

Prédiction génomique

Timothée Flutre (INRA)

01/02/2020

Résumé

Ce document a pour but d’explorer par simulation l’intérêt de la prédiction génomique en sélection artificielle d’espèces domestiquées. Suivant [Fisher \(1918\)](#), il se focalise sur les architectures génétiques additives d’un unique caractère continu.

Table des matières

1	Contexte	1
2	Introduction	2
3	Ecrire le modèle	3
3.1	Notations	3
3.2	Vraisemblances d’extrêmes d’architecture génétique additive	4
4	Simuler des données	6
4.1	Génotypes	6
4.2	Effets additifs des allèles	7
4.3	Erreurs puis phénotypes	7
5	Réaliser l’inférence	8
5.1	Visualisation graphique	8
5.2	SNP à SNP (“GWAS”)	10
5.3	Tous les SNPs conjointement (“ridge”)	10
6	Evaluer les résultats	11
7	Autres points importants	14
7.1	Eviter le sur-ajustement	14
7.2	Intermédiaires d’architecture génétique additive	16
7.3	Au-delà de l’architecture génétique additive	16
8	Explorer les simulations possibles	17
9	Analyser de vrais jeux de données disponibles	17
10	Perspectives	17
11	Références	18
12	Annexe	18

1 Contexte

Ce document fait partie de l’atelier “Prédiction Génomique” organisé et animé par Jacques David et Timothée Flutre depuis 2015, avec l’aide de Julie Fiévet et Philippe Brabant, à [Montpellier SupAgro](#) dans le cadre de l’option [APIMET](#) (Amélioration des Plantes et Ingénierie végétale Méditerranéennes et Tropicales) couplée

à la spécialité SEPMET (Semences Et Plants Méditerranéens Et Tropicaux) du [Master 3A](#) (Agronomie et Agroalimentaire), et de la spécialisation [PIST](#) du [Cursus Ingénieur d'AgroparisTech](#).

Le copyright appartient à Montpellier SupAgro et à l'Institut National de la Recherche Agronomique. Le contenu du répertoire est sous license [Creative Commons Attribution-ShareAlike 4.0 International](#). Veuillez en prendre connaissance et vous y conformer (contactez les auteurs en cas de doute).

Les versions du contenu sont gérées avec le logiciel git, et le dépôt central est hébergé sur [GitHub](#).

Il est recommandé d'avoir déjà lu attentivement le document “Premiers pas” de l'atelier.

De plus, ce document nécessite de charger des paquets additionnels (ceux-ci doivent être installés au préalable sur votre machine, via `install.packages("pkg")`):

```
suppressPackageStartupMessages(library(MM4LMM))
suppressPackageStartupMessages(library(rrBLUP))
```

Il est également utile de savoir combien de temps est nécessaire pour exécuter tout le code R de ce document (voir l'annexe):

```
t0 <- proc.time()
```

2 Introduction

Le modèle fondamental de la génétique quantitative (voir les références en fin de document) considère une population d'individus plus ou moins génétiquement apparentés. Le terme “individu” est utilisé ici comme synonyme de “génotype”, c'est-à-dire pour distinguer deux organismes biologiques d'une même espèce, animale ou végétale, ayant des génomes “suffisamment” différents car étant séparés par au moins un événement de reproduction sexuée (c'est-à-dire que ce ne sont pas des clones, même si ceux peuvent différer par quelques mutations somatiques).

Pour chaque individu i parmi les N que compte la population, on écrit:

$$y_i = g_i + \epsilon_i \quad (1)$$

où:

- y_i : valeur phénotypique de l'individu i pour le caractère d'intérêt, considérée ici comme continu;
- g_i : **valeur génotypique** de l'individu i , en unité du phénotype, interprétée comme étant le phénotype moyen de l'individu s'il était cloné dans tous les environnements possibles;
- ϵ_i : composante non-génétique pour l'individu i (“déviations environnementales”).

Si l'on suppose que les valeurs génotypique et composante non-génétique ne sont pas corrélées, alors la variance phénotypique est égale à $\sigma_p^2 = \sigma_g^2 + \sigma_\epsilon^2$, et l'**héritabilité au sens large** est définie comme étant $H^2 = \frac{\sigma_g^2}{\sigma_g^2 + \sigma_\epsilon^2}$.

La valeur génotypique peut également se décomposer en **composantes additive, de dominance et d'épistasie**: $g_i = a_i + d_i + \zeta_i$. La première est particulièrement importante car elle correspond à la part de la valeur génotypique qui est héritable, c'est-à-dire transmissible à la descendance. C'est donc surtout la **valeur génotypique additive** (*breeding value*) qui est d'intérêt pour le sélectionneur. En effet, celui-ci peut s'en servir comme critère pour trier les individus de son programme de sélection et identifier ceux à utiliser préférentiellement comme géniteurs.

On suppose généralement aussi que ces composantes ne sont pas corrélées, et donc $\sigma_g^2 = \sigma_a^2 + \sigma_d^2 + \sigma_\zeta^2$. Ceci amène à définir l'**héritabilité au sens strict**: $h^2 = \frac{\sigma_a^2}{\sigma_g^2 + \sigma_\epsilon^2}$.

Sur le plan fondamental, l'un des buts de la génétique quantitative est de quantifier la part de la variation phénotypique au sein de la population expliquée par la composante génétique, c'est-à-dire d'estimer l'héritabilité. Sur le plan appliqué, l'un des buts consiste à quantifier la valeur génotypique de chaque individu.

Le même modèle que (1) mais en notation matricielle:

$$\mathbf{y} = \mathbf{g} + \boldsymbol{\epsilon} \quad (2)$$

- G : matrice de variance-covariance $N \times N$ des valeurs génotypiques;
- R : matrice de variance-covariance $N \times N$ des composantes non-génétiques.

Sous certaines hypothèses (panmixie, etc), la matrice G se décompose aussi en contributions additives, de dominance et d'épistasie, même si les premières sont généralement les seules utilisées en pratique car contribuant majoritairement à la variance génétique. Dans ce cas, $G = \sigma_a^2 A$ où σ_a^2 est estimé et A , la matrice des relations génétiques additives, est calculée à partir de l'arbre généalogique (pédigrée) des individus. D'où le fait que l'on parle de matrice d'**apparentement** (*kinship*). De plus, la matrice R est généralement diagonale, telle que $R = \sigma_e^2 I$ où σ_e^2 est estimé simultanément à σ_a^2 , et I est la matrice identité.

Si l'on suppose que $\mathbf{g} \sim \mathcal{N}_N(\mathbf{0}, G)$ et $\boldsymbol{\epsilon} \sim \mathcal{N}_N(\mathbf{0}, R)$, alors $\hat{\mathbf{g}} = E[\mathbf{g}|\mathbf{y}] = G(G + R)^{-1}\mathbf{y}$ où $\hat{\mathbf{g}}$ est le **meilleur prédicteur linéaire sans biais** de \mathbf{g} (*best linear unbiased predictor*, BLUP) et $H = G(G + R)^{-1}$ est une généralisation matricielle de l'héritabilité.

Or il faut bien remarquer que la généalogie ne permet de calculer que la matrice d'apparentement attendue, celle-ci pouvant donc différer de la matrice d'apparentement réalisée. En effet, bien qu'en moyenne le coefficient d'apparentement (identité par descendance) entre un allèle d'un parent et un allèle de son enfant soit de 1/4, cette proportion varie le long du génome, à cause, entre autres, de l'échantillonnage mendélien des chromosomes et de la variation du taux de recombinaison le long des chromosomes. De plus, la généalogie seule ne permet pas d'identifier quelles régions du génome ont une variation génétique plus ou moins associée à la variation phénotypique, les fameux **locus influençant les traits quantitatifs** (*quantitative trait locus*, QTL).

Si maintenant on dispose d'information génomique pour l'individu i , par exemple ses génotypes $\{\mathbf{x}_i\}$ à un ensemble de P marqueurs, le modèle (1) devient:

$$y_i = g(\mathbf{x}_i) + \epsilon_i \quad (3)$$

où la fonction g correspond à l'**architecture génétique** du caractère d'intérêt, détaillée dans la section suivante. (L'erreur est différente du modèle précédent, mais gardons la même notation par simplicité.)

On peut n'utiliser les marqueurs que pour estimer la matrice d'apparentement plus précisément, mais on peut aussi les inclure explicitement dans le modèle comme variables explicatives et tenter d'estimer les effets de leurs allèles.

Avec le toujours plus grand débit des technologies de séquençage, il est fréquent qu'il y ait beaucoup plus de marqueurs que d'individus: $P \gg N$. Dans de tels cas, le modèle de régression multiple correspondant à l'extension de la régression simple présentée dans le document "Premiers Pas" ne donne plus de bonnes estimations des effets alléliques. La **vraisemblance** doit être **pénalisée** (on dit aussi **régularisée**). Cela se traduit par un **rétrécissement des estimations des effets** (*shrinkage*).

En plus des effets alléliques additifs à tous les marqueurs, explicitement incorporer dans le modèle les effets de dominance est un peu plus difficile, et explicitement incorporer les effets d'épistasie est impossible étant donné l'explosion combinatoire qui en résulte. Certains auteurs ont donc proposé d'autres types de modèles, dits "semi-/non-paramétriques", mais qui ne seront pas abordés ici.

Quoi qu'il en soit, une abondance d'articles existe sur ces sujets (voir les références listées en fin de document) et, pour se familiariser avec ces questions à moindre coût, rien de mieux que de faire des simulations !

3 Ecrire le modèle

3.1 Notations

De manière similaire au document "Premiers Pas":

- N : nombre d'individus (diploïdes, plus ou moins apparentés);
- i : indice indiquant le i -ème individu, donc $i \in \{1, \dots, N\}$;
- P : nombre de marqueurs génétiques de type SNP (*single nucleotide polymorphism*), tous supposés bi-alléliques;
- p : indice indiquant le p -ème SNP, donc $p \in \{1, \dots, P\}$;
- y_i : phénotype de l'individu i pour le caractère d'intérêt;
- μ : moyenne globale du phénotype des N individus;
- $x_{i,p}$: génotype de l'individu i au SNP p , codé comme le nombre de copie(s) de l'allèle minoritaire à ce SNP chez cet individu ($\forall i, p, x_{i,p} \in \{0, 1, 2\}$);
- X : matrice à N lignes et P colonnes contenant les génotypes de tous les individus à tous les SNPs, codés "en additif" (c'est-à-dire en nombre de copies d'un des deux allèles, souvent le minoritaire); les génotypes de l'individu i à tous les SNPs sont réunis dans le vecteur \mathbf{x}_i^T , et les génotypes du SNP p pour tous les individus sont réunis dans le vecteur \mathbf{x}_p ;
- β_p : effet additif de chaque copie de l'allèle compté du SNP p , en unité du phénotype; tous ces effets sont réunis dans le vecteur β ;
- a_i : valeur génotypique additive de l'individu i ;
- σ_a^2 : variance génétique additive;
- A : matrice $N \times N$ de variance-covariance des a_i 's, contenant les relations génétique additives entre les N individus deux-à-deux;
- ϵ_i : erreur pour l'individu i ;
- σ^2 : variance des erreurs.

3.2 Vraisemblances d'extrêmes d'architecture génétique additive

L'**architecture génétique** d'un caractère se définit comme étant la fonction mathématique modélisant la relation entre les génotypes des individus de la population et leurs phénotypes (*genotype-phenotype map*):

- à l'échelle de l'individu, son étude vise à découvrir quels sont les locus impliqués dans la construction d'un caractère donné et, de là, à décrypter les mécanismes moléculaires et développementaux sous-jacents;
- à l'échelle de la population, son étude vise à quantifier la part de variation phénotypique contributive par la variation génotypique aux locus impliqués dans la construction du caractère, et à expliquer son évolution.

Ces deux axes de recherche sont complémentaires, mais ce document se focalise sur le deuxième, de surcroît en se limitant au cas simple d'architectures génétiques additives et en considérant principalement deux cas extrêmes.

Dans le premier, un seul SNP a un effet non-nul, par exemple un SNP non-synonyme dans le seul gène causal. On parle alors de **caractère monogénique**. Donc, si l'on teste chaque SNP un par un à la manière du document "Premier Pas", on devrait pouvoir identifier ce SNP particulier:

$$\forall p, \mathbf{y} = \mathbf{1}\mu + \mathbf{x}_p\beta_p + \epsilon \text{ avec } \epsilon \sim \mathcal{N}_N(\mathbf{0}, \sigma^2 I) \quad (4)$$

La matrice de variance-covariance phénotypique vaut:

$$\text{Var}(\mathbf{y}) = \text{Var}(\epsilon) = \sigma^2 I \quad (5)$$

Mais par rapport au document précédent, il faut maintenant prendre en compte l'apparentement entre individus. En effet, des individus apparentés génétiquement ont plus de chance de partager des allèles aux locus causaux, et donc d'avoir des phénotypes similaires. La prise en compte de cette contribution génétique à la covariance phénotypique peut se faire en ajoutant un **effet aléatoire**, noté ici u_i pour l'individu i , avec K comme matrice de variance-covariance. Alors que, jusqu'à maintenant, seule la moyenne des phénotypes était modélisée, maintenant sa variance-covariance l'est aussi, et on peut écrire le **modèle mixte** suivant:

$$\forall p, \mathbf{y} = \mathbf{1}\mu + \mathbf{x}_p\beta_p + \mathbf{u} + \epsilon \text{ avec } \epsilon \sim \mathcal{N}_N(\mathbf{0}, \sigma^2 I) \text{ et } \mathbf{u} \sim \mathcal{N}_N(\mathbf{0}, \sigma_u^2 K) \quad (6)$$

où (σ_u^2, σ^2) sont appelés les **composants de la variance** (*variance components*).

En supposant $Cov(\mathbf{u}, \boldsymbol{\epsilon}) = 0$, on obtient:

$$Var(\mathbf{y}) = Var(\mathbf{u}) + Var(\boldsymbol{\epsilon}) = \sigma_u^2 K + \sigma^2 I \quad (7)$$

Si l'on connaît le pédigrée reliant tous les individus, il est possible de calculer les apparentements deux-à-deux attendus: $K = A_{\text{ped}}$. La fonction `kinship` du paquet R `kinship2` fait exactement cela. Sinon, on peut utiliser les génotypes aux marqueurs: $K = A_{\text{mark}}$ (plus de détails ci-après).

Ce modèle (6) a cependant le désavantage d'utiliser les marqueurs pour estimer l'apparentement (A_{mark}), tout en testant leurs effets un par un (les \mathbf{x}_p), un peu comme s'il voulait faire deux choses à la fois sans se décider entre utiliser les marqueurs un par un ou tous ensemble. Il existe bien certaines astuces, mais d'autres modèles plus élégants évitent d'utiliser deux fois la même information, en incluant explicitement tous les marqueurs dans la régression.

Passons donc au deuxième cas extrême d'architecture génétique additive, dans lequel tous les SNPs ont un effet non-nul. On parle alors de **caractère polygénique**. Comme il y a vraiment beaucoup de SNPs ($P \gg N$), l'hypothèse habituelle est que leurs allèles ont tous des effets très faibles, donc chercher à les estimer individuellement n'est pas une stratégie pertinente. Il vaut mieux tenter d'estimer leur effet global, par exemple en supposant qu'ils s'additionnent tous: $\forall i, \sum_{p=1}^P x_{ip} \beta_p = \mathbf{x}_i^T \boldsymbol{\beta}$. On parle alors d'architecture génétique **additive infinitésimale**. De plus, sans connaissance plus précise a priori, il est habituel de supposer que les effets alléliques sont tous indépendants les uns des autres (attention, les génotypes, eux, ne sont généralement pas indépendants à cause du déséquilibre de liaison). Au final, le modèle mixte s'écrit:

$$\mathbf{y} = \mathbf{1}\mu + X\boldsymbol{\beta} + \boldsymbol{\epsilon} \text{ avec } \boldsymbol{\epsilon} \sim \mathcal{N}_N(\mathbf{0}, \sigma^2 I) \text{ et } \boldsymbol{\beta} \sim \mathcal{N}_P(\mathbf{0}, \sigma_\beta^2 I) \quad (8)$$

En modélisation statistique, ce modèle est connu sous le nom de **régression d'arête** (*ridge regression*). Dans la régression linéaire classique, maximiser la vraisemblance revient à minimiser la somme des carrés des erreurs ($\sum_i \epsilon_i^2$). Sous forme vectorielle, cette somme de carrés s'écrit comme une norme euclidienne au carré ($\|\boldsymbol{\epsilon}\|_2^2$), où la norme $\|\cdot\|_2$ est aussi appelée "norme L^2 ". Dans le cas de la régression d'arête, un terme de pénalité est ajouté à la vraisemblance, terme qui dépend aussi de la norme L^2 des effets pour la minimiser: $\|\boldsymbol{\epsilon}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_2^2$, où λ , paramètre à estimer, correspond à la pénalité à appliquer aux effets. C'est cette pénalité qui induit le rétrécissement des estimations des effets vers 0. Cela introduit du biais dans les estimations $\hat{\boldsymbol{\beta}}$ mais au bénéfice de réduire leur variance.

En supposant $Cov(X\boldsymbol{\beta}, \boldsymbol{\epsilon}) = 0$, on obtient:

$$Var(\mathbf{y}) = \sigma_\beta^2 X X^T + \sigma^2 I \quad (9)$$

où nous avons utilisé la formule mathématique $Var(M\boldsymbol{\theta}) = M Var(\boldsymbol{\theta}) M^T$ qui est l'équivalent matriciel de $Var(m\theta) = m^2 Var(\theta)$ où m est un coefficient connu et θ est une variable aléatoire.

Mais surtout, remarquez que $X X^T$ apparaît dans (9). Ceci permet de faire le lien entre l'apparentement attendu calculé à partir du pédigrée (A_{ped}) et l'apparentement réalisé estimé à partir des marqueurs (A_{mark}). En effet, considérons les génotypes dans X comme des variables aléatoires et suivons [Habier et coll. \(2007\)](#) pour calculer l'espérance du produit des génotypes aux marqueurs pour les individus i et j :

$$\begin{aligned} \mathbb{E}[\mathbf{x}_i^T \mathbf{x}_j] &= \sum_{p=1}^P \mathbb{E}[X_{ip} X_{jp}] \\ &= \sum_{p=1}^P (\text{Cov}[X_{ip}, X_{jp}] + \mathbb{E}[X_{ip}]\mathbb{E}[X_{jp}]) \end{aligned}$$

Remarquez que $\text{Cov}[X_{ip}, X_{jp}] = A_{ij} \times 2 f_p (1 - f_p)$, où A_{ij} est la relation génétique additive entre les individus i et j , égale à deux fois leur coefficient de simple apparentement (aussi souvent noté ϕ_A), et où les f_p 's sont

les fréquences alléliques des P SNPs. Par ailleurs, $\mathbb{E}[X_{ip}] = 2f_p$. Il s'avère donc que l'espérance $\mathbb{E}[XX^T]$ est égale à $A_{\text{ped}} \times 2 \sum_p f_p(1 - f_p)$ à une constante près. (Pendant l'inférence, la constante $\mathbf{1}\mathbf{1}^T 4 \sum_p f_p^2$ se retrouvera dans l'estimation de la moyenne globale μ , et n'a donc pas grande importance.)

Un estimateur de l'apparentement génétique additif deux-à-deux à partir des génotypes aux SNPs est donc:

$$A_{\text{mark}} = \frac{XX^T}{2 \sum_p f_p(1 - f_p)} \quad (10)$$

Un autre estimateur, celui de [VanRaden \(2008\)](#), centre d'abord la matrice X avec les fréquences alléliques, de telle sorte que A_{mark} , sous Hardy-Weinberg, est centrée le long de sa diagonale sur 1 et hors de sa diagonale sur 0:

$$A_{\text{mark},VR} = \frac{X_{\text{centered}} X_{\text{centered}}^T}{2 \sum_p f_p(1 - f_p)}$$

Parmi plusieurs estimateurs d'apparentement, celui proposé par VanRaden a été considéré comme un choix robuste ([Toro et coll., 2011](#)).

Ainsi, le modèle de régression d'arête (8) est équivalent au modèle suivant:

$$\mathbf{y} = \mathbf{1}\mu + \mathbf{a} + \boldsymbol{\epsilon} \text{ avec } \boldsymbol{\epsilon} \sim \mathcal{N}_N(\mathbf{0}, \sigma^2 I) \text{ et } \mathbf{a} \sim \mathcal{N}_N(\mathbf{0}, \sigma_a^2 A_{\text{mark}}) \quad (11)$$

Cette équivalence permet d'utiliser (8) pour:

- estimer les effets alléliques, $\hat{\beta}$, et leur variance, $\hat{\sigma}_{\beta}^2$;
- prédire les valeurs génotypiques additives, $\hat{\mathbf{a}} = X\hat{\beta}$;
- estimer la composante génétique additive de la variance, $\hat{\sigma}_a^2 = \hat{\sigma}_{\beta}^2 \times 2 \sum_p f_p(1 - f_p)$;
- et estimer l'héritabilité au sens strict, $\hat{h}^2 = \frac{\hat{\sigma}_a^2}{\hat{\sigma}_a^2 + \sigma^2}$.

Il y aurait encore beaucoup à dire sur ces questions, mais passons maintenant aux exercices pratiques.

4 Simuler des données

Fixons la graine du générateur de nombres pseudo-aléatoires pour la reproductibilité des simulations:

```
set.seed(1953) # année de publication de la découverte de la structure de l'ADN
```

4.1 Génotypes

Simulons des génotypes, en supposant qu'ils sont tous indépendants (c'est-à-dire sans déséquilibre de liaison):

```
N <- 500
inds.id <- sprintf(fmt=paste0("ind%0", floor(log10(N))+1, "i"), 1:N)
P <- 5000
snps.id <- sprintf(fmt=paste0("snp%0", floor(log10(P))+1, "i"), 1:P)

calcGenoFreq <- function(maf){ # assuming Hardy-Weinberg equilibrium
  c((1 - maf)^2, 2 * (1 - maf) * maf, maf^2)
}

X <- matrix(sample(x=c(0,1,2), size=N*P, replace=TRUE, prob=calcGenoFreq(0.3)),
            nrow=N, ncol=P, dimnames=list(inds.id, snps.id))
```

Les fréquences alléliques s'estiment facilement:

```
afs <- colMeans(X) / 2
summary(afs)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.252   0.290   0.300   0.300   0.310   0.346
```

La matrice des relations génétiques additives peut s'estimer avec (10):

```
A.mark <- (X %*% t(X)) / (2 * sum(afs * (1 - afs)))
```

Une simulation moins simpliste avec du déséquilibre de liaison (indispensable à une bonne précision de prédiction) nécessiterait un véritable scénario évolutif, par exemple en utilisant le processus stochastique du coalescent avec recombinaison. Pour cela, vous pouvez vous inspirer du document “Exemple de simulation pour explorer la prédiction génomique”.

4.2 Effets additifs des allèles

— Caractère monogénique:

Commençons par choisir le seul SNP causal, de telle sorte que sa fréquence allélique ne soit ni trop faible ni trop élevée:

```
mafs <- apply(rbind(afs, 1 - afs), 2, min) # fréquences de l'allèle minoritaire
(snp.qtl <- sample(x=snps.id[mafs >= 0.25 & mafs <= 0.35], size=1))
```

```
## [1] "snp1384"
```

Puis fixons son effet allélique additif à une valeur élevée, les autres SNPs ayant un effet nul:

```
beta.mono <- setNames(rep(0, P), snps.id)
beta.mono[snp.qtl] <- 3
```

— Caractère polygénique:

L'effet allélique additif à chaque marqueur, β_p , vient de $\mathcal{N}(0, \sigma_\beta^2)$:

```
sigma.beta2.poly <- 10^(-3)
beta.poly <- setNames(rnorm(n=P, mean=0, sd=sqrt(sigma.beta2.poly)), snps.id)
```

4.3 Erreurs puis phénotypes

Fixons la moyenne globale, et simulons les erreurs:

```
mu <- 36
sigma.epsilon2 <- 3
epsilon <- matrix(rnorm(n=N, mean=0, sd=sqrt(sigma.epsilon2)))
```

Les phénotypes, y , sont calculés à partir de la formule (8). Seul le vecteur des effets alléliques additifs, β , est différent selon l'architecture génétique concernée.

— Caractère monogénique:

```
y.mono <- matrix(1, nrow=N) * mu + X %*% beta.mono + epsilon
```

— Caractère polygénique:

```
y.poly <- matrix(1, nrow=N) * mu + X %*% beta.poly + epsilon
```

Dans ce cas, on s'attend à une héritabilité au sens strict de:

```
sigma.a2 <- sigma.beta2.poly * 2 * sum(afs * (1 - afs))
sigma.g2 <- sigma.a2
(h2 <- sigma.a2 / (sigma.g2 + sigma.epsilon2))
```

```
## [1] 0.412
```

Ce que l'on retrouve dans les données simulées:

```
(var(X %%% beta.poly) / (var(X %%% beta.poly) + var(epsilon)))
```

```
##      [,1]
```

```
## [1,] 0.417
```

Notez qu'on aurait aussi pu directement simuler les valeurs génotypiques additives via $\mathbf{a} \sim \mathcal{N}_N(\mathbf{0}, \sigma_a^2 A_{\text{mark}})$. En R, en utilisant la fonction `mvrnorm` du paquet [MASS](#), cela aurait donné:

```
if(requireNamespace("MASS", quietly=TRUE)){
  a <- MASS::mvrnorm(n=1, mu=rep(0, N), Sigma=sigma.a2 * A.mark)
  g <- a
  y.poly <- matrix(1, nrow=N) * mu + g + epsilon
}
```

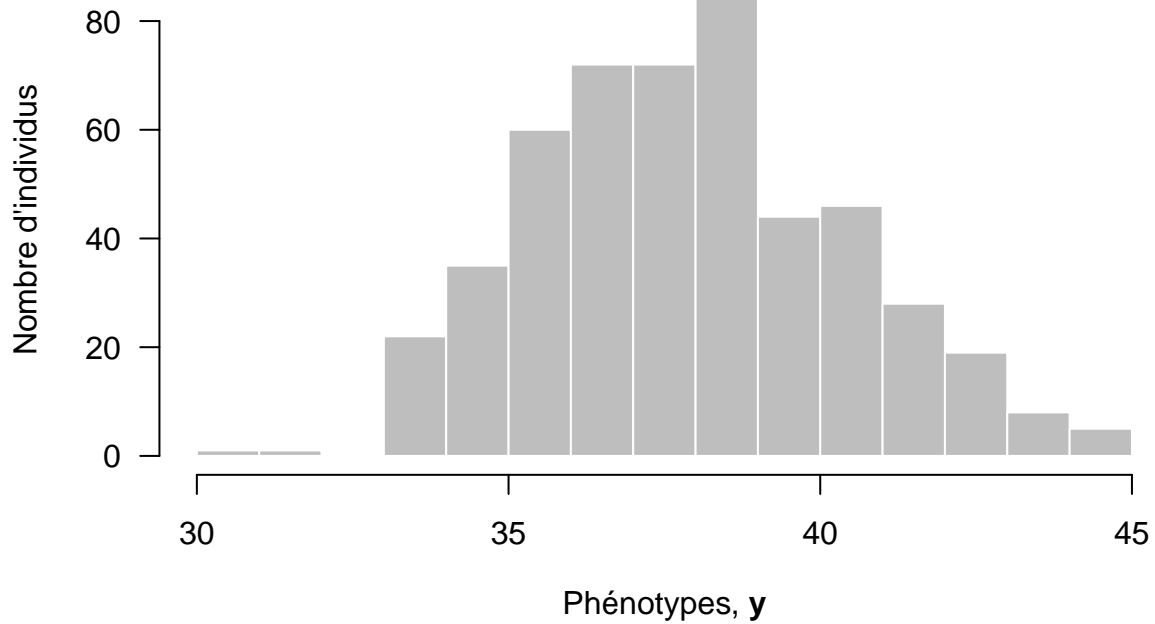
5 Réaliser l'inférence

5.1 Visualisation graphique

Avant toute autre chose, regardons à quoi ressemblent les données:

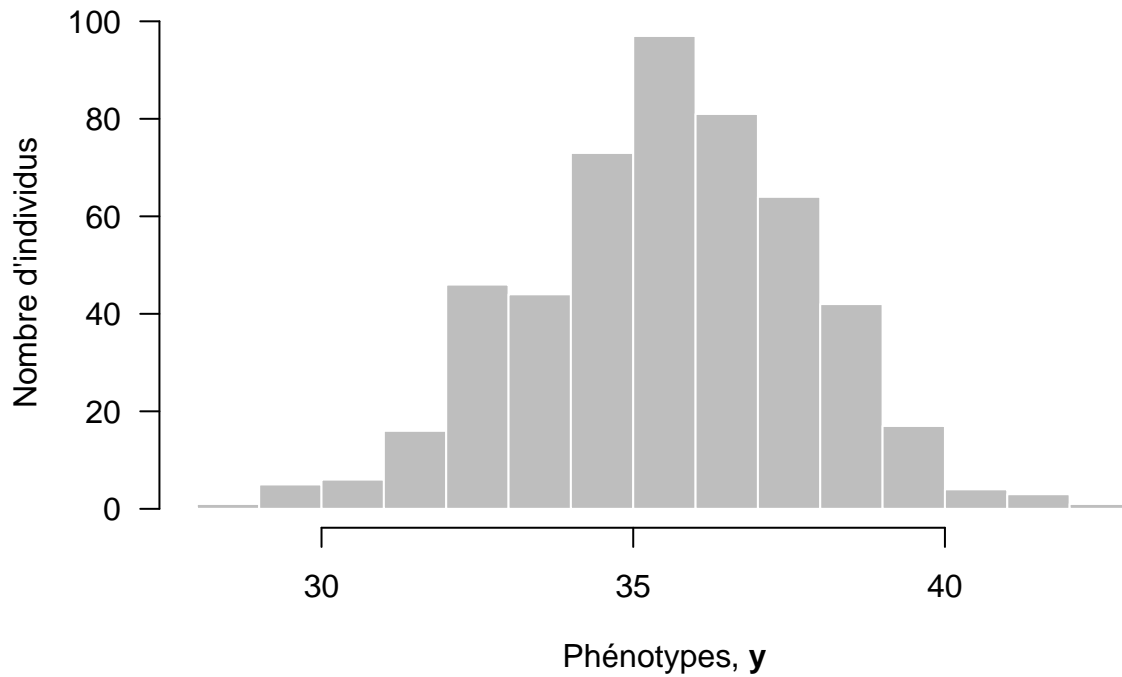
```
hist(y.mono, breaks="FD", las=1, col="grey", border="white",
     main="Caractère monogénique", ylab="Nombre d'individus",
     xlab=expression(paste("Phénotypes, ", bold(y))))
```


Caractère monogénique



```
hist(y.poly, breaks="FD", las=1, col="grey", border="white",  
     main="Caractère polygénique", ylab="Nombre d'individus",  
     xlab=expression(paste("Phénotypes, ", bold(y))))
```

Caractère polygénique



Il est aussi recommandé de regarder la matrice d'apparentement estimée à partir des marqueurs via [\(10\)](#).

Pour cela, les fonctions `seriate` et `pimage` du paquet `seriation` peuvent vous être utile:

```
library(seriation)
A.mark.reorder <- seriate(A.mark)
pimage(A.mark, A.mark.reorder)
```

5.2 SNP à SNP (“GWAS”)

Sur le plan informatique, nous allons tester seulement un sous-ensemble des $P = 5000$ SNPs pour aller plus vite. Echantillons donc uniformément un sous-ensemble de SNPs à tester (mais incluant le causal):

```
nb.subset.snps <- 20
subset.snps <- unique(sort(c(snp.qtl, sample(snps.id, nb.subset.snps))))
```

Le paquet `MM4LMM` implémente un algorithme MM pour ajuster le modèle (6) par ML ou ReML.

Ajustons le modèle SNP à SNP (6) sur les données du caractère monogénique:

```
res.mono.gwas <- list()
out.mmest <- MMEst(Y=y.mono[,1], X=X[,subset.snps],
                  VarList=list(Additive=A.mark, Error=diag(N)))

## Aie
## iteration = 3
## Aie
## iteration = 3

out.anovatest <- AnovaTest(out.mmest, Type="TypeI")
tmp <- t(sapply(out.anovatest, function(x){x["Xeffect",]}))
res.mono.gwas$scan <- list(pval=setNames(tmp[, "pval"], rownames(tmp)))
```

Ajustons ce même modèle sur les données du caractère polygénique:

```
res.poly.gwas <- list()
out.mmest <- MMEst(Y=y.poly[,1], X=X[,subset.snps],
                  VarList=list(Additive=A.mark, Error=diag(N)))

## Aie
## iteration = 3

out.anovatest <- AnovaTest(out.mmest, Type="TypeI")
tmp <- t(sapply(out.anovatest, function(x){x["Xeffect",]}))
res.poly.gwas$scan <- list(pval=setNames(tmp[, "pval"], rownames(tmp)))
```

5.3 Tous les SNPs conjointement (“ridge”)

Le paquet `rrBLUP` implémente la régression d’arête (8) permettant d’estimer tous les effets alléliques conjointement.

Ajustons le modèle conjoint (8) sur les données du caractère monogénique:

```
res.mono.ridge <- mixed.solve(y=y.mono, Z=X)
```

Ajustons ce même modèle sur les données du caractère polygénique:

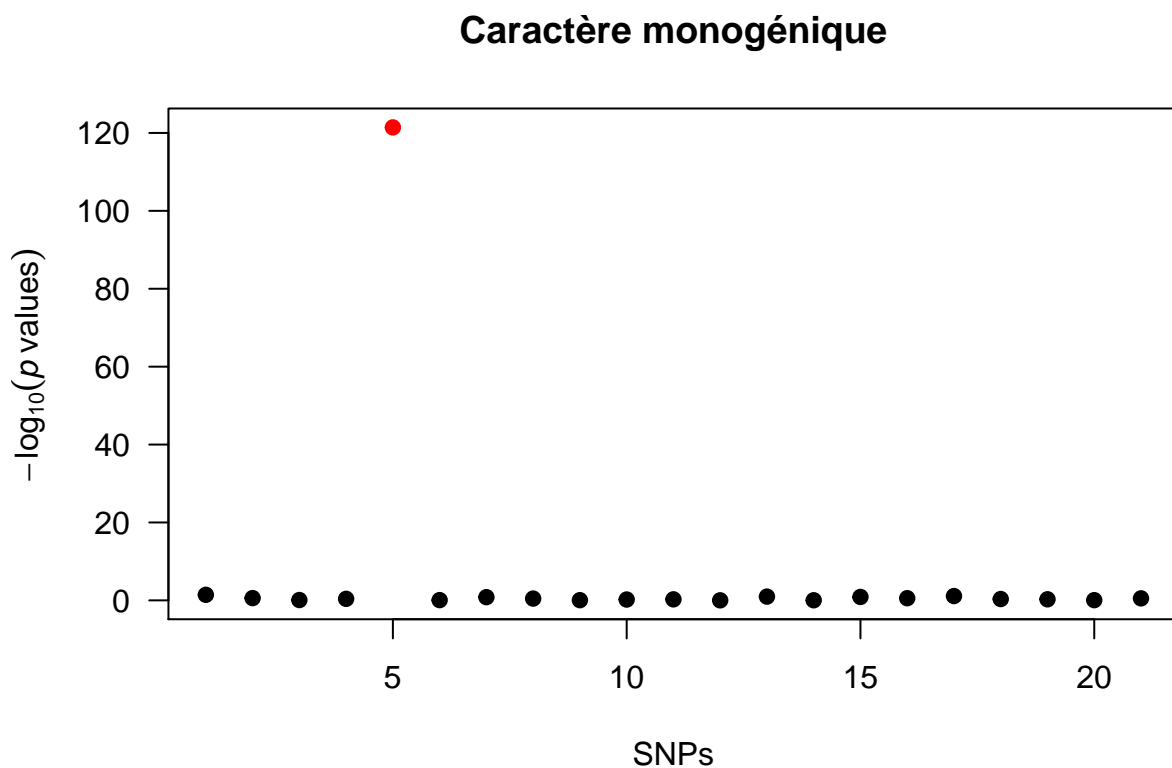
```
res.poly.ridge <- mixed.solve(y=y.poly, Z=X)
```

6 Evaluer les résultats

La manière habituelle de regarder les résultats des tests du modèle d'inférence SNP à SNP (6) est de tracer un *Manhattan plot* (regardez la fonction `manhattan` du paquet `qqman`).

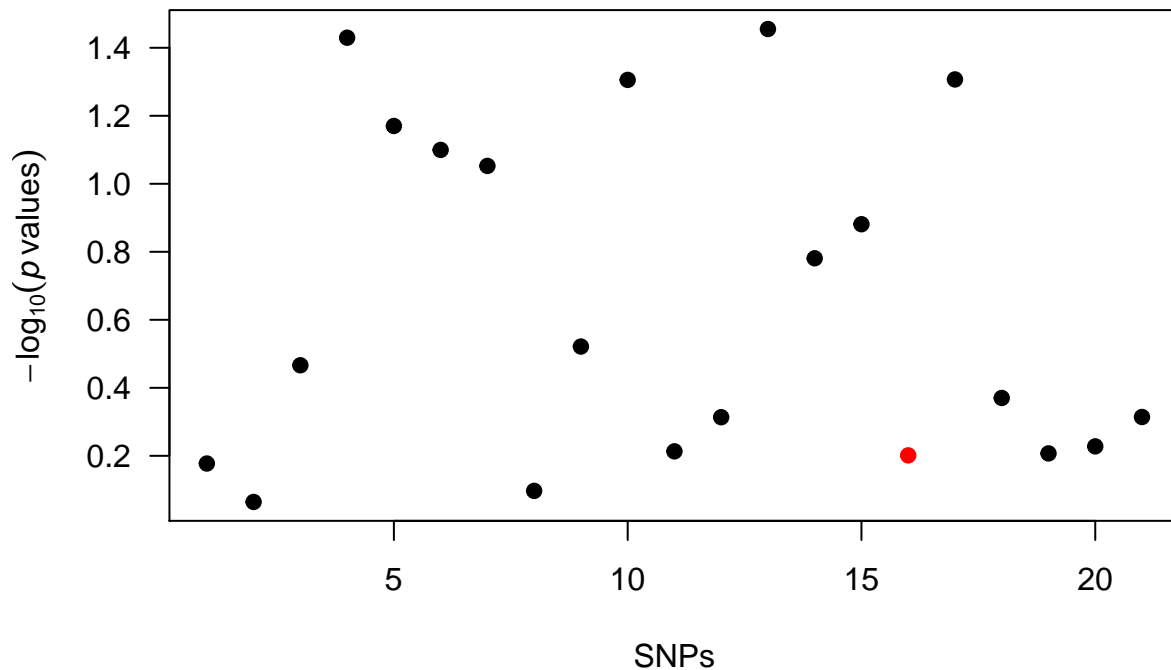
Dans tous les cas, comme les données sont simulées, nous connaissons le SNP p avec l'effet β_p le plus grand. Il sera indiqué d'un point rouge dans les graphiques ci-dessous.

```
plot(x=1:length(subset.snps), y=-log10(res.mono.gwas$scan$pval),
     main="Caractère monogénique", las=1, type="n",
     xlab="SNPs", ylab=expression(-log[10](italic(p)~values)))
idx <- which(names(res.mono.gwas$scan$pval) == snp.qtl)
points(x=idx, y=-log10(res.mono.gwas$scan$pval[idx]), col="red", pch=19)
points(x=which(names(res.mono.gwas$scan$pval) != snp.qtl),
       y=-log10(res.mono.gwas$scan$pval[-idx]), col="black", pch=19)
```



```
plot(x=1:length(subset.snps), y=-log10(res.poly.gwas$scan$pval),
     main="Caractère polygénique", las=1, type="n",
     xlab="SNPs", ylab=expression(-log[10](italic(p)~values)))
idx <- which(names(res.poly.gwas$scan$pval) == names(which.max(beta.poly[subset.snps])))
points(x=idx, y=-log10(res.poly.gwas$scan$pval[idx]), col="red", pch=19)
points(x=which(names(res.poly.gwas$scan$pval) != names(which.max(beta.poly[subset.snps]))),
       y=-log10(res.poly.gwas$scan$pval[-idx]), col="black", pch=19)
```

Caractère polygénique



Le modèle d'inférence SNP à SNP parvient bien à détecter le SNP causal dans le cas du caractère monogénique, mais l'interprétation du *Manhattan plot* est beaucoup moins claire dans le cas du caractère polygénique. En effet, dans ce cas, tous les SNP ayant un effet non-nul de faible grandeur, aucun ne ressort distinctement, et celui avec le plus grand β n'a pas le plus grand $\hat{\beta}$.

A l'inverse, le modèle d'inférence conjoint estime relativement précisément les composants de la variance et la moyenne globale dans le cas du caractère polygénique:

```
c(mu, res.poly.ridge$beta)
```

```
## [1] 36.0 35.4
```

```
c(sigma.epsilon2, res.poly.ridge$Ve)
```

```
## [1] 3.00 1.93
```

```
c(sigma.beta2.poly, res.poly.ridge$Vu)
```

```
## [1] 0.00100 0.00144
```

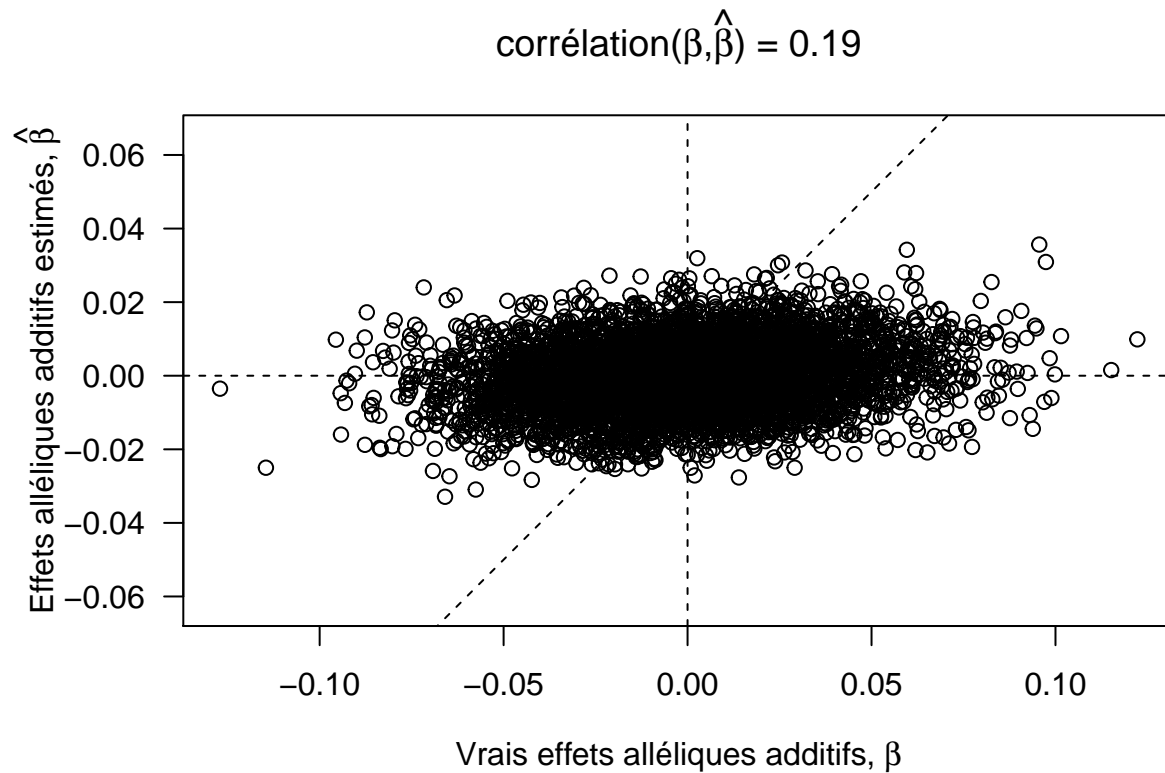
Il est ensuite intéressant de remarquer que les effets aux marqueurs sont relativement mal estimés individuellement (ceci étant dû au rétrécissement opéré par la pénalité λ):

```
(c <- cor(beta.poly, res.poly.ridge$u))
```

```
## [1] 0.193
```

```
par(mar=c(5, 4.5, 4, 2) + 0.1)
plot(beta.poly, res.poly.ridge$u, las=1, asp=1,
      xlab=expression(paste("Vrais effets alléliques additifs, ", bold(beta))),
      ylab=expression(paste("Effets alléliques additifs estimés, ",
                             hat(beta))),
      main=bquote(paste("corrélation(", bold(beta), ",", hat(bold(beta)), ") = ",
                        .(format(c, digits=2))))
```

```
abline(v=0, lty=2); abline(h=0, lty=2); abline(a=0, b=1, lty=2)
```

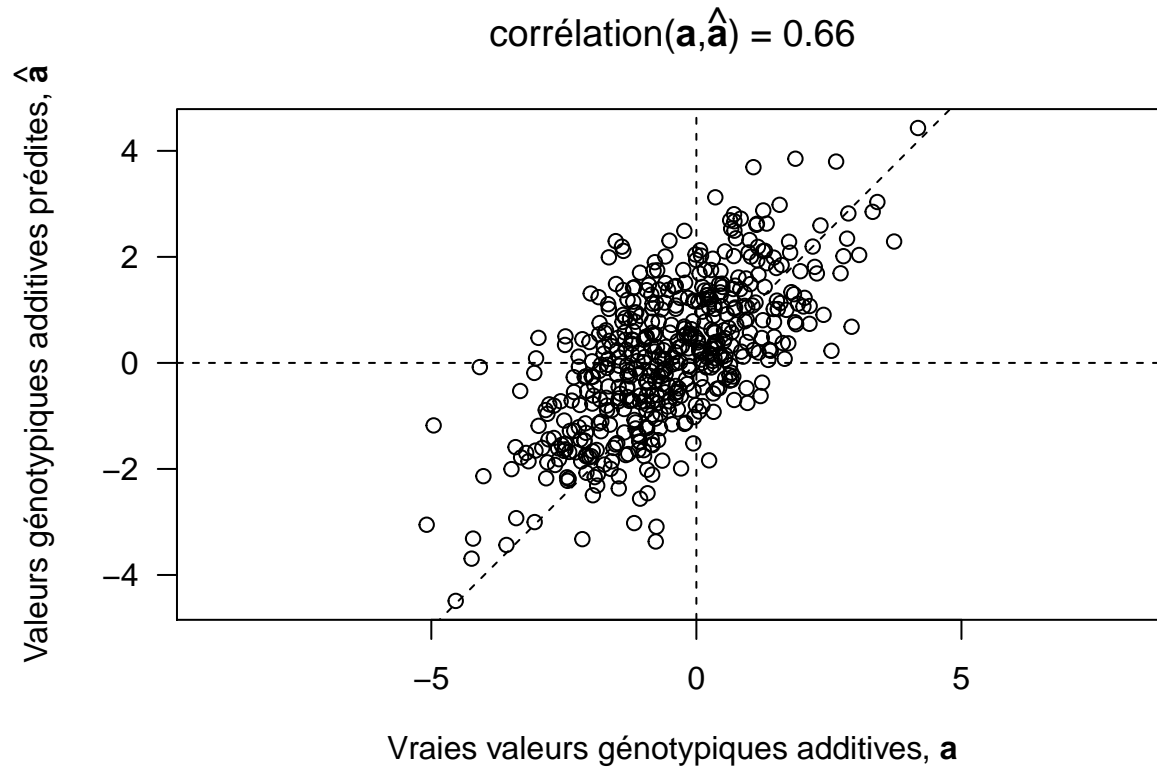


Par contre, les valeurs génotypiques additives, elles, sont bien mieux prédites:

```
(c <- cor(X %%% beta.poly, X %%% res.poly.ridge$u))
```

```
##      [,1]
## [1,] 0.659
```

```
par(mar=c(5, 4.5, 4, 2) + 0.1)
plot(X %%% beta.poly, X %%% res.poly.ridge$u, las=1, asp=1,
     xlab=expression(paste("Vraies valeurs génotypiques additives, ", bold(a))),
     ylab=expression(paste("Valeurs génotypiques additives prédites, ",
                           hat(bold(a)))),
     main=bquote(paste("corrélation(", bold(a), ",", hat(bold(a)), ") = ",
                       .(format(c, digits=2))))))
abline(v=0, lty=2); abline(h=0, lty=2); abline(a=0, b=1, lty=2)
```



C'est en cela qu'analyser conjointement tous les marqueurs est pertinent. Pour les caractères polygéniques, le modèle d'inférence SNP à SNP (6) n'est pas efficace car les effets alléliques, pris individuellement, sont trop faibles. En estimant tous les effets conjointement avec (8), même si chacun d'eux est biaisé, leur somme, elle, est estimée bien plus précisément.

Notez que je parle d'effets alléliques "estimés" et de valeurs génotypiques "prédites", même si les deux sont des effets aléatoires dans les modèles mixtes (8) et (11). L'une des raisons vient du fait que le nombre de valeurs génotypiques dépend du nombre d'individus. De plus, dans le modèle (11), les inconnues \mathbf{a} sont les *breeding values* et les résultats $\hat{\mathbf{a}}$ sont les *Best Linear Unbiased Predictions* (BLUPs) des *breeding values*.

C'est la raison pour laquelle on parle de **prédiction génomique**, qui mène ensuite tout naturellement à la **sélection génomique** qui se base sur les valeurs génotypiques additives "prédites" grâce aux génotypes aux marqueurs.

7 Autres points importants

7.1 Eviter le sur-ajustement

Avec de "vraies" données, c'est-à-dire non simulées, il est important de réaliser que les estimations des effets alléliques ont le risque d'être sur-ajustées aux individus particuliers pour lesquels on dispose de génotypes et phénotypes. Or, en sélection génomique, ces estimations sont utilisées pour prédire la valeur génotypique additive d'individus pour lesquels on ne dispose que de génotypes et pas de phénotypes. Un sur-ajustement a pour conséquence de mal généraliser les estimations du jeu d'entraînement pour effectuer des prédictions sur différents jeux de test. Pour éviter cela, il est courant d'estimer les paramètres du modèle par **validation croisée**.

La variante fréquemment utilisée de cette procédure consiste à répartir aléatoirement les individus génotypés et phénotypés en $K = 10$ sous-ensembles de taille égale et n'ayant pas d'individus en commun. Pour chaque sous-ensemble, les 9 autres sont utilisés pour estimer les paramètres. Au final, pour chaque marqueur, on

dispose de 10 estimations de son effet allélique. Ainsi, lorsque de nouveaux individus non-phénotypés sont génotypés, on peut utiliser la moyenne des estimations de chaque effet allélique pour prédire leur valeur génotypique additive. Ces nouveaux individus peuvent donc être également sélectionnés selon ce critère alors même qu'ils n'ont pas été phénotypés.

De plus, la validation croisée est aussi utilisée pour sélectionner le meilleur modèle sur le jeu d'entraînement (modèle monogénique versus modèle additif infinitésimal versus ... voir ci-après). Pour chaque sous-ensemble k parmi les 10, on compare les prédictions aux phénotypes observés (ceux-ci ayant été corrigés au préalable des effets des facteurs autres que génétiques, telle l'hétérogénéité spatiale). Puis on estime l'**erreur quadratique moyenne de prédiction** (*mean squared prediction error*, MSPE):

$$\widehat{\text{MSPE}}_k = \frac{1}{N_k} \sum_{i_k=1}^{N_k} (y_{i_k} - (\hat{\mu}_k + \hat{a}_{i_k}))^2 \quad (12)$$

où N_k correspond au nombre d'individus dans le sous-ensemble n'ayant pas servi à estimer les effets alléliques utilisés pour calculer les \hat{a}_{i_k} .

Si l'on effectue cette procédure de validation croisée avec plusieurs modèles, le meilleur d'entre eux correspondra à celui ayant la plus petite $\overline{\text{MSPE}}$ (moyenne sur les 10 sous-ensembles).

Concernant la précision de prédiction, à la place des $\widehat{\text{MSPE}}_k$, les articles sur la sélection génomique indiquent habituellement la corrélation (coefficient de Pearson) entre les valeurs génotypiques additives prédites (\hat{a}_k) et les phénotypes corrigés pour les facteurs environnementaux (y_k). Le mot employé en anglais pour ce critère est *accuracy* (lorsque le carré de la corrélation est indiqué, le terme utilisé est *reliability*). Si ce coefficient est calculé, il est recommandé de regarder également les estimations des moyenne globale et pente de la régression linéaire simple $y_k = a + b \hat{a}_k$.

Concrètement, pour évaluer la précision de la prédiction par validation croisée en R avec **rrBLUP**, le paquet **cvTools** peut être utile. Mais il requiert une méthode **predict**, qui n'est pas fournie par **rrBLUP** version 4.5. Il faut donc d'abord encapsuler la fonction **mixed.solve** dans une autre fonction de telle sorte qu'elle renvoie un objet de classe "rr". Puis il faut ajouter la méthode **predict** à cette classe:

```
rr <- function(y, Z, K=NULL, X=NULL, method="REML"){
  stopifnot(is.matrix(Z))
  out <- mixed.solve(y=y, Z=Z, K=K, X=X, method=method)
  return(structure(out, class="rr"))
}
predict.rr <- function(object, newZ){
  stopifnot(is.matrix(newZ))
  out <- as.vector(newZ %*% object$u)
  if(! is.null(rownames(newZ)))
    names(out) <- rownames(newZ)
  return(out)
}
```

Une fois que c'est fait, on peut réaliser la validation croisée comme suit:

```
if(requireNamespace("cvTools", quietly=TRUE)){
  folds <- cvTools::cvFolds(n=nrow(X), K=5, R=10)
  callRR <- call("rr", y=y.poly, Z=X)
  system.time(
    out.cv <- cvTools::cvTool(call=callRR, x=X, y=y.poly, names=c("Z", "y"),
                             cost=cor, folds=folds))
}
```

Regardez le document "Exemple de simulation pour explorer la prédiction génomique" pour plus de détails.

7.2 Intermédiaires d’architecture génétique additive

Nous avons vu ci-dessus comment différents modèles d’inférence, (6) et (8), sont plus ou moins performants selon l’architecture génétique additive d’un caractère. Mais avec de “vraies” données, on ne connaît pas toujours l’architecture génétique des caractères d’intérêt. Ne pourrait-on donc pas avoir un seul modèle s’adaptant à toutes les architectures ?

C’est un problème plus compliqué, mais les modèles dits de **sélection de variables** vont dans ce sens en analysant conjointement tous les SNPs tout en testant lesquels ont des effets non-nuls, par exemple:

$$\mathbf{y} = \mathbf{1}\mu + \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \text{ avec } \boldsymbol{\epsilon} \sim \mathcal{N}_N(\mathbf{0}, \sigma^2 \mathbf{I}) \text{ et } \forall p \beta_p \sim \pi \mathcal{N}(0, \sigma_\beta^2) + (1 - \pi) \delta_0 \quad (13)$$

où δ_0 est la variable aléatoire qui prend la valeur 0 avec une probabilité de 1 (distribution de Dirac).

Dans ce modèle, π représente la proportion de SNPs dont les allèles ont un effet additif non-nul. On l’appelle souvent la “probabilité d’inclusion”. Estimer ce paramètre revient donc à s’adapter à différentes architectures génétiques additives. On fait cet effort-là lorsque l’on veut également identifier les SNPs ayant des effets alléliques non-nuls, en essayant de prédire précisément tout en étant parcimonieux, c’est-à-dire en n’incluant que les SNPs nécessaires. On cherche donc à ce que le vecteur $\hat{\boldsymbol{\beta}}$ soit **peu dense** (*sparse*). De manière générale, *shrinkage* et *sparsity* sont deux mots-clés des modèles statistiques en grande dimension.

Simulons des données:

```
beta.sparse <- setNames(rep(0, P), snps.id)
pi <- 0.2
snps.qtl <- sample(snps.id, floor(pi * P))
beta.sparse[snps.qtl] <- rnorm(n=length(snps.qtl), mean=0,
                             sd=sqrt(sigma.beta2.poly))
y.sparse <- matrix(1, nrow=N) * mu + X %*% beta.sparse + epsilon
```

Le paquet **BGLR** implémente le modèle (13) sous le nom de **BayesC**. Ce paquet utilise une méthode d’inférence bayésienne, l’échantillonneur de Gibbs (*Gibbs sampler*). Mais il est trop long d’en dire plus ici. De plus, comme l’inférence se réalise par échantillonnage, elle est (bien) plus longue que les méthodes présentées précédemment. La commande R ci-dessous n’est donc pas exécutée dans ce document:

```
nIter <- 1*10^5; burnIn <- 1*10^4; thin <- 5
res.BGLR <- BGLR(y=y.sparse, ETA=list(list(X=X, model="BayesC")),
                verbose=FALSE, nIter=nIter, burnIn=burnIn, thin=thin)
```

Pour information, comme la convergence de l’échantillonneur de Gibbs peut être lente à atteindre, d’autant plus que la taille du jeu de données est grande, des logiciels ont été développés spécifiquement pour la génétique quantitative. Ils sont souvent écrits en langage **Fortran**, tels que la suite **BLUPF90**, ainsi que le logiciel **GS3** (encapsulé dans le paquet R **rgs3**).

Beaucoup d’autres modèles de sélection de variables existent. Certains utilisent d’autres normes pour pénaliser la vraisemblance tout en sélectionnant certaines variables, comme la norme L^1 pour le modèle **Lasso** (Tibshirani, 1996; paquet **glmnet** et **lars**), une combinaison des normes L^1 et L^2 pour le modèle **Elastic Net** (Zou & Hastie, 2005; paquet **glmnet**), etc.

De plus, pour un même modèle, par exemple (13), d’autres méthodes algorithmiques sont utilisées, plus rapides que l’échantillonneur de Gibbs, mais au prix d’approximations, comme par exemple le **bayésien variationnel** (Carbonetto & Stephens, 2012; paquet **varbvs**).

7.3 Au-delà de l’architecture génétique additive

La matrice A_{mark} calculée via (10) est proportionnelle à $\mathbf{X}\mathbf{X}^T$. L’apparementement génétique additif entre deux individus i et j est donc $A_{ij} \propto \sum_p X_{ip}X_{jp}^T = \mathbf{x}_i^T \cdot \mathbf{x}_j$, appelé **produit scalaire** (*dot product*). D’un point

de vue géométrique, ce produit scalaire quantifie la distance linéaire entre les deux individus dans l'espace euclidien des génotypes.

Mais on peut bien sûr utiliser d'autres fonctions de distance, non-linéaires cette fois. On utilise alors le terme de **noyau** (*kernel*) pour dénoter ces fonctions. Afin de capturer la contribution des effets génétiques non-additifs, certains ont proposé d'utiliser le modèle (11) avec A_{mark} calculée via un noyau défini dans un **espace de Hilbert à noyau reproduisant** (*Reproducing Kernel Hilbert Space*, RKHS). A ce terme compliqué peut en fait simplement correspondre un noyau gaussien tel que $A_{ij} = \exp(-(D_{ij}/\theta)^2)$ où D_{ij} est la distance euclidienne entre \mathbf{x}_i et \mathbf{x}_j normalisée dans l'intervalle $[0, 1]$ et θ est un paramètre d'échelle influençant la vitesse à laquelle la covariance génétique décroît en fonction de la distance (celui-ci doit être estimé par validation croisée). Le paquet rrBLUP permet d'utiliser ce noyau, d'autres étant implémentés dans le paquet KRMM.

8 Explorer les simulations possibles

Voici certaines questions que vous pouvez vous poser:

- quel est l'impact de la fréquence allélique sur l'inférence des paramètres et la précision de la prédiction ?
- quel est l'impact de la taille du jeu d'entraînement sur l'inférence et la prédiction ?
- quel est l'impact de l'appareillement entre individus du jeu d'entraînement et individus du jeu de test ?
- etc

C'est à vous !

9 Analyser de vrais jeux de données disponibles

Comme l'a fait justement remarquer Zamir (PLoS Biology 2013, Science 2014), il est difficile de trouver des jeux de données avec phénotypes en libre accès. Cependant, en voici quelques uns:

- Crossa *et al* (Genetics, 2010): blé (599 lignées, 4 conditions, rendement en grains, pédigrée, 1279 marqueurs DArT) et maïs (300 lignées, 1148 marqueurs SNP, 3 caractères, deux conditions)
- Resende *et al* (Genetics, 2012): pin (951 individus de 61 familles, pédigrée, 4853 marqueurs SNP, phénotypes dérégressés)
- Cleveland *et al* (G3, 2012): porc (3534 animaux, pédigrée, 5 caractères, 53000 marqueurs SNP)

10 Perspectives

Les grandes simplifications de ce travail ont été de ne se concentrer que sur un seul caractère, continu de sucroût, et d'ignorer un grand nombre d'éléments tels le déséquilibre de liaison, les interactions génotype-environnement, etc.

Or tout ceci intervient dans la "vraie vie". C'est bien là le défi des sélectionneurs, qu'ils soient dans des entreprises semencières ou dans des collectifs de paysans: gérer la diversité et créer continuellement de nouvelles variétés combinant plusieurs caractères d'intérêt et adaptées à l'itinéraire technique, à la filière économique, à l'agriculteur, au consommateur, etc.

Mais ce sera pour le cours suivant !

11 Références

- Lynch, M., and B. Walsh (1998). Genetics and analysis of quantitative traits. Sinauer Associates, 1998.
- Barton, N. H. and P. D. Keightley (2002, January). Understanding quantitative genetic variation. *Nature Reviews Genetics* 3 (1), 11-21. [DOI](#)
- Weir, B. S., A. D. Anderson, and A. B. Hepler (2006, October). Genetic relatedness analysis: modern data and new challenges. *Nature Reviews Genetics* 7 (10), 771-780. [DOI](#)
- Visscher, P. M., W. G. Hill, and N. R. Wray (2008, March). Heritability in the genomics era — concepts and misconceptions. *Nature Reviews Genetics* 9 (4), 255-266. [DOI](#)
- Slatkin, M. (2008, June). Linkage disequilibrium — understanding the evolutionary past and mapping the medical future. *Nature Reviews Genetics* 9 (6), 477-485. [DOI](#)
- Stephens, M. and D. J. Balding (2009, October). Bayesian statistical methods for genetic association studies. *Nature Reviews Genetics* 10 (10), 681-690. [DOI](#)
- de los Campos, G., D. Gianola, and D. B. Allison (2010, December). Predicting genetic predisposition in humans: the promise of whole-genome markers. *Nature Reviews Genetics* 11 (12), 880-886. [DOI](#)
- Morrell, P. L., E. S. Buckler, and J. Ross-Ibarra (2012, February). Crop genomics: advances and applications. *Nature Reviews Genetics* 13 (2), 85-96. [DOI](#)
- Vitezica, Z., L. Varona, and A. Legarra (2013, December). On the additive and dominant variance and covariance of individuals within the genomic selection scope. *Genetics* 195 (4), 1223-30. [DOI](#)
- Howard, R., A. Carriquiry, and W. Beavis (2014, June). Parametric and nonparametric statistical methods for genomic selection of traits with additive and epistatic genetic architectures. *G3* 4 (6), 1027-46. [DOI](#)
- Rabier, C.-E., Barre, P., Asp, T., Charmet, G., Mangin, B. (2016, June). On the accuracy of genomic selection. *PLoS ONE* 11 (6): e0156086. [DOI](#)
- Scutari, M., Mackay, I., Balding, D. (2016, September). Using genetic distance to infer the accuracy of genomic prediction. *PLoS Genetics* 12 (9): e1006288. [DOI](#)
- Huang, W., and T. F. C. Mackay (2016, November). The genetic architecture of quantitative traits cannot be inferred from variance component analysis. *PLoS Genetics* 12 (11): e1006421. [DOI](#)

12 Annexe

```
t1 <- proc.time(); t1 - t0

##      user system elapsed
## 56.555   0.601  57.291

print(sessionInfo(), locale=FALSE)

## R version 3.5.3 (2019-03-11)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 18.04.3 LTS
##
## Matrix products: default
## BLAS: /usr/local/R/3.5.3/lib/R/lib/libRblas.so
## LAPACK: /usr/local/R/3.5.3/lib/R/lib/libRlapack.so
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods   base
##
## other attached packages:
## [1] rrBLUP_4.6      MM4LMM_1.1.5    knitr_1.23      rmarkdown_1.14
##
## loaded via a namespace (and not attached):
```

```
## [1] Rcpp_1.0.1      lattice_0.20-38 digest_0.6.20  grid_3.5.3
## [5] magrittr_1.5     evaluate_0.14  rlang_0.4.2    stringi_1.4.3
## [9] Matrix_1.2-15    tools_3.5.3   stringr_1.4.0  xfun_0.8
## [13] yaml_2.2.0       parallel_3.5.3 compiler_3.5.3 htmltools_0.4.0
```