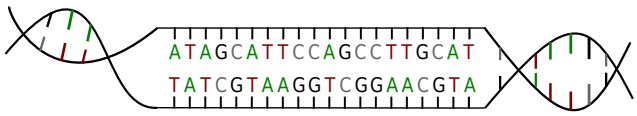

Figure 1.2 – ADN double hélice

Le *génome* constitue l'ensemble du matériel génétique codé dans l'ADN des êtres vivants. Il peut être extrait à partir de prélèvements sanguins, de poils ou de biopsies. L'ADN peut être analysé grâce au séquençage ou au génotypage.

1.1.3 Le séquençage

Le *séquençage* de l'ADN permet de déterminer l'ordre d'enchaînement des nucléotides pour un fragment d'ADN. Bien que la taille d'un génome soit de l'ordre de plusieurs millions de bases (3 milliards pour un bovin), le développement des techniques de séquençage ainsi que des moyens informatiques de traitement et de stockage des données, permet maintenant aux biologistes de séquencer des génomes complets : *Homo sapiens* (homme, 2001), *Mus musculus* (souris, 2005), *Gallus gallus* (poule, 2004), *Canis lupus familiaris* (chien, 2005), *Bos taurus* (bovin, 2006), *Equus caballus* (cheval, 2009), *Sus scrofa* (porcin, 2009), *Ovis aries* (mouton, 2010), *Oryctolagus cuniculus* (lapin, 2009). Alors que ce processus était encore très coûteux il y a quelques années, sont apparues en 2005 les méthodes de séquençage nouvelle génération (NGS - *Next-Generation Sequencing* - [Mardis 2008] [Ansorge 2009]) permettant un séquençage à très haut-débit (ou massif) et réduisant considérablement le coût. En effet, alors qu'il fallait 300 000 000 $ pour séquencer un génome humain en 2003, cela ne coûtait plus que 5 000 $ en 2010.
Dans le cadre de la sélection animale, la première étape consiste à séquencer une population d'individus (par exemple de bovins) sur son génome complet. Toutes les variations dans l'ADN observées entre les individus sont alors identifiées, ce sont les *marqueurs génétiques*. Différents types de marqueurs existent comme les polymorphismes de longueur des fragments (RFLP, AFLP), les marqueurs de séquence exprimée (EST) ou encore les séquences répétées (microsatellites, minisatellites) qui identifient des variations de segments d'ADN. Plus récemment, il est devenu possible d'identifier des marqueurs de polymorphisme nucléotidique (*Single Nucleotide*

Polymorphism - SNP), détaillés en section 1.2.1, qui sont des variations ponctuelles d'une paire de bases. Ce sont ceux que nous utiliserons dans cette thèse. Il y en a des millions le long du génome et ils serviront de base pour les analyses des nouveaux animaux.

1.1.4 Le génotypage

La première étape du *génotypage* consiste à déposer un fragment d'ADN sur des *puces à ADN* (ou *biopuces* - Figure 1.3). Sur ces puces sont positionnés des mar-

FIGURE 1.3 – Puce à ADN

queurs, sélectionnés parmi les millions de marqueurs identifiés lors du séquençage, de sorte qu'ils soient équidistants (qu'ils couvrent donc tout le génome) et qu'ils maximisent la MAF (*Minor Allele Frequency*), qui représente la proportion de variation d'un marqueur. Suivant la puce utilisée, le nombre de marqueurs sélectionnés peut aller jusqu'à 777 000 marqueurs pour le génotypage de bovins. Les puces les plus utilisées actuellement en génotypage de bovins, sont les puces 6K (de l'ordre de 6 000 SNPs) et 54K. Pour le génotypage humain, les puces sont de l'ordre du million de SNPs.
Une fois le fragment d'ADN déposé sur la puce, cette dernière est scannée afin d'extraire les différents allèles présents aux positions du génome prédéfinies sur la puce. Les *allèles* sont les différentes versions d'une séquence nucléotidique en une position donnée du génome.

Grâce au génotypage, de nouvelles méthodes de sélection animale basées sur les marqueurs génétiques sont proposées.

1.1.5 La Sélection Assistée par Marqueurs

Dans les années 2000, les technologies de génotypage ont permis l'apparition d'une nouvelle méthode de sélection, plus fiable, et permettant d'obtenir des résultats plus rapidement : la Sélection Assistée par Marqueurs (SAM ou SAM 1). Cette méthode repose sur l'hypothèse que, sur le génome, peu de régions ont un effet sur les caractères (*finite model*). Le principe général consiste à rassembler les marqueurs par régions chromosomiques appelées QTL (*Quantitative Trait Locus*). Ces régions, généralement proches du gène impliqué dans la variation du caractère étudié, sont connues pour avoir un effet sur ce dernier. Un marqueur sera lié à un QTL s'il est en fort *déséquilibre de liaison* (*Linkage Disequilibrium* - LD) avec ce

QTL. La présence de LD entre deux positions du génome représente une association non aléatoire des allèles à ces positions. La probabilité d'observer ces deux allèles ensemble ne sera pas égale au produit des probabilités de les observer individuellement. En effet, deux allèles proches sur le génome auront tendance à se transmettre ensemble et à avoir donc le même effet sur le caractère étudié. Les marqueurs de la

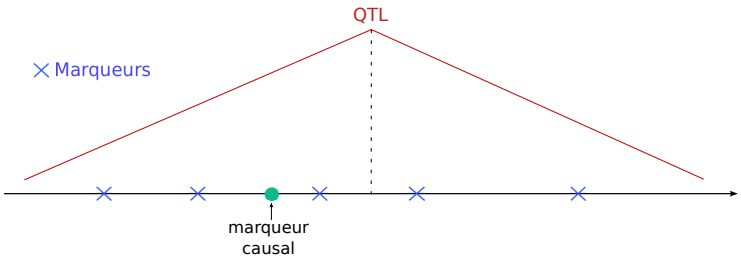

Figure 1.4 – QTL

Figure 1.4 sont tous en déséquilibre de liaison avec le QTL représenté et seront donc identifiés par ce même QTL. On obtient donc un ensemble de QTL (avec leurs effets respectifs) et le calcul de la performance d'un nouvel animal se fait alors en utilisant ces QTL qui sont connus pour contenir ou être lié à un ou plusieurs gènes ayant une influence sur les caractères étudiés. Une limite principale de cette méthode est l'étape de détection de QTL (*QTL mapping*) dont la précision sur la position peut être faible. Étant basée sur l'hypothèse qu'il faut rechercher peu de régions avec de gros effets, la SAM est une méthode intéressante uniquement si l'on travaille sur des marqueurs ayant des effets importants. Cependant, en 1988, Shrimpton et Robertson [Shrimpton 1988] ont démontré que, pour la majorité des caractères, seuls quelques gènes ont de gros effets et de nombreux ont de petits effets.

1.1.6 La Sélection Génomique

Grâce à une réduction importante du coût du génotypage, le génotypage haut-débit permet d'extraire un très grand nombre de marqueurs couvrant l'ensemble du génome de sorte que tous les QTL soient en déséquilibre de liaison avec au moins un marqueur. On peut donc étudier directement les effets des marqueurs en ne faisant aucune hypothèse sur la localisation des effets sur le génome. Contrairement à la SAM, la sélection génomique se base sur l'hypothèse qu'il y a beaucoup de marqueurs avec de petits effets (*infinitesimal model*). La SAM est désormais remplacée par la *Sélection Génomique* (SG - appelée aussi SAM 2 en France) dont le principe de base a été établi par Meuwissen, Hayes et Goddard en 2001 [Meuwissen 2001].
La sélection génomique est actuellement principalement développée en espèce bovine. En effet, l'utilisation de la génomique a un intérêt économique uniquement si la puce utilisée a un coût plus faible que le coût d'un animal. Or, contrairement à d'autres espèces (ovines, lapines, ...), les bovins coûtent très cher (de l'ordre de 2 000 €).

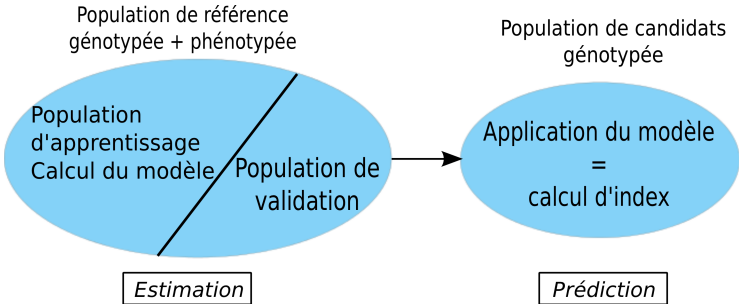

FIGURE 1.5 – Évaluation génomique

Le principe général de l'évaluation génomique, c'est-à-dire le calcul des performances basées sur la génomique qui permettent ensuite de sélectionner les animaux, est présenté en Figure 1.5 : une population de référence pour laquelle nous disposons des génotypes ainsi que des phénotypes (valeur du caractère sous étude - voir section 1.2.2) est découpée en deux échantillons appelés *apprentissage* et *validation*. Un modèle de prédiction du caractère en fonction des marqueurs est construit sur l'échantillon d'apprentissage puis évalué sur l'échantillon de validation. L'objectif est ensuite de prédire, à partir du modèle estimé sur la population de référence, la performance de nouveaux animaux dont on ne connaît que le génotype.
Le découpage de la population en deux n'est généralement pas aléatoire et dépendra de l'objectif de l'étude. En effet, l'apport principal de la sélection génomique en comparaison au testage sur descendance est qu'il est possible d'obtenir les performances d'un individus à sa naissance. L'objectif est donc d'être capable de prédire au mieux le caractère pour de jeunes animaux. Il est donc classique de mettre les animaux les plus jeunes dans l'échantillon de validation. Cependant, dans certaines études, l'objectif est de prédire des performances sur un animal n'ayant aucun lien de parenté avec les animaux présents dans l'étude. Pour cela, il semble donc plus judicieux d'extraire une famille complète pour constituer l'échantillon de validation.

De nos jours, l'objectif de la sélection génomique est d'être capable de sélectionner à partir de leur ADN, les animaux qui seront utilisés pour la génération suivante sans avoir de mesure de caractère pour ces derniers.

1.1.7 Testage sur descendance versus sélection génomique

Grâce aux nouvelles technologies et à la sélection génomique, on est maintenant capable d'obtenir la performance d'un animal sans mesurer de caractère ni faire du testage sur descendance. La Figure 1.6 trace les différentes étapes de la sélection génétique basée d'une part sur le testage sur descendance et d'autre part sur la sélection génomique. Nous prenons ici l'exemple de l'évaluation de la production laitière.
En testage sur descendance, à partir d'une population de veaux, et de mesures

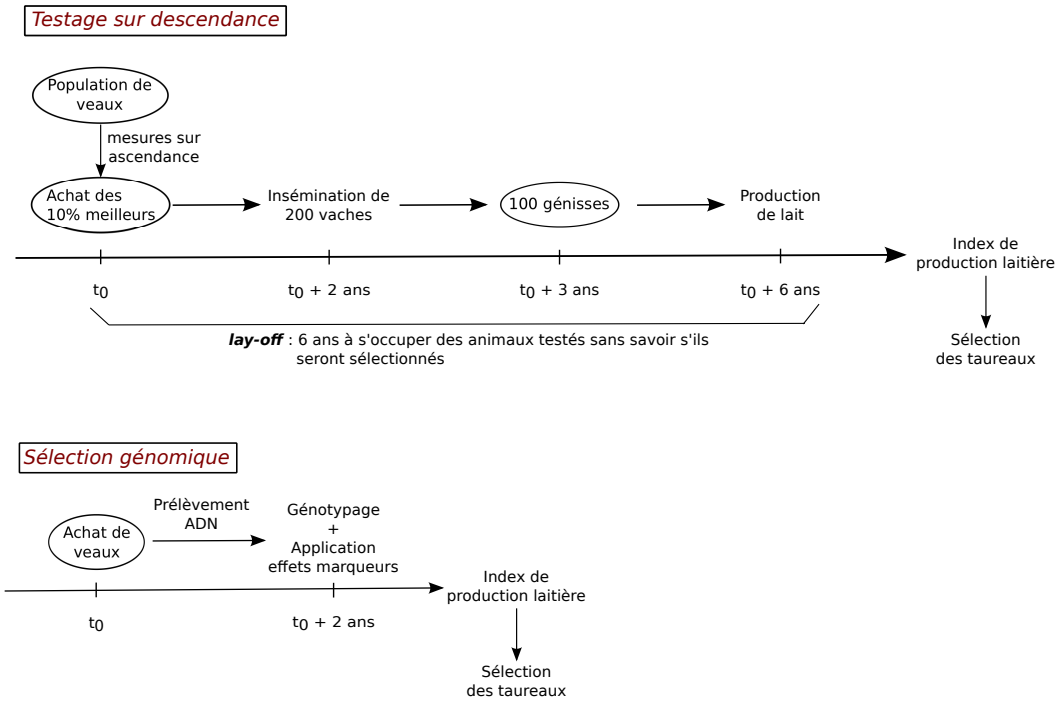

FIGURE 1.6 – Testage sur descendance versus sélection génomique

sur l'ascendance de ces derniers, les 10% meilleurs sont achetés pour l'insémination d'environ 200 vaches 2 ans après. Ces vaches mettront bas 1 an après, d'environ 100 génisses. Il faudra ensuite trois ans à ces génisses pour produire du lait. Il est donc possible de mesurer une production laitière 6 ans après l'achat des veaux et il faut donc pendant ces années s'occuper de ces animaux (les loger, les nourrir), sans savoir s'ils seront au final sélectionnés (*lay-off*).
En sélection génomique, lorsque les veaux sont achetés, leur ADN est prélevé, ils sont génotypés, le modèle de prédiction basé sur les marqueurs est appliqué ce qui permet d'obtenir un index (voir section 1.2.2) de production laitière et ce en 2 ans environ. L'obtention des performances en sélection génomique est donc beaucoup plus rapide qu'en testage sur descendance (2 ans contre 6 ans) et il n'y a pas de période de *lay-off* ce qui réduit considérablement le coût du programme de sélection qui est de l'ordre de 40 000 € par taureau en testage sur descendance alors que le coût du génotypage d'un taureau est de l'ordre de 500 €. Ce faible coût permet de génotyper plusieurs taureaux pour n'en choisir que quelques-uns (1 sur 50 par exemple) et donc augmenter l'intensité de sélection. Dans leur publication dans *Animal frontiers* sur l'application d'outils génomiques pour différentes espèces animales, Bagnato et Rosati [Bagnato 2012] expliquent l'importance de la génomique en sélection animale. Ils indiquent par exemple que l'information génomique peut réduire les coûts et accélérer le progrès génétique en réduisant les intervalles entre générations, c'est à dire en intégrant les veaux de plus en plus tôt dans le schéma de sélection.

Un autre avantage de la sélection génomique concerne les caractères étudiés. En effet, l'utilisation du testage sur descendance, ne permet pas de faire de la sélection sur un caractère peu *héritable*, c'est à dire peu dépendant de la génétique, comme la santé ou la longévité, lorsque l'animal étudié est un jeune taureau ayant peu de descendants. Avec l'apparition de ces méthodes basées sur l'ADN des animaux, les problèmes liés à l'héritabilité ne se posent plus puisque la performance d'un animal est évaluée par rapport à son ADN, les études pourront donc s'étendre à un plus grand nombre de caractères. Bagnato et Rosati [Bagnato 2012] soulignent le fait que la sélection génomique peut permettre d'identifier des individus supérieurs pour des caractères non encore considérés en reproduction animale car techniquement difficiles à relever. Cependant, les données phénotypiques doivent continuer à être collectées afin d'agrandir et/ou de renouveler la population de référence. En effet, même si pour certains caractères les effets des marqueurs sont déjà connus, ils peuvent changer sous l'influence de la sélection et devront donc être régulièrement ré-estimés.
Le testage sur descendance, s'il y a beaucoup de descendants, reste la méthode la plus précise, mais très longue comparée à la sélection génomique.
La mise en place de la sélection génomique nécessite l'analyse de nouvelles données : les données génomiques.

1.2 Les données génomiques

Dans cette section nous présentons la nature des données obtenues par le séquenceur. C'est en se basant sur ces données que le modèle de prédiction utilisé en sélection génomique sera généré.

1.2.1 Description

Un *locus* est une position du génome dont le texte génétique (séquence de nucléotides) peut avoir plusieurs versions appelées *allèles*. Un *polymorphisme* définit la présence en un locus de plusieurs allèles pour différents individus. Chaque individu possède k paires (avec $k = 22$ pour l'espèce humaine ou $k = 29$ pour l'espèce bovine par exemple) de chromosomes homologues (un hérité de la mère, l'autre du père), constitués de molécules d'ADN. Pour un individu donné on aura donc en un locus donné une combinaison de deux allèles, correspondant aux deux chromosomes de chaque paire, appelée *génotype*. Les marqueurs utilisés en sélection génomique sont des variations d'une paire de bases (nucléotides) en un locus, appelés SNPs (*Single Nucleotide Polymorphisms*). La Figure 1.7 montre un extrait de séquence d'ADN de trois individus. Pour chaque individu nous avons un extrait de son texte génétique sur 1 brin d'ADN de chacun des deux chromosomes homologues. L'individu 1 par exemple, diffère de l'individu 2 par un seul nucléotide (polymorphisme G/T : G sur le chromosome 2 de l'individu 1 remplacé par T sur le chromosome 2 de l'individu 2) et de l'individu 3 par deux nucléotides. Par conséquent, si les animaux sont génotypés à cette position, la différence entre les individus sera mise en valeur.

Pour un SNP donné, les deux bases possibles sont connues (généralement notées

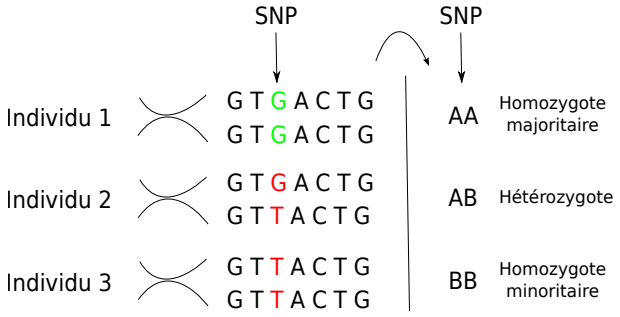

Figure 1.7 – Single Nucleotide Polymorphism

[C/G], [C/A], ...). Sur la Figure 1.7, les deux bases possibles pour le SNP sont G et T ([G/T]). L'intérêt ici est de distinguer les individus hétérozygotes, c'est à dire dont les deux bases du SNP sont différentes, des individus homozygotes (bases identiques) majoritaires et des individus homozygotes minoritaires peu importe les bases considérées. La paire de bases dont les deux bases sont différentes est notée 'AB', celle qui est majoritaire est généralement notée 'AA', c'est la combinaison allélique la plus fréquente dans la population pour un SNP donné, et la combinaison minoritaire 'BB'. En supposant que, sur la Figure 1.7, 'GG' est la combinaison la plus fréquente pour ce SNP dans la population étudiée, l'individu 1 aura la valeur 'AA', l'individu 2 la valeur 'AB' et l'individu 3 la valeur 'BB'.
Suivant la puce utilisée (6K, 54K, ...) chaque individu sera décrit suivant tous les marqueurs présents sur la puce. Nous aurons donc une matrice de dimension "individus × SNPs" contenant les valeurs 'AA', 'AB' et 'BB'.

1.2.2 Spécificités des données génomiques

Cette section présente quelques particularités des données de génomique au niveau du génotype dans un premier temps puis au niveau du caractère mesuré (phénotype). Nous présentons également une source d'information importante associée à ce type de données : le pedigree.

Génotype

Un échantillon d'individus est génotypé sur un ensemble de SNPs. Les données de génotypage directement issues du scanner ne sont pas utilisables telles quelles. En effet, un contrôle qualité de ces données est nécessaire avant de les analyser.

La ***fréquence de l'allèle mineur*** (*Minor Allele frequency* - MAF) ne doit pas être inférieure à une valeur donnée (10% généralement). En effet, pour un SNP, il faut un minimum de variations (polymorphismes) dans la population étudiée afin

que ce dernier ait un intérêt dans l'étude. Si ce n'est pas le cas il est supprimé.

L'***équilibre d'Hardy-Weinberg*** à un seuil k doit être respecté. Le principe d'Hardy-Weinberg stipule que, pour une population aléatoire, les fréquences alléliques restent les mêmes d'une génération à l'autre. L'équation $p^2+2pq+q^2=1$ illustre ce principe. Où p est la fréquence d'un allèle A, q celle d'un allèle B. p^2 est donc la fréquence du génotype $'AA'$, $2pq$ celle du génotype $'AB'$ et q^2 celle du génotype $'BB'$. Les fréquences génotypiques observées permettent grâce à l'équation d'Hardy-Weinberg de calculer les fréquences alléliques attendues puis d'en déduire les fréquences génotypiques attendues qui sont finalement comparées aux fréquences génotypiques observées. Les SNPs ne respectant pas cet équilibre sont supprimés de l'étude.

Le ***taux de succés du génotypage*** ne doit pas être inférieur à une valeur donnée (95% en général). C'est à dire qu'il ne doit pas y avoir plus de 5% de valeurs manquantes pour chaque SNP et chaque individu. En effet, cette contrainte doit être respectée au niveau des SNPs mais aussi au niveau des individus génotypés. Si le taux de valeurs manquantes pour un individu ou un SNP est élevé, soit il est supprimé, soit des méthodes d'imputation peuvent être utilisées [Marchini 2010], [Li 2009], consistant à prédire les SNPs manquants en analysant les SNPs voisins. Les méthodes d'imputation reposent généralement sur des modèles markoviens qui supposent donc que l'état d'un marqueur k dépendra uniquement du marqueur $k-1$.

Généralement, entre 10% et 20% des SNPs sont supprimés suite au processus de contrôle qualité.

Phénotype

Les processus d'amélioration génétique portent sur de nombreux critères de sélection et, pour ce faire, différents caractères appelés phénotypes sont mesurés. Le *phénotype* représente l'expression d'un caractère. Il peut être qualitatif s'il représente par exemple la couleur de la robe ou quantitatif comme la production laitière. Pour un caractère quantitatif on parlera également de performance de l'animal. Les variations phénotypiques (variations des caractères) sont dues aux variations génétiques mais aussi à des variations environnementales (Phénotype = Génotype + Environnement). Or, en *sélection animale*, on s'intéresse à la performance que l'animal sera susceptible de transmettre à sa descendance. Les effets environnementaux tels que la localisation géographique, la saison, le troupeau, doivent donc être pris en compte afin d'obtenir la *valeur génétique* de l'animal, c'est à dire la part du phénotype que sera transmise à la génération suivante, aussi appelée *index* de l'animal. Ce processus est appelé l'*évaluation génétique* ou l'*indexation*. Considérons le modèle :

$$\mathbf{y} = \boldsymbol{w} + \boldsymbol{a} + \boldsymbol{e}, \tag{1.1}$$

où $\mathbf{y}$ est la performance d'un animal, $\boldsymbol{w}$ un vecteur d'effets fixes (troupeau, saison par exemple), $\boldsymbol{a}$ est l'effet animal avec $\boldsymbol{a} = \frac{1}{2}a_m + \frac{1}{2}a_p + m$ (où a_m est l'effet de la mère, a_p est l'effet du père, et m l'aléa de la méïose), et $\boldsymbol{e}$ le résidu.

Différentes mesures de performances peuvent être utilisées en sélection génomique :

- les *phénotypes bruts* ($\mathbf{y}$), qui sont les performances sans correction (le modèle de sélection génomique intégrera alors les effets environnementaux).
- les *YD* (*Yield Deviation*) et *DYD* (*Daughter Yield Deviation*), qui sont les performances corrigées des effets environnementaux ainsi que de l'effet de la mère [Szyda 2008]. Pour les femelles c'est la YD qui est calculée : $YD = \mathbf{y} - (\hat{\boldsymbol{w}} + \frac{1}{2}\hat{a}_m)$. Pour un mâle on utilisera la DYD qui sera une moyenne des YD de sa progéniture (femelle).
- les *EBVs* (*Estimated Breeding Values* ou index), qui sont les performances corrigées des effets environnementaux et prenant en compte les effets familiaux (pedigree - voir paragraphe ci-dessous) : $EBV = \mathbf{y} - (\hat{\boldsymbol{w}} + \boldsymbol{e})$.
- les *DRP* (*DeRegressed Proofs*), qui sont les EBV auxquelles on enlève l'effet de l'ascendance et que l'on pondère (déregresse) par un coefficient de fiabilité (*EDC - Equivalent Daughter Contribution*) des EBV : $DRP = (EBV - (\frac{1}{2}\hat{a}_m + \frac{1}{2}\hat{a}_p) * EDC$.

Dans notre étude nous utiliserons les EBVs ou les DRP. En effet, ce sont les mesures principalement utilisées, de nombreuses études ayant prouvé leur fiabilité [Ostersen 2011], [Garrick 2009]. Thomsen *et al.* [Thomsen 2001] comparent ces différentes mesures dans leur publication et encouragent l'utilisation des DRP.

Pedigree

Une caractéristique importante des animaux sous étude est leur appartenance à un même troupeau pour lequel certains animaux font donc partie d'une même famille. Ces relations familiales entre animaux peuvent avoir une influence sur les prédictions. En effet, si dans la population d'apprentissage, plusieurs animaux sont hétérozygotes pour un SNP donné et ont une bonne performance, ce SNP sera considéré comme ayant un effet sur le caractère. Cependant, la présence de beaucoup d'hétérozygotes peut simplement provenir du fait que ces animaux sont issus de la même famille et ont donc hérité de cette hétérozygotie. Pour un animal qui n'est pas de la même famille la prédiction incluant un effet sur ce SNP ne sera donc pas forcément correcte. Il peut donc être important de prendre en compte ces relations afin de ne pas considérer des SNPs comme liés au caractère d'intérêt à tort (faux positifs). Les relations familiales sont représentées par le *pedigree* qui indique pour chaque animal l'identifiant de sa mère ainsi que celui de son père.

Dans le domaine animal nous disposons généralement d'un pedigree profond, c'est à dire remontant jusqu'à plusieurs générations, nous permettant ainsi d'intégrer

ces relations familiales aux études de sélection génomique. Cependant, l'intégration de ce facteur engendre l'utilisation de méthodes statistiques plus complexes du fait de la non indépendance des individus.

1.3 Problématique de Gènes Diffusion

Nous présentons dans cette section le cadre dans lequel s'inscrit notre travail au travers de la problématique soulevée par *Gènes Diffusion*.

1.3.1 Gènes Diffusion

Gènes Diffusion est une coopérative agricole spécialisée en génétique et reproduction animale sur les espèces bovines, équines, porcines et lapines. Sur les espèces équines, porcines et lapines, *Gènes Diffusion* prélève les semences de mâles sélectionnés par des entreprises extérieures et insémine les femelles. En revanche, *Gènes Diffusion* étant une coopérative appartenant à des éleveurs de bovins, son implication concernant cette espèce est plus importante et concerne autant la sélection des animaux que leur insémination.

Dans ce contexte de perfectionnement des technologies de génotypage (biopuces) et de développement des moyens informatiques, *Gènes Diffusion* a la possibilité d'analyser de plus en plus de SNPs à moindre coût. *Gènes Diffusion* utilise actuellement principalement des puces 54K, c'est à dire sur lesquelles environ 54 000 marqueurs sont mesurés. Ces puces offrent une précision correcte et sont utilisées dans la plupart des études de sélection génomique de bovins. Cependant, les méthodes d'analyse de ces données doivent être adaptées pour pouvoir gérer ces grands volumes de données.

1.3.2 Problématique

L'objectif principal de *Gènes Diffusion* est de sélectionner, pour un caractère donné, les meilleurs animaux pour constituer la génération suivante. Pour ce faire, il faut être capable de prédire les performances de la descendance des animaux présents dans le programme de sélection. Les schémas de sélection sont basés sur de nombreux caractères, à la fois quantitatifs comme la production de lait ou le rendement de carcasse et qualitatifs comme la résistance aux maladies. Cependant, malgré la diversité des caractères étudiés par *Gènes Diffusion*, les principaux sont quantitatifs et nous nous concentrons ici uniquement sur ces derniers. Nous sommes donc dans le cadre d'un modèle de régression permettant de prédire une variable quantitative.

L'objectif conjoint à cette prédiction est de sélectionner un nombre réduit de marqueurs significatifs pour le caractère étudié. Le modèle proposé devra donc également être capable de sélectionner un sous-ensemble de variables parmi un grand

ensemble de variables possibles (de l'ordre de 50 000 lorsqu'une puce 54K est utilisée). Cette problématique est basée sur plusieurs applications pratiques :

- sélectionner un nombre réduit de SNPs, de l'ordre de 100 à 300, soit pour créer une mini-puce soit pour les ajouter à une puce 6K existante.
- sélectionner des SNPs à partir d'une puce 800K pour les intégrer à une puce 54K ou 6K.

C'est actuellement la puce 54K qui est la plus utilisée par *Gènes Diffusion* car c'est aussi celle qui est utilisée par les organismes travaillant en sélection génomique comme l'INRA. Cependant, le génotypage avec des puces 54K reste coûteux et peut limiter le nombre d'animaux génotypés. C'est pourquoi, être capable d'obtenir de bonnes prédictions à partir d'un nombre réduit de SNPs peut être un atout majeur. En effet, une idée pourrait être de génotyper les animaux à bas coût avec une puce à faible densité (c'est à dire avec peu de SNPs) puis de faire un génotypage complet uniquement des meilleurs animaux pour obtenir une prédiction plus fiable. Un intérêt à prendre également en compte est l'utilisation de puces à faible densité pour les projets propres à l'entreprise, pour lesquels seule l'entreprise a des données et peut donc créer sa propre puce. C'est par exemple le cas en porcin où peu d'études sont menées actuellement mais *Gènes Diffusion* dispose de données sur lesquelles il serait intéressant de créer une puce dédiée.

1.4 Données utilisées dans la thèse

Dans cette section nous décrivons tout d'abord le codage général utilisé pour analyser les données dont nous disposons puis nous présentons ensuite les différents jeux de données utilisés pendant la thèse.

1.4.1 Description du codage

Nous avons vu en 1.2.1 que les données génomiques dont nous disposons se présentent sous la forme d'une matrice individus $\times$ SNPs contenant les valeurs 'AA', 'AB' et 'BB'. Pour être analysée, cette matrice est généralement recodée en {1, 0, -1} [Ogutu 2012] ou en {0, 1, 2} [Usai 2010], [Habier 2009], [Mai 2010]. Comme il n'y a pas d'avantage à choisir un codage plutôt que l'autre, nous choisissons de recoder les SNPs en {0, 1, 2}, 0 représentant la modalité la plus présente pour le SNP ('AA'), 1 une variation de la paire de base ('AB') et 2 deux variations ('BB'). Les valeurs prises pour le recodage ne sont pas arbitraires et représentent une quantité de variations, les SNPs peuvent donc être considérés comme des variables quantitatives.

Une autre possibilité serait de transformer les SNPs en variables binaires (0 ou 1). En effet, une variable qualitative à k modalités peut être représentée par $k-1$ variables binaires (la k-ième variables étant déduite des $k-1$ autres). Chaque SNP a trois modalités, il serait donc recodé en deux variables binaires. Cependant, cette transformation doublerait le nombre final de variables alors qu'elles sont déjà très nombreuses comparés au nombre d'individus.

1.4.2 Jeux de données utilisés dans la thèse

Afin de tester et valider les approches que nous proposons, nous utilisons deux benchmarks de la littérature, des jeux de données simulées se basant sur ces benchmarks, ainsi qu'un jeu de données réelles de *Gènes Diffusion*. Nous présentons ici les benchmarks de la littérature ainsi que les données simulées. Nous présenterons les données réelles dans le Chapitre 5.

1.4.2.1 Benchmarks de la littérature

QTLMAS est un workshop organisé depuis plusieurs années permettant aux chercheurs de confronter leurs approches sur un jeu de données simulées, proche de la réalité et adapté aux études génétiques animales et végétales. Ce workshop est organisé sous forme de challenge. En effet, une partie des phénotypes n'est pas fournie aux participants, ils doivent donc en trouver la meilleure prédiction. Nous utilisons ici les données des QTLMAS de 2010 et 2011. Les workshop étant passés, les phénotypes de validation sont maintenant disponibles et permettront donc l'évaluation des modèles proposés.

Le jeu de données du ***14ème workshop QTLMAS de 2010*** [Szydlowski 2011] est généré à partir de :

- 5 mâles,
- 15 femelles,
- 30 progénitures par femelle.

Les animaux sont extraits sur 5 générations pour un total de 3226 animaux dans le pedigree. Sur ces 3226 animaux, le phénotype de 900 animaux n'est pas donné lors du challenge. En effet, 4 fichiers textes sont fournis aux challengers :

- les génotypes de 3226 animaux sur 10 031 SNPs,
- les phénotypes de 2326 animaux,
- le pedigree de 3226 animaux,
- un fichier d'information sur les marqueurs.

Le challenge étant terminé nous disposons d'un fichier supplémentaire qui correspond aux phénotypes des 900 animaux, qui n'avaient pas été fournis initialement. Nous avons donc une population d'apprentissage de 2326 individus et une population de validation de 900 individus.
Le processus de contrôle qualité des données supprime 263 SNPs. Nous travaillons donc sur un ensemble de 9768 SNPs.

Le jeu de données du ***15ème workshop QTLMAS de 2011*** [Elsen 2012] est généré à partir de :

- 20 mâles,
- 10 femelles par mâles (200 femelles),
- 15 progénitures par femelle (3000 veaux).

Sur un total de 3220 individus, le phénotype de 1000 animaux (5 progénitures sur

15 par femelle) n'est pas donné lors du challenge. En effet, 4 fichiers textes sont fournis aux challengers :

- les génotypes des 3220 individus sur 9990 SNPs,
- les phénotypes de 2000 individus,
- le pedigree des 3220 animaux,
- un fichier d'information sur les marqueurs.

Le challenge étant terminé nous disposons d'un fichier supplémentaire qui correspond aux phénotypes des 1000 animaux, qui n'avaient pas été fournis initialement. Nous avons donc une population d'apprentissage de 2000 animaux et une population de validation de 1000 animaux.
Le processus de contrôle qualité des données supprime 2869 SNPs. Nous travaillons donc sur un ensemble de 7121 SNPs.

1.4.2.2 Données simulées

La simulation de génotype étant difficile à mettre en place du fait notamment de liaison entre variables (SNPs) ainsi qu'entre individus, nous proposons d'utiliser les génotypes de QTLMAS 2010 et 2011 et de générer uniquement le caractère $\mathbf{y}$, afin de connaître les variables significatives et leurs coefficients. La matrice de SNPs utilisée ici est donc composée de $n = 2326$ (resp. 3000) individus et $p = 9768$ (resp. 7121) variables pour les données simulées à partir de QTLMAS 2010 (resp. QTLMAS 2011). Notre simulation est basée sur le modèle de régression classique $\mathbf{y} = X\boldsymbol{\beta} + \mathbf{e}$ où $\mathbf{y}$ est le caractère à prédire, X est la matrice des génotypes (SNPs), $\boldsymbol{\beta}$ les effets de chaque marqueur et $\mathbf{e}$ l'erreur résiduelle.
Les étapes de la simulation des données sont les suivantes :

1. Chargement de X
2. Tirage aléatoire de 96 variables qui seront significatives parmi celles de X
3. Génération des $\boldsymbol{\beta}$ suivant trois lois normales :
   - $\beta_j \sim \mathcal{N}(4, 1)$ pour 32 variables parmi les 96,
   - $\beta_j \sim \mathcal{N}(16, 1)$ pour 32 autres,
   - $\beta_j \sim \mathcal{N}(64, 1)$ pour les 32 restantes.
4. Génération de e :
   $\mathbf{e} \sim \mathcal{N}(0, 4)$
5. Calcul de $\mathbf{y}$:
   $y_i = \sum_{j=1}^{96} \beta_j x_{ij} + e_i$, (les $\mathbf{x}_j$ sont ici les variables extraites de X à l'étape 2).

Nous choisissons 96 SNPs significatifs parmi tous les SNPs disponibles car 96 SNPs est une taille standard de mini-puce. Nous simulons les coefficients $\boldsymbol{\beta}$ suivant trois lois normales différentes afin d'obtenir des effets plus ou moins forts et donc des variables plus ou moins difficiles à retrouver.

1.5 Modélisation du problème

Comme présenté dans la section 1.3.2, l'objectif est d'élaborer un modèle prédictif pour un caractère quantitatif tout en sélectionnant un nombre réduit de variables. Nous sommes dans le cadre d'un modèle de sélection de variables en régression que nous proposons d'étudier en deux sous problèmes : la régression en grande dimension et la sélection de variables. Nous avons également vu en section 1.2.2 l'importance des relations familiales et nous proposerons donc un modèle les intégrant. Par la suite nous utiliserons les notations suivantes :

- n : nombre d'individus
- p : nombre de variables (SNPs)
- $x_{ij} \in \{0, 1, 2\}$: valeur du $j^{\text{ème}}$ SNP pour le $i^{\text{ème}}$ individu ($1 \leq j \leq p, 1 \leq i \leq n$)
- $y_i \in \mathbb{R}$: valeur du caractère étudié pour l'individu i. Comme vu dans la section 1.2.2, ce seront ici des EBVs ou DRPs.
- e_i : résidu gaussien. Les résidus sont supposés indépendants et identiquement distribués (i.i.d.) d'espérance nulle et de variance $\sigma^2_{\mathbf{e}}$.

1.5.1 Modèle de régression classique

La prédiction d'un caractère quantitatif à partir de variables quantitatives est classiquement représenté par le modèle de régression suivant :

$$y_i = \beta_0 + \sum_{j=1}^{p} (\beta_j x_{ij}) + e_i, \qquad i = 1, \ldots, n, \tag{1.2}$$

qui peut également s'écrire :

$$\begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_{1,1} & \ldots & x_{1,p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n,1} & \ldots & x_{n,p} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} + \begin{pmatrix} e_1 \\ \vdots \\ e_n \end{pmatrix} \tag{1.3}$$

soit :

$$\mathbf{y} = X\boldsymbol{\beta} + \mathbf{e}, \tag{1.4}$$

où X la matrice $n \times (p+1)$ des SNPs (la première colonne de 1 indique que c'est une régression avec constante), $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)^t$ est le vecteur des paramètres associés aux SNPs, que l'on doit estimer. La résolution de ce modèle engendre des difficultés lorsque nous étudions des données de grandes dimensions et en particulier lorsque $n < p$. Les limites ainsi que les méthodes classiques traitant cette problématique sont relatées dans la deuxième partie du chapitre suivant (cf. section 2.2).

1.5.2 Modèle de régression avec sélection de variables

L'objectif conjoint au problème de prédiction est de sélectionner un sous-ensemble de variables d'intérêt. Nous proposons donc une modélisation intégrant

au modèle précédent un paramètre γ qui représente les SNPs présents dans le modèle :

$$\begin{gathered} y_i = \beta_0 + \sum_{j=1}^{p} (\beta_j \gamma_j x_{ij}) + e_i, \qquad i = 1, \ldots, n, \\ \mathbf{y} = X(\boldsymbol{\beta} \cdot \boldsymbol{\gamma}) + \boldsymbol{e}, \end{gathered} \tag{1.5}$$

où $\gamma_j = (\gamma_1, \ldots, \gamma_p)^t$ vaut 1 si la variable j est dans le modèle 0 sinon. L'opérateur $\cdot$ correspond au produit terme à terme des vecteurs. Les paramètres $\boldsymbol{\gamma}$, $\sigma^2_{\mathbf{e}}$, β_0 et $\{\beta_j : \gamma_j = 1, 1 \leq j \leq p\}$ doivent être estimés. Les différentes méthodes de sélection de variables sont présentées dans la quatrième partie du chapitre suivant (cf. section 2.4).

Cependant, ce modèle de régression, prend pour hypothèse l'indépendance entre les individus. Or, en sélection animale, il existe des relations entre animaux, souvent connues (pedigree) qui peuvent avoir une influence sur la qualité des prédictions (voir section 1.2.2).

1.5.3 Intégration des relations familiales

Afin de prendre en compte les relations familiales connues entre les animaux, nous proposons de modéliser le modèle de prédiction avec intégration des relations familiales à l'aide d'un modèle mixte :

$$\begin{gathered} y_i = \beta_0 + \sum_{j=1}^{p} (\beta_j \gamma_j x_{ij}) + \sum_{k=1}^{q} (z_{ik} u_k) + e_i, \qquad i = 1, \ldots, n, \\ \mathbf{y} = X(\boldsymbol{\beta} \cdot \boldsymbol{\gamma}) + Z\mathbf{u} + \boldsymbol{e}, \end{gathered} \tag{1.6}$$

où $\mathbf{u} \sim N(0, \sigma^2_{\mathbf{u}} A)$ est une variable aléatoire représentant les effets individuels des animaux, avec A la matrice de relations entre animaux, calculée à partir du pedigree et Z est la matrice d'incidence de ces effets aléatoires. Les paramètres $\boldsymbol{\gamma}$, $\sigma^2_{\mathbf{e}}$, β_0 et $\{\beta_j : \gamma_j = 1, 1 \leq j \leq p\}$ sont à estimer, la variable aléatoire $\mathbf{u}$ à prédire. Les méthodes génomiques classiques dont celles prenant en compte les relations familiales entre les animaux sont exposées en troisième partie du chapitre suivant (cf. section 2.3).

1.6 Conclusion

Dans ce chapitre nous avons présenté le contexte de notre travail, avec notamment l'évolution des programmes de sélection animale avec le passage du testage sur descendance à la sélection génomique. Dans le cadre de la sélection génomique, nous avons décrit la problématique soulevée par *Gènes Diffusion*, les données que nous allons étudier pour finir par présenter la modélisation que nous proposons du problème de sélection de variable en régression. Des méthodes classiques permettant d'aborder cette problématique sont présentées dans le chapitre suivant.

Chapitre 2

Régression en grande dimension et sélection de variables

Sommaire

Avec le développement des technologies d'acquisition de données et des moyens informatiques notamment, les données mesurées disponibles sont de plus en plus nombreuses et ce dans différents domaines comme la finance, la météorologie, l'imagerie ou la biologie. C'est le cas des données génomiques. Cependant, dans le contexte de la génomique, le coût d'acquisition de ces données étant important et les mesures par prélèvement parfois difficiles, le nombre de mesures disponibles peut être limité. Nous avons donc en général un grand nombre de variables mesurées (p variables) pour un nombre limité d'individus (n) et sommes ainsi confrontés à un problème de régression en grande dimension, communément noté $n < p$ dans la littérature. Nous présentons ce problème dans la première partie de ce chapitre. Nous introduisons ensuite les méthodes classiques de régression en grande dimension et leur application en génomique. Puis dans une troisième partie, nous présentons les méthodes spécifiques à l'évaluation génétique. Une des solutions du problème de régression en grande dimension, qui constitue également un objectif de notre étude, consiste

à sélectionner un sous-ensemble de variables pertinentes, ce que nous proposons de faire en utilisant des techniques d'optimisation combinatoire. La quatrième partie de ce chapitre expose donc les différentes méthodes d'optimisation combinatoire pour la sélection de variables proposées dans la littérature.

2.1 Le cas $n < p$

Un de nos objectif est de déterminer un modèle prédictif pour un caractère quantitatif. Nous sommes donc dans le cadre d'une régression dont le modèle classique s'écrit sous la forme :

$$\mathbf{y} = X\boldsymbol{\beta} + \mathbf{e}, \tag{2.1}$$

- $\mathbf{y}$ est un vecteur colonne de taille n (nombre d'individus) et représente la variable à prédire,
- X est une matrice $n \times (p+1)$ des variables explicatives $(\mathbf{1}, \mathbf{x}_1, \ldots, \mathbf{x}_p)$,
- $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_p)^t$ est un vecteur colonne de taille $p+1$ représentant les effets associés à chaque variable explicative de la matrice X,
- et $\mathbf{e} \sim \mathcal{N}(\mu, \sigma^2_{\mathbf{e}})$ est une variable aléatoire, d'espérance μ et de variance $\sigma^2_{\mathbf{e}}$, représentant l'erreur résiduelle.

L'objectif est de déterminer une estimation des coefficients de régression $\boldsymbol{\beta}$ (notée $\hat{\boldsymbol{\beta}}$), ce qui est classiquement fait à l'aide de la méthode des moindres carrés (OLS - *Ordinary Least Square*) selon l'équation :

$$\hat{\boldsymbol{\beta}}_{OLS} = (X^t X)^{-1} X^t \mathbf{y}, \tag{2.2}$$

où X^t est la transposée de la matrice X. Une condition nécessaire à l'estimation des $\boldsymbol{\beta}$ est donc que la matrice $X^t X$ soit inversible. Or, dans le cas où le nombre d'individus n est plus faible que le nombre de variables p, comme dans notre étude, cette matrice est singulière (non inversible) et il n'y pas de solution unique pour les $\boldsymbol{\beta}$. De plus, même si $n > p$, en présence de fortes corrélations entre les variables, la matrice $X^t X$ est inversible mais mal conditionnée ce qui générera une grand variance des $\hat{\boldsymbol{\beta}}$. Des méthodes permettant de contourner ces problèmes ont donc été proposées et sont décrites dans les sections suivantes.

2.2 Les méthodes de régression en grande dimension

Le problème de régression en grande dimension est un problème classique pour lequel de nombreuses méthodes ont été proposées. Cette partie présente des méthodes classiquement utilisées dans la littérature, regroupées en trois catégories : les méthodes de sélection de variables, les méthodes de régression pénalisée et les méthodes de régression sur composantes (regroupement de variables). Le lecteur pourra se référer au livre de Hastie, Tibshirani et Friedman [Hastie 2009] pour une revue plus détaillée des approches couramment utilisées.

2.2.1 Méthodes de sélection d'un sous-ensemble de variables

Afin de diminuer la dimension des données à étudier, des méthodes de sélection d'un sous-ensemble de variables ont été proposées notamment pour faciliter l'interprétation du modèle obtenu. Une première méthode consiste à évaluer le modèle de régression avec tous les sous-ensembles de taille $k \in \{0, 1, 2, \ldots, p\}$ possibles afin de choisir le *meilleur sous-ensemble*. Cette méthode, qui nécessite de fixer au préalable la valeur de k, n'est calculatoirement pas réalisable lorsque le nombre de variables p est grand, le nombre de sous-ensembles possibles à tester étant trop élevé. Une alternative est l'utilisation de méthodes de sélection pas à pas comme la sélection *forward* qui démarre avec aucune variable et ajoute une à une les variables qui améliorent le plus le modèle. Mais cette approche est coûteuse en temps lorsque le nombre p de variables est très grand.

Application en génomique

Ces approches de sélection pas à pas, bien que coûteuses, ont été appliquées dans des contextes particuliers en génomique. Ainsi, dans leurs publications, [Mai 2010] et [Habier 2007] présentent une approche pas à pas *forward* simple marqueur où les SNPs sont considérés un à un. La significativité du SNP est évaluée par un test de Student (t-test) dont l'hypothèse nulle est que le marqueur n'a pas d'effet sur le trait, alors que l'hypothèse alternative est que le marqueur a un effet sur le trait (il est en déséquilibre de liaison avec un QTL). Celui dont la p-valeur est la plus faible est intégré au modèle. Les effets des marqueurs sont ensuite évalués par régression multiple sur l'ensemble des marqueurs sélectionnés. L'étude porte sur un grand nombre de marqueurs et une première limite de cette méthode concerne le problème des tests multiples. En effet, la répétition de tests de Student peut rendre des SNPs significatifs alors qu'en réalité ils ne le sont pas : ce sont des *faux positifs*. Plusieurs méthodes permettent de réguler ce taux de faux positifs dont la plus connue est la méthode de Bonferroni utilisée dans [Mai 2010] qui corrige le seuil de significativité (α). Mais cette correction reste conservative et ne tient pas compte d'éventuelles liaisons entre les tests étant donné que deux marqueurs peuvent être liés. D'autres alternatives à la correction de Bonferroni sont donc proposées, les plus connues étant présentées dans [Bar-hen 2005].

Le problème majeur de ces méthodes est que l'interaction possible entre marqueurs n'est pas prise en compte lors du choix des marqueurs à intégrer au modèle, puisque ce choix est basé sur des tests univariés.

2.2.2 Méthodes de régression pénalisée

En régression en grande dimension ($n < p$), le problème principal est l'impossibilité d'inverser la matrice X^tX nécessaire à l'estimation des $\boldsymbol{\beta}$ par moindres carrés ordinaires. Des méthodes de régularisation introduisant une pénalité $K(\boldsymbol{\beta})$ sur les coefficients à estimer sont proposées selon le schéma suivant :

$$\hat{\boldsymbol{\beta}} = argmin\{RSS(\boldsymbol{\beta}) + \lambda K(\boldsymbol{\beta})\}. \tag{2.3}$$

où $RSS(\boldsymbol{\beta}) = \sum_i (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2$ est la somme des carrés des résidus (*Residual Sum of Square* - RSS), λ est le paramètre de pénalisation ($\lambda \geq 0$) qui contrôle le compromis entre perte de précision et complexité du modèle, sa valeur étant généralement déterminée par validation croisée : plusieurs valeurs de λ sont testées, celle permettant d'obtenir la plus faible erreur de prédiction sur un échantillon de validation est conservée. Trois principales méthodes ont été proposées. Elles différent par la fonction de pénalité $K(\boldsymbol{\beta})$ utilisée.

Ridge

La *régression ridge* (Hoerl et Kennard 1970 [Hoerl 1970]) pénalise les coefficients de la régression en imposant une pénalité L_2 sur leur taille.

$$K(\boldsymbol{\beta})_{ridge} = \sum_{j=1}^{p} \beta_j^2. \tag{2.4}$$

L'estimation des coefficients de la régression ridge se fait alors suivant l'équation 2.5.

$$\hat{\boldsymbol{\beta}}_{ridge} = (X^t X + \lambda I)^{-1} X^t \mathrm{y}. \tag{2.5}$$

Dans le cas de la régression ridge, β_0 n'est pas pénalisé et est estimé préalablement par $\hat{\mathbf{y}} = \frac{1}{n}\sum_{i=1}^{n} y_i$, il n'y a donc pas de colonne de 1 dans la matrice X et $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)$. La principale différence par rapport à l'estimation des coefficients par moindres carrés ($\hat{\boldsymbol{\beta}}_{OLS}$) est l'ajout d'un paramètre λ sur la diagonale de la matrice $(X^t X)$, qui la rend inversible. Cette pénalité est particulièrement efficace pour le problème de corrélation entre variables. En effet, lorsque deux variables sont fortement corrélées, un fort coefficient positif sur l'une des deux peut être gommé par un fort coefficient négatif sur l'autre. En imposant une contrainte sur la taille des coefficients, ce problème est évité.
Une limite importante de cette méthode est qu'elle ne favorise pas la sélection de variables. Or la sélection d'un sous-ensemble de variables est un des objectifs de notre étude.

Lasso

La méthode *lasso*, introduite par Tibshirani [Tibshirani 1994] pénalise les coefficients de la régression à l'aide d'une pénalisation de type L_1 :

$$K(\boldsymbol{\beta})_{lasso} = \sum_{j=1}^{p} |\beta_j|. \tag{2.6}$$

Cette nouvelle pénalité L_1, contrairement à la pénalité L_2 utilisée par la méthode ridge, ne permet pas d'obtenir une solution analytique pour l'estimation des coefficients. Cependant, des algorithmes efficaces comme le *Least Angle Regression* -

LAR [Efron 2004] existent. Cette pénalisation permet de réduire la dimension du problème en forçant certains coefficients à être nuls. Cependant, bien qu'efficace pour sélectionner des variables, la régression lasso devient instable pour des données de grande dimension et ses performances sont réduites dans le cas de corrélation entre variables ([Zou 2005]).

Elastic net

Une méthode permettant de combiner les avantages des deux approches précédentes (ridge et lasso), appelée *elastic net* (EN) est proposée par Zou et Hastie [Zou 2005]. La fonction de pénalité associée à cette méthode est alors une combinaison des deux fonctions précédentes :

$$K(\boldsymbol{\beta})_{EN} = \alpha K(\boldsymbol{\beta})_{lasso} + (1-\alpha)K(\boldsymbol{\beta})_{ridge} = \sum_{j=1}^{p}(\alpha|\beta_j| + (1-\alpha)\beta_j^2). \qquad (2.7)$$

Elastic net sélectionne des variables comme lasso et réduit conjointement les coefficients des variables corrélées comme ridge. Notons que dans le cas particulier où $\alpha = 0$ on retrouve l'équation de la régression ridge et quand $\alpha = 1$ celle de la régression lasso. Un inconvénient de cette méthode est la nécessité de définir le paramètre α en plus du paramètre λ (généralement par validation croisée).

Application en génomique

Les méthodes de régression pénalisée sont largement utilisées dans divers problèmes de génomique. Waldron *et al.* [Waldron 2011] appliquent ces méthodes sur des données simulées ainsi que sur des données d'expressions géniques (*microarray*) relatives au cancer et sur des données métagénomiques pour étudier l'obésité. Ils montrent que les performances des méthodes vont dépendre de la structure du jeu de données. La méthode lasso donne de meilleurs résultats lorsqu'il y a peu de variables ayant un effet alors que ridge donnera de meilleurs résultats en présence de beaucoup de variables avec une forte corrélation. Cependant, généralement, elastic net donnera des résultats similaires à la meilleure des deux approches précédentes.

Dans leurs publications, Usai *et al.* [Usai 2010], [Usai 2009] présentent une application de la régression lasso en sélection génomique et montrent que cette méthode donne d'aussi bons voire meilleurs résultats que les méthodes plus classiquement utilisées (BLUP et BayesA présentées en section 2.3). Cette méthode est également appliquée par Xu *et al.* [Xu 2007], qui montrent qu'elle donne des résultats satisfaisants pour estimer les effets de QTL. En 2012, Ogutu *et al.* [Ogutu 2012] présentent une application des méthodes de régularisation à la sélection génomique. Ils considèrent les 3 méthodes présentées précédemment (ridge, lasso et EN) ainsi que 3 de leurs extensions (*ridge regression BLUP*, *adaptive lasso* et *adaptive EN*) et montrent que les méthodes de type lasso et elastic net donnent de meilleurs résultats, sur données simulées, que celles basées sur ridge.

2.2.3 Méthodes de régression sur composantes

Afin de réduire la dimension du problème étudié tout en considérant les fortes corrélations qui peuvent exister entre variables, des méthodes de régression sur composantes (regroupement de variables) sont proposées. Le principe de ces approches est de réduire la dimension des données en générant, à partir des données de base $(\mathbf{x}_1, \ldots, \mathbf{x}_p)$, un nombre réduit de composantes latentes $(\mathbf{z}_1, \ldots, \mathbf{z}_m$, $m < p)$, choisies à l'aide de critères comme la règle de Kaiser ou le test du coude (*scree test*) [Zwick 1986], sur lesquelles on peut ensuite appliquer une régression par moindres carrés. On trouve par exemple les méthodes de régression sur composantes principales et de régression par moindres carrés partiels.

Régression en composantes principales

La régression en composantes principales (*Principal Component Regression* - PCR [Jolliffe 1982]) est une régression sur les composantes d'une analyse en composantes principales (*Principal Component Analysis* - PCA [Pearson 1901]). Le principe de la PCA est de créer de nouvelles variables non corrélées, combinaisons linéaires des variables initiales, maximisant la perte d'information en projection.

Régression par moindres carrés partiels

Contrairement à la PCR dans laquelle la réduction de dimension est faite indépendamment de $\mathbf{y}$, la régression par moindres carrés partiels (*Partial Least Squares* - PLS [Wold 2004]) construit des *variables latentes* en fonction de X et de $\mathbf{y}$. L'objectif de cette approche est de maximiser la covariance entre les variables de X et la variable à prédire ($\mathbf{y}$). Boulesteix et Strimmer [Boulesteix 2007] montrent que cette méthode peut être utilisée autant pour de la classification que pour de la régression, de la sélection de variables ou encore de l'analyse de survie.

Application en génomique

Dans la revue de 2007 précédemment citée, [Boulesteix 2007] montrent que la régression PLS est parfaitement adaptée aux analyses de données génomiques de grande dimension. Croiseau *et al.* [Croiseau 2010] obtiennent, avec la régression PLS, des résultats comparables à certaines méthodes classiques d'évaluation génomique des bovins. Horne *et al.* [Horne 2004] montrent la pertinence d'utiliser la PCR pour sélectionner avec succès un sous-ensemble de SNPs. En 2011, Long *et al.* [Long 2011] montrent le potentiel de deux nouvelles méthodes combinant réduction de dimension et sélection de variables pour prédire la production de lait dans la race Holstein : la *PCR supervisée* qui pré-sélectionne des marqueurs, à l'aide d'une analyse simple marqueur et de la statistique de Student, avant de générer les variables latentes et la *sparse PLS* qui réduit la dimension et sélectionne des variables simultanément en appliquant une pénalité L_1 aux vecteurs de poids de la PLS.

Bien que donnant de bons résultats en prédiction, une limite principale de ces méthodes réside dans l'interprétation du modèle obtenu ; les nouvelles variables sur lesquelles est construit le modèle étant des combinaisons des variables initiales,

aucune sélection de variables n'est faite ce qui ne répond pas à notre problématique.

Une méthode spécifique aux données génomiques et basée sur le regroupement de variables consiste à utiliser des haplotypes. Un *haplotype* est un ensemble de marqueurs situés sur un même chromosome et habituellement transmis ensemble. Des marqueurs forment un haplotype s'ils sont en fort déséquilibre de liaison les uns avec les autres. L'utilisation d'haplotypes pour prédire un caractère [Meuwissen 2001] permet de réduire la dimension des données analysées, le nombre d'haplotypes étant inférieur au nombre de marqueurs. Cette approche, contrairement aux méthodes PCR ou PLS prend en compte une caractéristique biologique (le déséquilibre de liaison) pour regrouper les marqueurs.

2.3 Méthodes spécifiques à l'amélioration génétique

L'amélioration génétique des animaux repose notamment sur la sélection des "bons" animaux qui participeront à la formation de la génération suivante. Afin de déterminer si un animal sera bon pour un caractère donné, nous devons être capables de prédire la valeur de ce caractère à partir de ses marqueurs. Les caractères étudiés dans ce travail étant quantitatifs le modèle statistique est une régression. Cependant, comme vu dans la section 2.1, le modèle de régression classique n'est pas adapté. De plus, comme il sera expliqué ci-après, en génétique animale, contrairement aux principales études menées en humain, les individus étudiés ne sont pas indépendants ; il y a généralement beaucoup de frères et demi-frères et ces relations familiales sont connues. Les méthodes classiques d'amélioration génétique proposent donc d'utiliser les modèles mixtes permettant d'intégrer ces relations familiales au modèle de prédiction. Nous abordons dans un premier temps la possibilité de prendre en compte la non indépendance entre individus, puis nous présentons ce qu'est un modèle mixte et comment l'estimer. Enfin, nous exposons les méthodes d'évaluation génétique proposées dans la littérature.

2.3.1 Prise en compte de la non indépendance entre animaux

Les études menées en génétique animale peuvent être réalisées sur des animaux issus d'un même troupeau, sur plusieurs générations, avec donc certains animaux qui seront issus de la même famille. [Wientjes 2013] montrent l'impact des relations familiales sur la qualité des prédictions génomiques. Ces relations entre animaux sont généralement représentées sous forme d'une matrice $n \times n$ (n étant le nombre d'animaux), calculée soit à partir du pedigree, soit à partir de l'information des marqueurs.

Pedigree Le *pedigree*, comme présenté en section 1.2.2 permet d'identifier les relations familiales entre les animaux. Il peut être représenté sous forme de tableau (Table 2.1) correspondant à la liste des individus avec leurs parents lorsqu'ils sont

connus. Il peut également être représenté sous forme d'arbre de descendance (Figure 2.1) pour plus de lisibilité lorsqu'il y a peu d'individus.

ID animal	ID père	ID mère
A1	0	0
A2	0	0
A3	0	0
A4	0	0
A5	A1	A2
A6	A3	A4
A7	A1	A2
A8	A5	A7

TABLE 2.1 – Pedigree

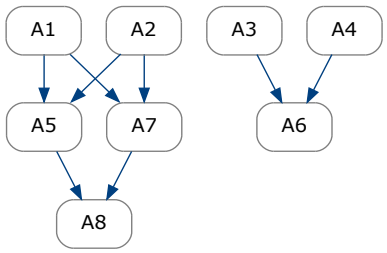

FIGURE 2.1 – Arbre de descendance

Afin d'identifier les relations familiales entre animaux, il est possible de construire une matrice de parenté A (Table 2.2) où : a_{ij} sera par exemple égal à $0,5$ si l'animal i est le fils de l'animal j puisque la moitié de l'information génétique d'un individu provient de sa mère, l'autre moitié de son père.

	A1	*A2*	*A3*	*A4*	*A5*	*A6*	*A7*	*A8*
A1	1	0	0	0	0.5	0	0.5	0.5
A2	0	1	0	0	0.5	0	0.5	0.5
A3	0	0	1	0	0	0.5	0	0
A4	0	0	0	1	0	0.5	0	0
A5	0.5	0.5	0	0	1	0	0.5	0.75
A6	0	0	0.5	0.5	0	1	0	0
A7	0.5	0.5	0	0	0.5	0	1	0.75
A8	0.5	0.5	0	0	0.75	0	0.75	1.25

TABLE 2.2 – Matrice de parenté A

Soit deux individus i et j, avec p et m les parents de i et p' et m' les parents de j, nous avons alors :

- $a_{ii} = 1 + 0.5 * a_{pm}$,
- si j est parent de i : $a_{ij} = 0.5 * a_{pj} + 0.5 * a_{mj}$,
- si i est parent de j : $a_{ij} = 0.5 * a_{ip'} + 0.5 * a_{im'}$,
- sinon : $a_{ij} = 0.25 * a_{pp'} + 0.25 * a_{pm'} + 0.25 * a_{mp'} + 0.25 * a_{mm'}$.

Notons que a_{ii} sera supérieur à 1 en présence de consanguinité entre individus. Sur l'exemple précédent $a_{88} = 1.25$ car $A8$ est le fils de $A5$ et $A7$ qui sont frères et sœurs. L'intégration de cette matrice de parenté dans le modèle étudié permettra de tenir compte des relations familiales entre animaux.

Information issue des marqueurs Une autre possibilité est de re-construire la matrice de liens entre individus à partir de la matrice des marqueurs (génotype). Cette matrice, notée G, est proportionnelle au produit croisé des génotypes ($G \propto XX^t$ où X est la matrice des marqueurs). Dans la matrice A, si deux animaux i et j sont frères, alors $a_{ij} = 0.5$. Or, on sait qu'en réalité la part d'information partagée par deux frères n'est pas exactement égale à 0.5 en raison de l' "aléa de méiose" (processus aléatoire de division des cellules). La matrice G permet d'obtenir une information plus précise, deux frères pourront par exemple avoir la valeur 0.58 ou 0.45 suivant qu'ils partagent plus ou moins d'information génomique.

2.3.2 Le modèle mixte

L'intégration des relations familiales implique l'utilisation d'un modèle mixte [Pinheiro 2000], qui est un modèle comportant à la fois au moins un effet fixe et au moins un effet aléatoire. Un *effet fixe* est un effet pour lequel les paramètres sont associés à une population entière. Un *effet aléatoire* est un effet pour lequel les données sont observées sur un échantillon de la population. L'introduction d'effets aléatoires permet de distinguer la variation dûe aux erreurs, de la variation dûe à ces effets aléatoires. Il n'est cependant pas toujours évident de savoir si un effet doit être considéré comme fixe ou aléatoire dans un modèle.

L'équation d'un modèle mixte peut s'écrire comme suit :

$$\mathbf{y} = X\boldsymbol{\beta} + Z\mathbf{u} + \mathbf{e}, \tag{2.8}$$

avec :

- $\mathbf{y} = (y_1, y_2, \ldots, y_n)^t$ est un vecteur de taille n (le nombre d'individus) représentant la variable à prédire,
- $X\boldsymbol{\beta}$ sont les effets fixes du modèle avec $X = (\mathbf{1}, \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p)$ une matrice supposée fixe et connue de taille $n \times (p+1)$ (p étant le nombre de variables) avec x_{ij} l'observation de l'individu i pour la variable j et $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)^t$ le vecteur des effets associés à estimer,
- $Z\mathbf{u}$ sont les effets aléatoires avec Z la matrice des indicatrices de taille $n \times q$ et $\mathbf{u} \sim \mathcal{N}(0, \sigma^2_{\mathbf{u}}\Sigma)$ un vecteur aléatoire de taille q, Σ étant la matrice de variances co-variances associée,
- et $\mathbf{e} \sim \mathcal{N}(0, \sigma^2_{\mathbf{e}}R)$ le vecteur aléatoire des erreurs résiduelles de matrice de variances-covariances R.

À partir du modèle précédent (équation 2.8), l'objectif est d'estimer les variances $\sigma^2_{\mathbf{u}}$ et $\sigma^2_{\mathbf{e}}$, d'estimer les paramètres $\boldsymbol{\beta}$ et de prédire la variable aléatoire $\mathbf{u}$.
Dans la suite de cette partie nous présentons dans un premier temps deux méthodes d'estimation des paramètres d'un modèle mixte (la meilleure prédiction linéaire non biaisée et l'échantillonnage de Gibbs). Puis, nous introduisons les méthodes classiques utilisées en évaluation génétique que nous comparons ensuite.

2.3.3 Estimation des paramètres du modèle mixte

Lors de l'utilisation d'un modèle mixte, deux principales méthodes permettent d'estimer les paramètres du modèle.

Meilleur prédicteur linéaire non biaisé

Si l'on considère l'équation 2.8, l'objectif est d'estimer le vecteur de paramètres $\boldsymbol{\beta}$ et de prédire la variable aléatoire $\mathbf{u} \sim \mathcal{N}(0, \Sigma)$. Des méthodes de résolution adaptées à ces données sont donc proposées, permettant d'obtenir les meilleurs estimateurs (resp. prédicteurs) linéaires sans biais de $\boldsymbol{\beta}$ (resp. de $\mathbf{u}$) (*Best Linear Unbiased Prediction - BLUP, Best Linear Unbiased Estimation - BLUE*).

Une première méthode de résolution [Henderson 1963] est basée sur les moindres carrés généralisés et consiste à estimer les $\boldsymbol{\beta}$ suivant l'équation $X^t V^{-1} X\hat{\boldsymbol{\beta}} = X^t V^{-1}\mathbf{y}$ où $V = R + Z\Sigma Z^t$ [Ducrocq 1990]. Cependant, la matrice V pouvant être de grande dimension ($n \times n$), son inversion peut être très coûteuse ($O(n^3)$ par la méthode du pivot de Gauss). Une alternative est proposée par [Henderson 1950], qui montre que $\boldsymbol{\beta}$ et $\mathbf{u}$ sont solutions du système suivant :

$$\begin{pmatrix} X^t R^{-1} X & X^t R^{-1} Z \\ Z^t R^{-1} X & Z^t R^{-1} Z + \Sigma^{-1} \end{pmatrix} \begin{pmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\mathbf{u}} \end{pmatrix} = \begin{pmatrix} X^t R^{-1}\mathbf{y} \\ Z^t R^{-1}\mathbf{y} \end{pmatrix} \tag{2.9}$$

L'avantage de ce système est qu'il ne nécessite pas l'inversion de V. La matrice R, de même dimension que V doit quant à elle être inversée mais sera généralement une matrice identité. Il est ensuite possible d'obtenir simultanément les effets fixes et aléatoires à l'aide d'une méthode itérative [Henderson 1975] : supposons par exemple initialement que tous les $\mathbf{u}$ sont nuls et tous les $\boldsymbol{\beta}$ sauf β_1 sont nuls. Le système d'équations nous permet de déterminer une estimation de la valeur de β_1. En utilisant cette valeur pour β_1 et en supposant tous les autres coefficients nuls sauf β_2 on peut retrouver une estimation de la valeur de β_2. Une fois une estimation obtenue pour chaque paramètre ($\mathbf{u} = (u_1, \ldots, u_q)^t$ et $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_p)^t$), on ré-estime la valeur de chacun en utilisant les estimations des autres, et ce itérativement jusqu'à ce que les estimations ne changent plus.

Échantillonnage de Gibbs

Une autre approche permettant d'estimer les paramètres d'un modèle mixte se base sur l'*échantillonnage de Gibbs* (*Gibbs sampling*) qui est l'un des algorithmes de Monte Carlo par Chaîne de Markov (*Markov Chain Monte Carlo* [Robert 2004]) les plus couramment utilisés. Cette approche permet, à partir des distributions conjointes, de générer des échantillons aléatoires, par échantillonnage des distributions conditionnelles réajustées itérativement [Wang 1994].

En évaluation génétique, les approches principalement utilisées s'appuient sur ces deux types de méthodes. En particulier, une famille de méthodes s'appuie sur la méthode BLUP (pedigree-BLUP, G-BLUP, ...). D'autre part, les approches bayésiennes s'inspirent de l'échantillonnage de Gibbs. Ces méthodes sont présentées ci-

après. Pour plus de détails, une revue des méthodes de prédiction utilisées en sélection génomique est présentée par [De Los Campos 2013]. Des publications confrontant les différentes méthodes y sont répertoriées.

2.3.4 Les méthodes basées sur BLUP

Pedigree-BLUP

Avant la génomique, la méthode principalement utilisée pour l'évaluation génétique était le ***pedigree-BLUP***. Dans cette approche, les prédictions sont uniquement basées sur le phénotype et le pedigree.

$$\mathbf{y} = \mathbf{u} + \mathbf{e}, \tag{2.10}$$

- $\mathbf{y}$ est la caractère étudié,
- $\mathbf{u} \sim \mathcal{N}(0, \sigma^2_{\mathbf{u}} A)$ est l'effet aléatoire animal, A étant la matrice de parenté basée sur le pedigree,
- et $\mathbf{e} \sim \mathcal{N}(0, \sigma^2_{\mathbf{e}})$ sont les résidus gaussiens.

Cette approche permet de prendre en compte les relations familiales entre individus au travers du pedigree (matrice A).

G-BLUP

En 2008 VanRaden [VanRaden 2008] propose la méthode ***G-BLUP*** (Genomic BLUP), dans laquelle les marqueurs interviennent uniquement dans la construction de la matrice de variance-covariance des effets aléatoires.

$$\mathbf{y} = \mathbf{u} + \mathbf{e}, \tag{2.11}$$

- $\mathbf{y}$ est la caractère étudié,
- $\mathbf{u} \sim \mathcal{N}(0, \sigma^2_{\mathbf{u}} G)$ est l'effet aléatoire animal, G étant la matrice de liens entre les individus, obtenue grâce à la matrice des SNPs,
- et $\mathbf{e} \sim \mathcal{N}(0, \sigma^2_{\mathbf{e}})$ sont les résidus gaussiens.

Cette méthode permet de prendre en compte les liens entre animaux, obtenus grâce aux marqueurs. Cependant, le génotype n'intervenant pas directement dans le modèle, il n'est pas possible d'estimer d'effets pour ces marqueurs.

H-BLUP

Plus récemment, la méthode ***H-BLUP*** (*one-step* ou *single step* [Legarra 2009]) propose une nouvelle modélisation permettant d'utiliser à la fois la matrice de parenté A (basée sur le pedigree) et la matrice de relation entre individus G (basée sur les marqueurs) :

$$\mathbf{y} = \mathbf{u} + \mathbf{e}, \tag{2.12}$$

- $\mathbf{y}$ est la caractère étudié,

- $\mathbf{u} \sim \mathcal{N}(0, \sigma^2_{\mathbf{u}} H)$ est l'effet animal, H étant la matrice de liens entre les individus, combinaison des matrices A et G (cf. [Legarra 2009] pour plus de détails sur la construction de la matrice),
- et $\mathbf{e} \sim \mathcal{N}(0, \sigma^2_{\mathbf{e}})$ sont les résidus gaussiens.

Les méthodes G-BLUP et H-BLUP utilisent les marqueurs pour générer les matrices G et H de liens entre les individus mais ne permettent pas d'obtenir un effet pour chaque marqueur. Pour cela, une alternative consiste à les considérer comme effets aléatoires dans le modèle, en supposant une distribution *a priori* de l'effet de ces marqueurs.

RR-BLUP

La méthode ***Random Regression-BLUP (RR-BLUP)*** [Meuwissen 2001], propose de considérer les marqueurs comme effets aléatoires distribués selon une loi normale avec une variance égale pour tous les marqueurs. Cette méthode est également appelée ***SNP-BLUP***.

$$\mathbf{y} = Z\mathbf{u} + \mathbf{e}, \tag{2.13}$$

- $\mathbf{y}$ est la caractère étudié,
- Z est la matrice des marqueurs [1],
- $\mathbf{u} \sim \mathcal{N}(0, \sigma^2_{\mathbf{u}})$ est le vecteur des effets aléatoires associés aux marqueurs,
- et $\mathbf{e} \sim \mathcal{N}(0, \sigma^2_{\mathbf{e}})$ sont les résidus gaussiens.

Une comparaison des performances des méthodes SNP-BLUP, G-BLUP et H-BLUP en prédiction génomique sur des taureaux est réalisée dans l'étude de [Koivula 2012]. Ils montrent un léger avantage de l'utilisation de l'approche H-BLUP comparée aux deux autres approches.

2.3.5 Les méthodes bayésiennes

Les méthodes basées sur BLUP considèrent une distribution normale des marqueurs avec les mêmes effets pour tous les marqueurs. Cependant, pour certains caractères, seuls quelques gènes ont de gros effets et de nombreux ont de petits effets [Shrimpton 1988]. Des méthodes prenant en compte ce précepte sont donc proposées. C'est par exemple le cas des méthodes de régressions linéaires bayésiennes [Meuwissen 2001] qui supposent des distributions *a priori* différentes des effets des marqueurs. Le modèle utilisé est le suivant :

$$\mathbf{y} = Z\mathbf{u} + \mathbf{e}, \tag{2.14}$$

- $\mathbf{y}$ est la caractère étudié,
- Z est la matrice des marqueurs[1],
- $\mathbf{u} \sim \mathcal{N}(0, \sigma^2_{\mathbf{u}})$ est le vecteur des effets aléatoires associés aux marqueurs,

1. Notons qu'ici $Z = X$ mais nous conservons les notations classiques de la littérature.

– et $\mathbf{e} \sim \mathcal{N}(0, \sigma^2_{\mathbf{e}})$ sont les résidus gaussiens.

BayesA

La méthode ***BayesA*** suppose que tous les SNPs ont un effet mais que cet effet est différent d'un SNP à l'autre. Elle suppose donc une distribution normale des marqueurs avec une moyenne nulle et une variance spécifique à chaque marqueur. Chaque variance de marqueur est supposée distribuée selon une loi du Khi-deux inverse.

$$\sigma^2_{\mathbf{u}} \sim \chi^{-2}(v, S),$$

– v est le nombre de degrés de liberté, supposé connu,
– S est un paramètre de réduction, supposé connu.

BayesB

La méthode ***BayesB*** fait l'hypothèse que la variance des effets des SNPs est variable le long du génome et est nulle pour une proportion connue de SNPs. Cette approche suppose généralement que beaucoup de SNPs ont un effet nul.

$$\begin{array}{rl} \sigma^2_{\mathbf{u}} = 0, & \text{avec une probabilité } \pi, \\ \sigma^2_{\mathbf{u}} \sim \chi^{-2}(v, S), & \text{avec une probabilité } (1-\pi). \end{array}$$

– π est une proportion connue de SNPs nuls,
– v est le nombre de degrés de liberté, supposé connu,
– S est un paramètre de réduction, supposé connu.

Les principaux inconvénients des méthodes BayesA et BayesB sont exposés dans [Gianola 2009] : 1) les paramètres v et S doivent être fixés par l'utilisateur et vont avoir une influence majeure sur le taux de réduction appliqué aux effets des marqueurs, 2) la probabilité π qu'un SNP ait un effet nul est supposée connue ($\pi = 0$ dans BayesA de sorte que tous les SNPs aient un effet non nul, $\pi > 0$ dans BayesB qui suppose que certains SNPs ont un effet nul).

BayesCπ, BayesDπ

Considérant ces limites, [Habier 2011] définit récemment les approches ***BayesCπ*** et ***BayesDπ*** (ou BayesD), qui supposent que la proportion de SNPs nuls π est inconnue. BayesDπ suppose également que le paramètre S est inconnu.

$$\begin{array}{rl} \sigma^2_{\mathbf{u}} = 0 & \text{avec une probabilité } \pi, \\ \sigma^2_{\mathbf{u}} \sim \chi^{-2}(v, S) & \text{avec une probabilité } (1-\pi). \end{array}$$

– π est une proportion inconnue de SNPs nuls,
– v est le nombre de degrés de liberté, supposé connu,
– S est un paramètre de réduction, supposé connu dans BayesCπ et inconnu dans BayesDπ.

BayesR

Plus récemment, la méthode ***BayesR*** a été introduite par [Erbe 2012] et suppose qu'une proportion de marqueurs a un effet nul et que les autres ont des effets faibles à modérés. Pour cela les effets des SNPs sont supposés distribués selon un mélange de quatre lois normales ayant des variances différentes, la première avec une variance à 0 jusqu'à une variance égale à 1% de la variance génétique pour la dernière. La variance d'un SNP a donc 4 valeurs possibles.

$$\begin{array}{cl} \sigma^2_{\mathbf{u}} = 0 & \text{avec une probabilité } \pi_1, \\ \sigma^2_{\mathbf{u}} \sim \mathcal{N}(0, 10^{-4}\sigma^2_g) & \text{avec une probabilité } \pi_2, \\ \sigma^2_{\mathbf{u}} \sim \mathcal{N}(0, 10^{-3}\sigma^2_g) & \text{avec une probabilité } \pi_3, \\ \sigma^2_{\mathbf{u}} \sim \mathcal{N}(0, 10^{-2}\sigma^2_g) & \text{avec une probabilité } \pi_4. \end{array}$$

- π_1 est une proportion inconnue de SNPs nuls et π_2, π_3 et π_4 sont des proportions inconnues de SNPs suivant chaque loi,
- $\sigma^2_g = r^2\sigma^2$ est la variance génétique supposée, avec r^2 la fiabilité supposée du caractère et σ^2 la variance du caractère. La variance génétique est la variance des écarts à la moyenne dûs aux différences génétiques.

Intégration des relations familiales Les méthodes bayésiennes présentées dans cette section sont également adaptables pour tenir compte des relations familiales en ajoutant un effet aléatoire représentant ces relations. C'est ce que font par exemple Erbe *et. al* [Erbe 2012] dans leur application de la méthode BayesR.

2.3.6 Comparaison des différentes méthodes

Depuis une dizaine d'années, de nombreuses études des méthodes de prédiction sur données génomique ont été réalisées, dans un premier temps sur données simulées puis plus récemment sur données réelles. Nous allons discuter ici des performances de chacune, leurs limites et leurs utilisations.

Des études sur données simulées ont montré que la précision de prédiction sera fortement influencée par certaines caractéristiques du jeu de données étudié comme :

- la taille et les particularités de la population de référence (échantillon d'apprentissage),
- le degré de relations entre les animaux de l'échantillon d'apprentissage et de l'échantillon de validation [Calus 2010], [Habier 2007],
- le nombre de SNPs ayant un effet [Habier 2011],
- ou encore l'héritabilité du caractère étudié.

Le choix de la méthode à utiliser dépendra de l'architecture génétique du caractère étudié [Hayes 2010]. En effet, si on sait, pour un caractère donné, que la distribution des effets est normale, on préférera utiliser une méthode de type BLUP alors que si on sait que le caractère dépend de zones du génome ayant de gros effets, les méthodes bayésiennes seront privilégiées.

Les études sur données réelles [De Los Campos 2013] ont montré que la différence de qualité de prédiction entre différentes méthodes (BayesA, BayesB, G-BLUP,

lasso) est très faible avec un léger avantage pour les méthodes qui pénalisent les coefficients tout en sélectionnant des variables comme BayesB.
En 2009, Moser *et al.* [Moser 2009] comparent notamment une régression pas à pas (sélection *forward*), une méthode bayésienne (BayesR) et une régression PLS sur données réelles de taureaux laitiers. Ils montrent que toutes les méthodes donnent des résultats similaires à l'exception de la régression pas à pas, nettement moins performante. Une comparaison des méthodes pas à pas *forward*, BLUP, BayesA et BayesB est réalisée par Meuwissen *et al.* en 2001 [Meuwissen 2001] sur données simulées. Ils concluent également que les résultats de la méthode pas à pas sont moins bons que ceux des autres méthodes et que les méthodes bayésiennes sont meilleures en terme de prédiction que la méthode BLUP, même quand la distribution supposée des marqueurs n'est pas correcte.

Comme indiqué dans la revue de [De Los Campos 2013], les méthodes bayésiennes ainsi que la méthode G-BLUP sont les méthodes les plus fréquemment utilisées. La méthode G-BLUP est très utilisée notamment par habitude et facilité puisque la procédure de calcul est la même que la méthode BLUP classique, utilisée pendant de nombreuses années, à l'exception du remplacement de la matrice A par la matrice G dans les équations de résolution du modèle mixte de Henderson. De plus, G-BLUP suppose que tous les SNPs ont la même variance, qui n'a donc pas à être estimée.

Le tableau 2.3 récapitule les différentes méthodes de sélection de variables en sélection génomique qui sont proposées dans la littérature. Nous proposons de les décrire suivant 6 critères : les effets fixes et aléatoires considérés, la distribution des variances des effets aléatoires, la prise en compte ou non des relations familiales et des liens entre individus et enfin la possibilité ou non de sélectionner des variables. Les deux dernières lignes du tableau montrent les caractéristiques des approches que nous proposons par la suite.

L'objectif de notre travail est d'élaborer un modèle prédictif basé sur un sous-ensemble de SNPs intéressants, tout en prenant en compte les relations familiales entre animaux. Afin de ne pas faire de supposition sur la distribution des SNPs, nous proposons de les considérer comme effets fixes dans notre modèle, de même que les 7 premières méthodes classiques présentées dans le tableau récapitulatif. Parmi ces méthodes existantes, certaines sélectionnent des variables mais aucune ne permet de prendre en compte les relations familiales. Nous proposons d'aborder le problème de sélection de variables sous un nouvel angle en utilisant des méthodes d'optimisation combinatoire, dans un premier temps sans tenir compte des relations familiales puis par la suite nous les intégrerons. Nous exposons donc dans la section suivante un état de l'art des méthodes d'optimisation combinatoire pour la sélection de variables.

Méthode	**Effets fixes**	**Effets aléatoires**	**Variances des effets aléatoires**	**Rel. fam.**	**Liens ind.**	**Sélect. var.**	**Réf.**
FR-LS	SNPs	$\emptyset$	$\emptyset$	N	N	O	[Meuwissen 2001]
Ridge	SNPs	$\emptyset$	$\emptyset$	N	N	N	[Hoerl 1970]
Lasso	SNPs	$\emptyset$	$\emptyset$	N	N	O	[Tibshirani 1994]
Elastic net	SNPs	$\emptyset$	$\emptyset$	N	N	O	[Zou 2005]
PCR	SNPs	$\emptyset$	$\emptyset$	N	N	N	[Jolliffe 1982]
PLS	SNPs	$\emptyset$	$\emptyset$	N	N	N	[Tenenhaus 1998]
sparse PLS	SNPs	$\emptyset$	$\emptyset$	N	N	O	[Long 2011]
Pedigree-BLUP (RR-BLUP)	$\emptyset$	rel. fam.	$\sigma^2 A$	O	N	N	[Meuwissen 2001]
Genomic-BLUP	$\emptyset$	liens ind.	$\sigma^2 G$	N	O	N	[VanRaden 2008]
H BLUP	$\emptyset$	rel. fam, liens ind.	$\sigma^2 H$	O	O	N	[Legarra 2009]
SNP-BLUP	$\emptyset$	SNPs	σ^2	N	N	O	[Meuwissen 2001]
BayesA	$\emptyset$	SNPs	$\sim \chi^{-2}$			N	[Meuwissen 2001]
BayesB	$\emptyset$	SNPs	$= 0$ avec proba π connue $\sim \chi^{-2}$ avec proba $(1-\pi)$			O	[Meuwissen 2001]
BayesCπ	$\emptyset$	SNPs	$= 0$ avec proba π inconnue χ^{-2} avec proba $(1-\pi)$			O	[Habier 2011]
BayesDπ	$\emptyset$	SNPs	$= 0$ avec proba π inconnue χ^{-2} avec proba $(1-\pi)$ et S inconnu			O	[Habier 2011]
BayesR	$\emptyset$	SNPs	mélange de 4 $\mathcal{N}$			O	[Erbe 2012]
Approche proposée 1	SNPs	$\emptyset$	$\emptyset$	N	N	O	
Approche proposée 2	SNPs	rel. fam.	$\sigma^2 A$	O	N	O	

Table 2.3 – Récapitulatif des méthodes classiques

Effets aléatoires : rel. fam. : relations familiales (pedigree), liens ind. : liens entre individus (obtenus à partir des marqueurs).
Variances : S : paramètre de réduction de la loi du Khi-deux.
Sélect. var. : sélection de variables.
O : oui, N : non.

2.4 Optimisation combinatoire pour la sélection de variables

2.4.1 Motivations

Comme indiqué dans l'article [Corne 2012], les méthodes d'optimisation combinatoire sont un moyen efficace de traiter le problème de sélection de variables (souvent appelé "attributs" en extraction de connaissances). En effet, rechercher un sous-ensemble de variables pertinentes, revient à se poser la question du meilleur sous-ensemble par rapport à un échantillon d'évaluation donné. C'est un problème de nature combinatoire. Une recherche exhaustive de tous les sous-ensembles peut être réalisée si le nombre de variables est faible. Cependant, dans cette thèse nous étudions des jeux de données dont le nombre de variables varie entre 5 000 et 50 000. Dans ce contexte de grande dimension, le problème est connu pour être NP-difficile [Amaldi 1997] et un algorithme exhaustif devient calculatoirement impossible à mettre en place. C'est pourquoi des méthodes approchées et en particulier des métaheuristiques ont été proposées pour aborder ce problème, et permettre de trouver de très bons sous-ensembles à défaut de trouver le meilleur.

Dans ce chapitre, nous présentons dans un premier temps les différentes métaheuristiques regroupées en deux catégories : les méthodes à solution unique et les méthodes à base de population de solutions. Dans la première catégorie, nous détaillons plus particulièrement la recherche locale itérée (ILS - *Iterated Local Search*) et dans la deuxième catégorie l'algorithme génétique (GA - *Genetic Algorithm*), qui sont les deux méthodes sur lesquelles nous nous appuyons dans ce travail. Ce chapitre se termine par un état de l'art des métaheuristiques pour la sélection de variables dans le cadre de problèmes de classification ou de régression. En effet, nous sommes dans le cadre d'un problème de régression, or, un plus grand nombre de méthodes ont été utilisées dans le contexte de la classification, mais elles sont facilement adaptables à la régression : la fonction d'évaluation de l'algorithme est modifiée, ce qui peut par exemple mener à des temps d'exécutions plus importants, mais le déroulement de la méthode reste identique.

2.4.2 Les métaheuristiques

Les métaheuristiques sont des algorithmes approximatifs et généralement non déterministes, permettant d'obtenir une solution de bonne qualité pour un problème d'optimisation difficile. L'objectif est d'explorer efficacement un espace de recherche afin d'obtenir une solution proche de la solution optimale. A l'inverse des heuristiques, les métaheuristiques ne sont pas spécifiques à un problème. Un grand nombre de métaheuristiques existent [Talbi 2009] et peuvent être regroupées en deux catégories : les métaheuristiques à solution unique et les métaheuristiques à base de populations (Figure 2.2).

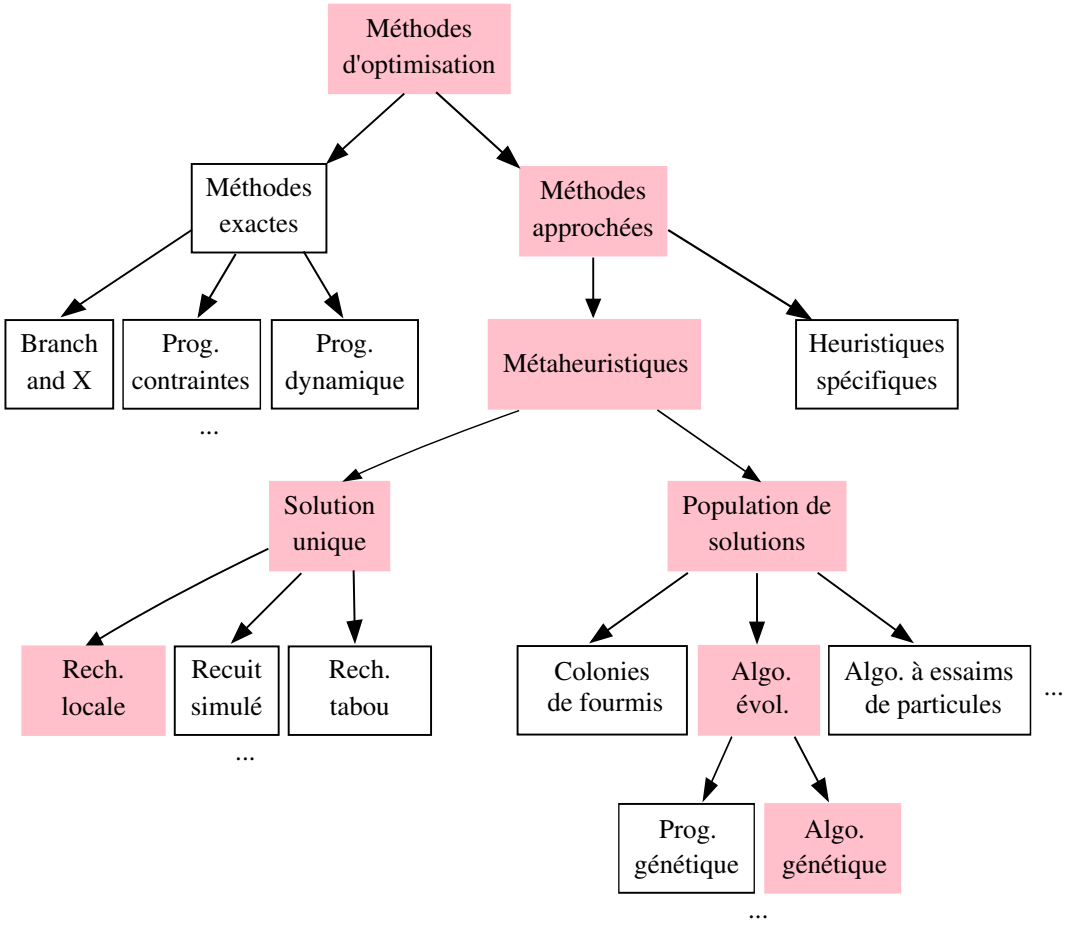

FIGURE 2.2 – Méthodes d'optimisation combinatoire

2.4.2.1 Métaheuristiques à solution unique

La recherche locale (*hill climbing*) est la plus ancienne et la plus simple des métaheuristiques [Papadimitriou 1976]. L'Algorithme 1 présente le principe général de la méthode. Partant d'une solution initiale donnée, à chaque itération, l'algorithme remplace la solution courante par une solution voisine (solution générée à partir de la précédente en appliquant une modification locale) qui améliore la fonction objectif. La descente s'arrête lorsqu'il n'y a plus de solution voisine permettant d'améliorer la solution courante : un optimum local est atteint.

Plusieurs stratégies peuvent être utilisées pour la sélection d'une meilleure solution voisine : choisir la première solution voisine améliorante (*first improvement*), choisir la meilleure parmi toutes les solutions voisines (*best improvement*), ou encore en choisir une aléatoirement parmi celles qui améliorent la solution courante (*random selection*). Une des principales limites de la recherche locale est qu'elle converge vers un optimum local. Pour éviter de rester bloqué sur ces optima, des mécanismes supplémentaires sont proposés menant à la création de nouveaux

Algorithme 1 Recherche locale

$\gamma = \gamma_0$ /* *Génération d'une solution initiale* */
tant que $\exists$ solution voisine(γ) **faire**
  Génération des solutions voisines candidates $\mathcal{N}(\gamma)$ /* γ' *dans le voisinage* */
  si il n'y a pas de meilleure solution voisine **alors**
    Stop
  fin si
  $\gamma = \gamma'$ /* *Sélection d'une meilleure solution voisine* $\gamma' \in \mathcal{N}(\gamma)$ */
fin tant que
retour Solution finale trouvée (optimum local)

algorithmes (recuit simulé, recherche tabou) ou à l'intégration de la recherche locale dans un processus itératif (ILS).

Le recuit simulé (*simulated annealing*) est un algorithme reposant sur un concept de mécanique statistique selon lequel il faut chauffer puis refroidir lentement une substance pour obtenir une structure cristalline. Les premières applications de cet algorithme à des problèmes d'optimisation ont été proposées par [Kirkpatrick 1983] et [Cerny 1985]. Si le processus de refroidissement de la substance est trop rapide, cette dernière présente des défauts, c'est l'équivalent d'un optimum local dans un problème d'optimisation. Pour sortir de ces optima locaux, un paramètre T, représentant la température est introduit au début de l'algorithme et va décroître au cours de l'algorithme pour tendre vers 0. La probabilité d'acceptation d'une solution voisine moins bonne dépend de T, plus la température sera élevée plus la probabilité sera grande. Des solutions voisines non améliorantes seront donc de moins en moins sélectionnées au cours de l'algorithme. Les performances de cet algorithme vont principalement dépendre du paramètre T et de sa vitesse de décroissance. En effet, s'il décroit trop lentement les temps de calcul vont être importants et s'il décroît trop rapidement l'algorithme risque d'atteindre un optimum local de mauvaise qualité. Régler ces paramètres est donc une étape importante de l'algorithme qui peut être fastidieuse.

La recherche tabou (*tabu search*) a été introduite par [Glover 1989] et est devenue une des métaheuristiques à solution unique les plus utilisées. Son principe général repose sur la notion de mémoire dans le processus de recherche. Le comportement de cet algorithme est similaire à une recherche locale mais il accepte des solutions non améliorantes pour sortir des optima locaux. En effet, lorsqu'un optimum local est atteint, la meilleure solution voisine est sélectionnée même si elle est moins bonne que la solution courante. Le risque de ce mécanisme est de revenir ensuite sur l'optimum local auquel on vient d'échapper. Pour éviter cela, l'algorithme garde en mémoire les solutions voisines visitées récemment (ou les mouvements appliqués récemment) dans une *liste tabou*. À chaque visite d'une nouvelle solution la solution la plus ancienne de la liste tabou est supprimée. La recherche d'une solu-

tion voisine améliorante se fait donc parmi les solutions du voisinage de la solution courante mais sans considérer les solutions qui sont dans la liste tabou. Cette méthode étant itérative elle nécessite la définition d'un critère d'arrêt qui peut par exemple être un nombre d'itérations ou un nombre d'itérations sans amélioration de la meilleure solution. Des processus de diversification et d'intensification sont généralement introduits dans l'algorithme de la recherche tabou.

La recherche locale itérée (ILS)

Lorsqu'on utilise une méthode de recherche locale, la qualité de l'optimum local est fortement dépendante de la solution initiale et la variabilité entre les optima peut donc être grande. La recherche locale itérée peut être utilisée pour palier ce problème en améliorant la qualité d'optima locaux successifs. Cet algorithme a dans un premier temps été appliqué par [Martin 1991] puis généralisé par [Lourenco 2001]. Le principe de l'algorithme ILS est décrit par la Figure 2.3. L'algorithme démarre par une recherche locale à partir d'une solution initiale. Lorsqu'un optimum local est atteint, il est perturbé, puis une recherche locale est appliquée à partir de cette solution perturbée. L'algorithme s'arrête lorsqu'il atteint un critère d'arrêt. Nous obtenons finalement un ensemble d'optima locaux dont le meilleur sera retourné.

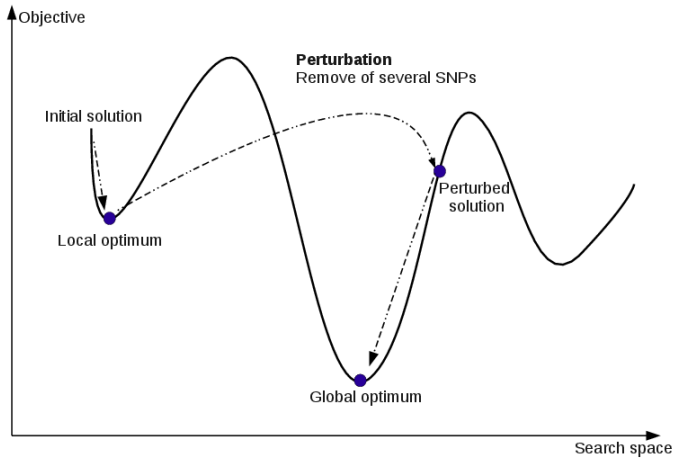

FIGURE 2.3 – Recherche locale itérée (ILS)

Pour mettre en place ces métaheuristiques, pour un problème donné, il est nécessaire de définir certains composants comme le codage des solutions, la notion de voisinage, la détermination de la valeur du paramètre T pour le recuit simulé ou la définition d'un liste tabou pour la recherche tabou, etc. Nous détaillerons dans le chapitre suivant ce que nous proposons pour le problème de sélection de variables en régression.

2.4.2.2 Métaheuristiques à base de population

Les métaheuristiques à base de population ont été introduites afin d'améliorer la diversité des solutions proposées par les algorithmes d'optimisation. Parmi les métaheuristiques à base de population on distingue les algorithmes évolutionnaires basés sur la théorie de l'évolution de Darwin [Darwin 1859], l'intelligence en essaim basée sur le comportement des espèces vivants en colonies comme les fourmis ou les abeilles, ou les systèmes immunitaires artificiels qui miment les systèmes immunitaires biologiques.

Les algorithmes évolutionnaires sont des algorithmes d'optimisation inspirés de l'évolution naturelle et sont les algorithmes à base de population les plus utilisés. Une revue d'algorithmes évolutionnaires appliqués à la bioinformatique est présenté dans [Pal 2006].
Les algorithmes évolutionnaires les plus populaires sont les **algorithmes génétiques**, développés dans les années 70 [Holland 1975]. Ils sont initialement associés à un codage binaire des solutions mais peuvent être utilisés avec d'autres représentations. Le principe général des algorithmes génétiques est illustré par la Figure 2.4. L'algorithme est initialisé par une population de n individus générés aléatoirement où chaque individu est une version codée d'une solution candidate. Chaque solution est évaluée puis $n/2$ couples de solutions sont sélectionnés. Chaque couple va générer 2 solutions filles par l'intermédiaire du croisement puis un opérateur de mutation sera éventuellement appliqué à ces solutions filles pour les diversifier. Une stratégie de remplacement décide ensuite des solutions, parmi les solutions initiales et les filles générées, qui constitueront la nouvelle population. Cet enchaînement d'étapes représente une génération. L'algorithme s'arrête lorsqu'il atteint un critère d'arrêt donné. Différents opérateurs de sélection, croisement, mutation, remplacement, peuvent être utilisés et nous verrons ceux que nous choisissons d'utiliser dans le Chapitre 4.

On trouve également dans cette classe d'algorithmes évolutionnaires **la programmation génétique**, qui est une approche récente de programmation évolutionnaire [Koza 1998] dont le processus de reproduction est basé uniquement sur la mutation (pas de croisement). Sa différence majeure avec les autres algorithmes évolutionnaires est le fait que les individus sont des programmes, souvent représentés sous forme d'arbres.

Parmi les métaheuristiques à base de populations, les algorithmes à base d'essaims sont basés sur le comportement en collectivité de certaines espèces comme les fourmis ou les abeilles. Les algorithmes d'optimisation basés sur l'intelligence en essaims les plus performants sont les colonies de fourmis et l'optimisation en essaims de particules.
L'idée de base des **colonies de fourmis**, proposé par [Dorigo 1992], repose sur le comportement coopératif de vraies fourmis qui recherchent un chemin de leur

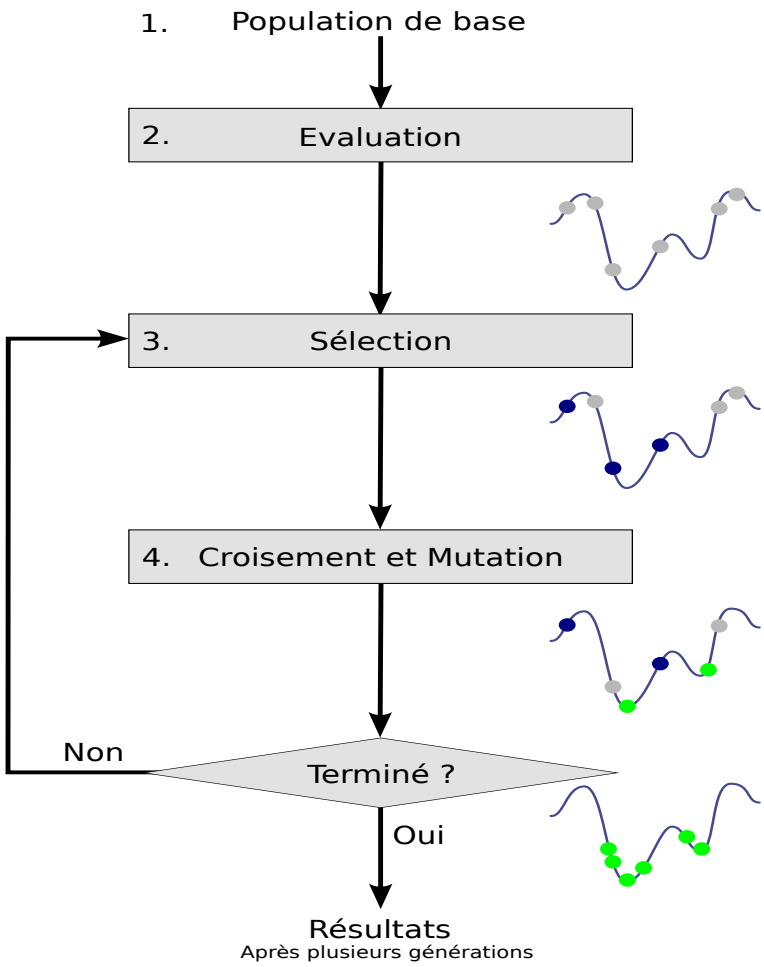

Figure 2.4 – Les algorithmes génétiques
(Source : wikimedia)

nid à une source de nourriture. L'observation de fourmis a permis de constater qu'elles sont capables de trouver le chemin le plus court les menant à cette source. En effet, lorsque les fourmis se déplacent d'une source de nourriture à leur nid et inversement, elles déposent une substance chimique appelée phéromone que les autres vont pouvoir sentir. Les fourmis passant à proximité vont donc avoir tendance à regagner ce chemin tout en re-déposant des phéromones afin de renforcer le chemin puisqu'elles seront attirées par la plus grande quantité de substance, la phéromone s'évaporant. Si deux pistes sont possibles, les fourmis seront attirées par le plus court chemin car la phéromone y sera plus persistante. Ce chemin constituera donc à terme le chemin emprunté par toute la colonie.

Les **algorithmes en essaims de particules** [Kennedy 1995], quant à eux, sont basés sur le comportement social d'organismes naturels comme les bans de poissons

ou les nuées d'oiseaux et plus particulièrement leurs déplacements coordonnés. Ces algorithmes utilisent le concept de déplacement en groupe pour rechercher les solutions d'un algorithme d'optimisation. La population de l'algorithme est appelée "essaim" et les individus sont des "particules". Une particule, qui constitue une solution candidate du problème, est définie par sa position et sa vélocité, c'est à dire sa direction de "vol" et la distance qu'elle va parcourir. Le mouvement d'une particule va dépendre de sa vélocité, de sa meilleure position (qu'elle garde en mémoire), et de la meilleure solution obtenue dans son voisinage. La nouvelle vélocité de la particule sera fonction de sa tendance à suivre son propre chemin, à revenir vers sa meilleure position atteinte ou à aller vers sa meilleure solution voisine.

Après avoir présenté les métaheuristiques classiques de résolutions de problème d'optimisation combinatoire, nous allons maintenant nous intéresser à leurs applications en sélection de variables.

2.4.3 Les métaheuristiques pour la sélection de variables

La sélection de variables est un domaine de recherche actif en statistiques et *data mining*. Le principe général consiste à choisir un sous-ensemble de variables en éliminant les variables qui ont peu ou pas d'influence sur l'information que l'on souhaite prédire. Ce processus est particulièrement adapté pour réduire la dimension des données et donc les temps d'exécution. Il permet aussi de réduire le sur-apprentissage et d'améliorer la précision de prédiction puisque l'intégration de variables sans intérêt peut induire un bruit dans le modèle. Enfin, le modèle final est généralement plus facilement interprétable avec un nombre restreint de variables.
Utiliser des méthodes d'optimisation pour la sélection de variables nécessite de définir comment combiner la méthode d'optimisation et le critère d'évaluation permettant de mesurer la qualité du modèle. Pour cela, différentes approches ont été proposées.

2.4.3.1 Classification des approches de sélection de variables

Les méthodes de sélection de variables sont classiquement présentées suivant trois classes en fonction de la manière dont elles combinent l'algorithme de sélection et la construction du modèle (Figure 2.5).

Les méthodes *filter*

Les méthodes de type *filtre* [Pudil 1998] sélectionnent les variables indépendamment du modèle utilisé. Elles se basent uniquement sur les caractéristiques générales comme la corrélation avec la variable à prédire. Généralement les variables les moins intéressantes sont supprimées, les autres feront partie du modèle de classification/régression utilisé pour classer/prédire les données. Ces méthodes sont particulièrement efficaces en temps de calcul et robustes au sur-apprentissage (*over-fitting*).

Cependant, ne tenant pas compte des relations entre variables, elles vont avoir tendance à sélectionner des variables redondantes et seront donc généralement utilisées en pré-process.

Les méthodes *wrapper*

Les méthodes *wrapper* [Kohavi 1997] évaluent des sous-ensembles de variables ce qui permet, contrairement aux approches *filter*, de prendre en compte les éventuelles intéractions entre variables. Une procédure de recherche des sous-ensembles possibles est définie et différents sous-ensembles de variables sont évalués à l'aide du modèle. Les principaux inconvénients de ces méthodes sont le risque de sur-apprentissage lorsque le nombre d'observations est insuffisant ainsi que le temps de calcul qui devient important lorsque le nombre de variables est grand.

Les méthodes *embedded*

Récemment, les méthodes *embedded* [Lal 2006] ont été proposées en classification pour diminuer le sur-apprentissage. Elles ont pour objectif de combiner les avantages des deux types de méthodes précédents. L'algorithme d'apprentissage utilise son propre algorithme de sélection de variables et nécessite donc de pouvoir caractériser a priori ce que serait une bonne sélection, ce qui limite leur utilisation.

Dans la plupart des problèmes de sélection de variables l'objectif est que les attributs sélectionnés soient pertinents. L'évaluation de la pertinence d'un sous-ensemble de variables se fait généralement par évaluation de la qualité de la classification ou de la régression engendrée par le modèle sous-jacent sur des données non utilisées pour la génération du modèle. Dans notre contexte de régression, nous souhaitons obtenir un modèle de prédiction intéressant à l'aide du sous-ensemble de variables sélectionnées. La pertinence d'un sous-ensemble de variables pourra donc être évaluée en terme de qualité de prédiction (erreur de prédiction) du modèle de régression obtenu à l'aide de ces variables.
Les approches *wrapper*, qui tiennent compte du modèle de régression, sont donc généralement plus utilisées que les méthodes *filter*.

2.4.3.2 Revue de la littérature

La Table 2.4 présente quelques travaux de la littérature sur les applications d'algorithmes de sélection de variables. Ces travaux sont décrits selon le domaine d'application, l'algorithme utilisé, l'approche (*filter*, *wrapper* ou *embedded*), le classifieur (s'il y en a un) et la fonction d'évaluation.
Peu d'études ont combiné une méthode d'optimisation combinatoire pour sélection de variables avec un problème de régression (en gras dans le tableau) comparé à la classification. C'est par exemple le cas de [Meiri 2006] ou [Kapetanios 2005] pour des

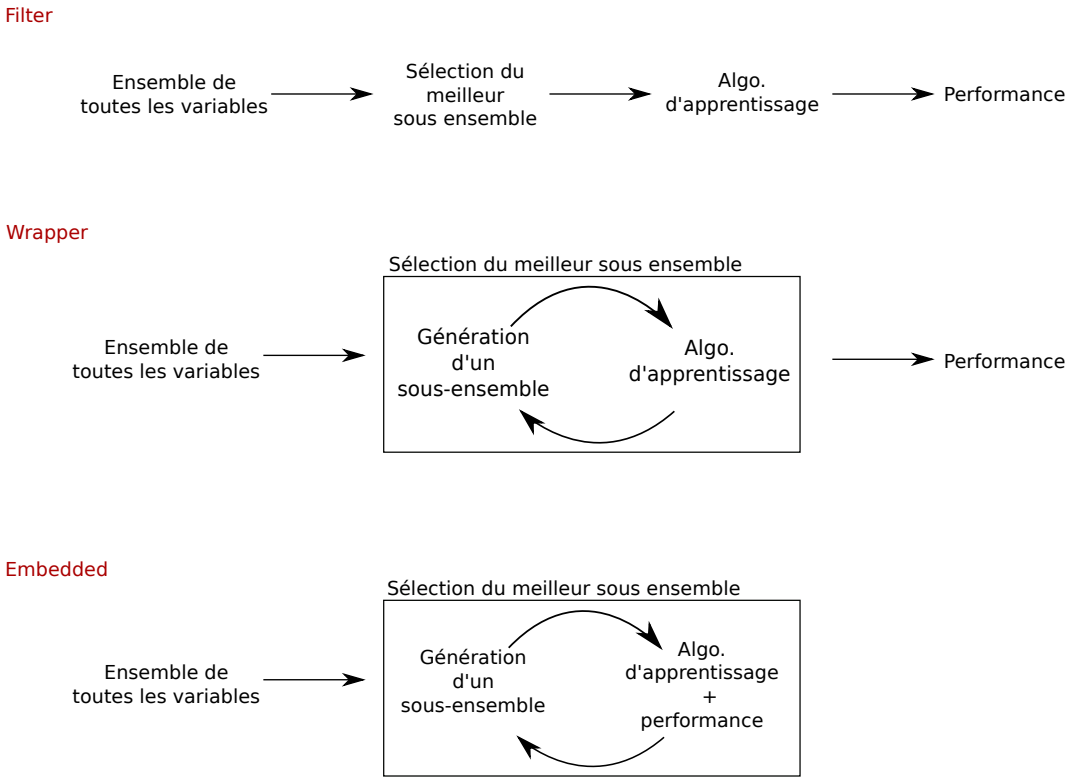

FIGURE 2.5 – Les approches *filter*, *wrapper* et *embedded*

applications en marketing et économie avec l'utilisation du recuit simulé et l'algorithme génétique avec évaluation des modèles de régression par AIC [Akaike 1974], BIC [Schwarz 1978] ou coefficient de corrélation (r^2). Dans [Hans 2007], une recherche locale itérée est considérée dans le contexte de régression sur données d'expression génétique et les modèles sont évalués par leurs probabilités *a posteriori*. En 1997, Broadhurst *et al.* [Broadhurst 1997] proposent d'utiliser un algorithme génétique associé à une régression linéaire multiple (RLM) et à une régression PLS pour l'analyse de spectres de masse. Le calcul de l'erreur de prédiction moyenne leur permet d'évaluer leurs modèles et leurs approches obtiennent de bonnes performances.

La fonction d'évaluation, que ce soit en classification ou en régression doit permettre d'évaluer la capacité d'un modèle à classer ou prédire de nouvelles données. En régression, un modèle peut par exemple être évalué à l'aide des critères AIC [Akaike 1974] comme utilisé par [Meiri 2006] ou BIC [Schwarz 1978] comme [Kapetanios 2005]. Cependant, en classification la méthode la plus utilisée, et qui peut également être utilisée en régression, est la validation croisée (*k-fold* ou *leave-one-out* principalement).

[Alba 2007] et [Duval 2009] utilisent par exemple de la validation croisée *10-fold* et [Jirapech-Umpai 2005] et [Peng 2003] de la validation croisée *leave-one-out* (LOO). [Broadhurst 1997] proposent dans le cadre de la régression, de découper l'échantillon en 3 groupes (apprentissage, validation et test) dont le premier servira à calibrer le modèle, le second à tester sa validité, et enfin le troisième à tester la validité du modèle final, ce qui permet de limiter le sur-apprentissage.
Malgré quelques utilisations des algorithmes à solution unique (hill climbing, recuit simulé, ILS), des colonies de fourmis, ou d'algorithmes en essaim de particules (PSO), l'algorithme génétique (AG) reste majoritairement utilisé avec de nombreuses applications sur données de microarray, notamment pour des études sur le cancer ou sur le diabète.
Concernant les applications sur données génomiques et plus particulièrement les SNPs, quelques études ont été proposées. [Long 2007] appliquent une approche *filter* suivie d'une approche *wrapper* utilisant un classifieur bayésien naïf ([Elkan 1997]) et montrent des résultats encourageant.

Application	Algo.	Approche	Classifieur	Fonction d'évaluation	Ref.
SNPs	FSFS	F		r^2	[Phuong 2005]
SNPs	AG	W	Arbre décision	CA (10-fold), spécificité, ...	[Shah 2004]
SNPs	Hill Climbing	F+W	Bayésien naïf	PRESS	[Long 2007]
SNPs	Recuit simulé		Bayésien naïf	CA (5-fold)	[Ustunkar 2011]
Segments parole	Colonies fourmis	W	ANN	MSE	[Al-ani 2005]
Marketing	Recuit simulé	W	**Régression**	AIC, r^2	[Meiri 2006]
Économie	Recuit simulé, AG	W	**Régression**	BIC	[Kapetanios 2005]
Spectres de masse	AG	W	**RLM, PLS**	RMSEP (test set)	[Broadhurst 1997]
Microarray	Rech. tabou + PSO	W	KNN, SVM	dist. euclidienne (LOOCV)	[Chuang 2009]
Microarray	PSO, AG	W	SVM	CA (10-fold)	[Alba 2007]
Microarray	AG + ILS	E	SVM	CA (10-fold)	[Duval 2009]
Microarray	ILS	W	**Régression**	Probabilité a posteriori (10-fold)	[Hans 2007]
Microarray	AG	W	KNN	CA (LOOCV)	[Jirapech-Umpai 2005]
Microarray	AG hybride	W	KNN	CA (LOOCV), # attributs	[Oh 2004]
Microarray	AG	W	SVM	Sensibilité, spécificité, ...	[Xuan 2011]
Microarray	AG	W	AP, SVM	CA (LOOCV)	[Peng 2003]
Microarray	AG	E	SVM	CA (10-fold), # attributs	[Hernandez 2007]
Microarray	AG	H	SVM	CA (LOOCV)	[Huerta 2006]
Microarray	PG			CA (10-fold), # attributs	[Muni 2006]
Microarray	AG parallèle	W		EH-DIALL, CLUMP	[Jourdan 2004]

TABLE 2.4 – Applications de métaheuristiques à la sélection de variables.

AG : algorithme génétique, PG : programmation génétique, FSFS : feature selection using feature similarity. Approche : wrapper(W), embedded(E), hybrid(H). Classifieurs : KNN : K nearest neighbor, ANN : artificial neural network, SVM : support vector machine, AP : all paired, RLM : Régression Linéaire Multiple, PLS : partial least square. Fonctions d'évaluation : CA : classification accuracy, LOOCV : leave-one-out cross-validation, RMSEP : root-mean-square error of prediction, PRESS : predicted residual sum of squares.

2.5 Conclusion

Dans ce chapitre, nous avons tout d'abord introduit les méthodes de régression en grande dimension, dont certaines comme lasso ou elastic net font également de la sélection de variables. Nous avons ensuite présenté les méthodes spécifiques à la génomique qui pour la plupart considèrent les effets des marqueurs comme aléatoires. Enfin, un des objectifs de ce travail étant de sélectionner un sous-ensemble de variables significatives, ce que nous proposons de faire en utilisant des méthodes d'optimisation combinatoire, un état de l'art de ces méthodes a été proposé. Étant donné le succès des méthodes d'optimisation combinatoire pour la sélection de variables, nous proposons d'adopter ce genre d'approche dans le contexte de régression en grande dimension. C'est l'objet du chapitre suivant.

Chapitre 3

Sélection de variables en régression par recherche locale itérée

Sommaire

Dans ce chapitre, nous présentons une première approche que nous proposons dans le cadre de la prédiction d'un caractère quantitatif utilisant un nombre réduit de variables significatives à partir d'un grand ensemble de variables possibles. Cette approche est basée sur un modèle de régression classique auquel nous avons intégré un processus de sélection d'un sous-ensemble de variables intéressantes. Cette sélection peut être considérée comme un problème combinatoire, nous proposons donc d'utiliser une approche d'optimisation combinatoire pour rechercher efficacement le meilleur sous-ensemble de variables.
Une partie de ce travail a fait l'objet d'un poster pour les Journées Ouvertes en Biologie, Informatique et Mathématiques [Hamon 2011], d'une publication à la conférence ROADEF 2012 [Hamon 2012], ainsi que d'une publication à la conférence

internationale CIBB 2013 [Hamon 2013a]. Nous commençons par présenter la modélisation proposée, puis nous décrivons la méthode d'optimisation utilisée. Enfin nous évaluons ses résultats et les comparons avec quelques méthodes classiques.

Par la suite nous utiliserons les notations suivantes :

- n : nombre d'individus
- p : nombre de variables (SNPs)
- $x_{ij} \in \{0, 1, 2\}$: valeur $j^{\text{ème}}$ SNP pour le $i^{\text{ème}}$ individu ($1 \leq j \leq p$, $1 \leq i \leq n$)
- $y_i \in \mathbb{R}$: valeur du caractère étudié pour l'individu i.
- e_i : résidu gaussien. Les résidus sont supposés indépendants et identiquement distribués (i.i.d) d'espérance nulle et de variance $\sigma^2_{\mathbf{e}}$.

3.1 Méthodologie

Dans cette section nous commençons par introduire la modélisation que nous proposons. Puis, ayant choisi d'utiliser une métaheuristique qui nécessite des opérateurs à valider, et pour laquelle nous proposons des processus d'amélioration, nous présentons ensuite le design expérimental de validation de ces opérateurs et processus.

3.1.1 Modélisation

L'objectif de ce travail est de sélectionner un sous-ensemble de variables significatives tout en élaborant un modèle prédictif pour un trait quantitatif. Pour cela, en se basant sur un modèle de régression classique, nous choisissons de modéliser le problème de la façon suivante :

$$\begin{aligned} y_i &= \beta_0 + \sum_{j=1}^{p} (\beta_j \gamma_j x_{ij}) + e_i, \qquad i = 1, \ldots, n, \\ \mathbf{y} &= X(\boldsymbol{\beta} \cdot \boldsymbol{\gamma}) + \boldsymbol{e}, \end{aligned} \tag{3.1}$$

où X est la matrice des SNPs, $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)^t$ est le vecteur des paramètres associés aux SNPs, $\gamma = (\gamma_0, \gamma_1, \ldots, \gamma_p)$ est un vecteur de paramètres linéaires, avec $\gamma_0 = 1$ et pour $j = 1, \ldots, p$, $\gamma_j = 1$ si la variable j est dans le modèle, 0 sinon. Les paramètres γ, σ^2_e, β_0 et $\{\beta_j : \gamma_j = 1, 1 \leq j \leq p\}$ doivent être estimés. Le paramètre γ étant discret, prenant ses valeurs dans un ensemble fini $\{0, 1\}^p$, son estimation par maximum de vraisemblance nécessiterait de calculer la vraisemblance du modèle pour toutes les combinaisons possibles, ce qui est impossible lorsque p est grand (exemple : si $p = 100$, le nombre de combinaison est $2^p \simeq 10^{30}$). Déterminer les valeurs des γ_j revient à déterminer les variables qui sont inclues dans le modèle de régression. Ce problème est un problème classique de sélection de variables, connu en analyse de données et peut être traité par des méthodes d'optimisation combinatoire. Dans la littérature, les méthodes d'optimisation traitent plus souvent des problèmes de classification que de régression, ce qui fait la particularité de notre approche.

Nous proposons dans ce chapitre d'utiliser une métaheuristique à solution unique : la recherche locale itérée (ILS - *Iterated Local Search* - cf. section 2.4.2.1).

3.1.2 Design expérimental

Nombre de simulations

Comme nous l'avons vu dans le chapitre précédent, les métaheuristiques sont des approches stochastiques qui ne donneront pas les mêmes résultats d'une exécution à l'autre. De plus, lorsque nous simulons des données comme présentées dans la section 1.4.2.2, à chaque simulation, les variables significatives sont différentes et le caractère à prédire est donc également différent. Dans l'optique de comparer différents opérateurs d'une métaheuristiques, se pose la question du nombre d'instances (jeux de données) à utiliser et du nombre d'exécutions de l'algorithme par instance. Ceci a été étudié par [Birattari 2004] dans le cadre d'études de performances de métaheuristiques, où il conclu que si l'on s'autorise N exécutions, alors "N instances, une exécution par instance" est le meilleur scénario, en comparaison à N exécutions pour 1 instance, par exemple. Nous proposons donc de simuler 30 jeux de données.

Scénarios de simulations

Nous avons proposé 2 types de données simulées suivant le génotype sur lequel elles sont basées : 1) celui de QTLMAS 2010 ou 2) celui de QTLMAS 2011 (plus de détails en section 1.4.2.2). Nous simulons donc 30 jeux de données par type de simulation (1 et 2) sur lesquels nous exécutons l'algorithme étudié une fois. La génération des données est schématisée sur la figure 3.1 : à partir de la matrice des marqueurs (génotypes) de QTLMAS (2010 ou 2011 suivant la simulation), nous simulons le vecteur $\boldsymbol{\beta}$ des effets des marqueurs, l'erreur résiduelle $\mathbf{e}$ puis nous calculons la valeur du caractère $\mathbf{y}$, et ce 30 fois.

Nous utilisons également les jeux de données complets (génotypes + phénotypes) de QTLMAS 2010 et 2011 pour lesquels nous exécutons donc 30 fois l'algorithme puisque nous avons un jeu de données pour QTLMAS 2010 et un jeu pour QTLMAS 2011.

Indicateurs de qualité du modèle

Pour chaque jeu de données, que ce soit simulées ou pseudo-réelles, nous avons un échantillon d'apprentissage et un échantillon de validation. L'algorithme est exécuté sur l'échantillon d'apprentissage, puis nous étudions les performances en terme d'erreur de prédiction sur l'échantillon de validation. La qualité de prédiction sera également évaluée par la corrélation entre l'estimation du caractère et sa valeur réelle. Nous évaluons donc les résultats suivant la racine carrée de l'erreur moyenne de prédiction (*Root Mean Square Error of Prediction* - RMSEP), que l'on souhaite minimiser, ainsi que la corrélation entre le caractère prédit et le caractère réel, que l'on souhaite maximiser.

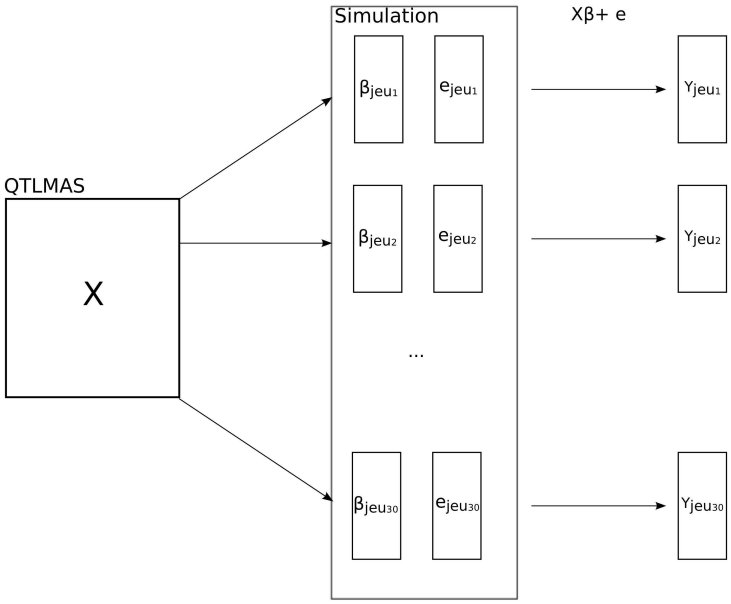

Figure 3.1 – Schéma de simulation des données

Présentation des résultats

Sauf expérimentations particulières, nous présentons les résultats sous forme de boîtes à moustaches avec une première figure rapportant les résultats sur données simulées puis une seconde rapportant ceux sur données pseudo-réelles. La première (resp. seconde) figure est divisée en 4 graphiques, les deux du haut présentent les résultats sur les données simulées 1 (resp. les données de QTLMAS 2010), et les deux du bas ceux sur données simulées 2 (resp. les données de QTLMAS 2011). Les graphiques de gauche présentent les performances en terme de RMSEP et ceux de droite en terme de corrélation. Les comparaisons ici se font au sein d'un même graphique, entre les différentes boîtes à moustaches présentées, mais aucune comparaison ne peut être faite entre les graphiques, les échelles étant différentes (les valeurs de caractères d'un jeu de données à un autre étant différentes).

Aspects informatiques

Les algorithmes sont développés en C++ en utilisant la plate-forme PARADISEO [Cahon 2004], développée en partie par l'équipe DOLPHIN d'Inria Lille-Nord Europe, pour la mise en œuvre de métaheuristiques. Pour les méthodes classiques (elastic net, lasso, pls, spls), nous utilisons le logiciel R avec les procédures "*glmnet*" (pour elastic net et lasso) et "*spls*". Nous utilisons la fonction "*cv.glmnet*" pour déterminer le paramètre λ des méthodes lasso et elastic net, le paramètre α de la méthode elastic net est déterminé par validation croisée 3-fold.
Les résultats présentés ont été générés à l'aide du cluster de calcul régional financé par l'Université Lille 1, le CPER Nord-Pas-de-Calais/FEDER, France Grille et le

CNRS. Ce cluster est hébergé au Centre de Ressources Informatiques (CRI) de l'Université Lille 1, et administré par son service "calcul scientifique intensif" qui fournit également le support aux utilisateurs. Le cluster de calcul est composé de 2016 cœurs CPU et 7168 cœurs GPU avec un espace disque total de 156 To.

3.2 Approche par recherche locale itérée

Nous proposons dans une première approche de sélectionner les variables de notre modèle à l'aide d'une recherche locale itérée (ILS). Nous avons choisi d'utiliser cette méthode pour sa simplicité, qui nous permet d'évaluer indépendamment la pertinence de l'approche combinatoire et de la fonction d'évaluation. L'algorithme ILS est basé sur une succession de recherches locales et de perturbations dont le principe général est décrit dans le chapitre précédent (cf. section 2.4.2.1). L'algorithme démarre d'une solution (un sous-ensemble de variables) initiale puis applique un algorithme de descente. Lorsqu'un optimum local est atteint, il est perturbé puis une recherche locale est de nouveau appliquée en démarrant de cette solution perturbée, et ce jusqu'à ce qu'un critère d'arrêt soit atteint.
L'algorithme ILS de base sur lequel nous allons nous appuyer est défini à l'aide des opérateurs suivants :

- initialisation : uniforme,
- voisinage : opérateur *bit-flip* (défini en section 3.2.2),
- perturbation : suppression aléatoire de 20% des variables de la solution courante,
- critère d'arrêt : nombre maximal fixé d'évaluations.

Le choix des opérateurs dépend du problème traité, nous allons donc en étudier plusieurs, évaluer leurs performances et éventuellement améliorer l'algorithme en conséquence.

3.2.1 Représentation d'une solution

La représentation d'une solution joue un rôle majeur dans la mise en œuvre d'une métaheuristique puisqu'elle influence le choix des opérateurs et de la fonction d'évaluation. Plusieurs codages sont possibles pour les problèmes de sélection de variables : un vecteur binaire indiquant les variables sélectionnées, un vecteur de réels indiquant le poids de chaque variable (β_j), une liste des variables sélectionnées, etc. Nous choisissons d'utiliser un vecteur binaire indiquant si une variable est sélectionnée (1) ou non (0) puisque c'est très proche du modèle statistique présenté précédemment (c'est équivalent au vecteur $\gamma = (\gamma_1, \ldots, \gamma_p)$). De plus, ce codage permet un design simple mais efficace du voisinage.
Exemple d'un individu :

1	0	0	1	1	0	1	0

Dans cette solution, les variables 1, 4, 5 et 7 sont sélectionnées. La taille d'une solution (ici 8) est égale au nombre total de variables p du jeu de données utilisé.

Dans notre étude, la taille des solutions variera donc entre 6 000 et 54 000 suivant le jeu de données étudié.

3.2.2 Voisinage

La fonction de voisinage attribue à chaque solution un ensemble de solutions obtenues par l'application d'un opérateur de voisinage (modification locale de la solution). Le voisinage joue un rôle important dans la performance d'une recherche locale puisqu'il définit l'ensemble des solutions à explorer à chaque étape. La fonction de voisinage classiquement utilisée avec un vecteur binaire est l'opérateur de *bit-flip*. Cet opérateur sélectionne aléatoirement une variable et modifie le bit correspondant dans le vecteur. Par conséquent, si cette variable est sélectionnée dans la solution courante, alors elle ne le sera plus dans la solution voisine, et inversement.
Exemple d'une solution courante :

1	0	0	1	**1**	0	1	0

Une solution voisine :

1	0	0	1	**0**	0	1	0

La variable 5, qui faisait partie de la solution courante est supprimée pour générer une solution voisine. Notons que pour une solution donnée, il y a peu de variables sélectionnées comparé au nombre total de variables, l'opérateur *bit-flip* aura donc tendance à ajouter des variables plus qu'en retirer. De plus, l'objectif de Gènes Diffusion est de sélectionner un nombre réduit de variables afin de créer une mini-puce. Nous fixons donc à l'algorithme un nombre maximal de variables à sélectionner, et lorsque ce nombre est atteint, la génération d'une solution voisine consiste à supprimer aléatoirement une variable présente dans la solution courante. Nous aurions également pu utiliser un opérateur d'échange (*swap*) entre une variable sélectionnée (1) et une variable non sélectionnée (0). Cependant, avec cet opérateur, toutes les solutions testées ont exactement le même nombre de variables or il peut parfois être intéressant d'en sélectionner moins que la limite fixée au départ de l'algorithme.

Les stratégies classiques d'exploration de voisinage consistent à choisir soit la meilleure solution voisine, soit la première qui améliore la qualité de la solution. Choisir la meilleure nécessite la génération, à chaque étape, de l'ensemble du voisinage ce qui peut être gourmand en temps. En pratique, il a été observé dans de nombreuses applications que la stratégie de choix de la première solution voisine améliorante permet d'atteindre la même qualité de solution que la stratégie de choix de la meilleure solution voisine [Whitley 2013]. Nous choisissons donc d'utiliser la stratégie de choix de la première solution voisine améliorante (*first improvement*).

3.2.3 Évaluation de la qualité d'une solution

L'objectif de la méthode d'optimisation est d'explorer efficacement un grand espace de recherche qui correspond ici à tous les sous-ensembles possibles de va-

riables. Une telle méthode utilise donc un critère d'évaluation (fonction de fitness) capable d'associer à chaque solution une mesure de qualité. Dans notre contexte, l'objectif est d'identifier le meilleur sous-ensemble de variables, c'est-à-dire celui qui fournira le meilleur modèle prédictif. Une difficulté bien connue en *data mining* est d'être capable d'évaluer la capacité du modèle à prédire un caractère à partir de données qui n'ont pas été utilisées pour élaborer ce modèle (échantillon de validation). Un panel des méthodes d'évaluation de sélection de modèle est disponible dans [Hastie 2009]. Les critères les plus couramment utilisés sont le critère AIC (*Akaike information criterion*) [Akaike 1974], le critère BIC (*Bayesian information criterion*) [Schwarz 1978] et la validation croisée. Le critère BIC, contrairement à AIC, aura tendance à pénaliser plus fortement les modèles complexes [Lebarbier 2006] et semble donc plus approprié à notre optique de sélection de variables en (très) grande dimension. Pour ce travail nous comparons donc 3 critères : BIC, et deux types de validation croisée (k-fold et *leave-one-out*).

3.2.3.1 Les différents critères

Critère BIC (*Bayesian Information Criterion*) Ce critère, comme le critère AIC (*Akaike information criterion*), est basé sur la vraisemblance du modèle et se calcule selon l'équation suivante :

$$BIC = -2 \cdot ln(\hat{L}) + d \cdot ln(n). \tag{3.2}$$

où $\hat{L}$ est le maximum de vraisemblance (*likelihood*) du modèle, n est le nombre d'individus et d le nombre de paramètres à estimer. Dans notre étude, $d = q+2$ avec q le nombre de variables sélectionnées par la solution évaluée. En effet il y a autant de paramètres $\boldsymbol{\beta}$ à estimer qu'il y a de variables dans le modèle, auxquels s'ajoutent les estimations de β_0 et de la variance de l'erreur ($\sigma^2_{\mathbf{e}}$). Le meilleur modèle est celui pour lequel le critère BIC est le plus petit. Nous sommes donc dans un problème de minimisation.

Calcul de la vraisemblance du modèle : Pour un sous-ensemble de q variables sélectionnées, nous avons le modèle suivant :

$$y_i = \beta_0 + \sum_{j=1}^{q} (\beta_j x_{ij}) + e_i. \tag{3.3}$$

L'objectif est d'estimer $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_q)^t$ et $\sigma^2_{\mathbf{e}}$ (la variance de $\boldsymbol{e} = (e_1, \ldots, e_n)^t$). Nous sommes ici dans l'évaluation d'une solution (donc pour un γ fixé), les variables initiales sont donc redéfinies par rapport à l'équation 3.1 puisque nous avons ici uniquement les q variables sélectionnées.

La vraisemblance de l'échantillon $\underline{\mathbf{x}} = (\mathrm{x}_1, \ldots, \mathrm{x}_n)$, avec $\mathrm{x}_i = (x_{i1}, \ldots, x_{iq})$ est :

$$L(\underline{\mathbf{y}}, \underline{\mathbf{x}}; \boldsymbol{\beta}, \sigma^2_{\mathbf{e}}) = \prod_{i=1}^{n} \frac{1}{\sigma_{\mathbf{e}}\sqrt{2\pi}} exp^{-\frac{(y_i - \beta_0 - \sum_{j=1}^{q} \beta_j x_{ij})^2}{2\sigma^2_{\mathbf{e}}}}, \tag{3.4}$$

avec $\underline{\mathbf{y}} = (y_1, \ldots, y_n)$. En pratique, on maximise généralement la log-vraisemblance :

$$ln(L) = -nln(\sigma_{\mathbf{e}}\sqrt{2\pi}) - \frac{1}{2\sigma_{\mathbf{e}}^2}\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{q}\beta_j x_{ij})^2. \tag{3.5}$$

Une fois la log-vraisemblance estimée, nous sommes alors capables de calculer la valeur du critère BIC, qui devient la fitness de la solution évaluée.

La validation croisée k-fold La validation croisée permet d'estimer la qualité d'un modèle en évaluant sa capacité à prédire de nouvelles données. Plusieurs types de validation croisée existent dont : la *test set validation* qui divise l'échantillon en 2 parties, la validation croisée k-fold, qui divise l'échantillon en k sous-échantillons. La Figure 3.2 illustre le principe de la validation croisée k-fold dans le cas où $k = 3$. Tout d'abord les individus sont répartis en k groupes puis le processus se décompose en k itérations. À chaque itération un des k groupes va constituer l'échantillon de validation, les $k-1$ autres constitueront l'échantillon d'apprentissage. Chaque groupe doit servir d'échantillon de validation une et une seule fois. Le modèle est construit sur l'échantillon d'apprentissage puis appliqué sur l'échantillon de validation.

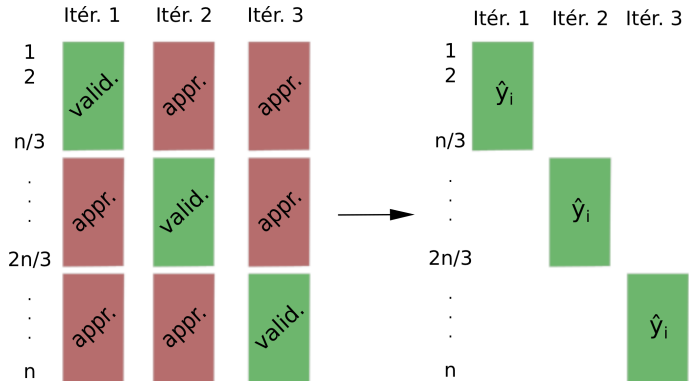

FIGURE 3.2 – Validation croisée 3-fold

Dans notre contexte de régression, à chaque itération, les coefficients $\boldsymbol{\beta}$ sont estimés sur l'échantillon d'apprentissage par leur estimateur du maximum de vraisemblance :

$$\hat{\boldsymbol{\beta}} = (X_{appr}^t X_{appr})^{-1} X_{appr}^t \ \mathbf{y}_{appr},$$

avec X_{appr} la matrice de données d'apprentissage et $\mathbf{y}_{appr}$ le caractère associé. Le modèle ainsi obtenu est appliqué sur l'échantillon de validation permettant d'obtenir une estimation du caractère $(\hat{\mathbf{y}})$ pour les individus de cet échantillon de validation et une estimation de la variance de l'erreur ${\sigma_{\mathbf{e}}}^2$:

$$\hat{\mathbf{y}}_{valid} = X_{valid}\hat{\boldsymbol{\beta}} \qquad \Rightarrow \qquad \hat{\sigma_{\mathbf{e}}} = \tfrac{1}{n-p-1}\textstyle\sum_{i=1}^{n}(y_i - \hat{y}_i)^2,$$

avec X_{valid} la matrice de données de validation et $\hat{\mathbf{y}}_{valid}$ l'estimation du caractère associé. Chaque individu faisant partie une fois de l'échantillon de validation, nous obtenons finalement une estimation du caractère pour tous les individus. Nous sommes donc capables de calculer la racine de l'erreur moyenne de prédiction : $RMSEP = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$. Cette erreur de prédiction représentera la fitness d'une solution (correspondant à un sous-ensemble de variables sélectionnées) dans l'algorithme d'optimisation et nous serons dans le cadre d'un problème de minimisation.

La validation croisée *leave-one-out* La validation croisée *leave-one-out* (LOO) est un cas particulier de la validation croisée k-fold où $k = n$. Ainsi, cette méthode se décompose en n itérations, à chaque itération le modèle est construit sur $n - 1$ individus puis validé sur l'individu restant. De même que précédemment, on obtient une estimation du caractère pour tous les individus, ce qui nous permet de calculer l'erreur de prédiction à minimiser.

3.2.3.2 Comparaison expérimentale des différents critères

Afin de définir le critère de qualité du modèle que nous allons utiliser dans la suite de l'étude, nous comparons sur données simulées et pseudo-réelles, les performances de l'algorithme utilisant chacun de ces trois critères en tant que fonction objectif. Pour chaque critère, l'ILS est exécuté sur l'échantillon d'apprentissage puis la racine carrée de l'erreur de prédiction est calculée sur l'échantillon de validation, et ce pour les 30 jeux simulés. Nous évaluons également la corrélation, sur l'échantillon de validation, entre les valeurs estimées du caractère et les valeurs réelles. L'objectif étant d'avoir la plus faible erreur de prédiction et la plus forte corrélation. Nous représentons les résultats sous forme de boîtes à moustaches.
Nous constatons que sur ces données simulées (Figure 3.3), le critère BIC donne les meilleurs résultats que ce soit en terme d'erreur de prédiction ou de corrélation. Nous observons parfois, comme c'est le cas ici, une symétrie entre les graphiques présentant les résultats en terme de RMSEP et ceux en terme de corrélation, le meilleur critère en terme de RMSEP (le plus faible) étant le même que le meilleur en terme de corrélation (le plus fort). Cependant, nous présenterons toujours les résultats suivant ces deux critères car nous verrons par la suite que ce n'est pas toujours symétrique.

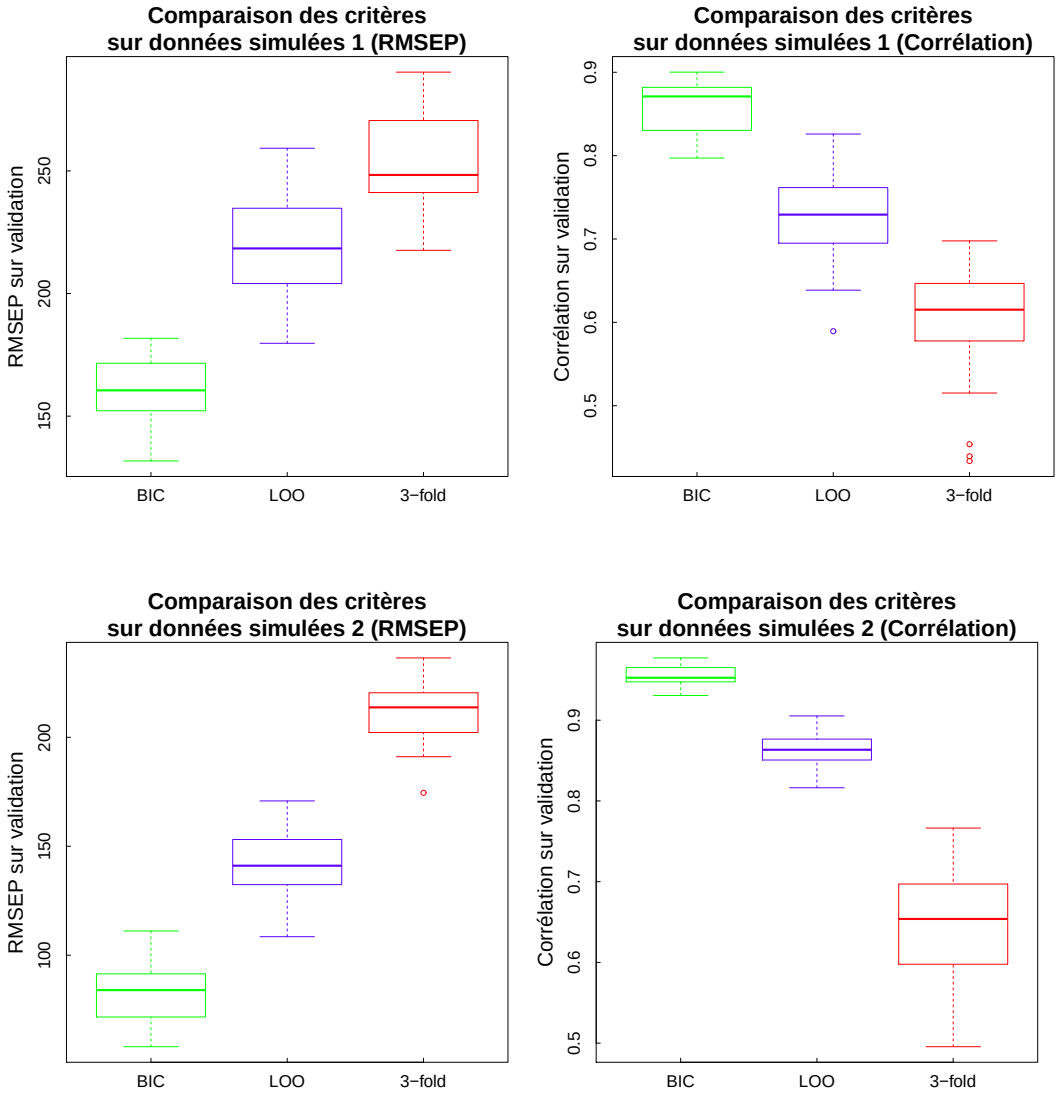

FIGURE 3.3 – Comparaison des 3 critères d'évaluation sur données simulées

La Figure 3.4 présente les résultats sur données pseudo-réelles. En terme d'erreur de prédiction, les résultats sont similaires avec les trois critères, avec un léger avantage pour la validation croisée 3-fold sur les données de QTLMAS 2011 et pour la validation croisée leave-one-out sur les données de QTLMAS 2010 (sachant que pour le critère LOO, 2 valeurs extrêmes, correspondant à des erreurs $\simeq 18$, ont été enlevées du graphique pour ne pas perdre en lisibilité). Cependant, en terme de corrélation, le critère BIC semble donner de meilleurs résultats.

De plus, pour un même nombre d'évaluations, l'algorithme avec le critère BIC est deux fois plus rapide qu'avec la validation croisée 3-fold et huit fois plus rapide qu'avec la validation croisée leave-one-out avec des temps d'exécution d'environ 1h, 2h et 8h pour 500 000 évaluations.

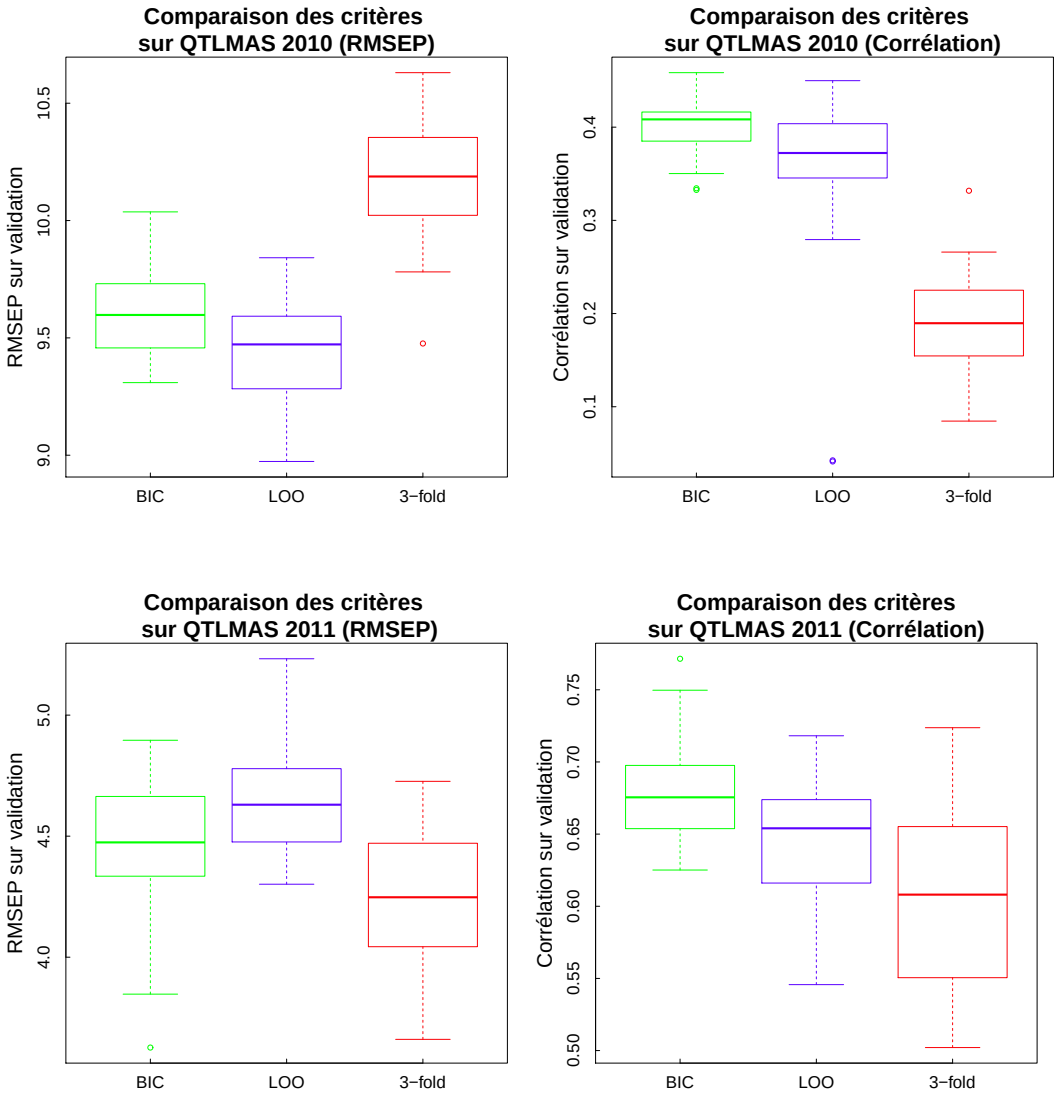

FIGURE 3.4 – Comparaison des 3 critères d'évaluation sur données pseudo-réelles

Conclusion ILS 1 **: Nous choisissons d'évaluer la qualité d'une solution par le critère BIC.**

3.2.4 Choix de l'initialisation de l'algorithme

La solution initiale qui correspond au premier sous-ensemble de variables sélectionnées, est basiquement générée aléatoirement, suivant une distribution uniforme donnant autant de chance à chaque variable d'être sélectionnée. Cependant, la recherche locale itérée est un algorithme stochastique dont la solution finale peut dépendre de la solution initiale. La question de l'initialisation est donc importante. Nous proposons ici de guider le choix des variables de la solution initiale à l'aide de la corrélation de chaque variable avec le caractère à prédire.
Ainsi, une probabilité de sélection π_j, proportionnelle à sa corrélation relative avec le caractère étudié, est attribuée à chaque variable $\mathbf{x}_j$:

$$\pi_j = \frac{\rho(\mathbf{x}_j, \mathbf{y})}{\sum_{j=1}^{p} \rho(\mathbf{x}_j, \mathbf{y})}, \tag{3.6}$$

avec $\mathbf{x}_j$ la variable étudiée, $\mathbf{y}$ le caractère d'intérêt et p le nombre total de variables. Nous proposons de choisir les variables de la solution initiale de l'algorithme suivant les probabilités $(\pi_1, ..., \pi_p)$ avec π_j la probabilité de sélectionner le SNP j. L'objectif visé d'une telle initialisation est d'améliorer la qualité de la solution initiale de l'algorithme afin d'accélérer sa convergence et ainsi améliorer la qualité de la solution finale.
Afin d'évaluer l'intérêt de guider la solution initiale en utilisant les corrélations des variables explicatives avec le caractère étudié, nous évaluons dans un premier temps les performances de l'algorithme en terme d'erreur de prédiction et de corrélation sur données simulées (Figure 3.5) puis sur données pseudo-réelles (Figure 3.6). Nous nous intéressons ensuite à la convergence de l'algorithme ainsi qu'à la qualité des solutions initiales (Figure 3.7).
En terme de qualité de la solution finale, que ce soit sur données simulées (Figure 3.5) ou pseudo-réelles (Figure 3.6), le guidage de l'initialisation par la corrélation avec le caractère étudié ne permet pas, contrairement à ce que nous avions pu penser, d'améliorer les résultats.

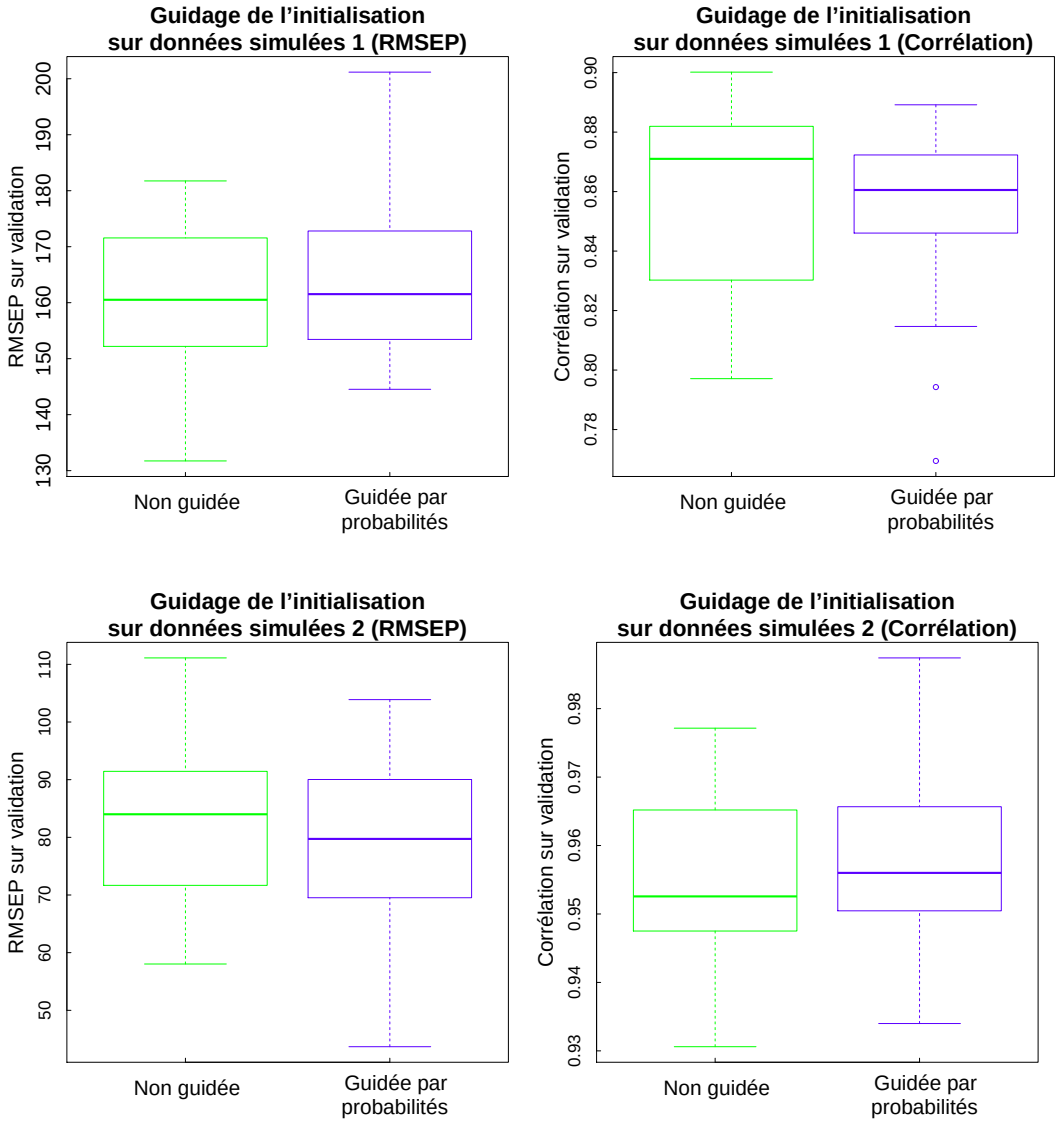

FIGURE 3.5 – Apport du guidage de la solution initiale sur données simulées

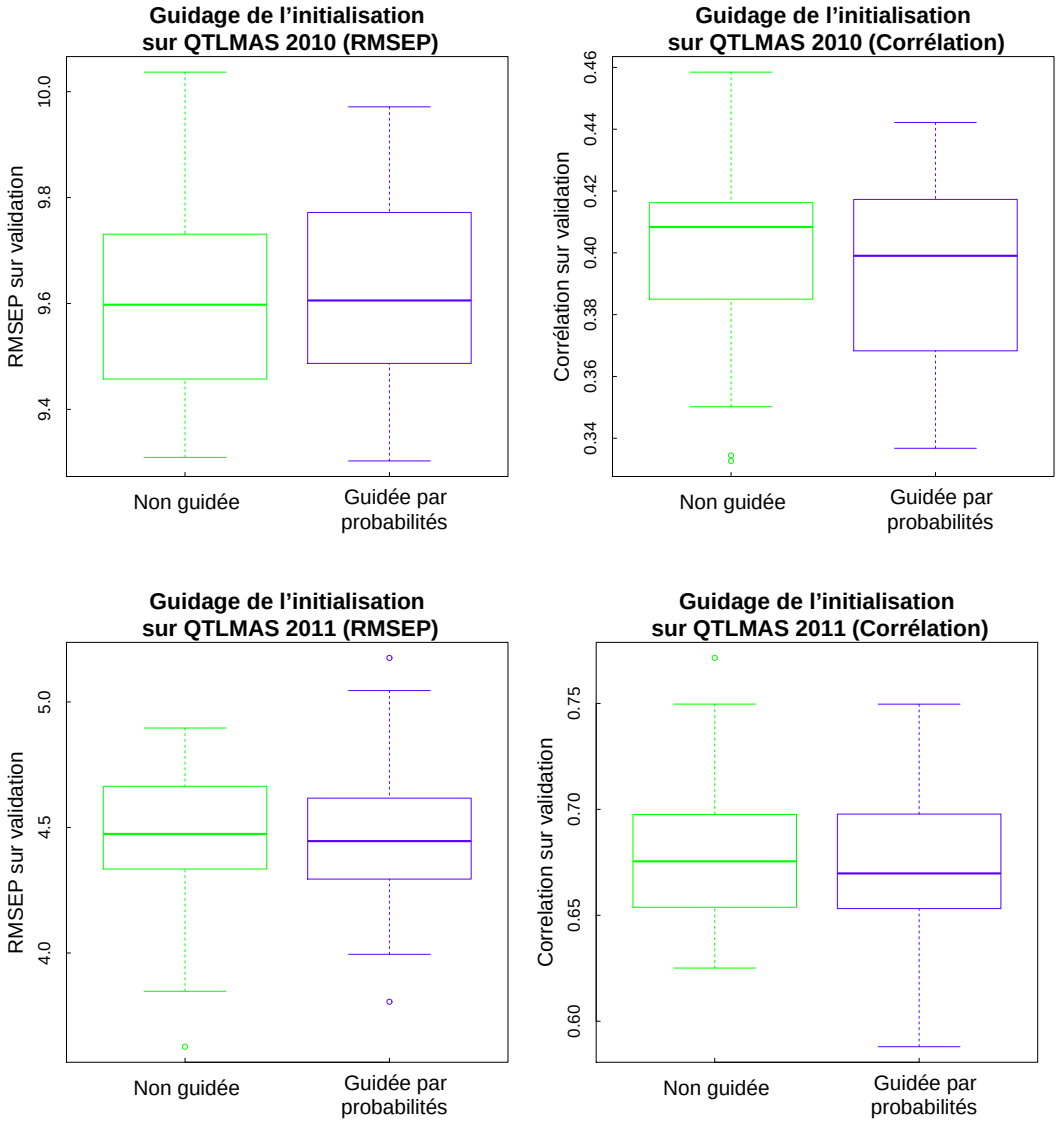

FIGURE 3.6 – Apport du guidage de la solution initiale sur données pseudo-réelles

La Figure 3.7 illustre les résultats en terme de convergence (partie gauche de la figure) et de qualité des solutions initiales (partie droite de la figure), sur l'échantillon d'apprentissage, sur données simulées 1 et sur les données pseudo-réelles de QTLMAS 2010 (les résultats sur données simulées 2 et sur QTLMAS 2011 étant similaires).
Les graphiques de la partie gauche de la figure représentent les courbes de convergence des 30 exécutions avec initialisation aléatoire (courbes vertes en pointillés) et des 30 exécutions avec initialisation guidée par les corrélations avec le caractère étudié (courbes bleues). Ce sont ici les optima locaux (solutions sans solution voisine améliorante) qui sont représentés, en terme de critère BIC sur l'échantillon d'appren-

tissage en fonction du nombre d'évaluations. Sur les données simulées, l'initialisation basée sur la corrélation avec le caractère étudié ne semble pas avoir d'influence sur la convergence de l'algorithme. En revanche, sur les données de QTLMAS 2010, nous remarquons que globalement, la convergence est plus rapide lorsque l'initialisation est guidée par les corrélations.
Les boîtes à moustaches de la Figure 3.7 (partie droite) représentent la qualité des solutions initiales générées uniformément ou avec les corrélations, en terme de critère BIC sur l'échantillon d'apprentissage.

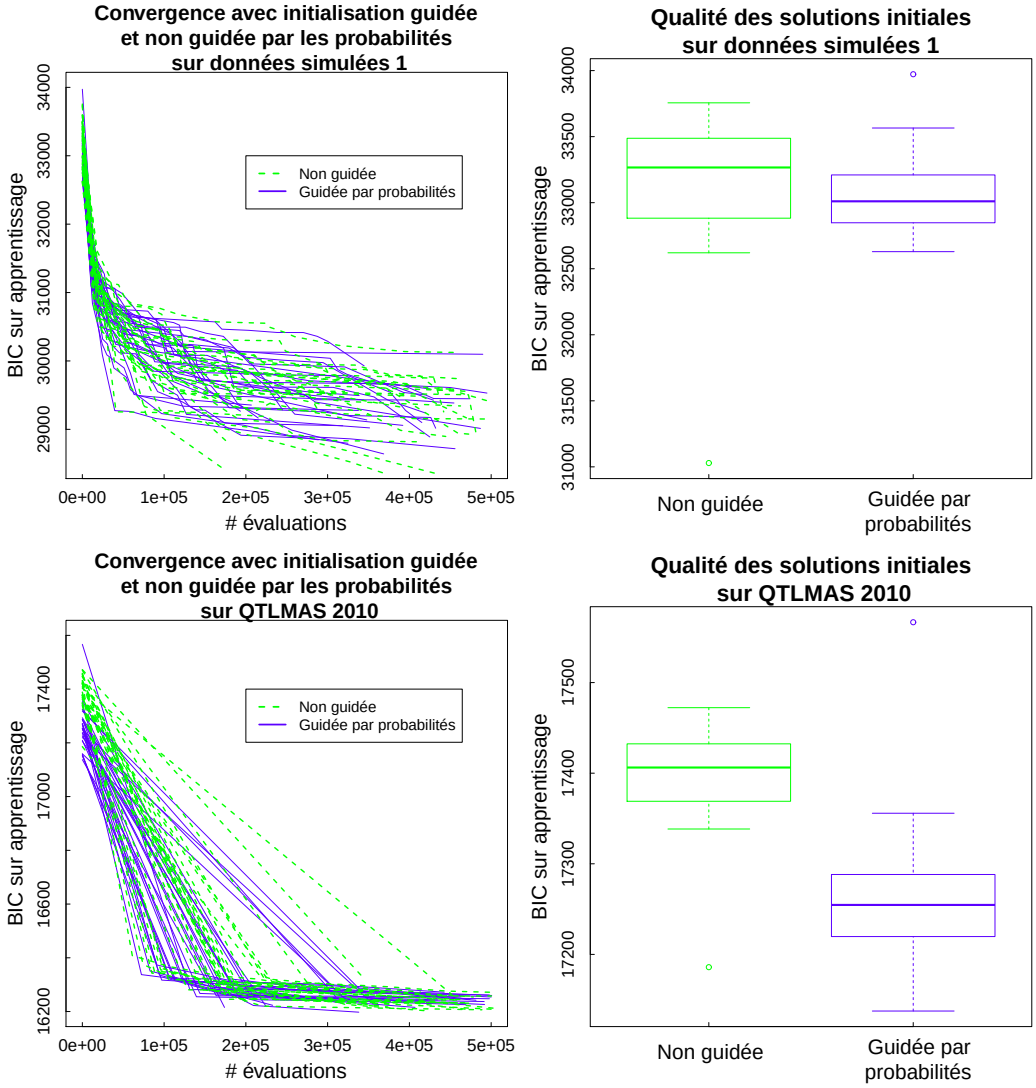

FIGURE 3.7 – Etude de la convergence et de la qualité des solutions initiales

Nous observons une différence significative entre les deux types d'initialisation, sur les données de QTLMAS 2010. Ce n'est en revanche pas le cas sur les données simulées. En effet, étant donné le grand nombre de variables étudiées,

les probabilités individuelles de tirage d'une variable donnée sont très faibles. Ainsi, même pour des variables fortement corrélées avec le caractère étudié, il sera difficile de les identifier lors du choix des variables à intégrer dans la solution initiale.

Conclusion ILS 2 **: L'initialisation avec les probabilités basées sur la corrélation ne permettant pas d'améliorer les résultats, nous décidons de conserver une initialisation uniforme.**

3.2.5 Choix de la perturbation

La recherche locale itérée consiste à appliquer plusieurs recherches locales successives. En effet, lorsqu'une recherche locale atteint un optimum local (une solution n'ayant pas de solution voisine améliorante), l'algorithme applique une perturbation à cet optimum et continue le parcours de l'espace de recherche à partir de cette nouvelle solution perturbée. La solution perturbée est acceptée même si elle n'améliore pas la qualité de la solution courante. L'idée principale de l'algorithme ILS est que la méthode de perturbation soit plus efficace qu'une approche redémarrant d'une solution uniforme. La perturbation est donc souvent basée sur l'opérateur de voisinage et consiste généralement en plusieurs applications de ce dernier. Dans notre approche, nous avons constaté que le nombre maximal de variables à sélectionner (fixé par l'utilisateur) est rapidement atteint (cf. discussion sur l'opérateur *bit-flip* en section 3.2.2). C'est pourquoi, plutôt que d'appliquer plusieurs fois l'opérateur de mutation (*bit-flip*), nous proposons de supprimer un pourcentage k des variables présentes dans la solution à perturber (l'optimum local). Nous comparons ici les résultats obtenus en utilisant 4 valeurs de k représentant des pourcentages faibles à élevés de variables supprimées : 5%, 10%, 20% et 40%. La Figure 3.8 présente les résultats sur données simulées et la Figure 3.9 sur données pseudo-réelles.

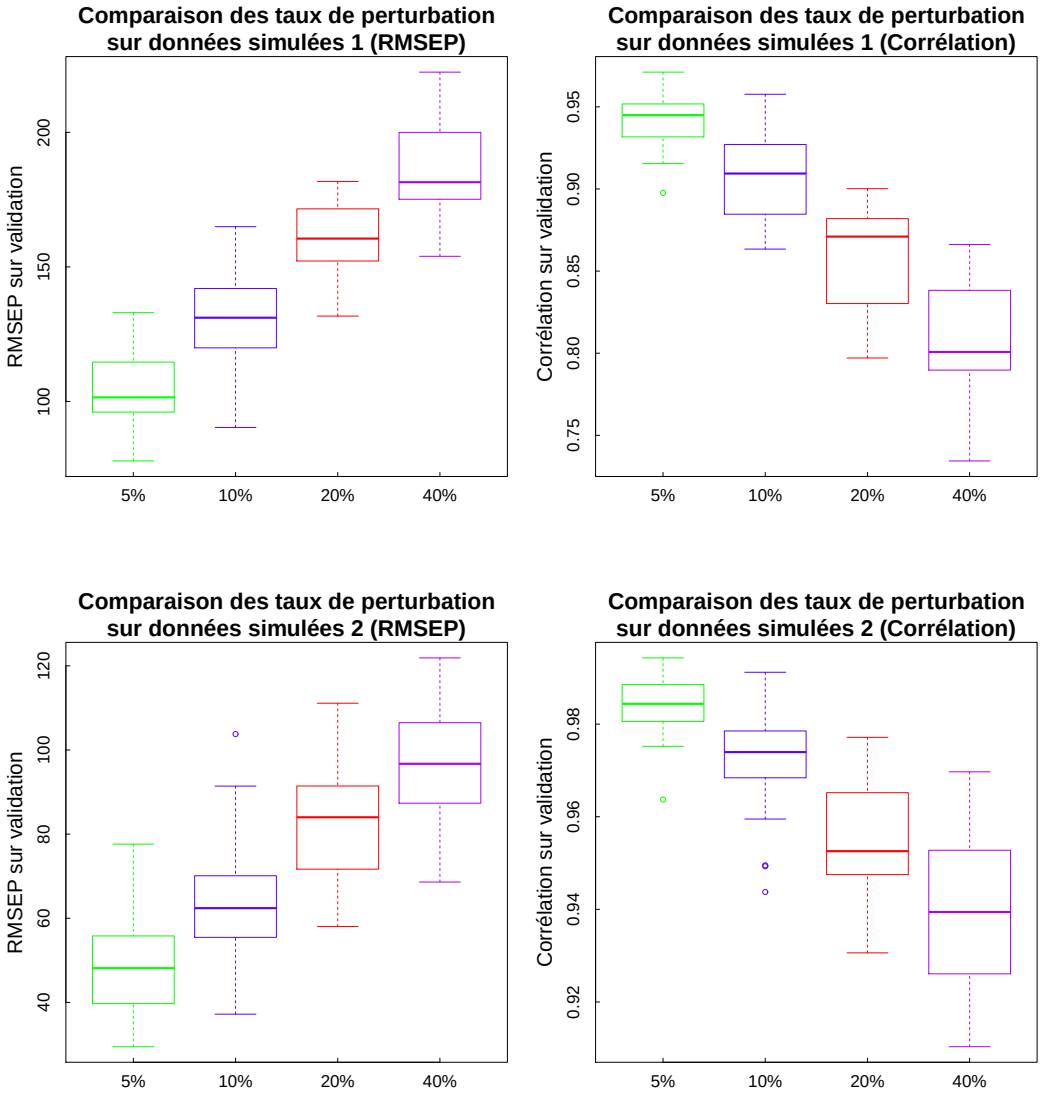

FIGURE 3.8 – Comparaison des différents taux de perturbation sur données simulées

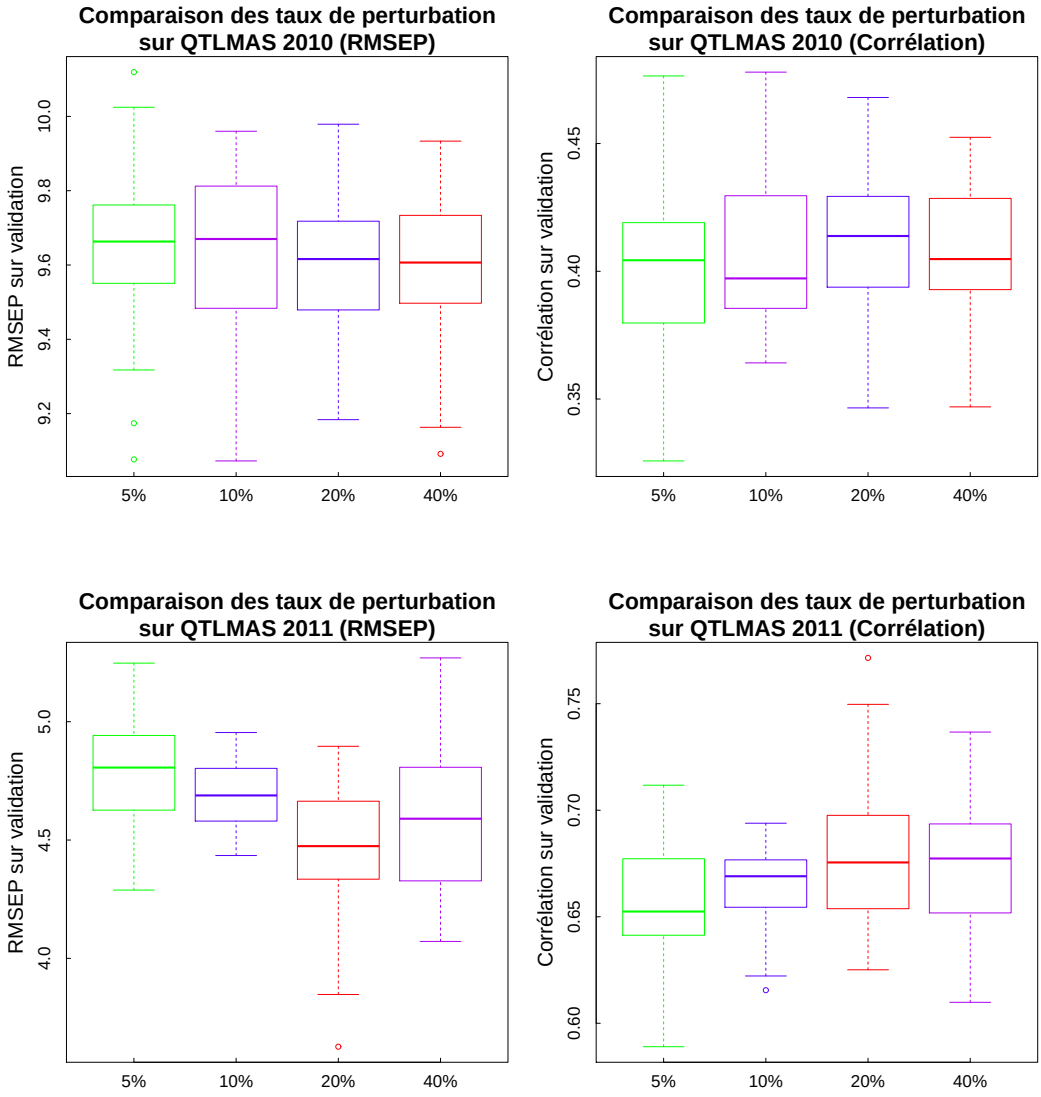

FIGURE 3.9 – Comparaison des différents taux de perturbation sur données pseudo-réelles

Les résultats sur données simulées montrent que les meilleures solutions sont obtenues avec le plus faible taux de perturbation (5%), tandis qu'il ne semble pas y avoir de différence significative entre l'utilisation des différents taux de perturbation sur données pseudo-réelles. Pour mieux comprendre cela, nous étudions la convergence de l'algorithme en fonction des différents taux de perturbation.

La Figure 3.10 illustre la convergence de l'algorithme suivant les différents taux de perturbation, avec un même départ (une même solution initiale et les mêmes mouvements aléatoires), sur les données simulées 1. La première descente est donc identique jusqu'à la première perturbation. Sur la partie gauche de la figure, les fitness (BIC) de toutes les solutions acceptées sont représentées. Ces courbes illustrent donc la convergence de l'algorithme sur les 500 000 évaluations que nous lui avons fixées. Nous remarquons que globalement, plus le taux de perturbation est faible plus les solutions explorées sont intéressantes. Sur la partie droite, un zoom sur les 15 000 premières évaluations permet de constater, comme attendu, que plus le taux de perturbation est faible moins la solution est dégradée lors de la perturbation. Cependant, on remarque également l'intérêt de perturber plus fortement une solution pour aller explorer d'autres zones de l'espace de recherche. En effet aux alentours de 10 000 évaluations par exemple, l'algorithme perturbé de 10% est parti d'une solution beaucoup moins intéressante que celle de l'algorithme perturbé de 5% mais l'optimum local qu'il atteint finalement est de bien meilleure qualité.

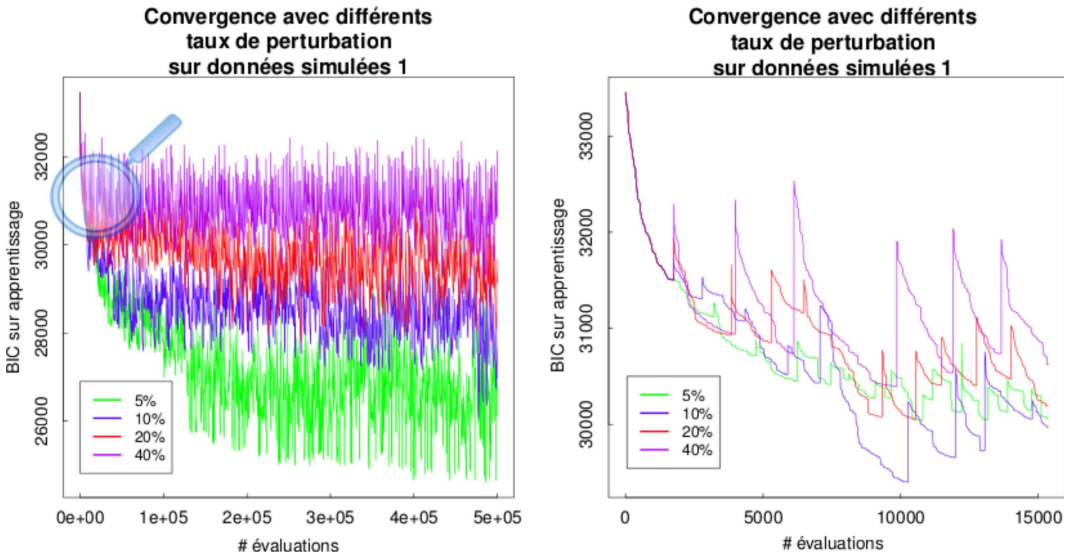

FIGURE 3.10 – Convergence en fonction des différents taux de perturbation sur données simulées 1

Sur la Figure 3.11, nous représentons les mêmes courbes que précédemment mais sur les données pseudo-réelles de QTLMAS 2010. Comme sur les données simulées nous remarquons que plus le taux de perturbation est élevé plus la solution perturbée est de mauvaise qualité. En revanche, contrairement aux données simulées nous ne remarquons pas de distinction globale entre les différents taux. Notons que l'écart entre la fitness de la solution aléatoire de départ et la fitness de la solution finale est beaucoup moins important sur ce jeu de données pseudo-réelles (amélioration d'environ 12%) que sur les données simulées (amélioration d'environ 70%).

Cette faible différence de fitness entre une mauvaise et une bonne solution pourrait donc expliquer la faible différence observée entre l'utilisation des différents taux de perturbations.

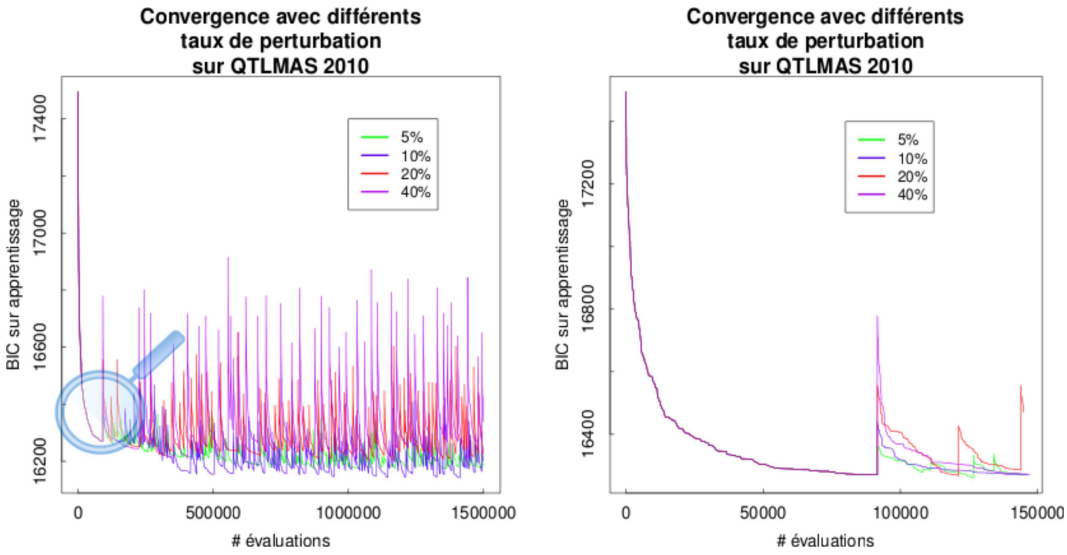

Figure 3.11 – Convergence en fonction des différents taux de perturbation sur QTLMAS 2010

Conclusion ILS 3 **: Le taux de perturbation de 5% nous permettant d'obtenir les meilleurs résultats sur données simulées, nous décidons de l'utiliser pour la suite.**

3.2.6 Critère d'arrêt

La recherche locale itérée est une méthode itérative ; elle ne s'arrête pas par elle-même. Un critère d'arrêt doit donc être défini. Nous proposons ici de laisser la méthode converger, et de l'arrêter lorsqu'un nombre maximum donné d'évaluations est atteint. La Figure 3.12 représente, pour 5 exécutions de l'algorithme, l'évolution du critère BIC de la meilleur solution (mise à jour dès qu'un nouveau meilleur optimum local est trouvé).
Nous constatons que sur les données pseudo-réelles, 1 500 000 évaluations semblent permettre à l'algorithme de converger. En revanche, sur les données simulées, 10 000 000 évaluations sont nécessaires à l'algorithme pour converger. La convergence est plus ou moins rapide suivant le jeu de données, il faut donc faire des expérimentations afin de pouvoir déterminer le nombre d'évaluations nécessaires.

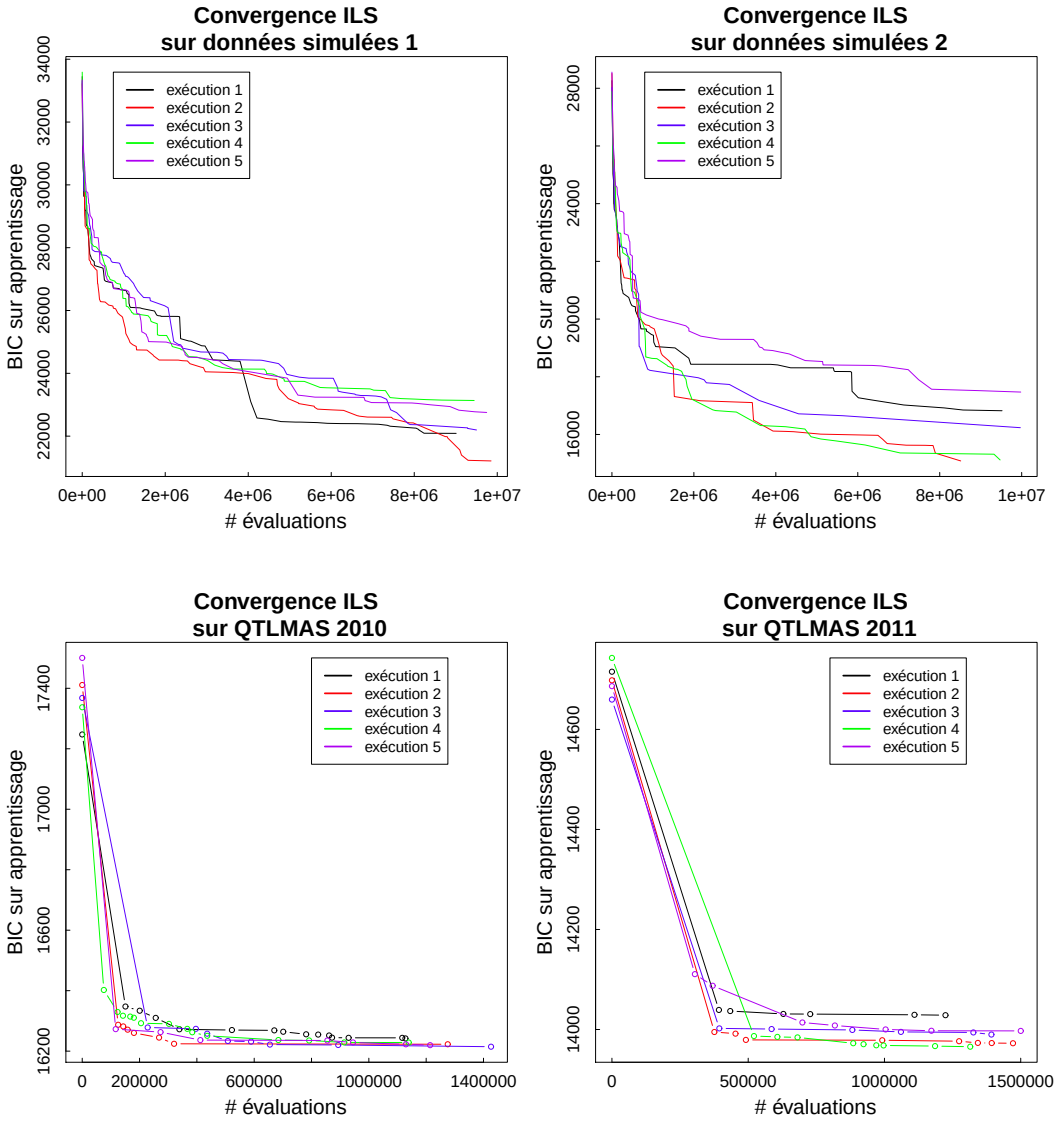

FIGURE 3.12 – Convergence de l'algorithme

3.2.7 Conclusion

Jusqu'ici, les différents composants de la recherche locale itérée ont été étudiés et leur pertinence évaluée. Nous pouvons nous poser la question de la meilleure utilisation de cet algorithme ILS :

- utilisation seule en partant d'une solution uniforme,
- hybridation avec une autre méthode de la littérature faisant de la sélection de variables.

Nous proposons donc d'étudier la pertinence d'une hybridation de la recherche locale itérée avec une approche classique de sélection de variables.

3.3 Hybridation

En référence à la taxonomie proposée pour des métaheuristiques hybrides [Jourdan 2009], afin d'accélérer la convergence et d'améliorer les performances de notre approche, nous proposons de mettre en place une hybridation de type relais haut niveau (*high level relay heuristic*). Ce type d'hybridation consiste à appliquer deux méthodes séquentiellement (Figure 3.13).

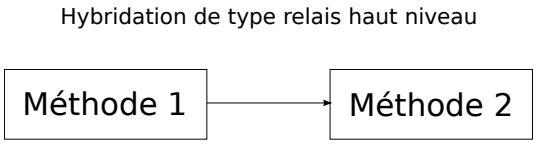

Figure 3.13 – Hybridation de type relais haut niveau

Dans la méthode lasso, présentée dans le chapitre précédent (section 2.2.2), il est possible de fixer un nombre maximal k de variables sélectionnées en incluant le paramètre $dfmax = k$ dans la fonction *glmnet* du logiciel *R*. Cela nous permet alors d'obtenir un modèle prédictif à partir d'un sous-ensemble de variables de taille fixée et ainsi de proposer une réponse à notre problématique.
Nous proposons ici d'hybrider notre approche avec cette méthode. En effet, nous commençons par appliquer la méthode lasso pour laquelle nous fixons le nombre maximal de variables à sélectionner. Nous obtenons donc un sous-ensemble de variables intéressantes. Nous initialisons ensuite notre algorithme avec ce sous-ensemble généré par la méthode lasso.
Nous comparons les résultats de notre approche seule avec ceux de notre approche hybridée avec lasso ainsi qu'avec ceux de la méthode lasso seule.

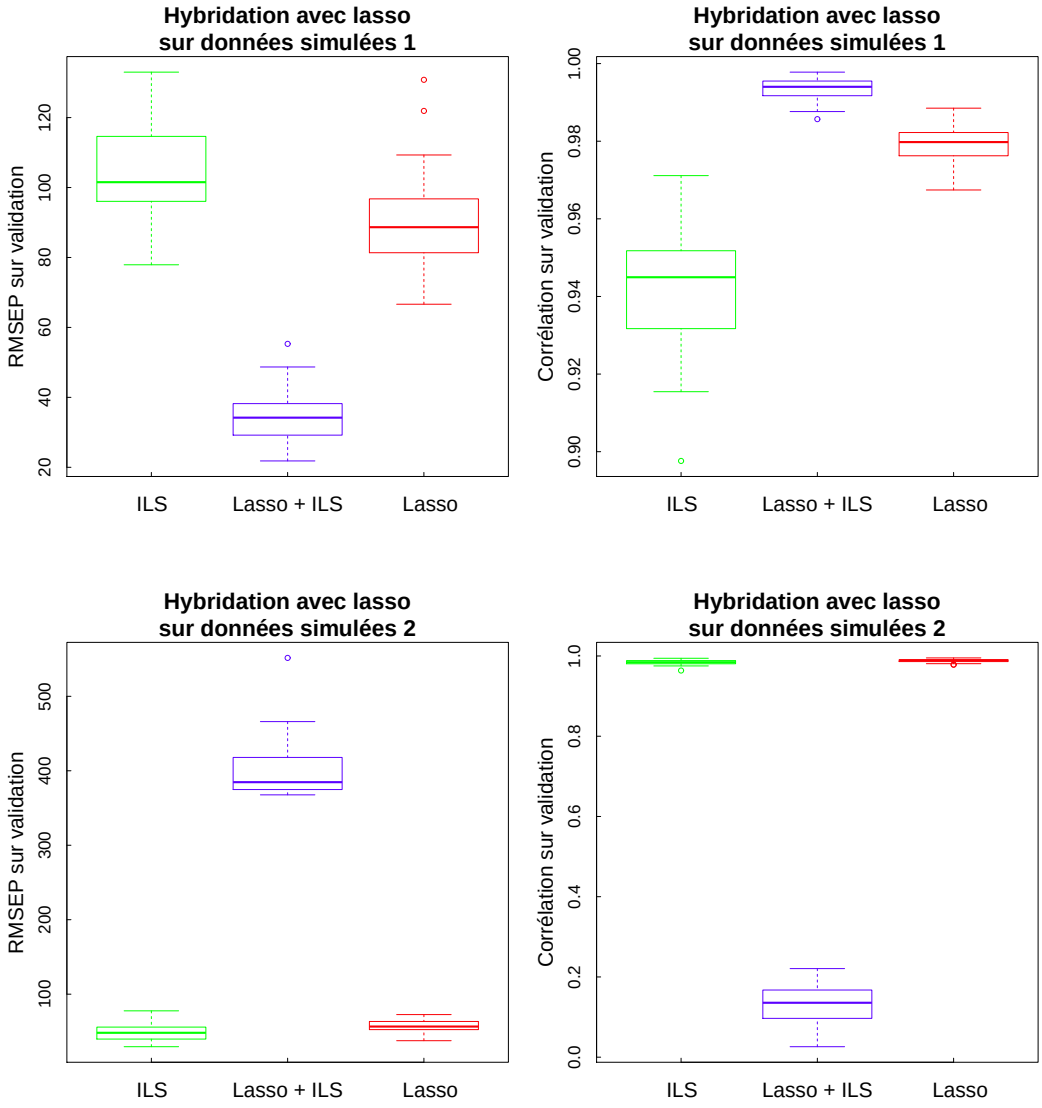

FIGURE 3.14 – Hybridation avec lasso sur données simulées

Sur les données simulées 1, la recherche locale itérée hybridé avec la méthode lasso donne des résultats significativement meilleurs que l'algorithme ILS seul ou que la méthode lasso. En revanche, ce n'est pas le cas sur les données simulées 2 où l'hybridation avec la méthode lasso dégrade la qualité finale de l'algorithme notre approche.

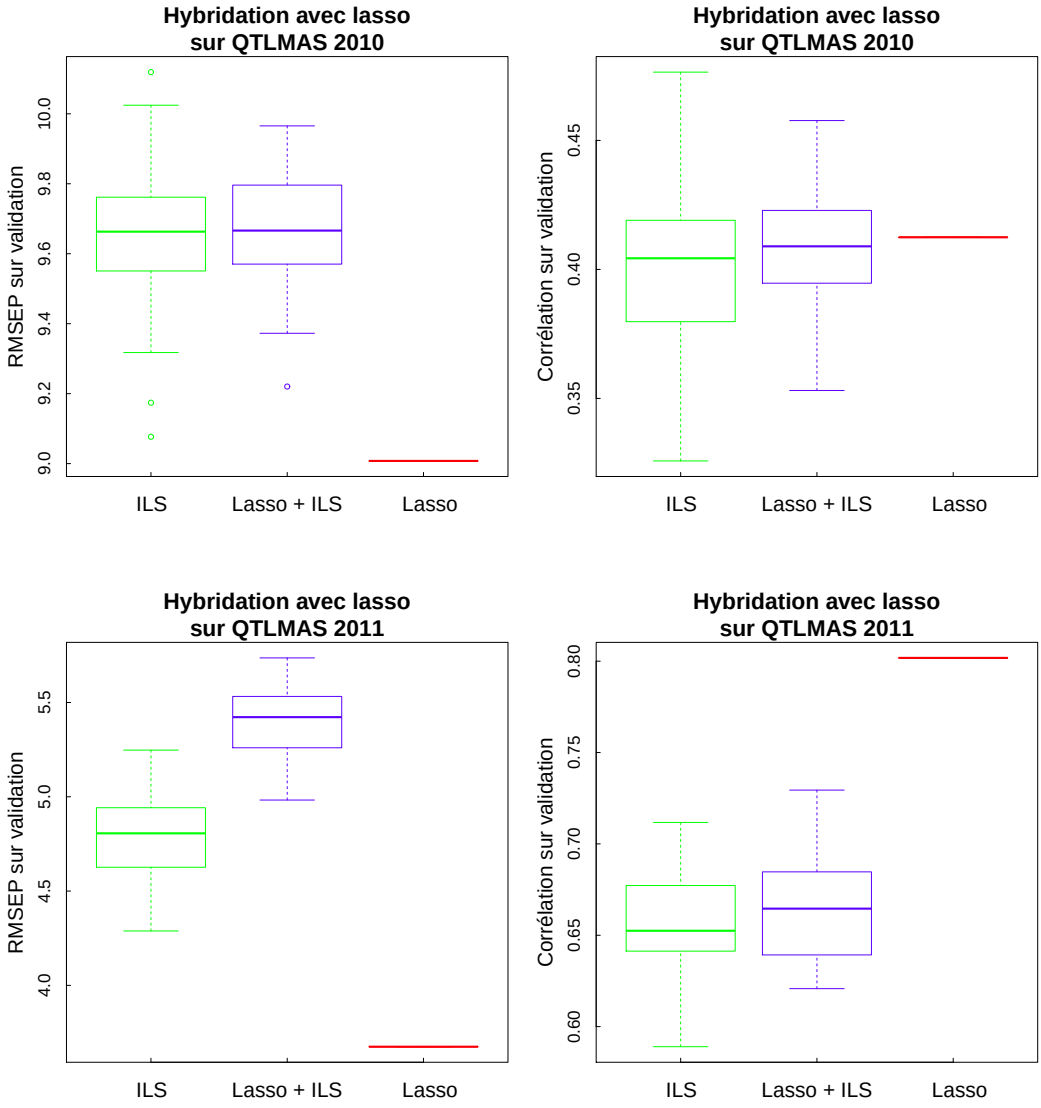

FIGURE 3.15 – Hybridation avec lasso sur données pseudo-réelles

Sur les données de QTLMAS 2010 (Figure 3.15), l'hybridation avec la méthode lasso n'améliore pas les résultats par rapport à l'algorithme ILS seul. De plus, les résultats de l'hybridation sur QTLMAS 2011 sont significativement moins bons. Étant donné que nous nous attendions à une amélioration des résultats grâce à l'hybridation, et afin de mieux comprendre le comportement de l'algorithme, nous nous intéressons aux résultats sur l'échantillon d'apprentissage (Figure 3.16).

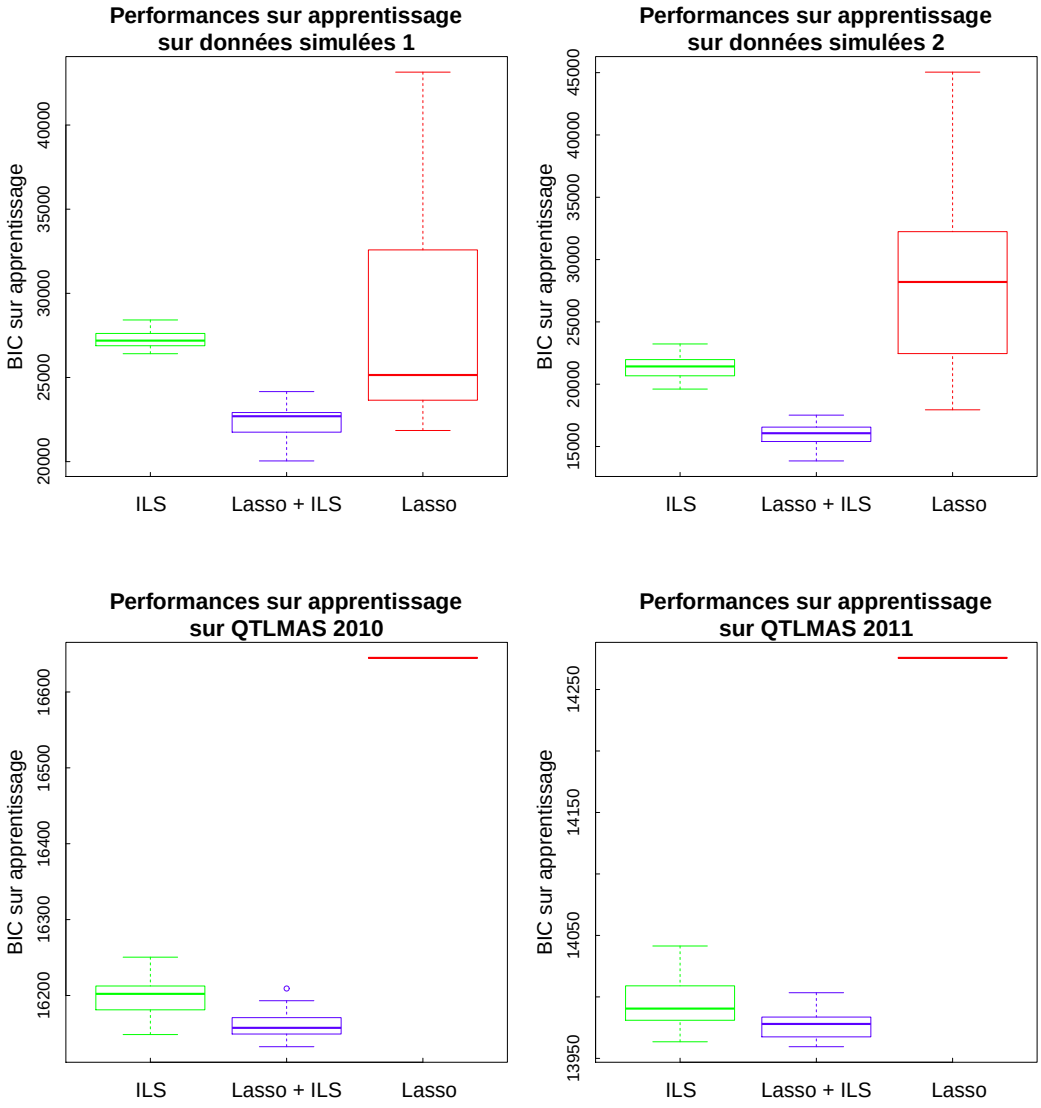

FIGURE 3.16 – Hybridation avec lasso - résultats sur apprentissage

Nous remarquons que l'utilisation de notre algorithme à la suite de la méthode lasso permet d'en améliorer les résultats sur l'échantillon d'apprentissage. Remarquez qu'en aucun cas la méthode ILS n'aurait pu dégrader la solution initiale si celle ci était réalisable. Nous améliorons également les performances de notre algorithme non hybridé. Cependant, cette amélioration visible sur les données d'apprentissage n'est pas répercutée sur les données de validation. Nous sommes donc dans le cas du sur-apprentissage ou sur-ajustement (*overfitting*) où le modèle colle trop aux données d'apprentissage et se généralise donc difficilement, donnant de moins bons résultats sur les données de validation. Ce problème de sur-apprentissage est principalement dû au fait que nous avons un faible nombre d'individus, ne nous

permettant donc pas d'utiliser des individus différents au cours de l'algorithme pour valider sa capacité à prédire.

Conclusion ILS 4 **: Étant donné les résultats non concluants de cette hybridation, nous choisissons de ne pas conserver ce processus.**

3.4 Analyses expérimentales

Après avoir étudié différentes améliorations possibles de notre algorithme de base, nous sélectionnons finalement la meilleure configuration de l'ILS adaptée à notre problème :

- initialisation : uniforme,
- voisinage : opérateur bit-flip,
- perturbation : suppression de 5% des variables de la solution courante,
- évaluation d'une solution : modèle de régression sur lequel le critère BIC est calculé et devient la fitness (à minimiser) de la solution,
- critère d'arrêt : nombre maximal fixé d'évaluations.

Sur les jeux de données simulées que nous avons générés, ainsi que sur les données pseudo-réelles de QTLMAS 2010 et 2011, nous menons dans un premier temps une étude qualitative puis une étude quantitative. L'étude qualitative consiste à évaluer les performances de l'approche en terme de variables sélectionnées et l'étude quantitative en terme d'erreur de prédiction sur l'échantillon de validation et de corrélation entre le caractère estimé et le caractère réel (sur l'échantillon de validation).
Nous comparons nos résultats avec ceux des méthodes classiques de la littérature : elastic net (EN), lasso et sparse PLS (sPLS). L'objectif de notre travail étant de sélectionner un sous-ensemble de variables de taille maximale fixée (ici 96 variables car c'est une taille standard de mini-puce pour marqueurs SNPs), cette limite est imposée à notre algorithme. Il est possible également pour les méthodes lasso et elastic net de leur imposer un nombre limite de variables à sélectionner en utilisant l'option $dfmax$ de la procédure *glmnet* du logiciel R, nous nous comparons donc également à ces approches limitées à 96 variables. Nous les notons EN96 et L96 dans la suite. Sur les graphiques, nous séparons par une ligne verticale en pointillé noir les méthodes non limitées en nombre de variables (à gauche) des méthodes limitées à 96 variables (à droite).

3.4.1 Sélection de variables

Pour analyser la pertinence des variables sélectionnées nous nous basons dans un premier temps sur les jeux de données simulées. En effet, sur ces jeux de données nous avons 96 variables significatives à retrouver (ligne verte en pointillé sur les graphiques). La Figure 3.17 représente le nombre de vrais positifs retrouvés par chacune des approches. Pour une approche donnée, un vrai positif est une variable

sélectionnée qui est réellement significative, c'est-à-dire qui fait partie des 96 variables significatives fixées. Les graphiques nous permettent de constater que, sur les données simulées 1 (partie gauche) comme sur les données simulées 2 (partie droite), notre approche sélectionne un plus grand nombre de vrais positifs que les approches classiques limitées à 96 variables. Les méthodes lasso et elastic net pour lesquelles nous ne limitons pas le nombre de variables participant au modèle retrouvent assez facilement les variables significatives mais sélectionnent beaucoup plus de variables que les autres approches (Table 3.1 et Table 3.2).

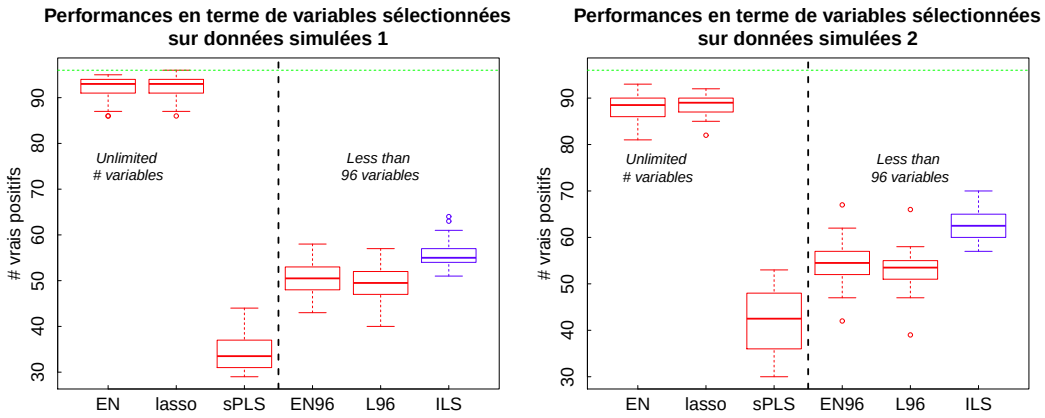

FIGURE 3.17 – Capacité des approches à sélectionner les bonnes variables

Nous indiquons dans la Table 3.1 et la Table 3.2 le nombre moyen de variables sélectionnées par les différentes approches. Nous indiquons ensuite le pourcentage de faux positifs, c'est à dire le nombre de variables sélectionnées qui ne sont en réalité pas significatives divisé par le nombre total de variables sélectionnées par la méthode. Ces tableaux détaillent également la répartition des différents effets trouvés. En effet, sur les jeux de données simulées, nous savons que nous avons 32 variables à effets faibles ($\beta \simeq 4$), 32 à effets moyens ($\beta \simeq 16$) et 32 à effets forts ($\beta \simeq 64$). Nous nous intéressons ici aux pourcentages de variables trouvées (rapport entre le nombre de variables sélectionnées par l'approche et le nombre de variables réellement significatives) pour chaque type d'effet (faible, moyen, fort) et au total.

		EN	Lasso	sPLS	EN96	L96	ILS
# var. select.		205,3	201,6	89,3	98,5	98,5	96
% faux positifs		56	55	62	50	48	**42**
% variables trouvées	*Effets faibles*	88,8	88,8	0,9	1,9	2,1	**9,1**
	Effets moyens	99,8	99,6	12,4	52,3	57,1	**68,1**
	Effets forts	100	99,9	93,8	**98,7**	**98,7**	96,3
	Total	96,2	96	35,8	51,1	52,6	**57,8**

Table 3.1 – Qualité des variables sélectionnées sur données simulées 1

		EN	Lasso	sPLS	EN96	L96	ILS
# var. select.		189,2	181,7	173,3	98	98,1	96
% faux positifs		53	52	75	46	45	**35,1**
% variables trouvées	*Effets faibles*	79	78,7	5,3	4,8	5,9	**22,8**
	Effets moyens	97,4	97	36,9	64,7	67,2	**78,1**
	Effets forts	100	99,8	91,8	**96,7**	96,6	94,7
	Total	92,2	91,8	44,7	55,4	56,6	**65,1**

Table 3.2 – Qualité des variables sélectionnées sur données simulées 2

Nous pouvons tout d'abord constater que les approches lasso et elastic net non limitées en nombre de variables sélectionnées ont un nombre de faux positifs élevé. Ceci implique qu'il est difficile, étant donné un ensemble de variables sélectionnées, de savoir si celles-ci sont pertinentes ou non. Concernant la capacité des différentes méthodes à retrouver les effets faibles, moyens ou forts, comme attendu, quelque soit l'approche utilisée les effets forts sont toujours plus facilement retrouvés que les effets faibles. Parmi les approches limitées à 96 variables, sur les deux scénarios de simulation, notre approche semble légèrement plus performante pour retrouver les variables ayant un effet faible ou moyen mais légèrement moins performante pour les effets forts.

Concernant les données de QTLMAS, nous connaissons la position des QTL (*Quantitative Trait Locus*), qui sont des régions du génome ayant une influence sur la caractère d'intérêt. Il y a 37 QTL pour les données de QTLMAS 2010 et 8 pour les données de QTLMAS 2011. Nous pouvons donc définir si une variable sélectionnée par une approche est réellement significative ou non. Nous considérons ici qu'un SNP est significatif s'il fait parti des 5 SNPs positionnés avant le QTL ou des 5 SNPs positionnés aprés (ce qui est équivalent à environ $\pm$100 000 paires de bases). La Figure 3.18 illustre cela :

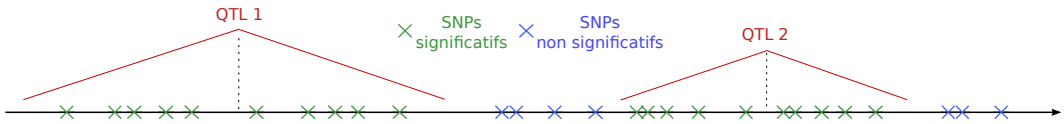

FIGURE 3.18 – SNPs significatifs et QTL

Nous rapportons dans les tables 3.3 et 3.4, le nombre de variables sélectionnées par chaque approche et le pourcentage de faux positifs (pourcentage de variables sélectionnées non significatives). Nous indiquons également le pourcentage de QTL trouvés par chacune des approches, c'est-à-dire le nombre de QTL pour lesquels au moins une variable (SNP) a été trouvée divisé par le nombre total de QTL. Rappelons que pour notre approche nous faisons une moyenne sur 30 exécutions de l'algorithme.

	EN	**Lasso**	**sPLS**	**EN96**	**L96**	**ILS**
# var. select.	297	191	3796	88	70	94
% faux positifs	66	61,8	63,4	**58**	61,4	64,5
% QTL trouvés	94,6	91,9	100	**64,9**	54,1	35,5

TABLE 3.3 – Qualité des variables sélectionnées sur QTLMAS 2010

Nous constatons dans la Table 3.3 que, sur les données de QTLMAS 2010, seule la méthode sparse PLS retrouve tous les QTL mais elle sélectionne pour cela un très grand nombre de variables et donc de faux positifs. Parmi les méthodes limitées à 96 variables, sur ces données, notre approche ne parvient pas à retrouver les QTL aussi bien que les méthodes classiques.

	EN	**Lasso**	**sPLS**	**EN96**	**L96**	**ILS**
# var. select.	36	33	78	55	41	70,7
% faux positifs	69,4	66,7	80,8	56,4	65,9	**56**
% QTL trouvés	37,5	37,5	37,5	**37,5**	**37,5**	25

TABLE 3.4 – Qualité des variables sélectionnées sur QTLMAS 2011

Sur les données de QTLMAS 2011 (Table 3.4), pour lesquelles il n'y a que 8 QTL, notre approche semble légèrement moins performante (on ne trouve en moyenne que 2 QTL) que les approches classiques (qui en trouvent 3). Cependant, aucune des méthodes ne semble retrouver les 8 QTL de ce jeu de données.

3.4.2 Temps d'exécution

La Table 3.5 présente les temps d'exécution des différentes méthodes sur les jeux de données simulées et pseudo-réelles.

	EN	**Lasso**	**sPLS**	**EN96**	**L96**	**ILS**
Simulation 1	17 min.	1 min.	2 min.	10 min.	35 sec.	3h50
Simulation 2	12 min.	26 sec.	1 min.	7 min.	15 sec.	2h50
QTLMAS 2010	30 min.	1 min	2 min.	8 min.	42 sec.	3h30
QTLMAS 2011	26 min.	77 sec.	1 min.	6 min.	27 sec.	2h

Table 3.5 – Temps d'exécution des différentes méthodes

Le temps d'exécution de notre approche basée sur une recherche locale itérée est nettement supérieur aux temps des méthodes classiques. Cependant, compte tenu du temps nécessaire à la collecte et au pré-traitement des données, la durée d'exécution de l'algorithme n'est pas un élément essentiel de notre problématique tant qu'il reste de l'ordre de la journée. Néanmoins, il paraît intéressant de paralléliser les évaluations de l'algorithme afin de diminuer son temps d'exécution. Nous verrons cela dans le chapitre suivant lors de l'utilisation d'un autre type de métaheuristique.

3.4.3 Évaluation des résultats

Comme précédemment, nous présentons sur une figure les résultats sur données simulées puis sur la suivante les résultats sur données pseudo-réelles. Chaque figure regroupe 4 graphiques, les deux du haut concernent les résultats sur 1 type de données (simulées 1 ou QTLMAS 2010) et les deux du bas sur l'autre type de données (simulées 2 ou QTLMAS 2011). Les graphiques de la partie gauche de la figure représentent les performances en terme d'erreur de prédiction (RMSEP), et ceux de la partie droite en terme de corrélation. Rappelons que la droite noire verticale en pointillé permet de distinguer les méthodes non limitées en nombre de variables sélectionnées des méthodes auxquelles nous avons fixé un nombre limite maximal (96) de variables à sélectionner.
Pour les jeux de données simulées, nous avons 30 jeux de données, nous illustrons donc les résultats au travers de boîtes à moustaches représentant les résultats d'une exécution par jeu de données (30 valeurs). En revanche, sur les données pseudo-réelles, nous avons un seul jeu de données donc une seule exécution pour les méthodes classiques (déterministes), ce qui donne une seule valeur (un trait sur les graphiques). Notre algorithme est quant à lui exécuté 30 fois sur ces jeux de données pseudo-réelles, ce qui nous permet d'illustrer les résultats sous forme de boîtes à moustaches. Les figure 3.19 et 3.20 comparent notre approche (ILS) avec les méthodes classiques précédemment citées, sur données simulées et pseudo-réelles, en terme d'erreur de prédiction et de corrélation sur l'échantillon de validation.

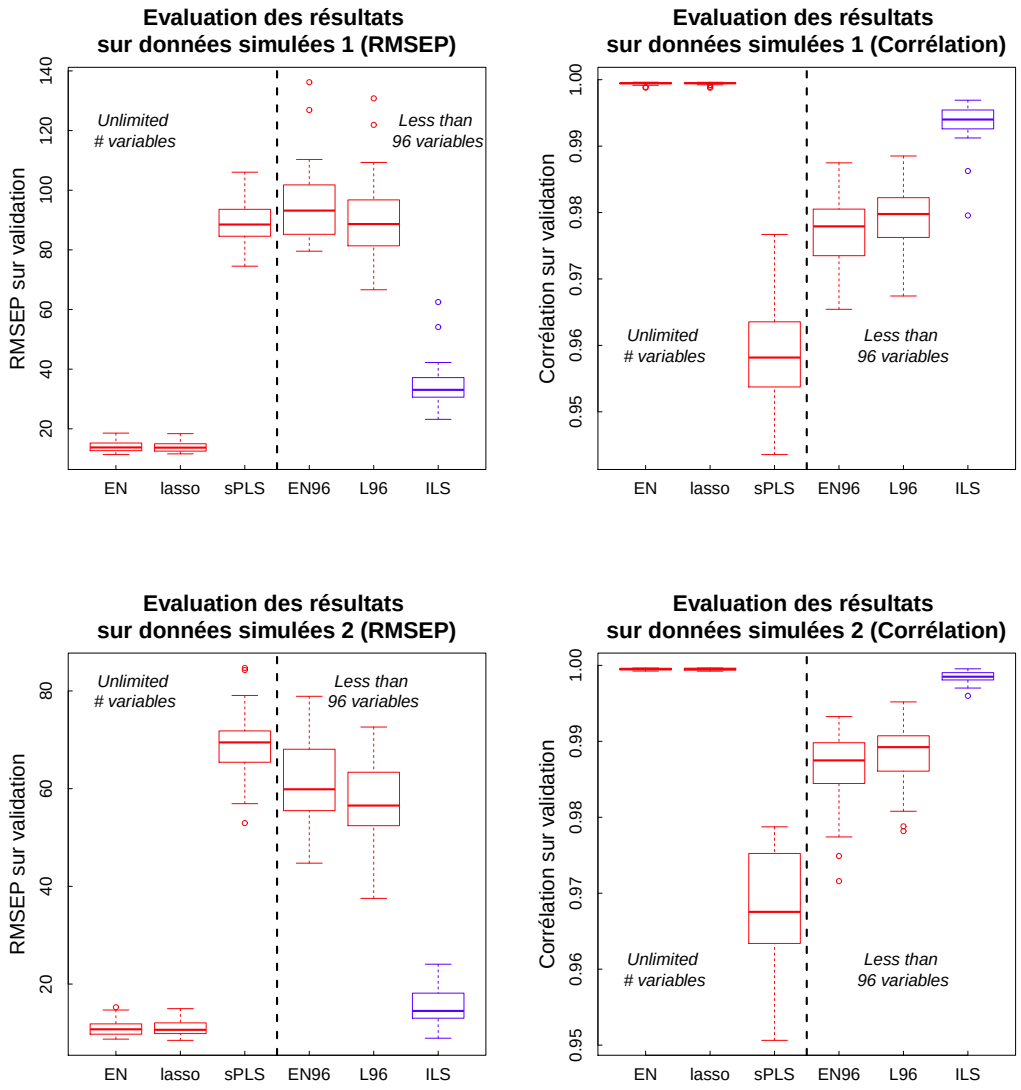

FIGURE 3.19 – Comparaison de l'algorithme avec des méthodes classiques sur données simulées

La Figure 3.19 montre qu'en terme d'erreur de prédiction, sur les deux scénarios de simulation, notre approche donne de meilleurs résultats que les méthodes classiques limitées à 96 variables. Les méthodes elastic net et lasso non limitées donnent de meilleurs résultats mais, comme nous l'avons vu précédemment, sélectionnent beaucoup de variables, ce qui ne répond pas à notre problématique. Nous obtenons également de meilleurs résultats que la méthode sparse PLS. En terme de corrélation, les résultats de notre approche sont meilleurs que ceux des méthodes classiques limitées à 96 variables. Ces résultats montrent clairement la pertinence de l'approche dans un contexte idéal (données simulées).

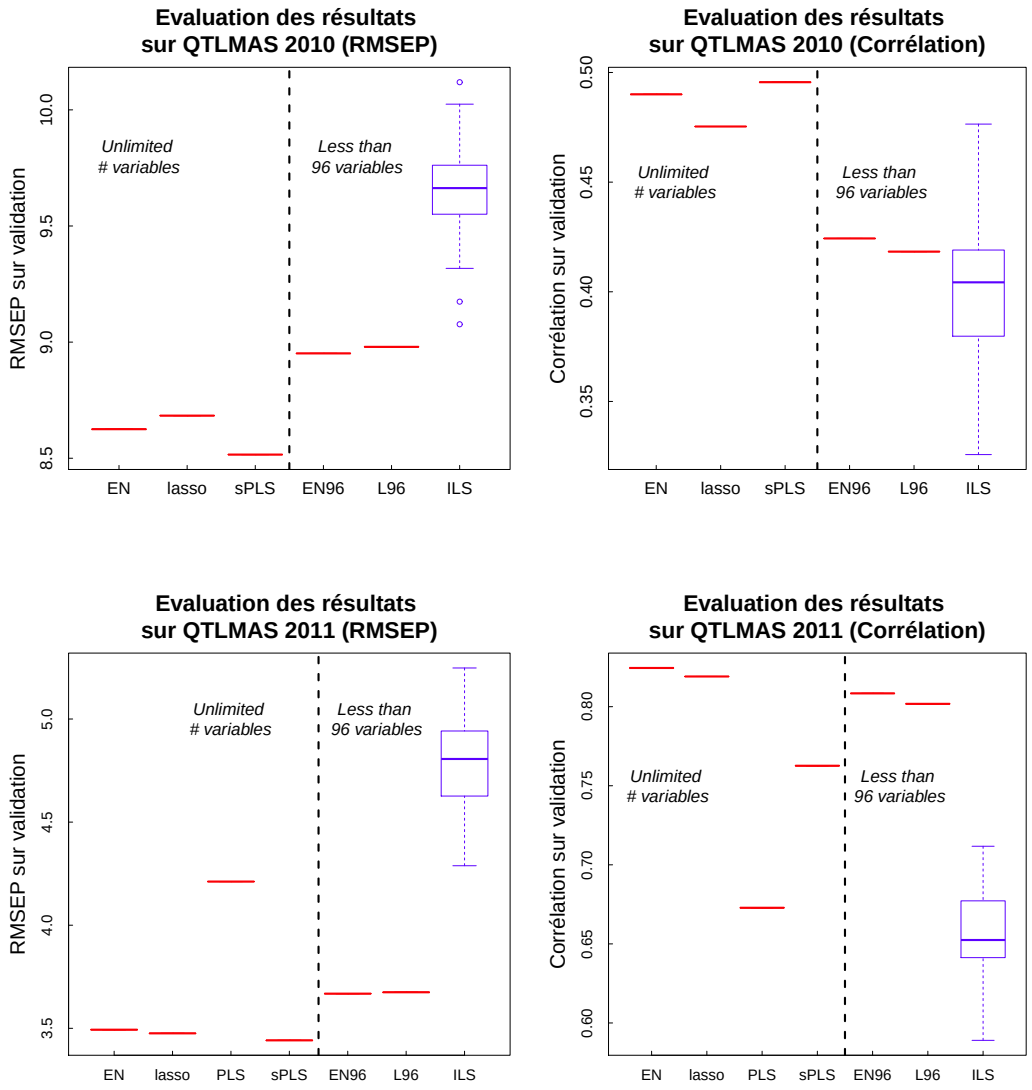

FIGURE 3.20 – Comparaison de l'algorithme avec des méthodes classiques sur données pseudo-réelles

Les résultats obtenus sur données pseudo-réelles (Figure 3.20) sont un peu plus mitigés. En effet, sur données pseudo-réelles, en terme d'erreur de prédiction, nous ne parvenons pas à égaler les méthodes classiques. En revanche, en terme de corrélation, sur QTLMAS 2010, nous obtenons de meilleurs résultats, sur certaines exécutions, que les méthodes classiques limitées à 96 variables.
La Table 3.6 rappelle le nombre de variables sélectionnées par chaque approche sur les deux jeux de données pseudo-réelles.

		EN	Lasso	sPLS	EN96	L96	ILS
# var. select.	QTLMAS 2010	297	191	3797	88	70	95,8
	QTLMAS 2011	36	33	79	55	41	70,7

TABLE 3.6 – Variables sélectionnées sur données pseudo-réelles

Sur les données de QTLMAS 2010, nous constatons que les approches elastic net, lasso et sparse PLS conservent un grand nombre de variables ce qui peut expliquer leurs bonnes performances en terme de prédiction.
En revanche, sur les données de QTLMAS 2011, les méthodes classiques sélectionnent peu de variables et génèrent de meilleurs résultats. En effet, sur ce jeu de données il n'y a que 8 QTL, le nombre de variables significatives est donc faible donc notre approche sélectionne des faux positifs. Le modèle généré par notre approche semble trop spécifique aux données d'apprentissage et devient donc difficilement généralisable à de nouvelles données.

3.5 Conclusion

Nous avons proposé dans ce chapitre d'aborder le problème de sélection de variables pour régression en grande dimension par une métaheuristique à solution unique : la recherche locale itérée (ILS). Nous avons étudié différents opérateurs et analysé leur pertinence par rapport à la problématique sous étude. Ceci nous a permis de proposer un modèle de l'algorithme ILS pour la recherche d'un sous-ensemble de variables pertinentes. Cette méthode a ensuite été comparée à des méthodes classiques sur données simulées et pseudo-réelles. Sur données simulées, nous obtenons de meilleurs résultats que certaines méthodes classiques ce qui est très encourageant. Cependant, sur données pseudo-réelles notre algorithme semble moins performant, tout en restant très prometteur. Cette étude nous a permis de valider l'approche par optimisation combinatoire et nous proposons d'améliorer la recherche en se basant sur une métaheuristique à population de solutions : l'algorithme génétique.

Chapitre 4

Sélection de variables en régression par algorithme génétique

Sommaire

Nous avons proposé dans le chapitre précédent d'aborder le problème de sélection de variables en régression par une approche d'optimisation combinatoire et nous avons obtenu des résultats prometteurs. En effet, les performances de notre approche basée sur une recherche locale itérée (ILS) sont du même ordre bien que sensiblement moins bonnes que les meilleures méthodes de la littérature. De plus, cette approche est facilement paramétrable pour obtenir un nombre maximal souhaité de variables sélectionnées. Afin d'améliorer les performances de l'approche basée sur la recherche locale itérée, nous proposons de mettre en œuvre une méthode d'optimisation plus sophistiquée.

L'algorithme génétique (AG) ayant prouvé son efficacité dans de nombreuses études (cf. Chapitre 2), nous proposons de l'utiliser pour résoudre notre problème de sélection de variables en régression.

Une partie de ce travail a fait l'objet d'une publication lors des 45ème journées de statistiques [Hamon 2013b].

Dans ce chapitre, nous commençons par exposer le design expérimental que nous

avons suivi. Nous présentons ensuite la méthode d'optimisation combinatoire que nous utilisons (algorithme génétique), et enfin nous évaluons les performances de notre approche et nous les comparons aux méthodes classiques de la littérature.

4.1 Design expérimental

Dans ce chapitre, comme dans le précédent, nous étudions les performances de notre approche basée sur un algorithme génétique sur deux types de données simulées (1 et 2) et deux types de données pseudo-réelles (QTLMAS 2010 et QTLMAS 2011). Pour plus de détails sur les jeux de données voir section 1.4.2. Pour les données simulées nous simulons 30 jeux de données par scénario de simulation, sur lesquels l'algorithme est exécuté une fois par jeu. En revanche, sur les données pseudo-réelles nous avons un seul jeu de données de chaque type, nous exécutons donc l'algorithme 30 fois (cf. section 3.1.2).
De même que dans le chapitre précédent, les résultats des expérimentations (sauf expérimentations particulières), sont rapportés dans l'ordre suivant : études en terme d'erreur de prédiction (graphiques de gauche) et de corrélation (graphiques de droite) sur les données simulées, puis sur les données pseudo-réelles de QTLMAS. Nous comparons dans un premier temps différentes configurations de l'algorithme d'optimisation combinatoire utilisé. Une fois la meilleure configuration trouvée nous évaluons ses performances, sur données simulées et pseudo-réelles, comparées à celles de méthodes classiques de la littérature : elastic net (EN), lasso, sparse PLS (sPLS), ainsi qu'aux méthodes EN et lasso limitées, comme notre approche, à 96 variables (notées EN96 et L96).
Le design expérimental étant similaire à celui du chapitre précédent, pour plus de détails voir la section 3.1.2.

4.2 Approche par algorithme génétique

Le principe général de l'algorithme génétique est décrit dans le Chapitre 2. Nous reprenons sur la Figure 4.1 les différentes étapes de cet algorithme, ce qui introduit également le plan de cette section.

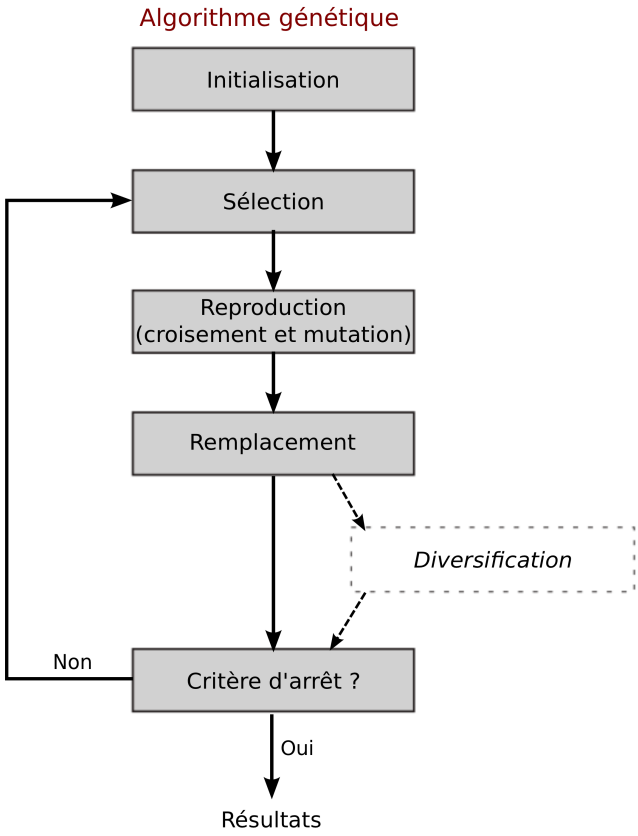

FIGURE 4.1 – Les différentes étapes de l'algorithme génétique

Cet algorithme fait évoluer une population de solutions, une solution étant un sous ensemble de variables, en appliquant des opérateurs de sélection, de croisement et mutation, de remplacement et éventuellement de diversification. Comme pour la recherche locale itérée présentée dans le chapitre précédent, nous définissons un algorithme génétique sur lequel nous allons nous appuyer avant d'étudier des alternatives et éventuellement le faire évoluer en conséquence. L'algorithme génétique sur lequel nous nous basons est défini à l'aide des opérateurs suivants :

- initialisation : uniforme,
- sélection : tournoi,
- reproduction :
  - croisement : opérateur SSOCF [Emmanouilidis 2000], détaillé en section 4.2.3, avec un taux de croisement de 0.8,
  - mutation : flip d'un pourcentage fixé de bits (faible, dépendant du nombre total de variables étudiées), avec un taux de mutation de 0.2,
- évaluation d'une solution : modèle de régression sur lequel le critère BIC est calculé et devient la fitness (à minimiser) de la solution,
- critère d'arrêt : nombre maximal de générations.

De même que pour l'algorithme ILS, nous choisissons une représentation binaire de sorte qu'une solution soit une chaîne de p bits (p étant le nombre total de variables) dont le bit j vaut 1 si la variable est sélectionnée, 0 sinon. Exemple de solution :

1	0	0	1	1	0	1	0

Cette solution représente le sous-ensemble constitué des variables 1, 4, 5 et 7. Notons que toutes les solutions (aussi appelées *chromosomes* dans l'algorithme génétique) ont la même taille.

4.2.1 Initialisation

L'initialisation classique des solutions d'un algorithme génétique se fait uniformément. La représentation d'une solution étant un vecteur binaire, il s'agit ici de positionner chacun des bits à 0 ou à 1. Nous souhaitons que le nombre k de variables sélectionnées (nombre de 1) dans chaque solution de la population initiale soit différent, tout en restant inférieur au nombre maximal autorisé (96 variables ici). Pour cela, pour chaque solution nous tirons uniformément ce nombre k de variables dans un intervalle $[min, max]$ fixé.
Pour accélérer la convergence de l'algorithme, nous proposons de guider le choix des variables constituant les solutions initiales. Pour cela, nous envisageons trois configurations :

- la première consiste à choisir de manière uniforme les variables qui feront partie de chaque solution de la population initiale.
- la deuxième configuration consiste à initialiser toutes les solutions de la population initiale à l'aide des variables sélectionnées par la méthode lasso. La méthode lasso (non limitée en nombre de variables sélectionnées) nous permet d'obtenir un sous-ensemble de variables *a priori* intéressantes. Pour chaque solution (individu) de notre population initiale, nous choisissons de manière uniforme le nombre de variables qu'elle va comporter, puis nous choisissons ces variables parmi celles obtenues par la méthode lasso. Si le nombre de variables souhaitées pour la solution est supérieur au nombre de variables extraites par la méthode lasso, nous choisissons toutes celles de la méthode lasso et rajoutons des variables sélectionnées uniformément parmi les autres.
- la troisième configuration consiste à mixer les deux précédentes. Pour cela, nous séparons la population initiale en deux, pour la moitié des solutions de la population initiale, nous choisissons uniformément le nombre de variables à intégrer dans chaque solution, puis nous sélectionnons ces variables parmi celles sélectionnées par la méthode lasso. Pour la seconde moitié des solutions de la population initiale, nous choisissons uniformément les variables parmi l'ensemble de toutes les variables. Cette solution permet de conserver de la diversité.

Les figures 4.2 et 4.3 comparent, sur données simulées et pseudo-réelles, les trois initialisations envisagées : uniforme, à partir de lasso pour toutes les solutions de la

population (Lasso 100%) et à partir de lasso pour 50% des solutions de la population (Lasso 50%).

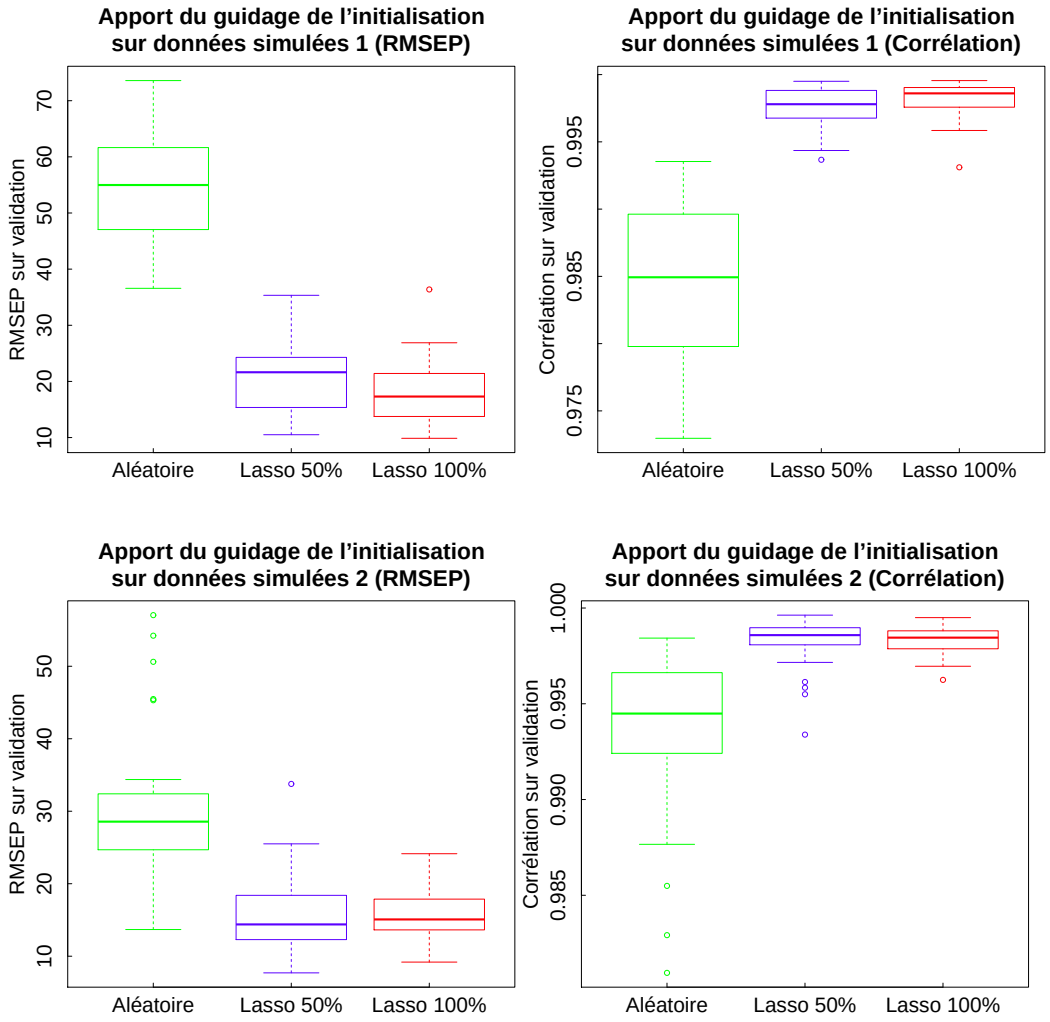

FIGURE 4.2 – Comparaison des différents processus d'initialisation sur données simulées

Sur la Figure 4.2, nous pouvons voir que, sur données simulées, lorsque l'initialisation de l'algorithme génétique est guidée par les variables sélectionnées par la méthode lasso, les résultats obtenus sur l'échantillon de validation sont meilleurs, que ce soit en terme d'erreur de prédiction ou de corrélation, que ceux obtenus par une initialisation purement uniforme. En revanche, il ne semble pas y avoir de différence significative entre une initialisation basée sur la méthode lasso pour toutes les solutions de la population initiale ou pour uniquement 50% des solutions.

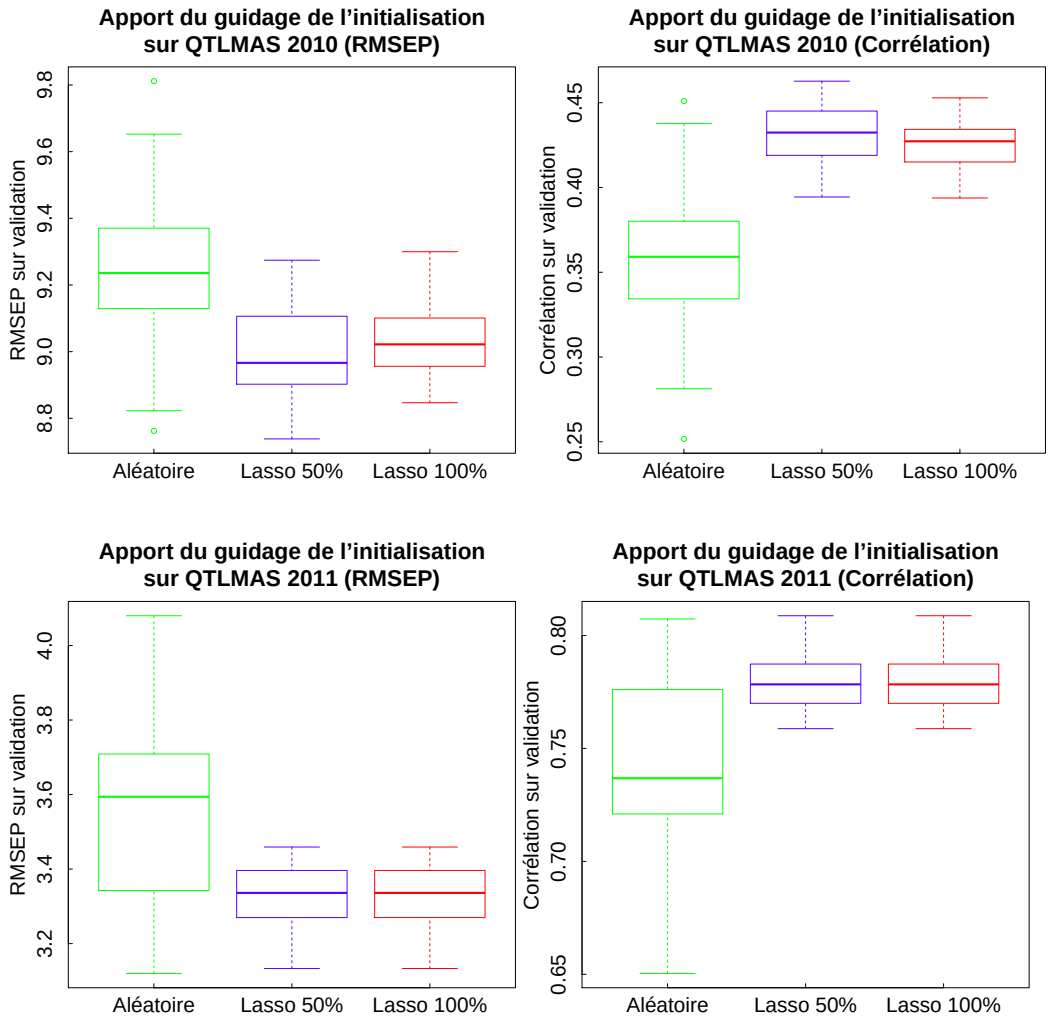

FIGURE 4.3 – Comparaison des différents processus d'initialisation sur données pseudo-réelles

Les résultats obtenus sur données pseudo-réelles (Figure 4.3) nous permettent de confirmer, pour l'initialisation de la population, ce que nous avons observé sur données simulées.

Conclusion AG 1 **: Nous choisissons donc d'initialiser 50% des solutions à l'aide de la méthode lasso et les 50% restantes uniformément afin de conserver de la diversité et pouvoir éventuellement s'adapter à d'autres types de jeux de données.**

4.2.2 Sélection

Le processus de sélection de l'algorithme génétique permet de déterminer les individus qui vont se reproduire et combien d'enfants chaque couple va générer. Ceci est équivalent à déterminer les sous-ensembles de variables qui vont servir à la création de nouveaux sous-ensembles. Le principe général repose sur l'idée que plus un individu est bon (en terme de fitness), plus il a de chance de devenir parent. Plusieurs stratégies de sélection sont possibles comme la sélection par roulette, l'échantillonnage stochastique universel ou encore la sélection par tournoi [Talbi 2009].

La sélection par roulette

La sélection par roulette (*roulette wheel selection*) affecte à chaque individu une probabilité de sélection proportionnelle à sa fitness relative ($f_s / \sum_l f_l$). La roulette peut être représentée comme un gâteau attribuant à chaque individu une part proportionnelle à sa fitness. Un tour de roulette va permettre de sélectionner un individu. Les individus ayant les plus grosses parts du gâteau ont plus de chance d'être sélectionnés. La sélection de k individus entraîne k tours de roulette. Ceci est équivalent à un tirage suivant une loi multinomiale $\mathcal{M}(k, f_1/\sum_l f_l, \ldots, f_s/\sum_l f_l)$ L'utilisation de cette méthode de sélection peut conduire à une convergence prématurée en présence d'individus extrêmes ayant de très fortes fitness relatives et pouvant être sélectionnées très souvent.

L'échantillonnage stochastique universel

L'échantillonnage stochastique universel fonctionne sur le même principe que la roulette mais permet de sélectionner le nombre d'individus souhaités en 1 tour de roulette en fixant plusieurs pointeurs uniformément répartis. Ce qui revient à des tirages sans remise parmi la population de solutions selon les probabilités $(f_1/\sum_l f_l, \ldots, f_s/\sum_l f_l)$.

La sélection par tournoi

La sélection par tournoi consiste à sélectionner aléatoirement m individus, m étant la taille du groupe de tournoi. Le meilleur individu parmi les m est celui qui sera conservé (Figure 4.4 où f est la fitness). La sélection de k individus fait appel à k générations de tournoi.

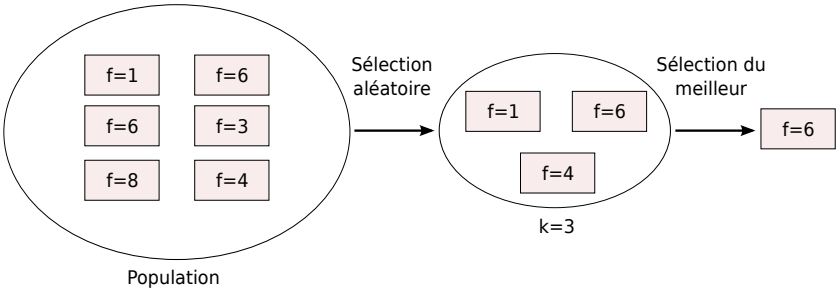

FIGURE 4.4 – Exemple de sélection par tournoi ($k = 3$)

Conclusion AG 2 **: La sélection par tournoi étant classiquement utilisée, nous la choisissons pour notre algorithme.**

4.2.3 Reproduction

Une fois la sélection des parents faite, la phase de reproduction applique des opérateurs de variation tel que le croisement et la mutation afin de générer les enfants.
Le choix de codage binaire des solutions que nous avons fait nous permet d'utiliser les opérateurs de croisement et mutation classiques de la littérature [Goldberg 1989]. Cependant, ces opérateurs ne sont pas toujours adaptés au problème spécifique sur lequel nous travaillons.

Croisement Le croisement (aussi appelé recombinaison) est un opérateur binaire (voire n-aire, c'est-à-dire avec plusieurs solution en entrée) dont l'objectif est que les enfants générés héritent de quelques (bonnes) caractéristiques de leurs parents. Les 3 types de croisement classiquement utilisés sont le croisement *1-point*, le croisement *2-points* et le croisement *uniforme*. La Figure 4.5 illustre le principe de ces trois croisements.

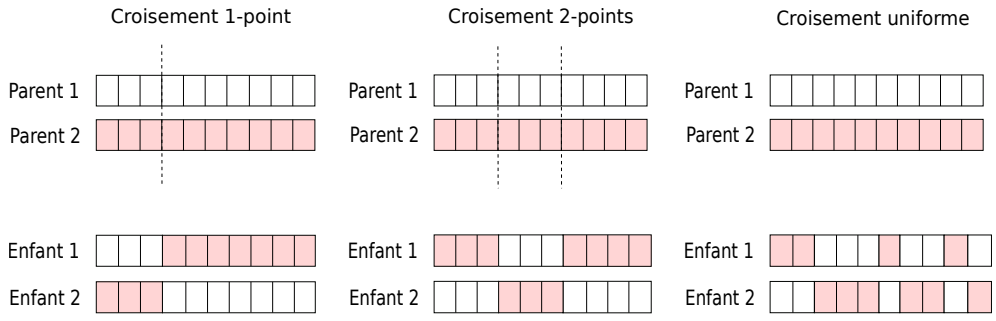

FIGURE 4.5 – Exemple de croisements 1-point, 2-points et uniforme

Le croisement 1-point définit un point de coupure aléatoirement et inverse la deuxième partie des deux parents. Le croisement 2-points (également généralisable à *n-points*), définit 2 points de coupure et inverse les parties des deux parents entre ces deux points. Enfin, le croisement uniforme sélectionne chaque élément d'un enfant uniformément parmi ceux des parents.
Le choix de l'opérateur de croisement va dépendre du problème étudié. En effet, dans le cadre de la sélection de variables, ces opérateurs classiques peuvent avoir un effet néfaste puisqu'ils peuvent "casser" des blocs intéressants. Le croisement 1-point par exemple aura tendance à ne pas générer d'enfant ayant à la fois le premier et le dernier élément d'un parent alors que ces deux éléments ensemble peuvent être intéressants. Nous choisissons donc d'utiliser ici un opérateur de croisement adapté au problème de sélection d'attributs, le *Subset Size-Oriented Common Feature* (SSOCF [Emmanouilidis 2000]). Le principe est décrit sur un exemple par la Figure 4.6.

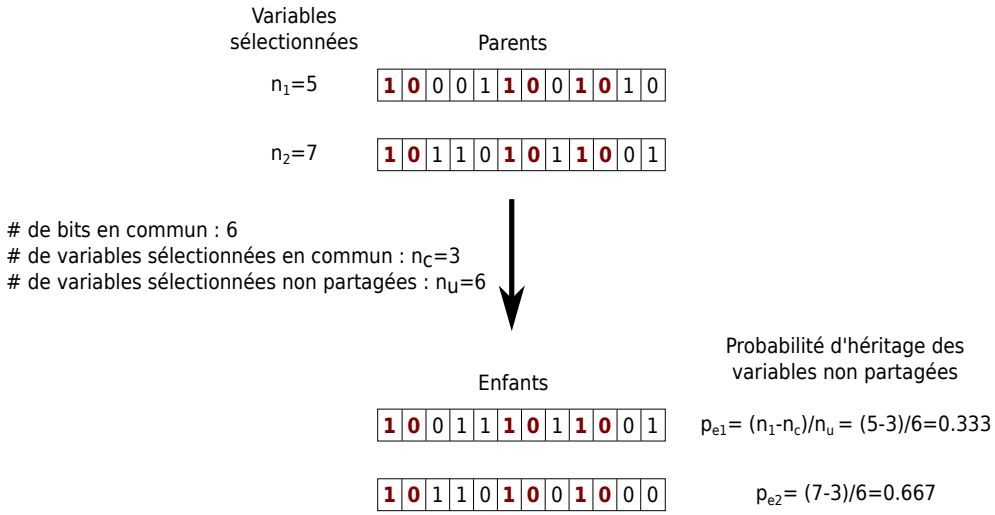

FIGURE 4.6 – Exemple de croisement SSOCF

Les variables communes aux deux parents sont conservées par les enfants. Les autres sont héritées du $i^{\text{ème}}$ parent avec la probabilité $(n_i - n_c)/n_u$ où n_i est le nombre de variables sélectionnées par le $i^{\text{ème}}$ parent, n_c est le nombre de variables sélectionnées en commun entre les deux parents et n_u le nombre de variables non partagées par les deux parents (variables sélectionnées par l'un ou l'autre des parents mais pas par les deux). L'objectif de cette méthode est d'une part de conserver les blocs d'informations utiles, et d'autre part que les enfants gardent les variables partagées par leurs parents.
Afin de confirmer la pertinence de l'utilisation du croisement SSOCF comparé au croisement 1-point nous avons comparé leurs performances sur les données de QTL-MAS 2011 (Figure 4.7).

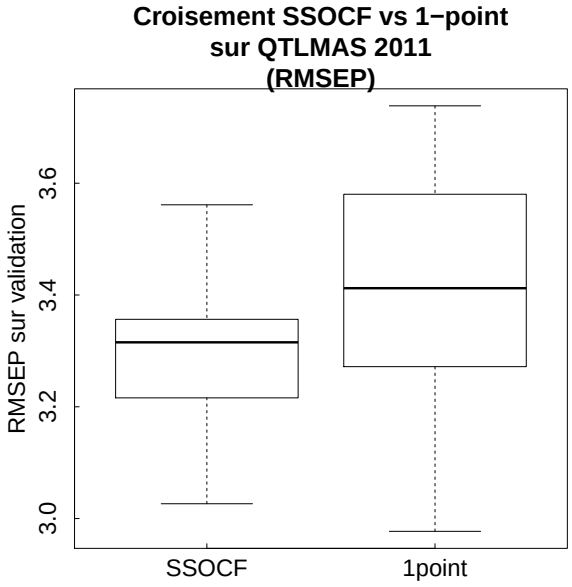

FIGURE 4.7 – Croisement SSOCF vs croisement 1-point

Conclusion AG 3 **: Les résultats obtenus avec le croisement SSOCF sont significativement meilleurs que ceux obtenus avec le croisement 1-point sur ces données.**

Mutation La mutation est un opérateur unaire (une seule solution en entrée) appliqué à un individu pour le modifier faiblement. Lors d'une représentation binaire des solutions, ce qui est notre cas, la mutation classiquement utilisée est le flip d'un bit. Deux types de mutations sont utilisés dans notre algorithme en fonction du nombre de variables sélectionnées dans la solution courante :

- le flip d'un pourcentage (faible) fixé de bits déterminés uniformément parmi l'ensemble des variables lorsque le nombre de variables sélectionnées dans la solution courante est inférieur au nombre maximal souhaité de variables ($\Rightarrow$ ajout ou suppression de variables).
- le flip d'un pourcentage (faible) fixé de bits déterminés uniformément parmi les variables sélectionnées (bit=1) lorsque le nombre maximal souhaité de variables est atteint ($\Rightarrow$ suppression de variables).

Lors de l'étape de reproduction, les opérateurs de croisement et de mutation ne sont pas appliqués systématiquement. En effet, le taux de croisement permet de définir la probabilité que deux parents sélectionnés soient croisés pour générer des enfants. De même, le taux de mutation représente la probabilité d'appliquer la mutation à une solution. Nous étudions donc ici l'influence de l'utilisation de taux

de croisement et mutation faibles (0.2) ou élevés (0.8).

Les figures 4.8 et 4.9 comparent, sur données simulées et pseudo-réelles les différentes combinaisons de taux de croisement (cross) et mutation (mut) faibles (0.2) et élevés (0.8).

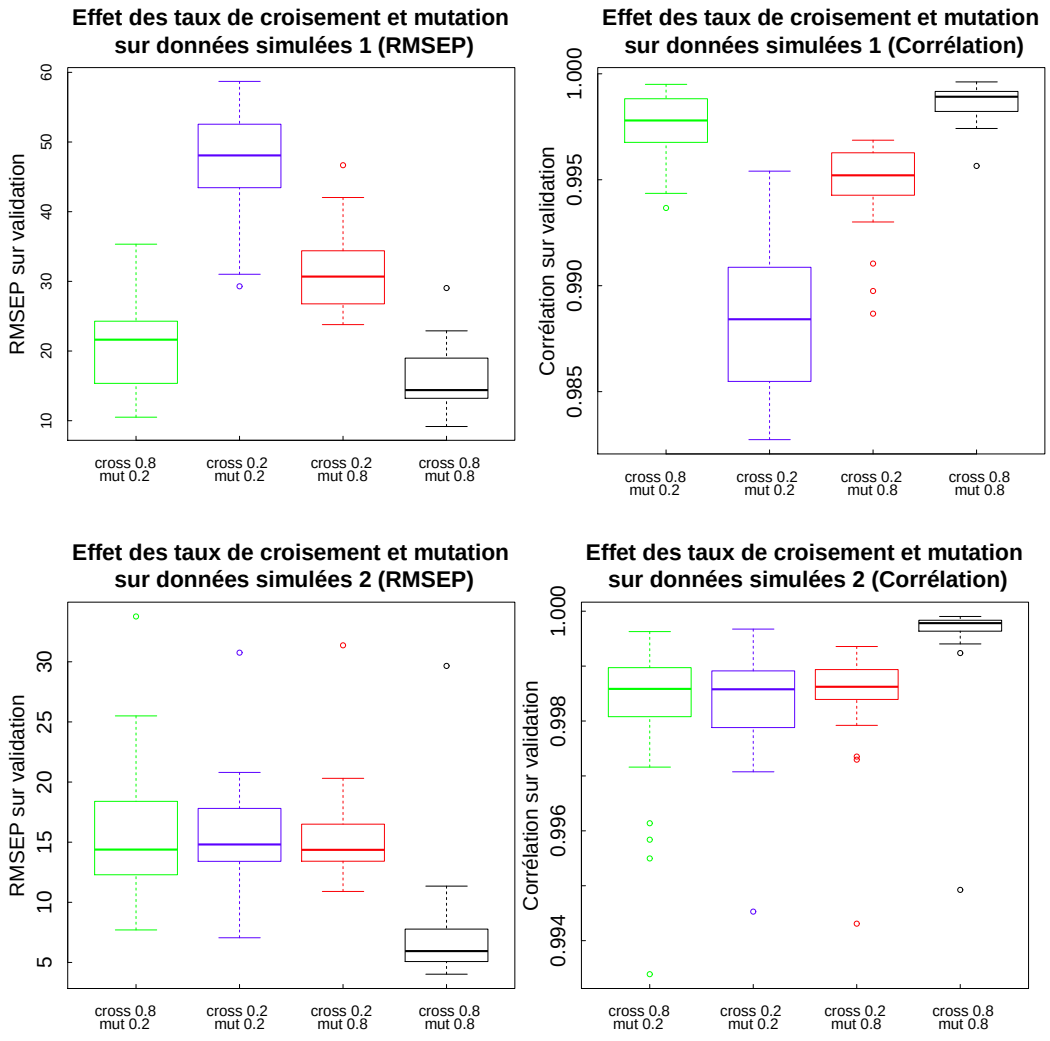

FIGURE 4.8 – Comparaison des différents taux de croisement et mutation sur données simulées

La Figure 4.8 nous montre que, sur les deux types de données simulées (1 et 2), la combinaison des taux de mutation et croisement élevés (0.8) permet d'obtenir les meilleurs résultats.

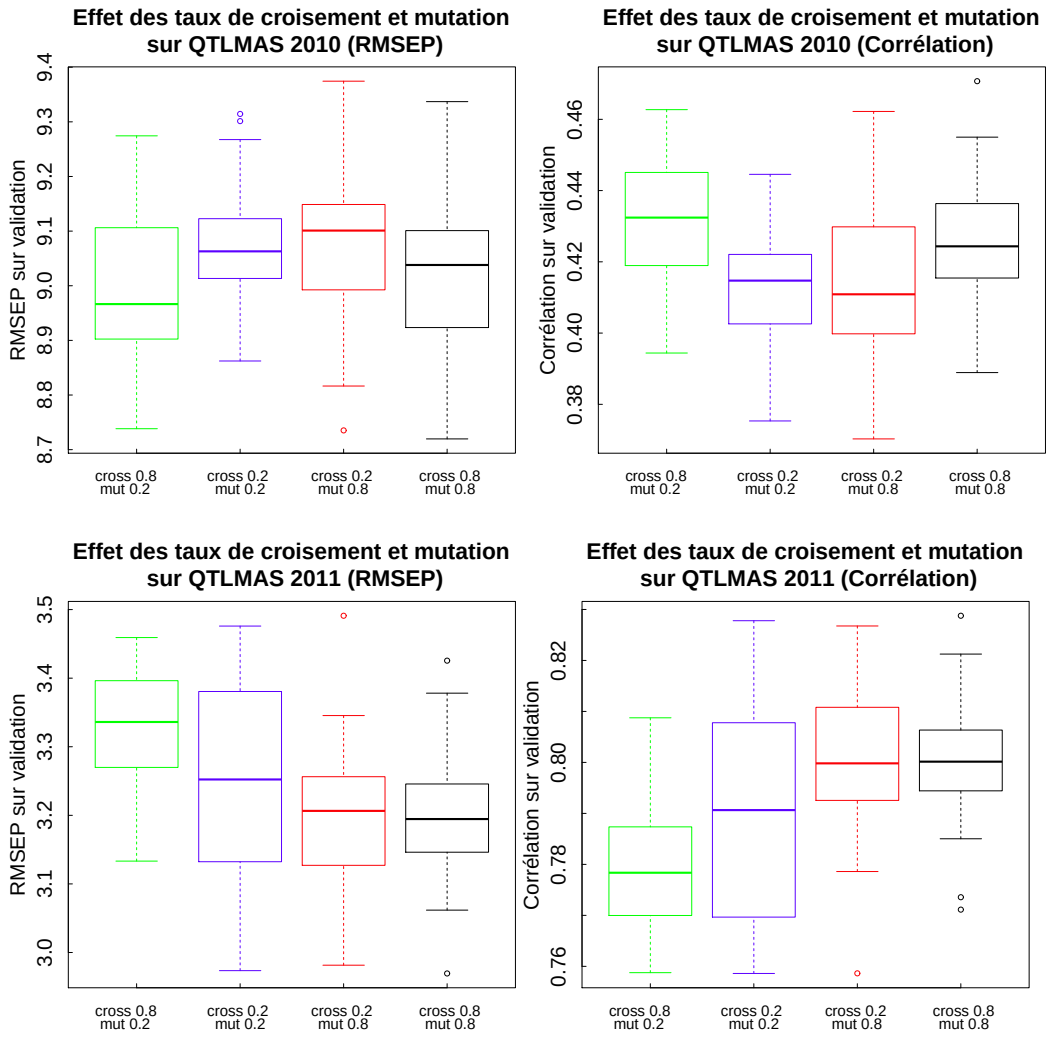

FIGURE 4.9 – Comparaison des différents taux de croisement et mutation sur données pseudo-réelles

Lorsque nous étudions les résultats sur données pseudo-réelles (Figure 4.9), il n'y a pas de différence significative entre l'utilisation des différentes combinaisons de taux.

Conclusion AG 4 **: Nous choisissons donc d'utiliser pour la suite un taux de mutation de 0.8 et un taux de croisement de 0.8, pour lesquels nous obtenons des résultats significativement meilleurs sur données simulées.**

4.2.4 Remplacement

La taille de la population étant constante au cours des générations, lorsque les enfants sont générés, tous les parents et enfants ne peuvent pas être conservés. La procédure de remplacement, dernière étape d'une génération, va permettre de définir les survivants parmi les parents et les enfants générés. Plusieurs procédures de remplacement ont été proposées comme le remplacement générationnel qui remplace systématiquement tous les parents par tous les enfants, ou l'élitisme qui garde uniquement les meilleurs parmi les parents et les enfants. L'élitisme permet une convergence plus rapide mais parfois prématurée. Il est parfois intéressant de conserver certains mauvais individus.
La procédure de remplacement que nous choisissons ici consiste à conserver un enfant uniquement s'il est meilleur que le moins bon des parents restants. Lorsqu'un enfant est conservé, le moins bon des parents est supprimé. Les parents les plus mauvais sont donc remplacés au fur et à mesure par des enfants meilleurs qu'eux. Cette procédure, contrairement à l'élitisme, va permettre de conserver certaines mauvaises solutions. La Figure 4.10 illustre la différence entre le remplacement élitiste et la procédure de remplacement que nous avons choisie.

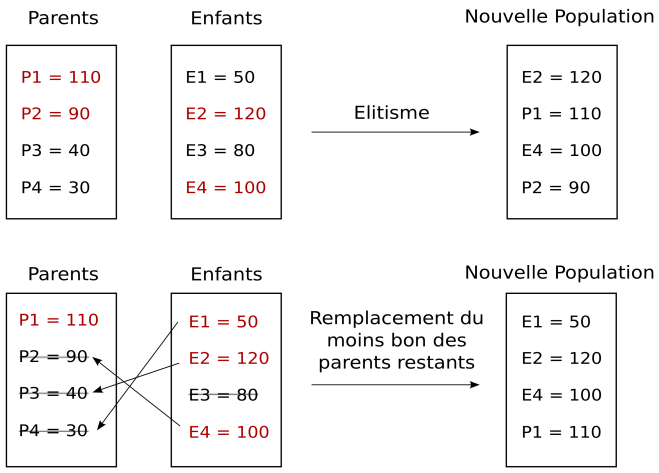

FIGURE 4.10 – Comparaison de l'élitisme avec la méthode de remplacement choisie

L'élitisme conserve les meilleures solutions parmi les parents et les enfants. En revanche, la procédure que nous avons choisi parcours les enfants un à un et les conserve uniquement s'ils sont meilleurs que le moins bon des parents restants. Sur la Figure 4.10 la solution E1 est meilleure que la solution P4 (qui est la moins bonne des parents), nous conservons donc la solution E1 et supprimons la solution P4. Nous comparons ensuite la solution E2 à la solution P3, nous conservons E2 et supprimons P3 et ainsi de suite. Dans la population finale nous avons donc conservé certaines solutions qui sont moins bonnes que d'autres que nous avons supprimées.

4.2.5 Critère d'arrêt

Comme la recherche locale itérée, l'algorithme génétique est un algorithme itératif pour lequel il est nécessaire de fixer un critère d'arrêt. Nous fixons ici un nombre de générations, déterminé empiriquement en fonction de la courbe d'évolution de la meilleure solution de la population. La Figure 4.11 représente, sur les différents jeux de données, pour 5 exécutions de l'algorithme, l'évolution de la meilleure solution de la population au cours des générations. La population initiale étant composée de 300 solutions.

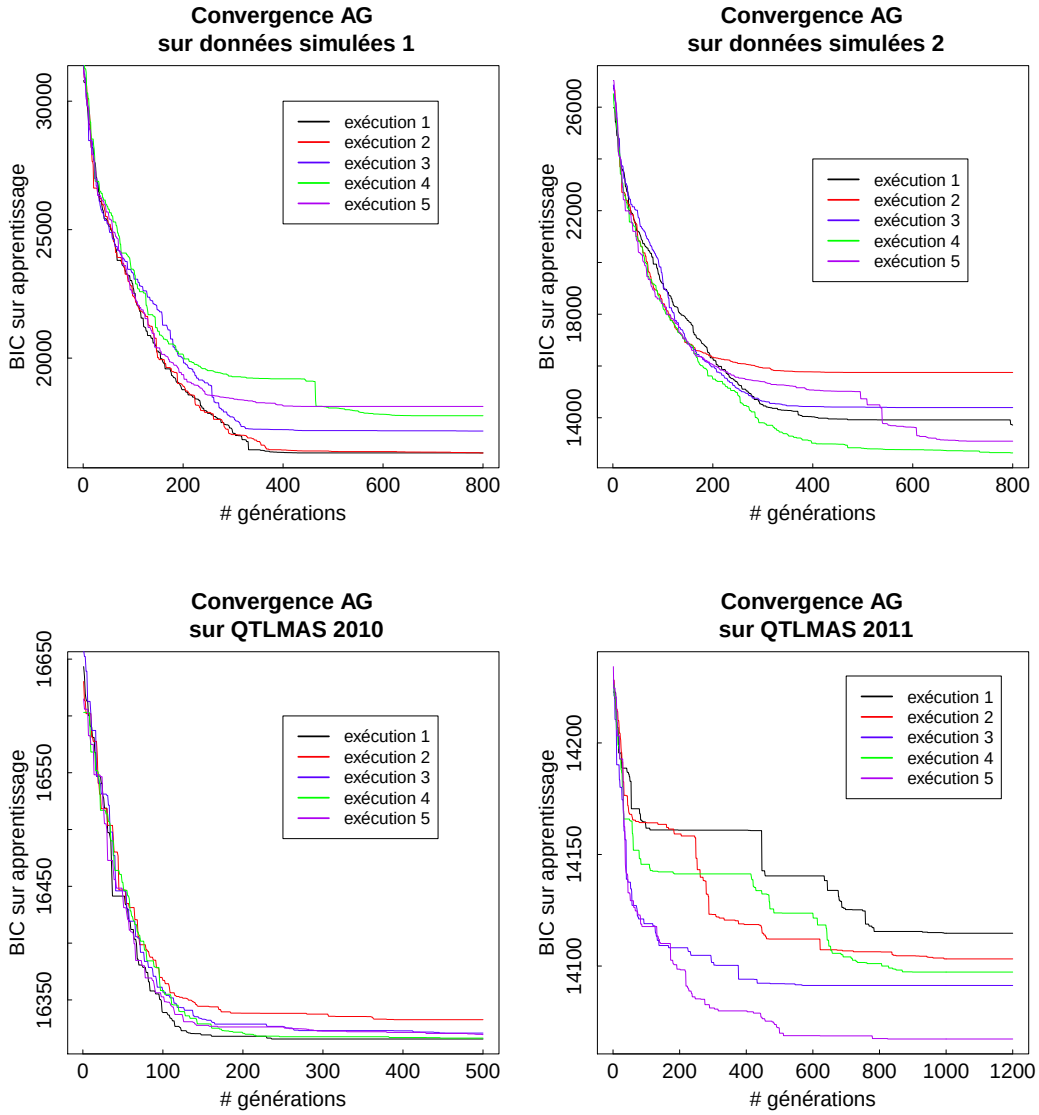

FIGURE 4.11 – Convergence de l'algorithme génétique

Sur les données simulées 1 et 2, 800 générations semblent permettre à l'algorithme de converger. En revanche, sur les données de QTLMAS 2010, 500 générations suffisent alors que pour QTLMAS 2011 il en faut 1 200. Le nombre de générations nécessaires à la convergence de l'algorithme est donc fortement dépendant du jeu de données étudié. Plusieurs expérimentations sont donc nécessaires avant de définir le bon nombre de générations.

4.2.6 Diversification

Lors de l'évolution de l'algorithme génétique, un défaut qui peut être observé est la stagnation de la recherche (ceci peut être observé sur les courbes de convergence précédentes, notamment celles de QTLMAS 2011, avec pour l'exécution 1 un palier entre les générations 200 et 400). Pour éviter cela, des méthodes de diversification sont proposées comme la migration de diversité stochastique ou "*Random Immigrant*" [Grefenstette 1992]. Le principe consiste à remplacer une partie de la population par des individus générés uniformément lorsque le meilleur individu de la population ne s'est pas amélioré pendant un nombre donné d'itérations.
Dans notre algorithme, lorsque le meilleur individu de la population n'évolue pas pendant un nombre fixé de générations, tous les individus dont la fitness est inférieure à la fitness moyenne de la population sont remplacés par de nouveaux individus générés uniformément.
Nous étudierons ici l'intérêt du processus de diversification en comparant la qualité des résultats de l'algorithme intégrant ou non ce processus.

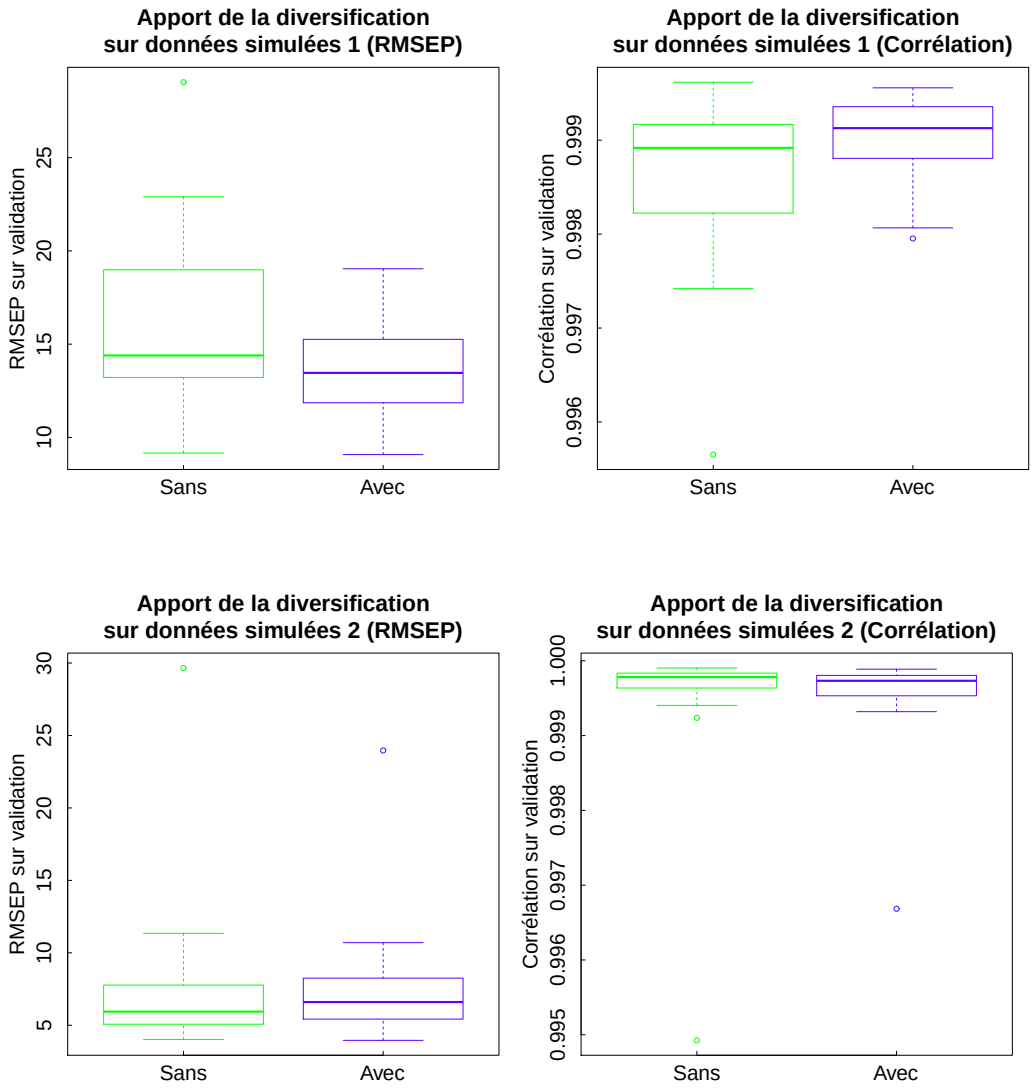

FIGURE 4.12 – Apport du processus de random immigrants sur données simulées

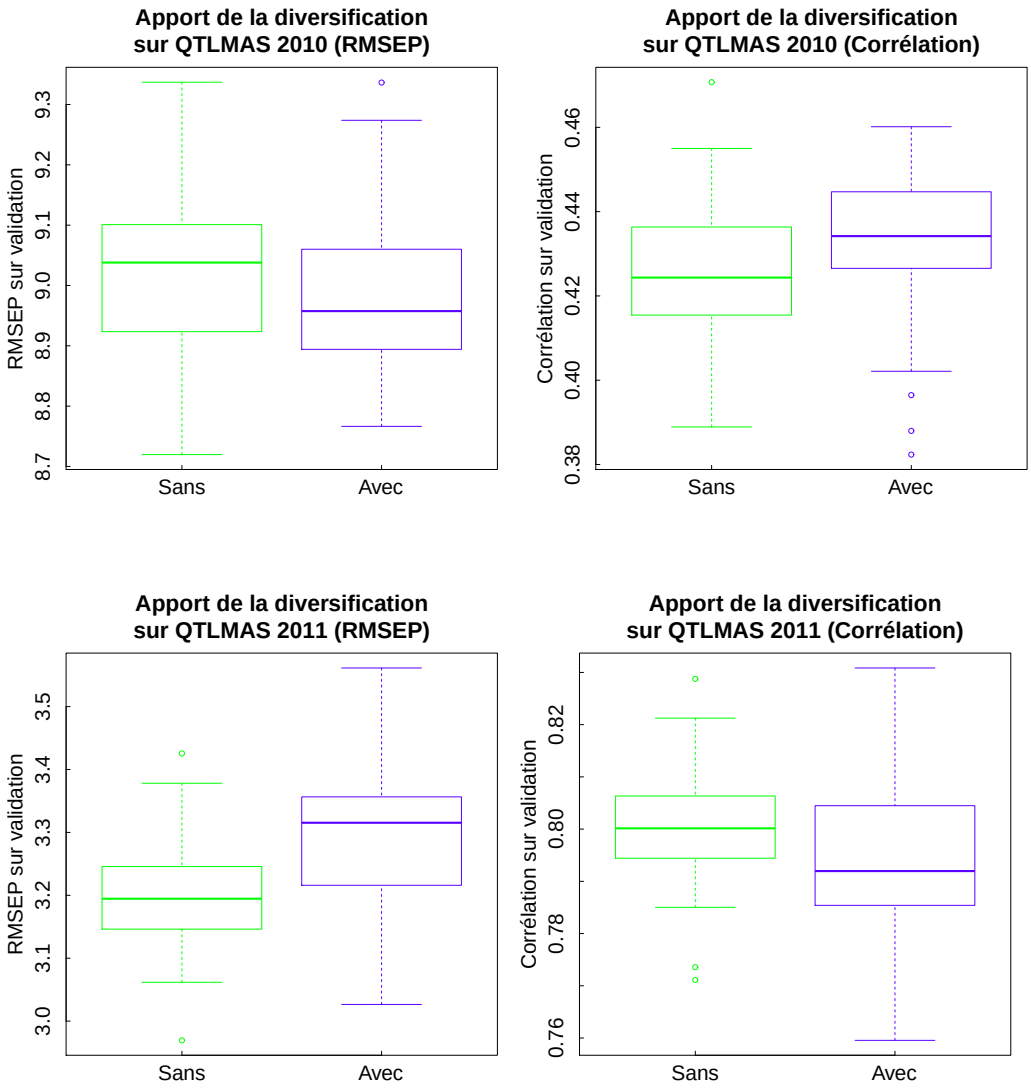

FIGURE 4.13 – Apport du processus de diversification sur données pseudo-réelles

Sur données simulées (Figure 4.12), comme sur données pseudo-réelles (Figure 4.13), il ne semble pas y avoir de différence entre l'utilisation ou non du processus de diversification en terme d'erreur de prédiction sur l'échantillon de validation. Afin de mieux comprendre ce phénomène, nous étudions l'évolution du critère d'optimisation (BIC) sur l'échantillon d'apprentissage (Figure 4.14). En effet, cela permet d'étudier la convergence, et de voir si la diversification permet à l'algorithme de continuer à converger et atteindre des solutions de meilleure qualité.

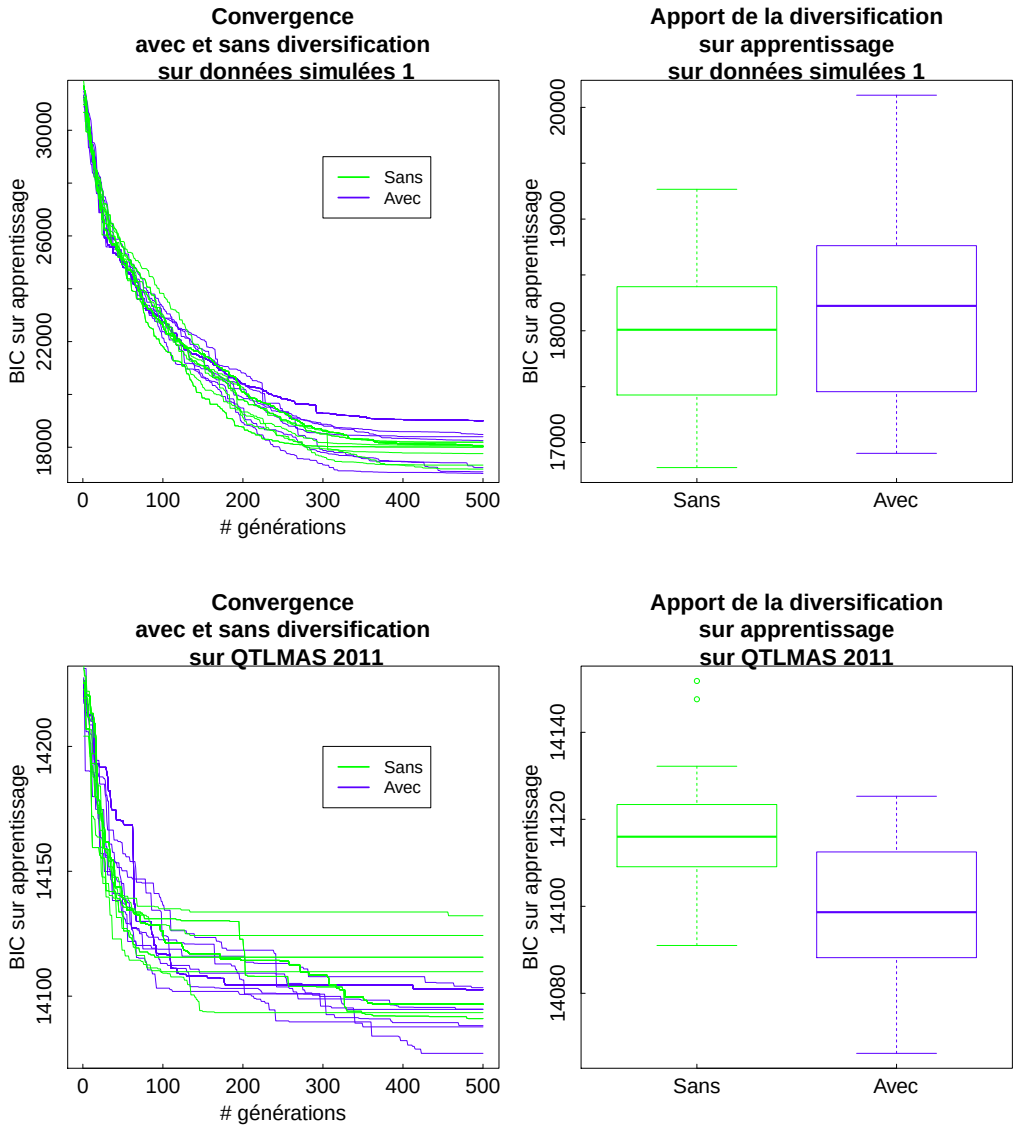

FIGURE 4.14 – Apport du processus de diversification sur données pseudo-réelles

Les graphiques de gauche représentent la convergence de l'algorithme et permettent de constater que sur les données simulées, il n'y a que très peu de plateaux et donc peu d'appels au processus de diversification. Ceci peut justifier le fait qu'il n'y ait pas de différence sur la qualité finale des solutions, même sur l'échantillon d'apprentissage (graphiques de la partie droite de la Figure 4.14). Sur les données pseudo-réelles (QTLMAS 2011), le processus de diversification a permis de continuer à converger en explorant plus largement l'espace de recherche lorsque des plateaux sont atteints, les résultats sont donc meilleurs sur l'échantillon d'apprentissage.

Bien que le processus de diversification ne permettent pas d'améliorer nos performances sur l'échantillon de validation, il ne dégrade pas non plus nos résultats et est très peu coûteux en temps de calcul.

Conclusion AG 5 **: Son efficacité ayant été montrée dans d'autres circonstances [Tinos 2007], nous décidons de conserver le processus de diversification. Ceci permettra éventuellement d'améliorer les résultats sur d'autre types de données.**

4.2.7 Parallélisation

Au cours de l'algorithme génétique, lors d'une génération plusieurs solutions candidates doivent être évaluées ce qui peut être fait parallèlement. Ainsi, afin de diminuer le temps d'exécution de l'algorithme, nous avons implémenté une version parallèle synchrone de l'algorithme à l'aide du module SMP de PARADISEO. L'objectif est de paralléliser, à chaque génération, les évaluations des enfants (solutions) de l'algorithme génétique grâce au schéma "maître/esclave". Une fois tous les enfants générés, leurs évaluations étant indépendantes elles sont effectuées en parallèle. Le principe est illustré par la Figure 4.15. Lors de la phase d'évaluation, le maître envoie une solution à évaluer par esclave, ces derniers lui retournent la fitness de la solution.

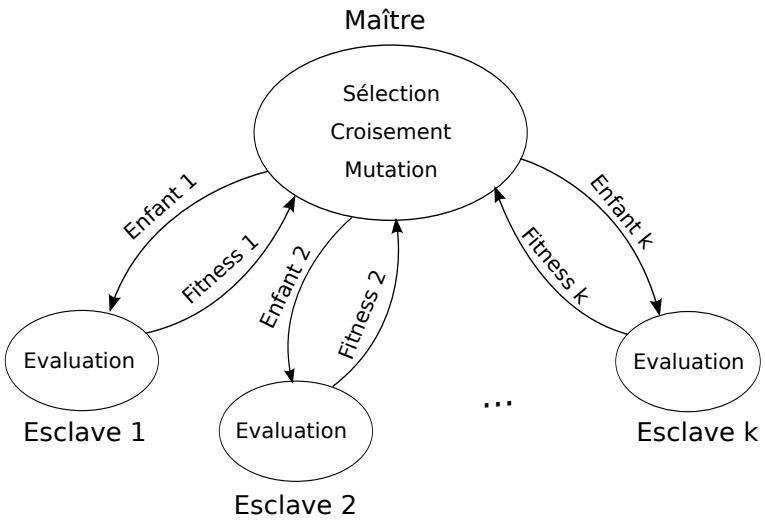

FIGURE 4.15 – Parallélisation maître/esclaves

Afin d'illustrer la pertinence de la parallélisation, nous comparons les temps d'exécution de l'algorithme, sans parallélisation puis en parallélisant sur 4, 8, 12 et 16 cœurs de calcul.

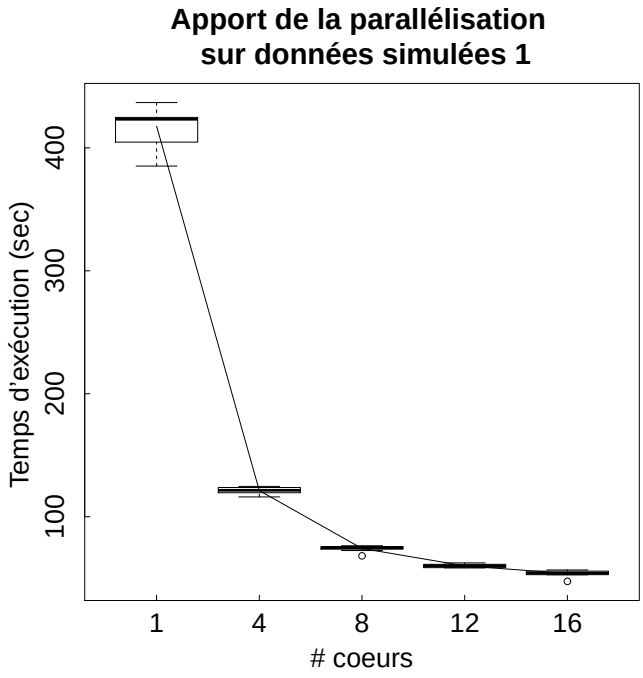

FIGURE 4.16 – Influence de la parallélisation

La Figure 4.16 montre la pertinence d'utiliser la parallélisation sur les données simulées 1 (les résultats sont similaires quelque soit le jeu de données). En effet, le passage de l'algorithme non parallélisé à l'algorithme parallélisé sur 4 cœurs permet de diviser par 3,4 le temps d'exécution de l'algorithme. Remarquons que le temps d'exécution n'est pas exactement divisé par le nombre de cœurs utilisés puisqu'il faut tenir compte des temps de communication entre les machines. Au delà de 8 cœurs, le gain de temps obtenus lors de l'augmentation du nombre de cœurs est moins important mais néanmoins significatif (coefficients de 5,6 entre 1 et 8 cœurs, 7 entre 1 et 12 cœurs et 7,8 entre 1 et 16 cœurs). Ceci est dû au fait qu'en augmentant le nombre de cœurs utilisés, nous augmentons également le temps accordés aux communications entre machines.

Conclusion AG 6 **: Nous choisissons d'utiliser la parallélisation sur 12 ou 16 cœurs suivant les ressources disponibles sur le cluster de calculs utilisé.**

4.3 Analyses expérimentales

Après avoir étudié différentes améliorations possibles de notre algorithme génétique de base, nous sélectionnons finalement la meilleure configuration adaptée à notre problème :

- initialisation : basée sur la méthode lasso pour 50% des solutions de la population initiale, uniforme pour les autres,
- sélection : tournoi,
- reproduction :
  - croisement : opérateur SSOCF [Emmanouilidis 2000], détaillé en section 4.2.3, avec un taux de croisement de 0.8,
  - mutation : flip d'un pourcentage fixé de bits (faible, dépendant du nombre total de variables étudiées), avec un taux de mutation de 0.8,
- diversification : *random immigrants*,
- évaluation d'une solution : modèle de régression sur lequel le critère BIC est calculé et devient la fitness (à minimiser) de la solution,
- critère d'arrêt : nombre maximal de générations, dépendant du jeu de données.

De même que dans le chapitre précédent, les expérimentations sont menées sur les jeux de données simulées 1 et 2 ainsi que sur les données pseudo-réelles de QTLMAS 2010 et 2011. Nous évaluons dans un premier temps la pertinence qualitative de notre approche en nous intéressant à la pertinence des variables qu'elle sélectionne, puis quantitative au travers de l'erreur de prédiction obtenue sur l'échantillon de validation et de la corrélation entre le caractère estimé et le caractère réel (sur l'échantillon de validation également).
Nous comparons nos résultats avec ceux des méthodes classiques de la littérature : elastic net (EN), lasso, sparse PLS (sPLS), EN96 et L96. Nous séparons également par une ligne verticale en pointillé noir les méthodes non limitées en nombre de variables (à gauche) des méthodes limitées à 96 variables (à droite).

4.3.1 Sélection de variables

Comme pour l'analyse de notre première méthode basée sur la recherche locale itérée, sur les données simulées nous connaissons les variables significatives et nous pouvons donc étudier les performances des différentes approches en terme de variables sélectionnées. La Figure 4.17, présente, à gauche sur les données simulées 1 et à droite sur les données simulées 2, le nombre de vrais positifs sélectionnés par chaque méthode. La ligne verte horizontale en pointillé correspond à l'objectif, qui est le nombre total réel de variables significatives (96).

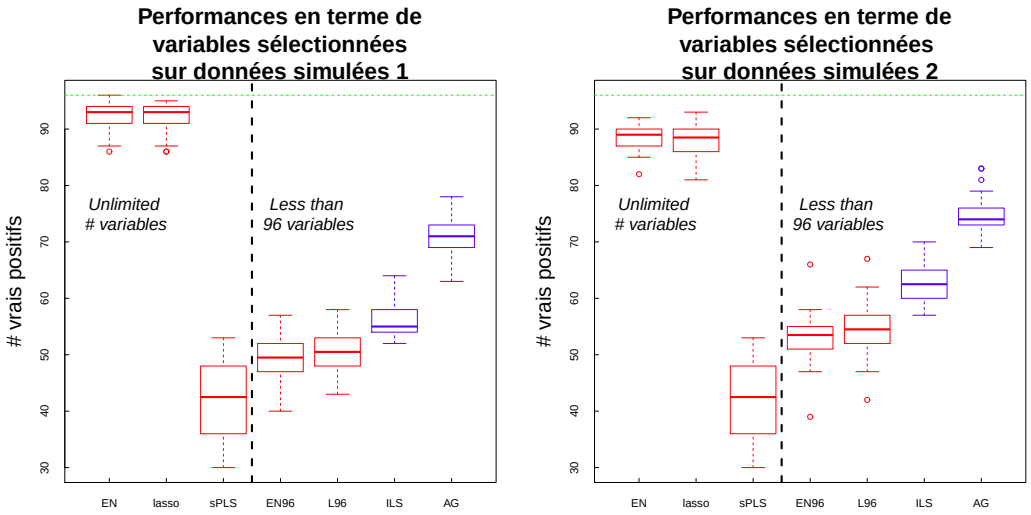

FIGURE 4.17 – Capacité à sélectionner les bonnes variables

Les graphiques de la figure 4.17 permettent de constater dans un premier temps que notre approche basée sur l'algorithme génétique permet d'obtenir des résultats nettement meilleurs que ceux obtenus avec notre première approche basée sur une recherche locale itérée. De plus, notre approche basée sur l'algorithme génétique semble être plus performante en terme de variables sélectionnées que les méthodes lasso et elastic net limitées à 96 variables. Les méthodes lasso et elastic net non limitées en nombre de variables retrouvent un plus grand nombre de variables mais en sélectionnent plus. Elles ont donc un taux de faux positifs plus important, comme nous pouvons le voir dans les tables 4.1 et 4.2. En effet, ces deux tableaux présentent le nombre total moyen de variables sélectionnées par chaque méthode, le pourcentage moyen de faux positifs (variables sélectionnées alors qu'elles ne sont pas significatives en réalité) ainsi que les pourcentages moyens de variables trouvées (variables significatives effectivement retrouvées parmi les 96) répartis en fonction des trois types d'effets (faibles, moyens et forts).

		EN	**Lasso**	**sPLS**	**EN96**	**L96**	**ILS**	**AG**
# var. select.		205,3	201,6	89,3	98,5	98,5	96	96
% faux positifs		56	55	62	50	48	50	**23**
% variables trouvées	*Effets faibles*	88,8	88,8	0,9	1,9	2,1	6,8	**40**
	Effets moyens	99,8	99,6	12,4	52,3	57,1	53,1	**93,1**
	Effets forts	100	99,9	93,8	**98,7**	**98,7**	91,25	97,8
	Total	96,2	96	35,8	51,1	52,6	50,4	**76,9**

TABLE 4.1 – Qualité des variables sélectionnées sur données simulées 1

		EN	Lasso	sPLS	EN96	L96	ILS	AG
# var. select.		189,2	181,7	173,3	98	98,1	96	96
% faux positifs		53	52	75	46	45	35,1	**18**
% variables trouvées	*Effets faibles*	79	78,7	5,3	4,8	5,9	22,8	**59,1**
	Effets moyens	97,4	97	36,9	64,7	67,2	78,1	**89,4**
	Effets forts	100	99,8	91,8	**96,7**	96,6	94,7	95,9
	Total	92,2	91,8	44,7	55,4	56,6	65,1	**81,5**

TABLE 4.2 – Qualité des variables sélectionnées sur données simulées 2

Sur ces données simulées, sur les effets faibles et moyens, notre approche basée sur l'algorithme génétique est plus performante en terme de vrais positifs que les méthodes classiques limitées à 96 variables. Elle est légèrement moins performante sur les effets forts mais sur la totalité elle permet de retrouver environ 25% de variables en plus par rapport aux méthodes classiques. De plus, les méthodes elastic net et lasso, non limitées en nombre de variables, ont certes un pourcentage de vrais positifs supérieur à notre méthode mais ont également un grand nombre de faux positifs. En revanche, l'approche que nous proposons basée sur l'algorithme génétique sélectionne très peu de faux positifs, ce qui signifie que la probabilité qu'une variable sélectionnée par cette approche soit réellement significative est plus élevée qu'avec les autres approches.
Concernant les données de QTLMAS, nous connaissons la position des QTL (*Quantitative Trait Locus*), qui sont des régions du génome ayant une influence sur la caractère d'intérêt. Il y a 37 QTL pour les données de QTLMAS 2010 et 8 pour les données de QTLMAS 2011. Nous pouvons donc définir si une variable sélectionnée par une approche est réellement significative ou non (cf. section 3.4.1 pour la définition d'une variable significative sur ces données). Les tables 4.3 et 4.4 rapportent le nombre de variables sélectionnées par chaque approche, le pourcentage de faux positifs (pourcentage de variables sélectionnées non significatives) ainsi que le pourcentage de QTL trouvés par chacune des approches, c'est-à-dire le nombre de QTL pour lesquels au moins une variable (SNP) a été trouvée, divisé par le nombre total de QTL.

	EN	Lasso	sPLS	EN96	L96	ILS	AG
# var. select.	297	191	3796	88	70	94	56,8
% faux positifs	66	61,8	63,4	**58**	61,4	64,5	60,9
% QTL trouvés	94,6	91,9	100	**64,9**	54,1	35,5	39,1

TABLE 4.3 – Qualité des variables sélectionnées sur QTLMAS 2010

Sur les données de QTLMAS 2010 (Table 4.3), nous constatons que notre ap-

proche par algorithme génétique est plus performantes en terme de sélection des bonnes variables que notre approche basée sur une recherche locale itérée. Les méthodes classiques retrouvent cependant plus de variables significatives (QTL) que nos approches.

	EN	**Lasso**	**sPLS**	**EN96**	**L96**	**ILS**	**AG**
# var. select.	36	33	78	55	41	70,7	28,5
% faux positifs	69,4	66,7	80,8	56,4	65,9	**56**	64,3
% QTL trouvés	37,5	37,5	37,5	**37,5**	**37,5**	26,6	28,8

Table 4.4 – Qualité des variables sélectionnées sur QTLMAS 2011

De même que sur les données de QTLMAS 2010, nous remarquons que sur les données de QTLMAS 2011 (Table 4.4) la deuxième approche que nous avons proposée (AG) permet de retrouver plus de QTL que notre première approche (ILS). Nous ne parvenons cependant pas à atteindre les performances des méthodes classiques.

4.3.2 Temps d'exécution

La Table 4.5 présente les temps d'exécution des différentes méthodes sur les jeux de données simulées et pseudo-réelles.

	EN	**Lasso**	**sPLS**	**EN96**	**L96**	**ILS**	**AG**
Simulation 1	17 min.	1 min.	2 min.	10 min.	35 sec.	3h50	4 min.
Simulation 2	12 min.	26 sec.	1 min.	7 min.	15 sec.	2h50	2 min.
QTLMAS 2010	30 min.	1 min	2 min.	8 min.	42 sec.	3h30	2 min.
QTLMAS 2011	26 min.	77 sec.	1 min.	6 min.	27 sec.	2h	2 min. 30

Table 4.5 – Temps d'exécution des différentes méthodes

Alors que la première approche que nous avons proposée, basée sur la recherche locale itérée (ILS), étaient très coûteuse en temps, cette nouvelle approche basée sur l'algorithme génétique avec parallélisation des évaluations, nous permet d'obtenir des résultats en un temps raisonnable et comparable aux approches classiques.

4.3.3 Évaluation des résultats

L'objectif de notre travail, conjointement à la sélection de variables, est d'élaborer un modèle de prédiction. Nous évaluons donc dans cette section les performances de notre approche comparées à des approches classiques couramment utilisées (elastic net (EN), lasso, sparse-PLS (sPLS)), en terme d'erreur moyenne de prédiction (RMSEP - à minimiser) et de corrélation (à maximiser) sur l'échantillon de validation. La Figure 4.18 présentent les résultats sur données simulées 1 et 2 et la Figure 4.19 sur données pseudo-réelles.

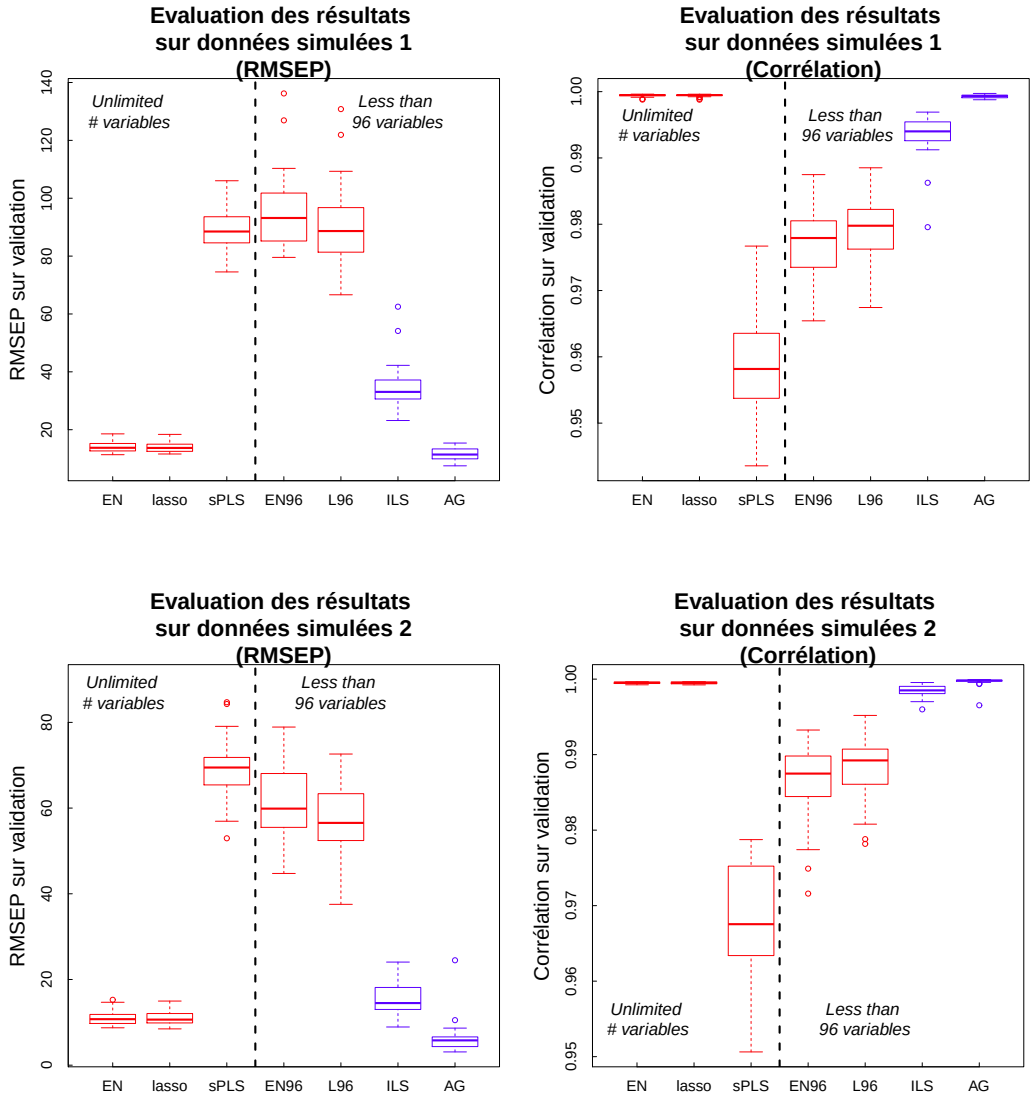

FIGURE 4.18 – Comparaison de l'AG avec des méthodes classiques sur données simulées

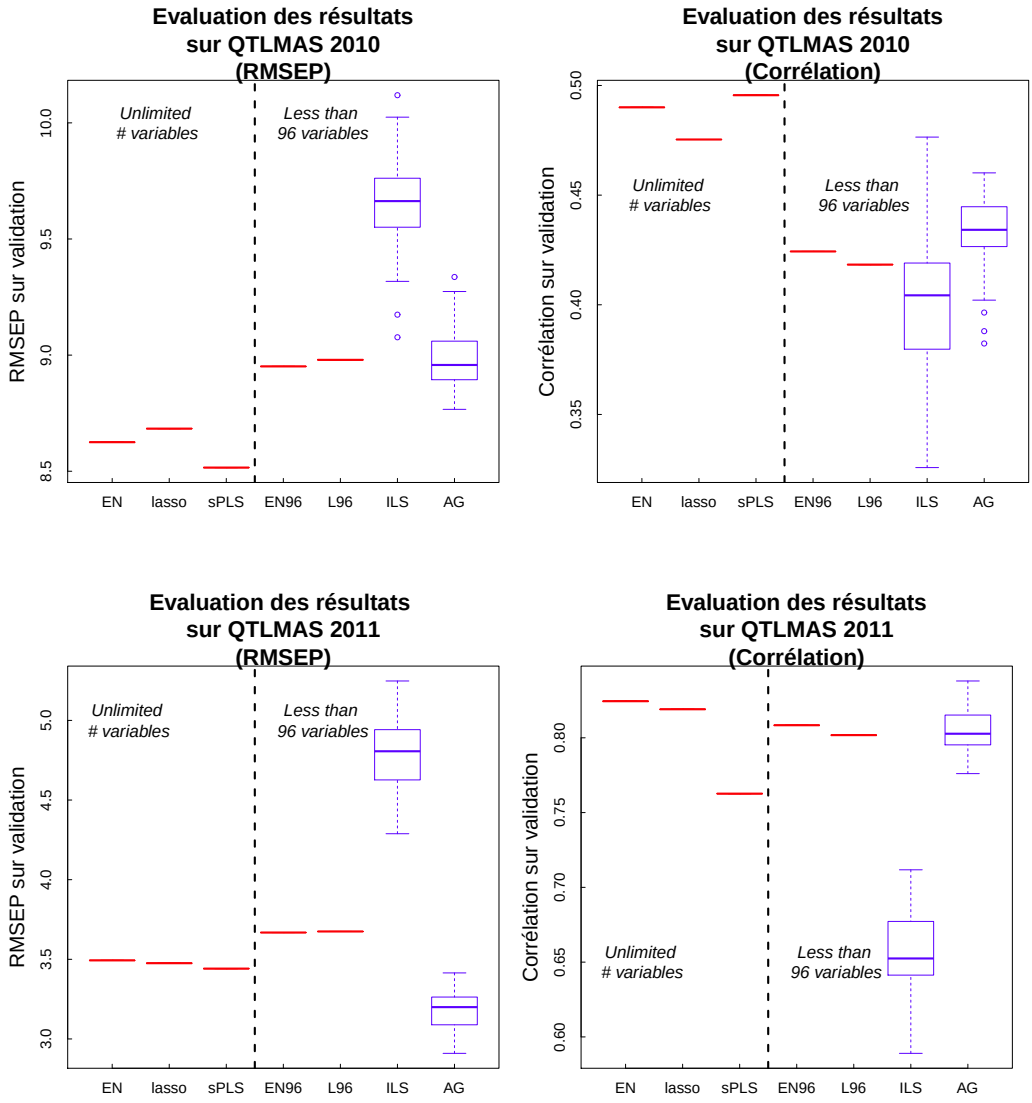

FIGURE 4.19 – Comparaison de l'algorithme avec des méthodes classiques sur données pseudo-réelles

Notre approche basée sur l'algorithme génétique permet d'obtenir des résultats nettement meilleurs que les approches classiques limitées à 96 variables (qui répondent à notre problématique) sur les jeux de données simulées ainsi que sur les données de QTLMAS 2011. De plus, nous obtenons des résultats comparables à ces méthodes classiques sur les données de QTLMAS 2010. Les résultats de notre approche sont également comparables à ceux des méthodes elastic net et lasso non limitées en nombre de variables. La Table 3.6 rappelle le nombre de variables sélectionnées par chaque approche sur les données pseudo-réelles.

		EN	Lasso	sPLS	EN96	L96	ILS	AG
# var. select.	QTLMAS 2010	297	191	3797	88	70	95,8	56,8
	QTLMAS 2011	36	33	79	55	41	70,7	28,5

TABLE 4.6 – Variables sélectionnées sur données pseudo-réelles

Nous remarquons que notre approche basée sur l'AG n'atteint pas la limite des 96 variables que nous lui avons fixée et construit donc un modèle sur un nombre plus faible de variables que les autres approches. Le faible nombre de QTL surs ces jeux de données, et particulièrement sur QTLMAS 2011, peu expliquer le faible nombre de variables sélectionnées par notre approche.

4.4 Conclusion

Dans ce chapitre, pour répondre à notre problématique de sélection de variables en régression, afin d'améliorer les performances prometteuses de notre première approche basée sur une recherche locale itérée, nous avons proposé d'utiliser un algorithme génétique. Nous avons dans un premier temps étudié les différents opérateurs de l'algorithme génétique afin d'en définir la meilleure configuration adaptée à notre problématique. Cette nouvelle approche nous a tout d'abord permis d'améliorer nettement les performances de notre première approche que ce soit en terme de variables sélectionnées et surtout en terme de qualité de prédiction sur l'échantillon de validation. En effet, cette approche nous permet d'obtenir de meilleurs résultats que les méthodes classiques de la littérature (elastic net, lasso, sparse-PLS) sur données simulées ainsi que sur données pseudo-réelles. Nous proposons donc dans le chapitre suivant une application sur données réelles.

Chapitre 5

Application

Sommaire

Ce travail s'inscrit dans le cadre d'une thèse CIFRE, financée par la société *Gènes Diffusion*, coopérative agricole spécialisée en génétique et reproduction animale sur les espèces bovines, équines, porcines et lapines. Les études menées par la société *Gènes Diffusion* nous permettent de tester nos approches sur données réelles. Nous présentons donc dans la première partie de ce chapitre, le projet Qualvigène dans lequel s'inscrit notre application. Puis nous évaluons les performances des deux approches que nous avons proposées précédemment. Ces nouvelles données semblant mener vers un nouveau comportement des différentes méthodes nous proposons d'améliorer les résultats obtenus sur cette application par l'intermédiaire de deux nouvelles approches. La première, présentée en deuxième partie de ce chapitre, permet de gérer le problème du sur-apprentissage particulièrement présent sur cette application, dû au très grand nombre de variables étudiées comparé au nombre d'individus disponibles ($n \ll p$). La seconde, que nous présentons en troisième partie de ce chapitre, consiste à modifier la fonction d'évaluation de l'algorithme de manière à proposer un modèle descriptif intégrant les relations familiales connues entre les animaux.

5.1 Le projet Qualvigène

Dans le chapitre précédent nous avons montré que notre approche basée sur un algorithme génétique avec évaluation des solutions par régression linéaire multiple permet d'obtenir de bons résultats sur données simulées et pseudo-réelles. Nous présentons dans cette section une application sur données réelles en commençant par décrire le projet à l'origine de cette application ainsi que les données dont nous disposons. Puis nous détaillons le design expérimental utilisé et enfin, les résultats obtenus sont comparés à des méthodes classiques de la littérature (elastic-net (EN), lasso, sparse PLS (sPLS)).

5.1.1 Description du projet

Après un constat selon lequel la consommation de viande en France est en baisse depuis une dizaine d'années, le projet Qualvigène est créé en 2003 dans l'objectif d'améliorer les qualités organoleptiques (tendreté, jutosité, flaveur et couleur de la viande) grâce à la génomique. En effet, la mesure de ces caractères étant coûteuse (mise en place d'un jury de dégustation en plus des coûts de prélèvement), l'exploration de la voie génomique semble pertinente. Ce programme Qualvigène est mis en œuvre par l'Institut National de la Recherche Agronomique (INRA), l'Institut de l'élevage, des centres d'insémination ainsi que des abattoirs. La société *Gènes Diffusion* fait partie des établissements partenaires de ce projet.
Parmi les animaux du programme, figurent des taureaux et des taurillons, qui sont des mâles élevés de façon à être abattus avant l'âge de 2 ans. Ils sont issus de 3 races :

- charolaise (avec 48 pères et 1114 taurillons),
- limousine (avec 36 pères et 1254 taurillons),
- blonde d'Aquitaine (avec 30 pères et 981 taurillons).

Sur ces animaux, les caractères étudiés sont regroupés en 3 catégories :

- les aptitudes bouchères (rendement de carcasse, développement musculaire, ...),
- les caractéristiques musculaires (taille et nombre de fibres, taux de lipides intramusculaires, ...),
- la qualité de la viande (force de cisaillement, note de tendreté, ...).

Dans ce travail, nous nous intéressons à la race charolaise, et au rendement de carcasse qui est un caractère ayant une forte héritabilité ($h^2 = 0.54$ - cf. Chapitre 1). Suite à un pré-traitement concernant les données sur les animaux disponibles (avec notamment la suppression des animaux non phénotypés pour ce caractère), nous obtenons finalement 1107 animaux (dont 48 pères) génotypés en 54K. Après la phase de contrôle qualité des données de génotypage (cf. section 1.2.2) 43 896 SNPs sont conservés pour l'étude. Nous disposons donc d'une matrice de données de taille 1107×43896, associée à un vecteur de taille 1107 correspondant au rendement de carcasse. Les valeurs du caractère étudié ici sont les performances dérégressées (DRP - cf. section 1.2.2). Nous disposons également d'un pedigree de 4741 animaux

indiquant les relations familiales entre ces animaux.

5.1.2 Design expérimental

Parmi les 1107 animaux génotypés, 100 taurillons sont sélectionnés aléatoirement pour constituer l'échantillon de validation afin de comparer les approches. L'échantillon d'apprentissage est donc constitué de 1007 animaux. Les 100 animaux de l'échantillon de validation sont sélectionnés uniquement parmi les taurillons car les performances des taureaux sont plus fiables, puisqu'elles sont issues de mesures sur un grand nombre de descendants, il est donc préférable de les conserver dans l'apprentissage. De plus, les animaux sur lesquels le caractère devra être prédit sont de jeunes animaux, constituer l'échantillon de validation avec uniquement des taurillons est donc plus représentatif des animaux sur lesquels nous allons devoir prédire le caractère. Afin de pouvoir généraliser les résultats obtenus, le découpage en échantillon d'apprentissage et de validation est réalisé 30 fois (génération de 30 instances).

Nous testons ici les deux approches que nous avons proposées : la première basée sur une recherche locale itérée (ILS) et la seconde basée sur un algorithme génétique (AG). Nous utilisons, pour chaque algorithme, la meilleure configuration adaptée à la problématique que nous étudions, définie expérimentalement sur données simulées et pseudo-réelles dans le Chapitre 3 pour l'algorithme ILS et dans le Chapitre 4 pour l'algorithme génétique.
Les paramètres de l'algorithme ILS sont les suivants :

- initialisation : uniforme,
- voisinage : opérateur bit-flip,
- perturbation : suppression de 5% des variables de la solution courante,
- évaluation d'une solution : régression linéaire multiple sur laquelle le critère BIC est calculé et devient la fitness (à minimiser) de la solution,
- critère d'arrêt : nombre maximal fixé d'évaluations (ce nombre sera déterminé dans la section 5.1.3).

Les paramètres de l'algorithme génétique (AG) sont les suivants :

- taille de la population : 300,
- initialisation : basée sur la méthode lasso pour 50% des solutions de la population initiale, uniforme pour les autres,
- sélection : tournoi,
- reproduction :
  - croisement : opérateur SSOCF [Emmanouilidis 2000], avec un taux de croisement de 0.8,
  - mutation : flip de 4 bits, avec un taux de mutation de 0.8,
- diversification : *random immigrants*,
- évaluation d'une solution : régression linéaire multiple sur laquelle le critère BIC est calculé et devient la fitness (à minimiser) de la solution,
- critère d'arrêt : nombre maximal de générations (ce nombre sera déterminé

dans la section 5.1.3).

Les algorithmes peuvent sélectionner 96 variables au maximum, comme dans les chapitres précédents, ce qui représente la taille standard d'une mini-puce. La création d'une mini-puce contenant uniquement les marqueurs significatifs permettrait de génotyper les animaux à moindres coûts.
Nous exécutons chaque algorithme (ILS, AG), ainsi que les différentes méthodes classiques auxquelles nous nous comparons, sur les 30 jeux de données dont nous disposons (30 découpages en échantillon d'apprentissage et de validation du jeu de données initial). Les résultats sont évalués, sur l'échantillon de validation, en terme d'erreur de prédiction ainsi qu'en terme de corrélation entre le caractère prédit et le caractère réel. De même que dans les deux chapitres précédents, nous nous comparons à des méthodes classiques de la littérature (elasticnet (EN), lasso, sparse PLS (sPLS), EN96 et L96).

5.1.3 Critère d'arrêt

Lors des expérimentations précédentes, nous avons constaté que le nombre d'évaluations (resp. de générations) nécessaires à l'algorithme ILS (resp. à l'algorithme génétique) pour converger dépend du jeu de données étudié. Nous étudions donc la convergence de chaque algorithme (ILS, AG) en visualisant l'évolution de la fitness (valeur du critère BIC sur les données d'apprentissage), afin de définir leur critère d'arrêt. La Figure 5.1 illustre la convergence des différentes approches. Le graphique de gauche illustre la convergence de l'algorithme ILS, qui est donc la représentation des optima locaux (solutions n'ayant pas de solution voisine améliorante) en fonction du nombre d'évaluations. Le graphique de droite illustre l'évolution de la meilleure solution de la population de l'algorithme génétique, en fonction du nombre de générations.

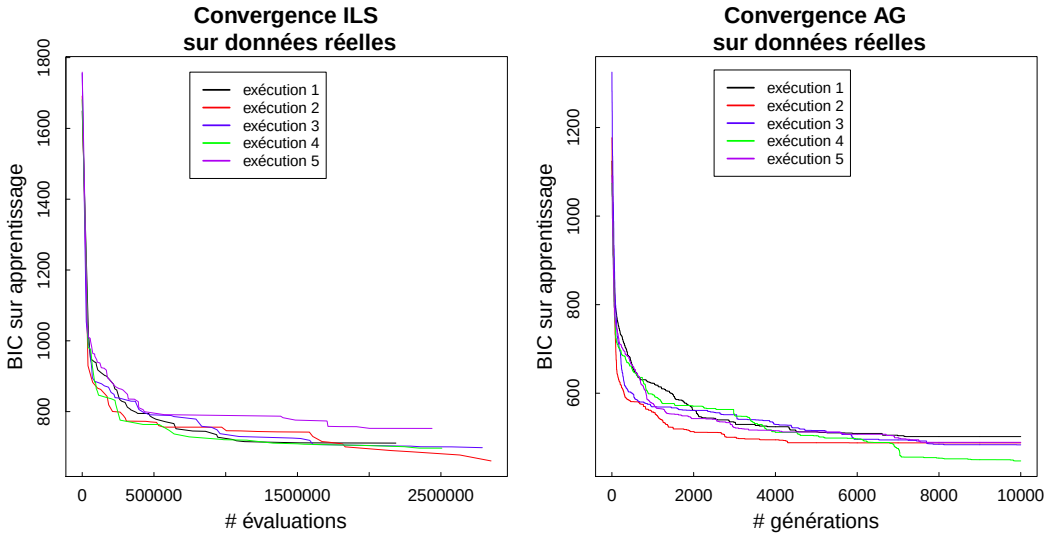

FIGURE 5.1 – Convergence sur données réelles

D'après la figure, pour la recherche locale itérée, nous jugeons que 3 000 000 d'évaluations suffisent à l'algorithme pour converger. Concernant l'algorithme génétique, 10 000 générations semblent nécessaires à l'algorithme pour converger. Nous pouvons remarquer dès à présent que l'algorithme ILS semble moins performant puisqu'il converge vers des solutions de moins bonnes qualités ($\simeq$ 800) que l'algorithme génétique ($\simeq$ 600).

5.1.4 Évaluation des résultats

Le nombre d'évaluations (resp. générations) étant fixé, nous pouvons comparer les résultats obtenus à ceux des approches de la littérature.
La Figure 5.2 illustre les résultats que nous obtenons sur les données réelles, sur le graphique de gauche en terme d'erreur de prédiction et sur le graphique de droite en terme de corrélation entre le caractère prédit et le caractère réel, sur l'échantillon de validation.

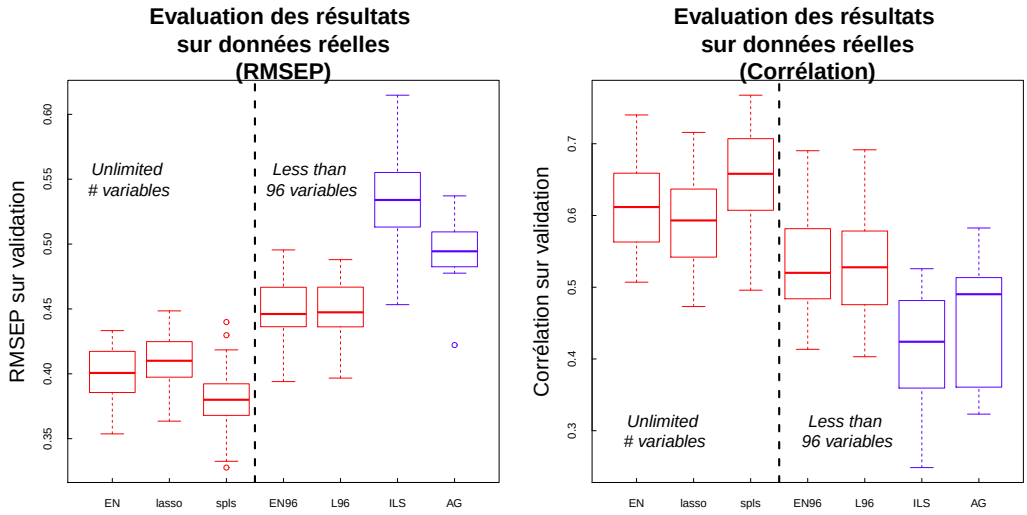

Figure 5.2 – Performances sur données réelles

Nous remarquons dans un premier temps que, de même que ce que nous avons constaté sur les données simulées et pseudo-réelles, les résultats de notre approche basée sur l'algorithme ILS sont nettement moins bons que ceux obtenus par l'algorithme génétique proposé.
Cependant, nous constatons que les résultats obtenus par notre approche la plus performante (AG) sont sensiblement moins bons que les méthodes classiques que ce soit en terme d'erreur de prédiction ou de corrélation. L'approche sparse-PLS est la méthode qui permet d'obtenir les meilleurs résultats comparés aux autres approches, cependant elle sélectionne un très grand nombre de variables (Table 5.1).

EN	**Lasso**	**sPLS**	**EN96**	**L96**	**ILS**	**AG**
578,8	297,3	31905	86,8	84,1	95,9	52,5

Table 5.1 – Nombre de variables sélectionnées

En effet, ce tableau montre que les approches non limitées en nombre de variables sélectionnées sont certes celles qui permettent d'obtenir les meilleurs résultats sur ces données, mais elles sélectionnent un grand nombre de variables ce qui ne répond pas à notre problématique de sélection de variables.
Notre approche basée sur une recherche locale itérée étant moins performante que l'approche basée sur un algorithme génétique nous utiliserons uniquement cette dernière approche pour les expérimentations suivantes.
Une question se pose ici : pourquoi l'algorithme génétique est-il moins performant sur ces données réelles que sur les données simulées et pseudo-réelles utilisées dans les chapitres précédents ? Nous proposons quelques éléments de réponse.

Sur les données de QTLMAS 2010 (resp. 2011) nous avions environ 2000 animaux génotypés sur 9768 (resp. 7121) marqueurs. Sur ces données réelles, alors que nous avons deux fois moins d'animaux, ils sont génotypés sur environ 40 000 marqueurs, nous sommes dans le cas où le nombre n d'individus est très faible comparé au nombre p de variables ($n \ll p$). Ce déséquilibre entre le nombre d'individus et le nombre de variables étudiés est plus favorable au sur-apprentissage, ce qui pourrait expliquer que nos performances soit moins bonnes sur cette application que sur les données de QTLMAS comparés aux méthodes classiques. Nous étudions donc le problème du sur-apprentissage dans la section suivante et proposons, afin de le diminuer, une nouvelle méthode de choix de la solution finale.
D'autre part, notre approche basée sur la régression linéaire, comme les méthodes classiques, suppose l'indépendance entre les individus. Or, nous savons qu'il existe des relations familiales connues entre les animaux. Il est possible que dans ce contexte de données réelles, les relations familiales aient un poids plus important et qu'il faille les prendre en compte. Notre approche d'optimisation étant généralisable à n'importe quel modèle de régression nous proposons d'utiliser un modèle mixte afin de prendre en compte ces relations familiales. Nous présentons cette approche en section 5.3.
Ainsi, les deux prochaines parties présentent des pistes d'amélioration pour la méthode proposée.

5.2 Problème du sur-apprentissage

Le sur-apprentissage (ou sur-ajustement - *overfitting*) est un problème connu dans la littérature [Hawkins 2004]. Nous y sommes confrontés lorsque l'algorithme génère un modèle trop spécifique aux données d'apprentissage ce qui diminue ses performances de prédiction sur de nouvelles données (échantillon de validation).
Pour visualiser ce problème dans le cadre de notre étude, nous étudions tout d'abord l'évolution de la qualité des solutions sur l'échantillon de validation (sur 10 instances) en fonction du nombre de générations fixé à l'algorithme (Figure 5.3).

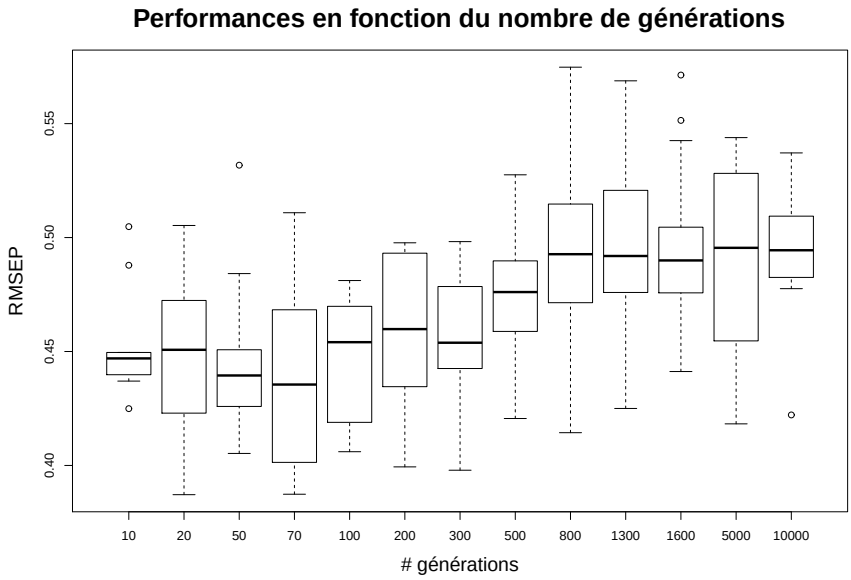

FIGURE 5.3 – Performances sur l'échantillon de validation en fonction du nombre de générations

Notons tout d'abord que, pour ne pas perdre en lisibilité, l'échelle de l'axe des abscisses n'a pas été normée. Sur ce graphique, nous constatons que laisser converger l'algorithme semble dégrader la qualité moyenne des solutions finales. En effet, en augmentant le nombre de générations, nous améliorons les résultats au début mais il y a un nombre de générations à partir duquel l'erreur sur l'échantillon de validation se dégrade (de plus de 10 %).
Il semble donc y avoir, pour chaque exécution, un point d'inflexion à partir duquel, plus on laisse converger l'algorithme, plus il va générer un modèle adapté aux données d'apprentissage, pour au final obtenir un modèle trop représentatif des données d'apprentissage et dont les performances sur l'échantillon de validation seront mauvaises : nous sommes face au problème du sur-apprentissage.
Pour visualiser ce problème, nous comparons également les performances de 5 exécutions, obtenues pour un même nombre de générations, sur l'apprentissage et sur la validation. Le graphique de gauche de la Figure 5.4 représente les résultats obtenus pour différentes exécutions ordonnées par qualité sur l'échantillon d'apprentissage. La solution finale de l'exécution 1 donne donc la meilleure performance sur l'échantillon d'apprentissage et celle de l'exécution 5 la moins bonne. Nous gardons les exécutions dans le même ordre et représentons les résultats sur l'échantillon de validation (graphique de droite). Nous constatons que la meilleure solution est celle de l'exécution 3 alors qu'elle n'était que 3$^{\text{ème}}$ sur l'apprentissage et la meilleure solution sur l'échantillon d'apprentissage (exécution 1) devient la moins bonne sur l'échantillon de validation.

Ces analyses expérimentales nous permettent alors de conclure que laisser l'algorithme converger et extraire la meilleure solution de la population finale n'est pas

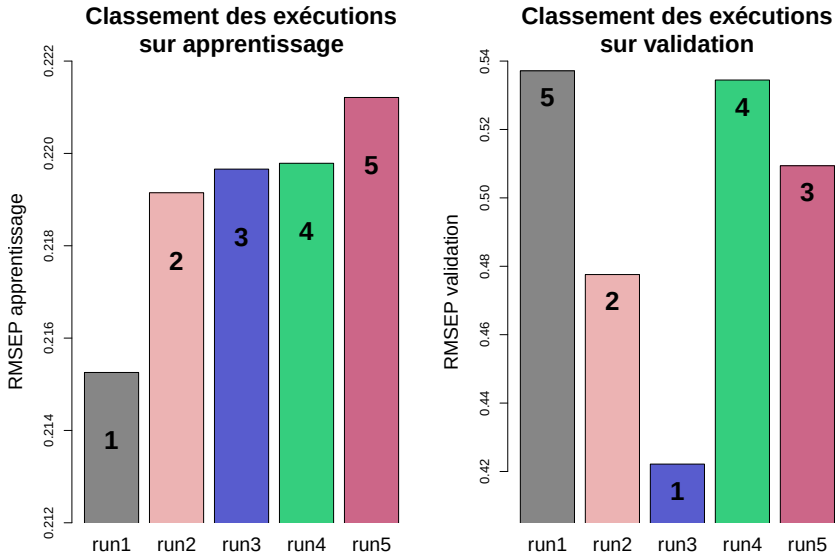

FIGURE 5.4 – Comparaison des performances sur apprentissage et validation

optimal. Nous proposons donc de modifier notre manière de choisir la solution finale.

5.2.1 Une réponse : choix de la solution finale

Dans la version présentée précédemment, une fois l'algorithme exécuté, nous choisissions la meilleure solution de la population finale pour constituer notre modèle de prédiction. Or, toutes les solutions de la population finale sont potentiellement intéressantes et choisir la meilleure, c'est-à-dire la plus performante sur l'échantillon d'apprentissage, ne permettra pas forcément d'obtenir les meilleurs résultats sur l'échantillon de validation. De plus, au cours de l'algorithme, des solutions permettant de générer un meilleur modèle de prédiction sur l'échantillon de validation ont sans doute été visitées mais peuvent avoir ensuite été remplacées par des solutions plus performantes sur l'échantillon d'apprentissage mais moins performantes sur la validation.
Afin d'éviter ce sur-apprentissage, nous proposons d'extraire un sous-ensemble d'individus de notre échantillon d'apprentissage pour constituer un échantillon de test [Broadhurst 1997] permettant de mesurer la qualité de prédiction des solutions au cours de l'algorithme mais n'intervenant pas dans son déroulement. En effet, jusqu'à présent nous avions un échantillon d'apprentissage sur lequel nous exécutions l'algorithme et un échantillon de validation sur lequel nous évaluions les performances de la solution finale. Nous proposons maintenant de diviser l'échantillon d'apprentissage en deux de sorte à obtenir trois échantillons (Figure 5.5) :

- un échantillon d'apprentissage (A),
- un échantillon de test (T),

– et un échantillon de validation (V).

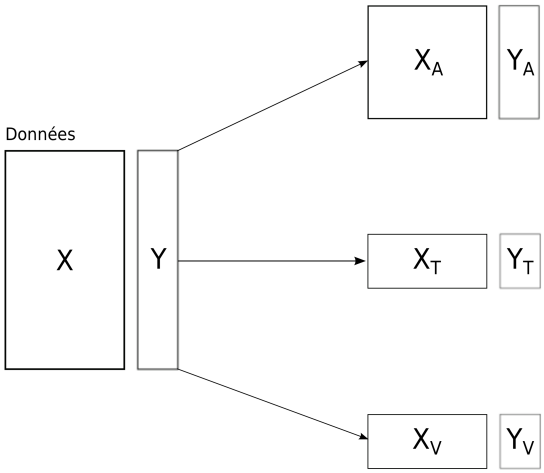

FIGURE 5.5 – Échantillons d'apprentissage, de test et de validation

Une fois les données divisées en trois groupes, notre approche se déroule en trois étapes successives, selon le schéma présenté sur la Figure 5.6.

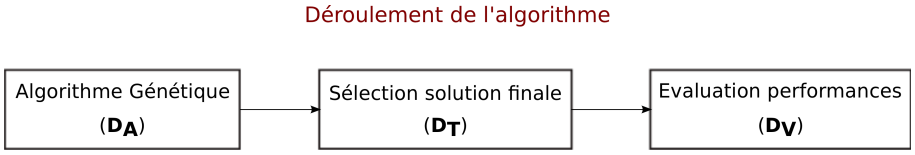

FIGURE 5.6 – Déroulement de l'algorithme

L'algorithme est exécuté sur les données d'apprentissage (D_A). Les solutions améliorantes sur l'échantillon d'apprentissage sont conservées puis évaluées sur l'échantillon de test afin de sélectionner la meilleure solution sur les données de test (D_T). Enfin, cette solution finale est évaluée sur les données de validation (D_V).

Une question se pose ici sur le nombre d'individus dans chaque échantillon. Nous avons au total pour ce jeu de données 1107 animaux. Pour l'échantillon de validation, nous conservons la taille que nous avons utilisée jusqu'ici soit 100 individus. Quand aux échantillons d'apprentissage et de test, nous envisageons deux possibilités :

– 300 individus dans l'échantillon de test et donc 707 dans l'échantillon d'apprentissage,
– 100 individus dans l'échantillon de test et donc 907 dans l'échantillon d'apprentissage.

Nous comparons donc, suivant ces deux découpages, les résultats finaux c'est à dire les erreurs de prédiction, sur l'échantillon de validation, de la solution la plus performante sur l'échantillon de test.

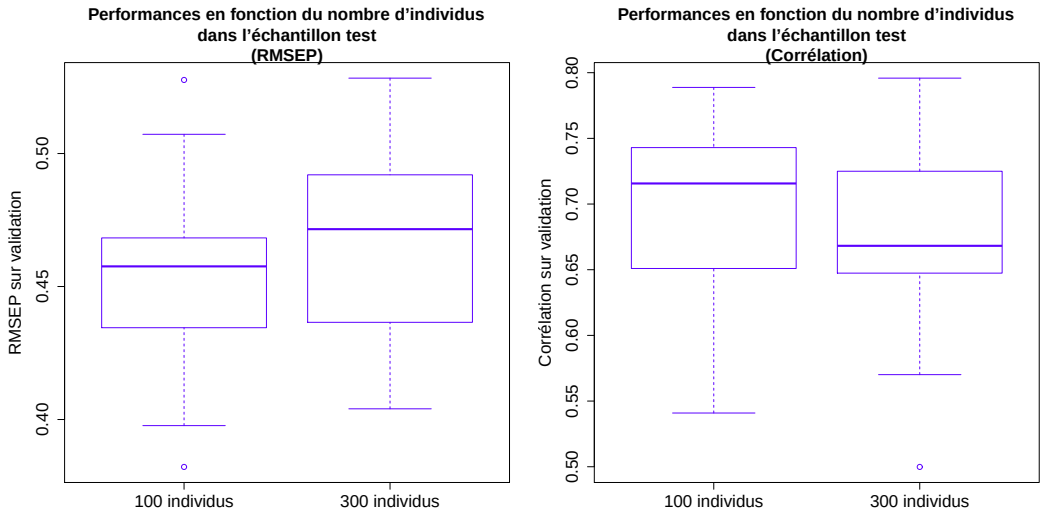

FIGURE 5.7 – Comparaison des performances suivant le nombre d'individus dans l'échantillon test

Nous remarquons sur la Figure 5.7, qu'il ne semble pas y avoir de différence significative entre les résultats obtenus avec un échantillon test de taille 100 et ceux obtenus avec un échantillon test de taille 300. La taille de cet échantillon test ne semble donc pas avoir d'influence. Nous proposons de conserver un échantillon test de taille 300, ce qui correspond sensiblement au découpage classique avec 2/3 des individus pour l'échantillon d'apprentissage, 1/3 pour l'échantillon test.

Pertinence du nouveau processus de choix de la solution finale

Afin d'étudier la pertinence de ce nouveau processus de choix de la solution finale, nous visualisons dans un premier temps, pour 4 exécutions de l'algorithme génétique, sur la Figure 5.8, l'évolution au cours des générations de la meilleure solution de la population et sa qualité sur l'échantillon d'apprentissage (courbe rouge) et sur l'échantillon de test (courbe bleue).

La Figure 5.8 permet de constater que d'une part, le modèle est plus adapté aux données d'apprentissage, et d'autre part que très peu de générations sont nécessaires avant que l'erreur sur l'échantillon test soit dégradée plutôt qu'améliorée. Il ne semble donc pas nécessaire d'exécuter l'algorithme sur un grand nombre de générations ce qui permet donc de diminuer les temps de calcul.

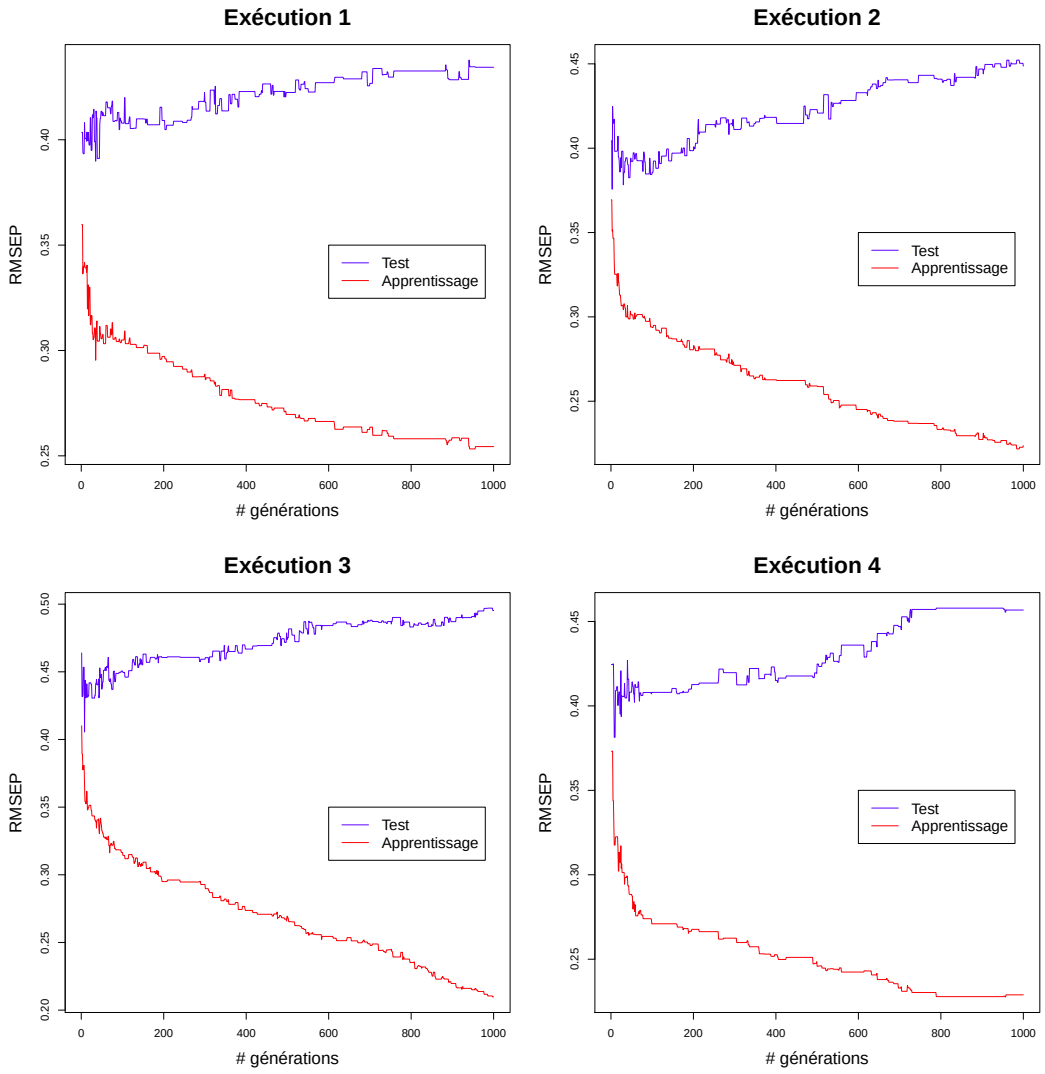

Figure 5.8 – Comparaison des erreurs sur échantillon d'apprentissage et de test

5.2.2 Évaluation des résultats

Nous étudions ici les performances de notre approche basée sur un algorithme génétique avec processus de gestion du sur-apprentissage (AG-2). Nous nous comparons à notre approche précédente (AG) ainsi qu'aux méthodes classiques de la littérature (elastic-net, lasso, sparse PLS, EN96 et L96). Les résultats sont présentés sur la Figure 5.9.

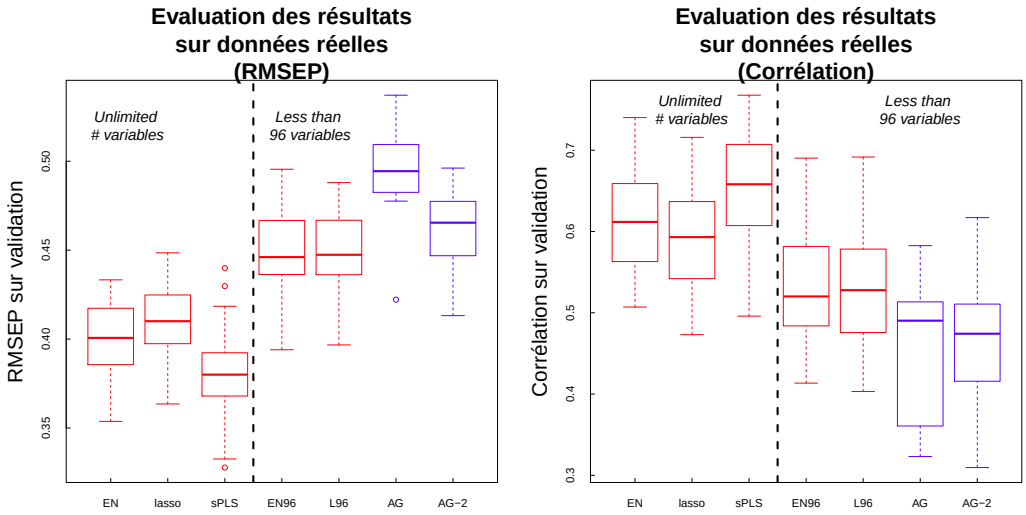

FIGURE 5.9 – Performances sur données réelles

Nous pouvons remarquer qu'en terme d'erreur de prédiction sur l'échantillon de validation, le processus de gestion du sur-apprentissage améliore considérablement les performances de notre approche. En revanche, cette amélioration est moins flagrante en terme de corrélation où il ne semble pas y avoir de différence entre l'utilisation ou non de ce processus. Ceci peut s'expliquer par le fait que nous choisissons la solution finale en fonction de l'erreur de prédiction sur l'échantillon de test et non en fonction de la corrélation.
Grâce à ce processus de gestion du sur-apprentissage, les performances de notre approche sont comparables à celles des méthodes classiques limitées à 96 variables. Lorsque nous nous intéressons au nombre de variables sélectionnées par chacune des approches (Table 5.2), nous constatons que l'utilisation du processus de gestion du sur-apprentissage diminue nettement le nombre moyen de variables sélectionnées par notre approche. Nous obtenons donc finalement des performances presque similaires aux méthodes EN96 et L96 en sélectionnant deux fois moins de variables.

	EN	**Lasso**	**sPLS**	**EN96**	**L96**	**AG**	**AG-2**
# **var. select.**	578,8	297,3	31905	86,8	84,1	89,4	48

TABLE 5.2 – Nombre de variables sélectionnées sur données réelles

5.3 Intégration des relations familiales

Nous avons vu dans la section précédente que le problème du sur-apprentissage est particulièrement présent sur ce jeu de données réelles et nous avons proposé une méthode permettant de le réduire.
Nous souhaitons maintenant analyser un deuxième axe d'amélioration de notre méthode basée sur un algorithme génétique et pour cela nous étudions la pertinence du modèle de prédiction utilisé dans notre approche. En effet, dans le cas du modèle de régression linéaire multiple que nous avons proposé précédemment (section 3.1.1), et comme dans un grand nombre d'approches de la littérature, les animaux sont considérés indépendants. Or, nous avons vu dans la section 2.3.1, qu'une spécificité des données de génomique animale, contrairement à l'humain par exemple, est que les animaux étudiés sont issus de troupeaux dont beaucoup d'animaux ont été croisés. Sur les données de réelles par exemple, il y a 48 géniteurs pour 1107 progénitures. Nous savons donc qu'il existe des relations familiales, généralement connues, entre les animaux, et que ne pas les prendre en compte peut amener à considérer des SNPs significatifs alors qu'ils ne le sont pas, c'est à dire augmenter le nombre de faux positifs (voir section 1.2.2 pour plus de détails). Afin d'améliorer les performances de notre modèle, que ce soit en terme de variables sélectionnées ou de prédiction, nous proposons d'intégrer ces relations familiales, en utilisant le pedigree (cf. section 2.3.1), et ce à l'aide d'un modèle mixte. Dans ce chapitre, nous commençons par présenter le modèle mixte que nous utilisons puis nous étudions la pertinence de l'intégration de ces relations familiales à l'aide de ce modèle.

5.3.1 Modélisation par un modèle mixte

Comme vu dans la section 2.3.2, un modèle mixte est un modèle de régression comportant à la fois au moins un effet fixe (dont les paramètres sont associés à une population entière) et un effet aléatoire (dont les données sont observées sur un échantillon de la population). Nous proposons ici de nous baser sur notre modèle de régression linéaire multiple proposé précédemment (équation 3.1) où les effets fixes sont les effets des marqueurs, auxquels nous ajoutons un effet aléatoire tenant compte de la non-indépendance des individus. Le modèle est donc le suivant :

$$\mathbf{y} = X(\boldsymbol{\beta} \cdot \boldsymbol{\gamma}) + Z\mathbf{u} + \mathbf{e}, \tag{5.1}$$

où :

- $\mathbf{y}$ est le caractère d'intérêt,
- X est la matrice des SNPs, de dimension $n \times (p+1)$, n étant le nombre d'animaux et p le nombre de SNPs, avec x_{ij} la valeur du SNP j pour l'individu i, la première colonne de X est une colonne de 1,
- $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)^t$ est le vecteur des effets fixes des SNPs,
- $\boldsymbol{\gamma} = (\gamma_0, \gamma_1, \ldots, \gamma_p)^t$ est un vecteur permettant de sélectionner les variables, avec $\gamma_0 = 1$ et pour $j = 1, \ldots, p$, $\gamma_j = 1$ si la variable j est dans le modèle, 0 sinon,

- $\boldsymbol{\beta} \cdot \boldsymbol{\gamma}$ est le produit terme à terme des vecteurs $\boldsymbol{\beta}$ et $\boldsymbol{\gamma}$,
- $\mathbf{u} \sim \mathcal{N}(0, \sigma^2_{\mathbf{u}} A)$ est un vecteur aléatoire de dimension q (nombre d'animaux dans le pedigree), A étant la matrice de parenté calculée à partir du pedigree (voir section 2.3.1). Le nombre q d'animaux dans le pedigree est plus important que le nombre n d'animaux génotypés, puisqu'on retrouve par exemple dans le pedigree, des parents non génotypés d'animaux génotypés.
- Z est la matrice indicatrice associée à $\mathbf{u}$, de dimension $n \times q$, q étant le nombre d'individus disponibles dans le pedigree, avec $z_{ij} = 1$ si le i$^{\text{ème}}$ animal génotypé est à la j$^{\text{ème}}$ position dans la liste des individus du pedigree et 0 sinon. La Figure 5.10 montre un exemple de matrice Z.

	A1	A2	A3	A4	A5	A6
A2	0	1	0	0	0	0
A3	0	0	1	0	0	0
A5	0	0	0	0	1	0
A6	0	0	0	0	0	1

FIGURE 5.10 – Exemple de matrice Z

  Sur cet exemple, les individus $A1$ et $A4$ sont dans le pedigree mais ne sont pas génotypés.
- $\mathbf{e} \sim \mathcal{N}(0, \sigma^2_{\mathbf{e}})$ est le vecteur aléatoire des erreurs résiduelles.

L'objectif est d'estimer les variances $\sigma^2_{\mathbf{e}}$ et $\sigma^2_{\mathbf{u}}$, les paramètres γ_j, le paramètre β_0 et les effets des marqueurs $\{\beta_j : \gamma_j = 1, 1 \leq j \leq p\}$ et de prédire le vecteur des effets aléatoires $\mathbf{u}$.

5.3.2 Méthode mise en œuvre

De même que dans le chapitre précédent, nous proposons d'utiliser l'optimisation combinatoire pour déterminer un sous-ensemble de variables intéressantes (estimer les γ_j). Nous avons montré dans le Chapitre 4 que l'algorithme génétique est plus performant que la recherche locale itérée sur les données simulées et pseudo-réelles que nous avons utilisées. Nous proposons donc ici d'estimer les paramètres γ_j du modèle mixte à l'aide d'un algorithme génétique.
Les paramètres de l'algorithme génétique sont les suivants :

- initialisation : basée sur la méthode lasso pour 50% des solutions de la population initiale, uniforme pour les autres,
- sélection : tournoi,
- reproduction :
  - croisement : opérateur SSOCF [Emmanouilidis 2000], avec un taux de croisement de 0.8,
  - mutation : flip de 4 bits, avec un taux de mutation de 0.8,
- diversification : *random immigrants*,

- évaluation : modèle mixte sur lequel nous utilisons la validation croisée 3-fold (définie section 3.2.3) pour évaluer sa qualité. En effet, le calcul du critère BIC nécessite le calcul de la vraisemblance du modèle. Or, comme expliqué dans [Foulley 2002], la méthode du maximum de vraisemblance n'est pas adaptée aux modèles mixtes, l'utilisation du maximum de vraisemblance restreint (REML) est préconisée sur ce type de modèle. Une adaptation du critère BIC a été proposée par [Delattre 2012] dans le cadre de données répétées mais ce n'est pas le cas de nos données, nous choisissons donc d'utiliser de la validation croisée 3-fold.
- critère d'arrêt : nombre maximal de générations.

L'évaluation d'une sélection de variables par un modèle mixte est réalisée à l'aide du programme fortran *BLUPF90* fourni par Misztal [Misztal 2002] pour l'utilisation de modèles mixtes en reproduction animale. Classiquement, ce programme est utilisé pour des modèle soù les effets fixes sont des effets environnementaux et les effets aléatoires sont les effets des SNPs. Nous n'avons pas d'effets environnementaux à prendre en compte dans notre modèle puisque nous utilisons comme caractère d'intérêt des performances corrigées de ces effets (DRP - cf. section 1.2.2). Dans notre approche, nous fournissons donc au programme, les SNPs en effets fixes et les effets aléatoires représentent les relations familiales. Nous lui fournissons également la variance résiduelle, estimée à partir du programme *AIREMLF90* de Misztal.
De même que précédemment nous avons validé expérimentalement la pertinence des différents opérateurs de l'algorithme sur ces nouvelles données. Suite à ces analyses expérimentales, nous avons fixé la taille de la population à 100 solutions et le nombre de générations à 10. En effet, l'analyse du problème du sur-apprentissage présenté de la section précédente a montré que peu de générations permettaient d'obtenir de bonnes performances et après quelques expérimentations 10 générations nous a semblé être le plus pertinent. Nous verrons dans les perspectives que, pour une détermination plus fine du nombre de générations nécessaires, il serait intéressant d'intégrer le processus de gestion du sur-apprentissage proposé précédemment à cette nouvelle approche basée sur un modèle mixte.

5.3.3 Évaluation des résultats

Afin d'étudier la pertinence de l'intégration des relations familiales, nous comparons dans cette section les performances de l'approche par modèle mixte (AG-MM) avec celles des trois méthodes que nous avons proposée précédemment : la première basée sur une recherche locale itérée (ILS), la seconde basée sur un algorithme génétique avec évaluation par régression linéaire multiple (AG) et la troisième intégrant le processus de gestion du sur-apprentissage (AG-2). Comme précédemment, nous nous comparons également aux méthodes classiques elastic-net (EN), lasso, sparse PLS (sPLS) et aux méthodes EN et lasso limitées à 96 variables (EN96 et L96).

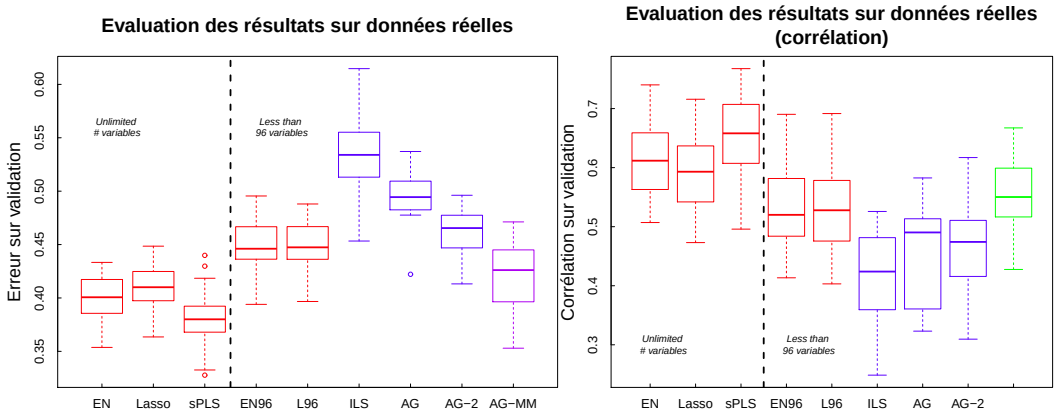

FIGURE 5.11 – Performances sur données réelles

Sur la Figure 5.11 nous remarquons dans un premier temps que, sur les données réelles, cette nouvelle approche intégrant les relations familiales améliore nettement les résultats des approches que nous avions proposées précédemment, que ce soit en terme d'erreur de prédiction ou de corrélation. De plus, nous obtenons des résultats sensiblement meilleurs que les approches elastic-net et lasso limitées à 96 variables. Les méthodes elastic-net, lasso et sparse PLS restent meilleures que nos approches mais sélectionnent beaucoup trop de variables (Table 5.3).

EN	**Lasso**	**sPLS**	**EN96**	**96**	**ILS**	**AG**	**AG-2**	**AG-MM**
578,8	297,3	31905	86,8	84,1	95,9	89,4	48	48,9

TABLE 5.3 – Nombre de variables sélectionnées

Afin d'évaluer la significativité de la différence observée graphiquement entre notre meilleure approche (AG-MM) et les méthodes classiques limitées à 96 variables (EN96 et L96), nous avons réalisé un test de Student de comparaison des moyennes. En terme de corrélation, les tests concluent à une différence non significative entre les performances de notre approche et celles des méthodes classiques ($p-valeur = 0.19$ lors de la comparaison avec EN96 et $p-valeur = 0.23$ avec L96). En revanche, en terme d'erreur de prédiction notre approche (AG-MM) est significativement meilleure que les approches classiques (p-valeur$<$0.001 pour EN96 et L96) sur ces données réelles. Les expérimentations mettent donc en évidence l'intérêt d'utiliser un modèle mixte pour ce type de données en terme de qualité de prédiction.

5.3.4 Temps d'exécution

Les modèles mixtes nécessitent l'estimation d'un plus grand nombre de paramètres ce qui rend la méthode plus coûteuse en temps d'exécution. Nous comparons donc ici les temps d'exécution de chacune des approches.

EN	**Lasso**	**sPLS**	**EN96**	**L96**	**ILS**	**AG**	**AG-2**	**AG-MM**
35 min.	3 min.	5h40	16 min.	1 min.	2h55	20 min.	3 min. 10	10 min.

Table 5.4 – Temps d'exécution des différentes méthodes

Notons tout d'abord que pour nos méthodes basées sur l'algorithme génétique (AG, AG-2 et AG-MM), étant donné que l'initialisation de l'algorithme est basée sur la solution de la méthode lasso, les temps d'exécution prennent en compte le temps d'exécution de la méthode lasso (3 min.). Nous remarquons que l'évaluation par modèle mixte est relativement longue comparée à la régression linéaire multiple. En effet, alors que les approches AG-2 et AG-MM ont sensiblement le même nombre de générations, le temps d'exécution propre à l'algorithme (une fois le temps nécessaire à l'initialisation enlevé) avec modèle mixte (AG-MM) est nettement supérieur (10 secondes pour AG-2 vs. 7 min. pour AG-MM). Cependant, quelque soit l'approche que nous avons proposée, les temps d'exécution restent raisonnables comparés à la durée de collecte et de pré-traitement des données.
La méthode sparse PLS est la plus performante sur ce jeu de données en terme d'erreur de prédiction mais elle est également la plus longue et sélectionnent trop de variables.

5.4 Généralisation à d'autres données

Ces deux nouvelles approches que nous avons proposées (AG-2 et AG-MM) permettant d'améliorer les résultats sur les données réelles, nous nous posons la question de l'intérêt de ces approches sur d'autres données et en particulier leurs performances sur les données pseudo-réelles de QTLMAS.
Nous avons donc réalisé différentes expérimentations et nos premières analyses montrent que le processus de gestion du sur-apprentissage ne semble pas améliorer les résultats. La dimension des données étant plus faible, l'impact du sur-apprentissage est probablement moins important que sur les données réelles. Contrairement aux données réelles, l'algorithme, en convergeant, continue à améliorer les résultats et le sur-apprentissage semble donc être moins important sur ce jeu de données pseudo-réelles.
De même, nous avons réalisé des expérimentations en utilisant l'algorithme génétique combiné à un modèle mixte (AG-MM) à la place de l'algorithme génétique combiné à une régression multiple (AG). Les expérimentations montrent que l'intégration des relations familiales (AG-MM) ne semble pas améliorer les résultats

obtenus par l'approche basée sur une régression multiple. Les relations familiales dans ce jeu de données ne semblent donc pas importantes au point de justifier la pertinence d'utiliser un modèle de régression plus complexe.
Ces analyses montrent également la difficulté de générer des données simulées ayant le même comportement que des données réelles. En effet, simuler des relations familiales cohérentes avec une matrice de SNPs est très difficile, ce qui peut aussi expliquer la non pertinence de l'utilisation du modèle mixte sur les données pseudo-réelles.

5.5 Conclusion

Dans ce chapitre, nous avons proposé une application de notre approche basée sur une sélection de variables à l'aide d'une méthode d'optimisation combinatoire utilisant une évaluation par régression multiple, sur données réelles. Les performances sur ces données étant moins bonnes que les méthodes classiques, nous avons étudié deux axes d'amélioration : la prise en compte du problème du sur-apprentissage d'une part et l'intégration des relations familiales d'autre part. Pour chaque axe, nous avons proposé une nouvelle méthode et évalué ses performances. La gestion du problème du sur-apprentissage (AG-2) permet d'améliorer nettement les résultats de notre approche (AG) de sorte à égaler les méthodes classiques (EN96 et L96). De plus, l'amélioration des performances lors du passage d'une régression linéaire multiple (AG) à un modèle mixte (AG-MM) montre l'importance de l'intégration des relations familiales sur ce jeu de données réelles. Cette approche intégrant les relations familiales permet d'obtenir de meilleurs résultats que l'approche proposée pour diminuer le sur-apprentissage (AG-2). Nous obtenons, grâce à cette dernière méthode proposée (AG-MM), des résultats statistiquement meilleurs que les méthodes classiques de la littérature (EN96, L96) sur cette application sur données réelles pour laquelle les relations familiales semblent donc jouer un rôle important.

Conclusion

Le travail que nous avons présenté dans ce mémoire est issu d'une problématique industrielle posée par la société *Gènes Diffusion*, spécialisée en génétique animale. L'objectif est de proposer une méthode permettant de sélectionner un sous-ensemble de variables pertinentes pour la prédiction d'un caractère quantitatif.

Nous avons introduit dans un premier temps le concept d'amélioration génétique ainsi que l'évolution des méthodes associées. Nous avons ensuite présenté notre problématique de sélection de variables en régression en grande dimension. Cette problématique faisant l'objet de nombreuses publications, nous avons étudié dans un premier temps les méthodes classiques de la littérature puis nous avons proposé de nouvelles approches. Nous reprenons ici nos principales contributions.

Nous avons suggéré dans cette thèse d'aborder notre problématique de sélection de variables en régression en grande dimension en combinant une approche d'optimisation combinatoire pour sélectionner des sous-ensembles de variables et une méthode statistique permettant d'évaluer ces sous-ensembles. Afin de valider cette approche, nous avons proposé dans le Chapitre 3 une première méthode combinant une métaheuristique à solution unique, la recherche locale itérée, avec une régression linéaire multiple. Nous avons étudié les différents opérateurs de l'algorithme de recherche locale itérée afin de l'adapter à notre problématique, et notamment la fonction objectif, qui permet de définir ce qu'est une bonne sélection de variable. Nous avons comparé cette méthode aux méthodes classiques de la littérature (elastic-net, lasso, sparse-PLS). Nous obtenons des résultats très encourageants, que ce soit au niveau des variables sélectionnées ou de la qualité des prédictions. Cette approche nous permet de valider la pertinence de combiner une méthode d'optimisation combinatoire et une régression pour répondre à notre problématique.

Afin d'améliorer les résultats que nous avons obtenus, nous proposons dans le Chapitre 4 une seconde approche basée sur une métaheuristique à population de solutions, l'algorithme génétique, permettant une meilleure exploration de l'espace de recherche. Cette nouvelle approche améliore nettement les résultats de la précédente que ce soit sur données simulées ou pseudo-réelles. De plus, sur données simulées, mais également sur le jeu de données du workshop QTLMAS 2011, nous obtenons de meilleurs résultats que les approches classiques de la littérature limitées en nombre de variables sélectionnées auxquelles nous nous comparons.

Étant donné les résultats intéressants obtenus sur données simulées et pseudo-réelles, nous proposons dans le Chapitre 5 une application sur données réelles issues d'un projet national en partenariat avec la société *Gènes Diffusion*. Sur cette application, les approches proposées dans les chapitres 3 et 4 ne se sont pas avérées aussi performantes que les approches classiques de la littérature. Nous proposons

donc deux nouvelles méthodes. D'une part, nous modifions le choix de la solution finale de l'algorithme de manière à réduire le sur-apprentissage, particulièrement visible sur cette application en raison du nombre important de variables (marqueurs) étudiées comparé au nombre d'animaux disponibles ($n \ll p$). Cette gestion du sur-apprentissage grâce à l'extraction d'un échantillon de test permettant de sélectionner la solution finale de l'algorithme, améliore nettement les résultats de notre précédente approche sur cette application. D'autre part, une caractéristique importante des données de génomiques animales est la présence de relations familiales connues entre les animaux étudiés. Notre approche étant généralisable à d'autres modèles de régression, nous proposons d'intégrer les relations familiales à notre approche par l'intermédiaire d'un modèle mixte. La modification de la fonction d'évaluation, c'est-à-dire le remplacement de la régression linéaire multiple par un modèle mixte, permet d'améliorer les performances de la méthode. Cette approche combinant un algorithme génétique et un modèle mixte est l'approche la plus performante que nous avons proposée. Elle permet, sur cette application sur données réelles, d'obtenir de meilleurs résultats que les méthodes classiques de la littérature.

Ainsi, cette thèse a permis d'une part de valider l'approche combinant optimisation combinatoire et statistiques, et s'est posé la question de la fonction objectif à utiliser. L'importance de la prise en compte des relations familiales a également été mise en évidence sur une application sur données réelles.

Les perspectives de ce travail concernent différents points.

Dans le dernier chapitre, nous avons proposé indépendamment de traiter le problème du sur-apprentissage d'une part et d'intégrer les relations familiales à l'aide d'un modèle mixte d'autre part. Une perspective directe serait l'intégration du processus de gestion du sur-apprentissage à notre approche par modèle mixte. En effet, lors de l'utilisation du modèle mixte nous avons choisi peu de générations suite à quelques expérimentations. Cependant, l'utilisation du processus de gestion du sur-apprentissage permettrait de déterminer plus finement le nombre optimal de générations. La détermination du critère d'arrêt serait donc intégrée à l'algorithme et ne nécessiterait donc plus d'expérimentations lors de l'application de l'approche sur un nouveau jeu de données.

L'évaluation des solutions par un modèle mixte augmente nettement le temps d'exécution de l'algorithme comparé à une évaluation par régression multiple. Une alternative pourrait donc consister à évaluer les solutions de l'algorithme avec une régression multiple au début de la recherche puis affiner ensuite la recherche par l'intermédiaire du modèle mixte.

Dans les approches que nous avons proposées deux méthodes d'évaluation des sous-ensembles de variables sélectionnées ont été proposées : une régression linéaire

multiple et un modèle mixte. Nous pourrions également envisager une évaluation à l'aide d'approches bayésiennes, présentées en section 2.3.5, l'efficacité de ces méthodes ayant été montrée dans de nombreuses applications.

Nous avons vu lors de l'étude du problème du sur-apprentissage que le choix de la solution finale de l'algorithme a une influence non négligeable sur les performances de l'approche proposée. Une autre possibilité ici serait de se baser sur une connaissance biologique liée aux données : le déséquilibre de liaison. Si deux variables ont un fort déséquilibre de liaison (elles font donc parties d'un même haplotype), sélectionner l'une ou l'autre n'aura *a priori* pas d'impact sur la qualité des prédictions. Dans notre approche basée sur l'algorithme génétique, nous avons constaté que les solutions de la population finale avaient tendance à conserver des variables en fort déséquilibre de liaison. Plutôt que de conserver un modèle uniquement basé sur les variables sélectionnées par une seule solution de la population, nous pourrions extraire les variables sélectionnées par toutes les solutions de la population, et lorsque deux variables sont en fort déséquilibre de liaison n'en garder qu'une seule. Le sous-ensemble final contiendrait donc toutes les variables potentiellement intéressantes conservées par l'algorithme.

Les haplotypes, groupes de variables en fort déséquilibre de liaison, constituent une connaissance intéressante sur les données qui pourrait également être utilisée au cours de l'algorithme. Deux possibilités sont alors envisageables pour prendre en compte ces haplotypes. Le voisinage d'une solution pourrait être modifié de sorte à réduire l'espace de recherche. En effet, au cours de l'algorithme, si une variable est conservée, il ne semble *a priori* pas intéressant de conserver une variable avec laquelle elle est en fort déséquilibre de liaison. Il n'est donc pas nécessaire de tester l'inclusion de cette variable dans le modèle ce qui permettrait de diminuer le nombre de solutions voisines explorées. Nous pourrions également étudier directement les haplotypes, comme le font certaines approches de la littérature [Meuwissen 2001], plutôt que de manipuler des marqueurs (SNPs). La difficulté ici consiste à déterminer le marqueur qui sera représentatif de l'haplotype. Cette approche permettrait également de diminuer l'espace de recherche.

Bibliographie

[Akaike 1974] H. Akaike. *A new look at the statistical model identification.* IEEE Transactions on Automatic Control, vol. 19, no. 6, pages 716–723, 1974. (Cité en pages 57 et 67.)

[Al-ani 2005] A. Al-ani. *Ant Colony Optimization for Feature Subset Selection.* In Proceedings of World Academy of Science, Engineering and Technology, pages 35–38, 2005. (Cité en page 59.)

[Alba 2007] E. Alba, J. Garcia-Nieto, L. Jourdan et E.-G. Talbi. *Gene Selection in Cancer Classification using PSO-SVM and GA-SVM Hybrid Algorithms.* Congress on Evolutionary Computation, Singapor : Singapore (2007), 2007. (Cité en pages 58 et 59.)

[Amaldi 1997] E. Amaldi et V. Kann. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. 1997. (Cité en page 49.)

[Ansorge 2009] W. J. Ansorge. *Next-generation DNA sequencing techniques.* New Biotechnology, vol. 25, no. 4, pages 195–203, Avril 2009. (Cité en page 18.)

[Bagnato 2012] A. Bagnato et A. Rosati. *Animal Selection : The Genomics Revolution.* Animal Frontiers, vol. 2, no. 1, pages 1–2, Janvier 2012. (Cité en pages 22 et 23.)

[Bar-hen 2005] A. Bar-hen, J.-J. Daudin et S. Robin. *Comparaisons multiples pour les microarrays.* Journal de la Societe Francaise de Statistique, vol. 146, no. 1-2, 2005. (Cité en page 35.)

[Birattari 2004] M. Birattari. *On the estimation of the expected performance of a metaheuristic on a class of instances : How many instances, how many runs ?* 2004. (Cité en page 63.)

[Boulesteix 2007] A.-L. Boulesteix et K. Strimmer. *Partial least squares : a versatile tool for the analysis of high-dimensional genomic data.* Briefings in bioinformatics, vol. 8, no. 1, pages 32–44, Janvier 2007. PMID : 16772269. (Cité en page 38.)

[Broadhurst 1997] D. Broadhurst, R. Goodacre, A. Jones, J. J. Rowland et D. B. Kell. *Genetic algorithms as a method for variable selection in multiple linear regression and partial least squares regression, with applications to pyrolysis mass spectrometry.* Analytica Chimica Acta, vol. 348, no. 1-3, pages 71–86, Août 1997. (Cité en pages 57, 58, 59 et 131.)

[Cahon 2004] S. Cahon, N. Melab et E.-G. Talbi. *ParadisEO : A Framework for the Reusable Design of Parallel and Distributed Metaheuristics.* Journal of Heuristics, vol. 10, no. 3, pages 357–380, Mai 2004. (Cité en page 64.)

[Calus 2010] M. P. L. Calus. *Genomic Breeding Value Prediction : Methods and Procedures.* animal, vol. 4, no. 02, pages 157–164, 2010. (Cité en page 46.)

[Cerny 1985] V. Cerny. *Thermodynamical approach to the traveling salesman problem : An efficient simulation algorithm.* Journal of Optimization Theory and Applications, vol. 45, no. 1, pages 41–51, Janvier 1985. (Cité en page 51.)

[Chuang 2009] L.-Y. Chuang, C.-H. Yang et C.-H. Yang. *Tabu search and binary particle swarm optimization for feature selection using microarray data.* Journal of computational biology : a journal of computational molecular cell biology, vol. 16, no. 12, pages 1689–1703, Décembre 2009. PMID : 20047491. (Cité en page 59.)

[Corne 2012] D. Corne, C. Dhaenens et L. Jourdan. *Synergies between operations research and data mining : The emerging use of multi-objective approaches.* European Journal of Operational Research, vol. 221, no. 3, pages 469–479, Septembre 2012. (Cité en page 49.)

[Croiseau 2010] P. Croiseau, C. Colombani, A. Legarra, F. Guillaume, S. Fritz, A. Baur, R. Dassonneville, C. Patry, C. Robert-Granie et V. Ducrocq. *Improving Genomic Evaluation Strategies In Dairy Cattle Through SNP Pre-Selection.* Liepzig, Germany, 2010. (Cité en page 38.)

[Darwin 1859] C. Darwin. On the origin of the species by means of natural selection : Or, the preservation of favoured races in the struggle for life. John Murray, 1859. (Cité en page 53.)

[De Los Campos 2013] G. De Los Campos, J. M. Hickey, R. Pong-Wong, H. D. Daetwyler et M. P. L. Calus. *Whole-Genome Regression and Prediction Methods Applied to Plant and Animal Breeding.* Genetics, vol. 193, no. 2, pages 327–345, Février 2013. (Cité en pages 43, 46 et 47.)

[Delattre 2012] Maud Delattre, Marc Lavielle et Marie-Anne Poursat. *BIC selection procedures in mixed effects models.* Mai 2012. (Cité en page 138.)

[Dorigo 1992] M. Dorigo. *Optimization, Learning and Natural Algorithms (in Italian).* 1992. (Cité en page 53.)

[Ducrocq 1990] V. Ducrocq. *Les techniques d'evaluation genetique des bovins laitiers.* INRA, Prod. Anim., no. 3 (1), pages 3–16, 1990. (Cité en page 42.)

[Duval 2009] B. Duval, J.-K. Hao et J. C. Hernandez Hernandez. *A memetic algorithm for gene selection and molecular classification of cancer.* In Proceedings of the 11th Annual conference on Genetic and evolutionary computation, GECCO '09, pages 201–208, New York, NY, USA, 2009. ACM. (Cité en pages 58 et 59.)

[Efron 2004] B. Efron, T. Hastie, L. Johnstone et R. Tibshirani. *Least angle regression.* The Annals of Statistics, vol. 32, no. 2, pages 407–499, 2004. Mathematical Reviews number (MathSciNet) : MR2060166 ; Zentralblatt MATH identifier : 02100802. (Cité en page 37.)

[Elkan 1997] C. Elkan. *Boosting And Naive Bayesian Learning.* Rapport technique, 1997. (Cité en page 58.)

[Elsen 2012] J.-M. Elsen, S. Tesseydre, O. Filangi, P. Le Roy et O. Demeure. *XVth QTLMAS : simulated dataset*. BMC Proceedings, vol. 6, no. Suppl 2, page S1, Mai 2012. (Cité en page 29.)

[Emmanouilidis 2000] C. Emmanouilidis, A. Hunter et J. MacIntyre. *A Multiobjective Evolutionary Setting for Feature Selection and a Commonality-Based Crossover Operator*. In in proc. of congress on evolutionary computation, pages 309–316, 2000. (Cité en pages 97, 103, 115, 125 et 137.)

[Erbe 2012] M. Erbe, B.J. Hayes, L.K. Matukumalli, S. Goswami, P.J. Bowman, C.M. Reich, B.A. Mason et M.E. Goddard. *Improving accuracy of genomic predictions within and between dairy cattle breeds with imputed high-density single nucleotide polymorphism panels*. Journal of Dairy Science, vol. 95, no. 7, pages 4114–4129, Juillet 2012. (Cité en pages 46 et 48.)

[Foulley 2002] J.-L. Foulley, C. Delmas et C. Robert-Granie. *Methodes du maximum de vraisemblance en modele lineaire mixte*, 2002. (Cité en page 138.)

[Garrick 2009] D. J. Garrick, J. F. Taylor et R. L. Fernando. *Deregressing estimated breeding values and weighting information for genomic regression analyses*. Genetics Selection Evolution, vol. 41, no. 1, page 55, Décembre 2009. PMID : 20043827. (Cité en page 26.)

[Gianola 2009] D. Gianola, G. De Los Campos, W. G. Hill, E. Manfredi et R. Fernando. *Additive Genetic Variability and the Bayesian Alphabet*. Genetics, vol. 183, no. 1, pages 347–363, Janvier 2009. PMID : 19620397. (Cité en page 45.)

[Glover 1989] F. Glover. *Tabu Search–Part I*. ORSA Journal on Computing, vol. 1, no. 3, pages 190–206, Juin 1989. (Cité en page 51.)

[Goldberg 1989] D. E. Goldberg. Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, 1989. (Cité en page 102.)

[Grefenstette 1992] J. Grefenstette. *Genetic Algorithms for Changing Environments*. In Parallel Problem Solving from Nature 2, pages 137–144. Elsevier, 1992. (Cité en page 109.)

[Habier 2007] D. Habier, R. L. Fernando et J. C. M. Dekkers. *The Impact of Genetic Relationship Information on Genome-Assisted Breeding Values*. Genetics, vol. 177, no. 4, pages 2389–2397, Janvier 2007. (Cité en pages 35 et 46.)

[Habier 2009] D. Habier, R. L. Fernando et J. C. M. Dekkers. *Genomic Selection Using Low-Density Marker Panels*. Genetics, vol. 182, no. 1, pages 343–353, Janvier 2009. (Cité en page 28.)

[Habier 2011] D. Habier, R. Fernando, K. Kizilkaya et D. J. Garrick. *Extension of the bayesian alphabet for genomic selection*. BMC Bioinformatics, vol. 12, no. 1, page 186, Mai 2011. PMID : 21605355. (Cité en pages 45, 46 et 48.)

[Hamon 2011] Julie Hamon, Clarisse Dhaenens, Julien Jacques et Gael Even. *Combining combinatorial optimization and statistics to mine high-throughput genotyping data*. In JOBIM - Journees Ouvertes Biologie Informatique Mathematiques, Juin 2011. (Cité en page 61.)

[Hamon 2012] Julie Hamon, Clarisse Dhaenens, Julien Jacques et Gael Even. *Cooperation entre Optimisation Combinatoire et Statistiques pour la Selection animale*. In 13e congres annuel de la Societe francaise de Recherche Operationnelle et d'Aide a la Decision, Février 2012. (Cité en page 61.)

[Hamon 2013a] Julie Hamon, Clarisse Dhaenens, Gael Even et Julien Jacques. *Feature selection in high dimensional regression problems for genomic*. In Tenth International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics, Juin 2013. (Cité en page 62.)

[Hamon 2013b] Julie Hamon, Clarisse Dhaenens, Gael Even et Julien Jacques. *Modeles mixtes en genetique animale : selection de variables par optimisation combinatoire*. In 45eme Journees De Statistiques, Mai 2013. (Cité en page 95.)

[Hans 2007] C. Hans, A. Dobra et M. West. *Shotgun stochastic search for 'large p' regression*. Journal of the American Statistical Association, 2007. (Cité en pages 57 et 59.)

[Hastie 2009] T. Hastie, R. Tibshirani et J. Friedman. The elements of statistical learning - data mining, inference, and prediction, second edition. 2009. (Cité en pages 34 et 67.)

[Hawkins 2004] D.M. Hawkins. *The Problem of Overfitting*. Journal of Chemical Information and Modeling, vol. 44, no. 1, pages 1–12, Janvier 2004. (Cité en page 129.)

[Hayes 2010] B. J. Hayes, J. Pryce, A. J. Chamberlain, P. J. Bowman et M. E. Goddard. *Genetic Architecture of Complex Traits and Accuracy of Genomic Prediction : Coat Colour, Milk-Fat Percentage, and Type in Holstein Cattle as Contrasting Model Traits*. PLoS Genet, vol. 6, no. 9, page e1001139, Septembre 2010. (Cité en page 46.)

[Henderson 1950] C. R. Henderson. *Estimation of genetic parameters*. Ann. Math. Stat, vol. 21, pages 309–310, 1950. (Cité en page 42.)

[Henderson 1963] C. R. Henderson. *Selection index and expected genetic advance*. Statistical genetics and plant breeding, no. 982, pages 141–63, 1963. (Cité en page 42.)

[Henderson 1975] C. R. Henderson. *Best Linear Unbiased Estimation and Prediction under a Selection Model*. Biometrics, vol. 31, No. 2., pages 423–447, 1975. (Cité en page 42.)

[Hernandez 2007] J. C. H. Hernandez, B. Duval et J.-K. Hao. *A genetic embedded approach for gene selection and classification of microarray data*. In Proceedings of the 5th European conference on Evolutionary computation, machine learning and data mining in bioinformatics, EvoBIO'07, pages 90–101, Berlin, Heidelberg, 2007. Springer-Verlag. (Cité en page 59.)

[Hoerl 1970] A. E. Hoerl et R. W. Kennard. *Ridge Regression : Biased Estimation for Nonorthogonal Problems*. Technometrics, vol. 12, no. 1, pages 55–67, 1970. (Cité en pages 36 et 48.)

[Holland 1975] J. H. Holland. Adaptation in natural and artificial systems : An introductory analysis with applications to biology, control, and artificial intelligence, volume viii. U Michigan Press, Oxford, England, 1975. (Cité en page 53.)

[Horne 2004] B. D. Horne et N. J. Camp. *Principal component analysis for selection of optimal SNP-sets that capture intragenic genetic variation.* Genetic Epidemiology, vol. 26, no. 1, pages 11–21, 2004. (Cité en page 38.)

[Huerta 2006] E. B. Huerta, B. Duval et J.-K. Hao. *A hybrid GA/SVM approach for gene selection and classification of microarray data.* evoworkshops 2006, LNCS, vol. 3907, pages 34–44, 2006. (Cité en page 59.)

[Innis 1990] M. A. Innis et D. H. Gelfand. *PCR protocols : a guide to methods and applications.* pages xviii + 482 pp., 1990. (Cité en page 17.)

[Jirapech-Umpai 2005] T. Jirapech-Umpai et S. Aitken. *Feature selection and classification for microarray data analysis : Evolutionary methods for identifying predictive genes.* BMC bioinformatics, vol. 6, no. 1, page 148, 2005. (Cité en pages 58 et 59.)

[Jolliffe 1982] I. T. Jolliffe. *A Note on the Use of Principal Components in Regression.* Applied Statistics, vol. 31, no. 3, page 300, 1982. (Cité en pages 38 et 48.)

[Jourdan 2004] L. Jourdan, C. Dhaenens et E.-G. Talbi. *Linkage disequilibrium study with a parallel adaptive GA.* International Journal of Foundations of Computer Science, 2004. (Cité en page 59.)

[Jourdan 2009] L. Jourdan, M. Basseur et E.-G. Talbi. *Hybridizing exact methods and metaheuristics : A taxonomy.* European Journal of Operational Research, vol. 199, no. 3, pages 620–629, 2009. 3 3. (Cité en page 82.)

[Kapetanios 2005] G. Kapetanios. *Variable Selection using Non-Standard Optimisation of Information Criteria.* Working Paper 533, Queen Mary, University of London, School of Economics and Finance, 2005. (Cité en pages 56, 57 et 59.)

[Kennedy 1995] J. Kennedy et R. Eberhart. *Particle swarm optimization.* In , IEEE International Conference on Neural Networks, 1995. Proceedings, volume 4, pages 1942–1948 vol.4, 1995. (Cité en page 54.)

[Kirkpatrick 1983] S. Kirkpatrick, C. D. Gelatt et M. P. Vecchi. *Optimization by simulated annealing.* Science (New York, N.Y.), vol. 220, no. 4598, pages 671–680, Mai 1983. PMID : 17813860. (Cité en page 51.)

[Kohavi 1997] R. Kohavi et G. H. John. *Wrappers for feature subset selection.* Artif. Intell., vol. 97, no. 1-2, pages 273–324, Décembre 1997. (Cité en page 56.)

[Koivula 2012] M. Koivula, I. Stranden, G. Su et E. A. Mantysaari. *Different methods to calculate genomic predictions–Comparisons of BLUP at the single nucleotide polymorphism level (SNP-BLUP), BLUP at the individual level (G-BLUP), and the one-step approach (H-BLUP).* Journal of Dairy Science, vol. 95, no. 7, pages 4065–4073, Juillet 2012. (Cité en page 44.)

[Koza 1998] J. R Koza. Genetic programming. MIT Press, Cambridge, Mass, 1998. (Cité en page 53.)

[Lal 2006] T. N. Lal, O. Chapelle, J. Weston et A. Elisseeff. *Embedded Methods*. In I. Guyon, M. Nikravesh, S. Gunn et L. A. Zadeh, editeurs, Feature Extraction, numéro 207 de Studies in Fuzziness and Soft Computing, pages 137–165. Springer Berlin Heidelberg, Janvier 2006. (Cité en page 56.)

[Lebarbier 2006] E. Lebarbier et T. Mary-Huard. *Le critere BIC : fondements theoriques et interpretation*, 2006. (Cité en page 67.)

[Legarra 2009] A. Legarra, I. Aguilar et I. Misztal. *A relationship matrix including full pedigree and genomic information*. Journal of dairy science, vol. 92, no. 9, pages 4656–4663, Septembre 2009. PMID : 19700729. (Cité en pages 43, 44 et 48.)

[Li 2009] Y. Li, C. Willer, S. Sanna et G. Abecasis. *Genotype Imputation*. Annual Review of Genomics and Human Genetics, vol. 10, no. 1, pages 387–406, 2009. PMID : 19715440. (Cité en page 25.)

[Long 2007] N. Long, D. Gianola, G. J. M. Rosa, K. A. Weigel et S. Avendano. *Machine learning classification procedure for selecting SNPs in genomic selection : application to early mortality in broilers*. Journal of animal breeding and genetics, vol. 124, no. 6, pages 377–389, Décembre 2007. PMID : 18076475. (Cité en pages 58 et 59.)

[Long 2011] N. Long, D. Gianola, G. J.M Rosa et K. A Weigel. *Dimension reduction and variable selection for genomic selection : application to predicting milk yield in Holsteins*. Journal of Animal Breeding and Genetics, vol. 128, no. 4, pages 247–257, Août 2011. (Cité en pages 38 et 48.)

[Lourenco 2001] H. R. Lourenco, O. C. Martin et T. Stutzle. *Iterated Local Search*. arXiv e-print math/0102188, Février 2001. In "Handbook of Metaheuristics", Ed. F. Glover and G. Kochenberger, ISORMS 57, p 321–353 (2002), Kluwer. (Cité en page 52.)

[Mai 2010] M. D. Mai, G. Sahana, F. B. Christiansen et B. Guldbrandtsen. *A genome-wide association study for milk production traits in Danish Jersey cattle using a 50K single nucleotide polymorphism chip*. Journal Of Animal Science, vol. 88, pages 3522–3528, 2010. (Cité en pages 28 et 35.)

[Marchini 2010] J. Marchini et B. Howie. *Genotype imputation for genome-wide association studies*. Nature Reviews Genetics, vol. 11, no. 7, pages 499–511, Juin 2010. (Cité en page 25.)

[Mardis 2008] E. R. Mardis. *Next-Generation DNA Sequencing Methods*. Annual Review of Genomics and Human Genetics, vol. 9, no. 1, pages 387–402, 2008. PMID : 18576944. (Cité en page 18.)

[Martin 1991] O. Martin, S. W. Otto et E. W. Felten. *Large-Step Markov Chains for the Traveling Salesman Problem*. Complex Systems, vol. 5, pages 299–326, 1991. (Cité en page 52.)

[Meiri 2006] R. Meiri et J. Zahavi. *Using simulated annealing to optimize the feature selection problem in marketing applications.* European Journal of Operational Research, vol. 171, no. 3, pages 842–858, Juin 2006. (Cité en pages 56, 57 et 59.)

[Meuwissen 2001] T. H. E. Meuwissen, B. J. Hayes et M. E. Goddard. *Prediction of Total Genetic Value Using Genome-Wide Dense Marker Maps.* Genetics Society of America, 2001. (Cité en pages 20, 39, 44, 47, 48 et 145.)

[Misztal 2002] I. Misztal, S. Tsuruta, T. Strabel, B. Auvray, T. Druet et D. H. Lee. *BLUPF90 and related programs (BGF90).* pages 1–2. Institut National de la Recherche Agronomique (INRA), 2002. (Cité en page 138.)

[Moser 2009] G. Moser, B. Tier, R. E. Crump, M. S. Khatkar et H. W. Raadsma. *A comparison of five methods to predict genomic breeding values of dairy bulls from genome-wide SNP markers.* Genetics, Selection, Evolution : GSE, vol. 41, no. 1, page 56, Décembre 2009. PMID : 20043835 PMCID : PMC2814805. (Cité en page 47.)

[Muni 2006] D. P. Muni, N. R. Pal et J. Das. *Genetic programming for simultaneous feature selection and classifier design.* IEEE Transactions on Systems, Man, and Cybernetics, Part B : Cybernetics, vol. 36, no. 1, pages 106–117, Février 2006. (Cité en page 59.)

[Ogutu 2012] J. O. Ogutu, T. Schulz-Streeck et H.-P. Piepho. *Genomic selection using regularized linear regression models : ridge regression, lasso, elastic net and their extensions.* BMC Proceedings, vol. 6, no. Suppl 2, page S10, 2012. (Cité en pages 28 et 37.)

[Oh 2004] I. S. Oh, J. S. Lee et B. R. Moon. *Hybrid genetic algorithms for feature selection.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 11, pages 1424–1437, Novembre 2004. (Cité en page 59.)

[Ostersen 2011] T. Ostersen, O. F. Christensen, M. Henryon, B. Nielsen, G. Su et P. Madsen. *Deregressed EBV as the response variable yield more reliable genomic predictions than traditional EBV in pure-bred pigs.* Genetics Selection Evolution, vol. 43, no. 1, page 38, Novembre 2011. PMID : 22070746. (Cité en page 26.)

[Pal 2006] S. K. Pal, S. Bandyopadhyay et S. S. Ray. *Evolutionary computation in bioinformatics : a review.* IEEE Transactions on Systems, Man, and Cybernetics, Part C : Applications and Reviews, vol. 36, no. 5, pages 601–615, 2006. (Cité en page 53.)

[Papadimitriou 1976] C. H. Papadimitriou. *The complexity of combinatorial optimization problems.* PhD thesis, Princeton University, Princeton, NJ, USA, 1976. AAI7704795. (Cité en page 50.)

[Pearson 1901] K. Pearson. *LIII. On lines and planes of closest fit to systems of points in space.* Philosophical Magazine Series 6, vol. 2, no. 11, pages 559–572, 1901. (Cité en page 38.)

[Peng 2003] S. Peng. *Molecular classification of cancer types from microarray data using the combination of genetic algorithms and support vector machines.* FEBS Letters, vol. 555, no. 2, pages 358–362, Décembre 2003. (Cité en pages 58 et 59.)

[Phuong 2005] T. M. Phuong, Z. Lin et R. B. Altman. *Choosing SNPs using feature selection.* Proceedings / IEEE Computational Systems Bioinformatics Conference, CSB. IEEE Computational Systems Bioinformatics Conference, pages 301–309, 2005. PMID : 16447987. (Cité en page 59.)

[Pinheiro 2000] J. C. Pinheiro et D. M. Bates. *Linear Mixed-Effects Models : Basic Concepts and Examples.* In Mixed-Effects Models in Sand S-PLUS, Statistics and Computing, pages 3–56. Springer New York, Janvier 2000. (Cité en page 41.)

[Pudil 1998] P. Pudil et J. Hovovicova. *Novel methods for subset selection with respect to problem knowledge.* IEEE Intelligent Systems and their Applications, vol. 13, no. 2, pages 66–74, 1998. (Cité en page 55.)

[Rendel 1950] J. M. Rendel et A. Robertson. *Estimation of genetic gain in milk yield by selection in a closed herd of dairy cattle.* Journal of Genetics, vol. 50, no. 1, pages 1–8, Juin 1950. (Cité en page 16.)

[Robert 2004] C. P. Robert et G. Casella. Monte carlo statistical methods. Springer, New York, 2004. (Cité en page 42.)

[Schwarz 1978] G. Schwarz. *Estimating the Dimension of a Model.* The Annals of Statistics, vol. 6, no. 2, pages 461–464, Mars 1978. Mathematical Reviews number (MathSciNet) : MR468014 ; Zentralblatt MATH identifier : 0379.62005. (Cité en pages 57 et 67.)

[Shah 2004] S. C. Shah et A. Kusiak. *Data mining and genetic algorithm based gene/SNP selection.* Artificial intelligence in medicine, vol. 31, no. 3, pages 183–196, Juillet 2004. PMID : 15302085. (Cité en page 59.)

[Shrimpton 1988] A. E. Shrimpton et A. Robertson. *The Isolation of Polygenic Factors Controlling Bristle Score in Drosophila Melanogaster. II. Distribution of Third Chromosome Bristle Effects within Chromosome Sections.* Genetics, vol. 118, no. 3, pages 445–459, Mars 1988. PMID : 17246417. (Cité en pages 20 et 44.)

[Szyda 2008] J. Szyda, E. Ptak, J. Komisarek et A. Zarnecki. *Practical application of daughter yield deviations in dairy cattle breeding.* Journal of applied genetics, vol. 49, no. 2, pages 183–191, 2008. PMID : 18436992. (Cité en page 26.)

[Szydlowski 2011] M. Szydlowski et P. Paczynska. *QTLMAS 2010 : simulated dataset.* BMC Proceedings, vol. 5, no. Suppl 3, page S3, Mai 2011. (Cité en page 29.)

[Talbi 2009] E.-G. Talbi. Metaheuristics. John Wiley & Sons, Inc., Hoboken, NJ, USA, Juin 2009. (Cité en pages 49 et 101.)

[Tenenhaus 1998] M. Tenenhaus. La regression PLS : theorie et pratique. Editions TECHNIP, Janvier 1998. (Cité en page 48.)

[Thomsen 2001] H. Thomsen, N. Reinsch, N. Xu, C. Looft, S. Grupe, C. Kuhn, G. A. Brockmann, M. Schwerin, B. Leyhe-Horn, S. Hiendleder, G. Erhardt, I. Medjugorac, I. Russ, M. Forster, B. Brenig, F. Reinhardt, R. Reents, J. Blumel, G. Averdunk et E. Kalm. *Comparison of estimated breeding values, daughter yield deviations and de-regressed proofs within a whole genome scan for QTL.* Journal of Animal Breeding and Genetics, vol. 118, no. 6, pages 357–370, 2001. (Cité en page 26.)

[Tibshirani 1994] R. Tibshirani. *Regression Shrinkage and Selection Via the Lasso.* Journal of the Royal Statistical Society, Series B, vol. 58, pages 267–288, 1994. (Cité en pages 36 et 48.)

[Tinos 2007] R. Tinos et S. Yang. *A self-organizing random immigrants genetic algorithm for dynamic optimization problems.* Genetic Programming and Evolvable Machines, vol. 8, no. 3, pages 255–286, Mai 2007. (Cité en page 113.)

[Usai 2009] M. G. Usai, M. E. Goddard et B. J. Hayes. *LASSO with cross-validation for genomic selection.* Genet Res (Camb)., vol. 91(6), pages 427–36., 2009. (Cité en page 37.)

[Usai 2010] M. G. Usai, M. E. Goddard et B. J. Hayes. *Using LASSO to estimate marker effects for Genomic Selection.* Italian Journal of Animal Science., 2010. (Cité en pages 28 et 37.)

[Ustunkar 2011] G. Ustunkar, S. Ozogur-Akyuz, G. W. Weber, C. M. Friedrich et Yesim Aydin Son. *Selection of representative SNP sets for genome-wide association studies : a metaheuristic approach.* Optimization Letters, Novembre 2011. (Cité en page 59.)

[VanRaden 2008] P. M. VanRaden. *Efficient Methods to Compute Genomic Predictions.* Journal of Dairy Science, vol. 91, no. 11, pages 4414–4423, Novembre 2008. (Cité en pages 43 et 48.)

[Waldron 2011] L. Waldron, M. Pintilie, M.-S. Tsao, F. A. Shepherd, C. Huttenhower et I. Jurisica. *Optimized application of penalized regression methods to diverse genomic data.* Bioinformatics (Oxford, England), vol. 27, no. 24, pages 3399–3406, Décembre 2011. PMID : 22156367. (Cité en page 37.)

[Wang 1994] C. S. Wang, J. J. Rutledge et D. Gianola. *Bayesian analysis of mixed linear models via Gibbs sampling with an application to litter size in Iberian pigs.* Genetics, Selection, Evolution : GSE, vol. 26, no. 2, pages 91–115, Avril 1994. PMID : null PMCID : PMC2709124. (Cité en page 42.)

[Whitley 2013] D. Whitley, A. Howe et D. Hains. *Greedy or Not ? Best Improving versus First Improving Stochastic Local Search for MAXSAT.* 2013. (Cité en page 66.)

[Wientjes 2013] Y. C. J. Wientjes, R. F. Veerkamp et M. P. L. Calus. *The Effect of Linkage Disequilibrium and Family Relationships on the Reliability of Genomic Prediction.* Genetics, vol. 193, no. 2, pages 621–631, Février 2013. PMID : 23267052. (Cité en page 39.)

[Wold 2004] H. Wold. *Partial Least Squares*. In Encyclopedia of Statistical Sciences. John Wiley & Sons, Inc., 2004. (Cité en page 38.)

[Xu 2007] S. Xu. *An empirical Bayes method for estimating epistatic effects of quantitative trait loci*. Biometrics, vol. 63, no. 2, pages 513–521, Juin 2007. PMID : 17688503. (Cité en page 37.)

[Xuan 2011] P. Xuan, M. Z. Guo, J. Wang, C. Y. Wang, X. Y. Liu et Y. Liu. *Genetic algorithm-based efficient feature selection for classification of pre-miRNAs*. Genetics and Molecular Research : GMR, vol. 10, no. 2, pages 588–603, 2011. PMID : 21491369. (Cité en page 59.)

[Zou 2005] H. Zou et T. Hastie. *Regularization and Variable Selection via the Elastic Net*. J. R. Statist. Soc. B, vol. 67, Part 2, pages 301–320, 2005. (Cité en pages 37 et 48.)

[Zwick 1986] W. R. Zwick et W. F. Velicer. *Comparison of five rules for determining the number of components to retain*. Psychological Bulletin, vol. 99, no. 3, pages 432–442, 1986. (Cité en page 38.)