Atelier prédiction génomique : prédiction génomique

Vincent Segura (INRAE)

d'après 'prediction-genomique.Rmd' de Timothée Flutre (INRAE)

Lundi 17 février 2020

Préambule

Packages

- Cette présentation nécessite le chargement des packages MM4LMM et rrBLUP (à installer au préalable par exemple via la fonction `install.packages`)

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

Introduction

Le modèle de la génétique quantitative

Le modèle de la génétique quantitative

- Pour chaque individu i parmi les N que compte la population :

$$y_i = g_i + \epsilon_i$$

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 ("déviation environnementale")

Décomposition de la variance

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

Décomposition de la valeur génotypique

- La valeur génotypique peut également se décomposer en **composantes additive**, **de dominance** et **d'épistasie** : $g_i = a_i + d_i + \zeta_i$
- La **valeur génotypique additive** (*breeding value*, a_i) est particulièrement importante car elle correspond à la part de la valeur génotypique qui est héritable, c'est-à-dire transmissible à la descendance
- On suppose généralement aussi que les composantes de la valeur génotypique 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** : $h^2 = \frac{\sigma_a^2}{\sigma_g^2 + \sigma_\epsilon^2}$

Ecriture matricielle du modèle

$$\boldsymbol{y} = \boldsymbol{g} + \boldsymbol{\epsilon}$$

- 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

- Si l'on ne considère que les contributions additives, alors $G = \sigma_a^2 A$
  où σ_a^2 est estimé et A est la **matrice d'apparentement** (*kinship*) calculée à partir du pédigrée

- La matrice R est généralement diagonale, telle que $R = \sigma_\epsilon^2 I$
  où σ_ϵ^2 est estimé simultanément à σ_a^2, et I est la matrice identité.

Notion de BLUP

- Si l'on suppose que $\boldsymbol{g} \sim \mathcal{N}_N(\mathbf{0}, G)$ et $\boldsymbol{\epsilon} \sim \mathcal{N}_N(\mathbf{0}, R)$,
- Alors $\hat{\boldsymbol{g}} = E[\boldsymbol{g}|\boldsymbol{y}] = G(G+R)^{-1}\boldsymbol{y}$

  où:
  - $\hat{\boldsymbol{g}}$ est le **meilleur prédicteur linéaire sans biais** de $\boldsymbol{g}$ (*best linear unbiased predictor*, BLUP)
  - $H = G(G+R)^{-1}$ est une généralisation matricielle de l'héritabilité

Matrice d'apparentement

- La généalogie permet de calculer la matrice d'apparentement **attendue** qui peut diffèrer de la matrice d'apparentement **réalisée**
- 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 **quantitative trait locus** (QTL)
- Si on dispose d'information génomique pour l'individu i, par exemple ses génotypes $\{\boldsymbol{x}_i\}$ à un ensemble de P marqueurs, le modèle précédent devient : $y_i = g(\boldsymbol{x}_i) + \epsilon_i$
  où g correspond à l'**architecture génétique** du caractère (détaillée ci-après)
- On peut donc utiliser les marqueurs pour estimer la matrice d'apparentement plus précisément

Estimation de l'effet des allèles aux marqueurs

- On peut aussi être intéressé par inclure les marqueurs explicitement dans le modèle comme variables explicatives pour estimer les effets de leurs allèles
- Mais il est fréquent qu'il y ait beaucoup plus de marqueurs que d'individus: $P >> N$
- Dans de tels cas, le modèle de **régression multiple** correspondant à l'extension de la **régression linéaire simple** présentée lors des "Premiers Pas" ne donne plus de bonnes estimations
- La **vraisemblance** doit être **pénalisée** (on dit aussi **régularisée**), ce qui se traduit par un **rétrécissement des estimations des effets** (*shrinkage*)

Ecrire le modèle

Notations

- N : nombre d'individus (diploïdes, plus ou moins apparentés)
- i : indice indiquant le i-ème individu, donc $i \in \{1, \ldots, N\}$
- P : nombre de marqueurs génétiques de type SNP (*single nucleotide polymorphism*), tous supposés bi-alléliques
- p : indice indiquant le p-ème SNP, donc $p \in \{1, \ldots, P\}$
- 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 ($\forall i, p, \quad x_{i,p} \in \{0, 1, 2\}$)

- X : matrice à N lignes et P colonnes contenant les génotypes de tous les individus à tous les SNPs
  - les génotypes de l'individu i à tous les SNPs sont dans le vecteur $\boldsymbol{x}_i^T$
  - les génotypes du SNP p pour tous les individus sont dans le vecteur $\boldsymbol{x}_p$
- 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 $\boldsymbol{\beta}$
- 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, contenant les relations génétique additives entre les N individus deux-à-deux
- ϵ_i : erreur pour l'individu i
- σ^2 : variance des erreurs

Vraisemblances d'extrêmes d'architecture génétique additive

- **Architecture génétique** (d'un caractère) : fonction mathématique modélisant la relation entre les génotypes des individus de la population et leurs phénotypes (*genotype-phenotype map*)
- On se limite à une **architecture génétique additive** et à deux cas extrêmes :
  1. **Caractère monogénique** déterminé par un seul SNP a un effet non-nul, par exemple un SNP non-synonyme dans le seul gène causal
  2. **Caractère polygénique** déterminé par un très grand nombre de SNPs ayant chacun un effet non-nul

Caractère monogénique

- Si l'on teste chaque SNP un par un avec une régression linéaire simple, on devrait pouvoir identifier le SNP causal

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

- La matrice de variance-covariance phénotypique vaut

$$Var(\boldsymbol{y}) = Var(\boldsymbol{\epsilon}) = \sigma^2 I$$

- Toutefois, il faut prendre en compte l'apparentement entre individus puisque des individus apparentés génétiquement ont plus de chance de partager des allèles aux locus causaux, et donc d'avoir des phénotypes similaires

Le modèle linéaire mixte

- Cela peut se faire en incluant dans le modèle un **effet aléatoire** (u_i) pour l'individu i, avec K comme matrice de variance-covariance

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

  où (σ_u^2, σ^2) sont les **composantes de la variance**

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

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

- La matrice K peut-être estimée à partir du pédigrée via par exemple la fonction `kinship` du package kinship2 ou bien à partir des génotypes aux marqueurs

Caractère polygénique

- Comme il y a vraiment beaucoup de SNPs ($P >> N$), l'hypothèse habituelle est que leurs allèles ont tous des effets très faibles
- Il vaut mieux tenter d'estimer leur effet global plutôt que leurs effets individuels, par exemple en supposant qu'ils s'additionnent tous :
  $\forall i,\ \sum_{p=1}^{P} x_{ip}\beta_p = \boldsymbol{x}_i^T \boldsymbol{\beta}$
- 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
- Alors, le modèle mixte s'écrit $\boldsymbol{y} = \mathbf{1}\mu + X\boldsymbol{\beta} + \boldsymbol{\epsilon}$,
  avec $\boldsymbol{\epsilon} \sim \mathcal{N}_N(\mathbf{0}, \sigma^2 I)$ et $\boldsymbol{\beta} \sim \mathcal{N}_P(\mathbf{0}, \sigma_\beta^2 I)$
- En modélisation statistique, ce modèle est connu sous le nom de **régression d'arête** (*ridge regression*)

Régression ‘ridge’

- 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 ‘ridge’, un terme de pénalité (λ) est ajouté à la vraisemblance, terme qui dépend aussi de la norme L^2 des effets pour la minimiser: $||\boldsymbol{\epsilon}||_2^2 + \lambda||\boldsymbol{\beta}||_2^2$
- C’est cette pénalité qui induit le rétrécissement des estimations des effets vers 0 (*shrinkage*)
- Cela introduit du biais dans les estimations $\hat{\boldsymbol{\beta}}$ mais au bénéfice de réduire leur variance

De la regression ‘ridge’ au modèle de génétique quantitative

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

$$Var(\boldsymbol{y}) = \sigma^2_\beta \, XX^T + \sigma^2 I$$

  où nous avons utilisé la formule mathématique

$$Var(M\boldsymbol{\theta}) = M \, Var(\boldsymbol{\theta}) \, M^T$$

- XX^T 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 al.* (2007) pour calculer l'espérance du produit des génotypes aux marqueurs pour les individus i et j:

$$\mathbb{E}[\boldsymbol{x}_i^T\,\boldsymbol{x}_j] = \sum_{p=1}^{P} \mathbb{E}[X_{ip}\,X_{jp}] = \sum_{p=1}^{P} \left(\mathrm{Cov}[X_{ip},\,X_{jp}] + \mathbb{E}[X_{ip}]\mathbb{E}[X_{jp}]\right)$$

- $\mathrm{Cov}[X_{ip},\,X_{jp}] = A_{ij} \times 2\,f_p\,(1 - f_p)$, où :
  - A_{ij} est la relation génétique additive entre les individus i et j, égale à deux fois leur coefficient de simple apparentement (ϕ_A),
  - f_p sont les fréquences alléliques des P SNPs
- Par ailleurs, $\mathbb{E}[X_{ip}] = 2\,f_p$

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

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

- 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 :

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

- Parmi plusieurs estimateurs d'apparentement, celui proposé par VanRaden est considéré comme un choix robuste (Toro *et al.*, 2011)

- Ainsi, le modèle de régression ‘ridge’ est équivalent au modèle suivant

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

- Cette équivalence permet d’utiliser le modèle de régression ‘ridge’ pour :
  - estimer les effets alléliques, $\hat{\boldsymbol{\beta}}$, et leur variance, $\hat{\sigma}^2_\beta$
  - prédire les valeurs génotypiques additives, $\hat{\boldsymbol{a}} = X\hat{\boldsymbol{\beta}}$
  - estimer la composante génétique additive de la variance, $\hat{\sigma}^2_a = \hat{\sigma}^2_\beta \times 2\sum_p f_p(1 - f_p)$
  - estimer l’héritabilité au sens strict, $\hat{h}^2 = \frac{\hat{\sigma}^2_a}{\hat{\sigma}^2_g + \hat{\sigma}^2}$

Simuler des données

Initialisation

- On fixe 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
```

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)
head(inds.id)
```

```
## [1] "ind001" "ind002" "ind003" "ind004" "ind005" "ind006"
```

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

```
## [1] "snp0001" "snp0002" "snp0003" "snp0004" "snp0005" "snp0006"
```

```
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))
dim(X)
```

```
## [1]  500 5000
```

```
X[1:5, 1:5]
```

```
##        snp0001 snp0002 snp0003 snp0004 snp0005
## ind001       1       0       1       1       0
## ind002       0       2       0       1       2
## ind003       0       0       1       0       1
## ind004       0       1       1       2       1
## ind005       1       0       1       0       1
```

- 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 la formule précédente de A_{mark}

```
A.mark <- (X %*% t(X)) / (2 * sum(afs * (1 - afs)))
A.mark[1:5, 1:5]
```

```
##        ind001 ind002 ind003 ind004 ind005
## ind001  1.900  0.925  0.865  0.862  0.836
## ind002  0.925  1.934  0.881  0.908  0.879
## ind003  0.865  0.881  1.861  0.860  0.860
## ind004  0.862  0.908  0.860  1.876  0.866
## ind005  0.836  0.879  0.860  0.866  1.836
```

- Une simulation moins simpliste avec du déséquilibre de liaison nécessiterait un véritable scénario évolutif
- Cela peut se faire par exemple en utilisant le processus stochastique du coalescent avec recombinaison
- Cf. “Simulations of population structure using the coalescent with recombination”

Effets additifs des allèles

Caractère monogénique

- On choisi l'unique 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] "snp1755"
```

- On fixe 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
head(beta.mono)
```

```
## snp0001 snp0002 snp0003 snp0004 snp0005 snp0006
##       0       0       0       0       0       0
```

```
table(beta.mono)
```

```
## beta.mono
##    0    3
## 4999    1
```

Effets additifs des allèles

Caractère polygénique

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

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

```
##  snp0001  snp0002  snp0003  snp0004  snp0005  snp0006
## -0.01199 -0.03162 -0.03751  0.00608  0.05056  0.00357
```

Erreurs

- On fixe la moyenne globale, et on simule les erreurs

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

Phénotypes

- Les phénotypes, $\boldsymbol{y}$, sont calculés à partir de la formule

$$\boldsymbol{y} = \mathbf{1}\mu + X\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

- Seul le vecteur des effets alléliques additifs, $\boldsymbol{\beta}$, est différent selon l'architecture génétique concernée

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

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

- Dans le cas du caractère polygénique, 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 $\boldsymbol{a} \sim \mathcal{N}_N(\mathbf{0}, \sigma_a^2 A_{\text{mark}})$

- Sous R, en utilisant la fonction `mvrnorm` du package MASS :

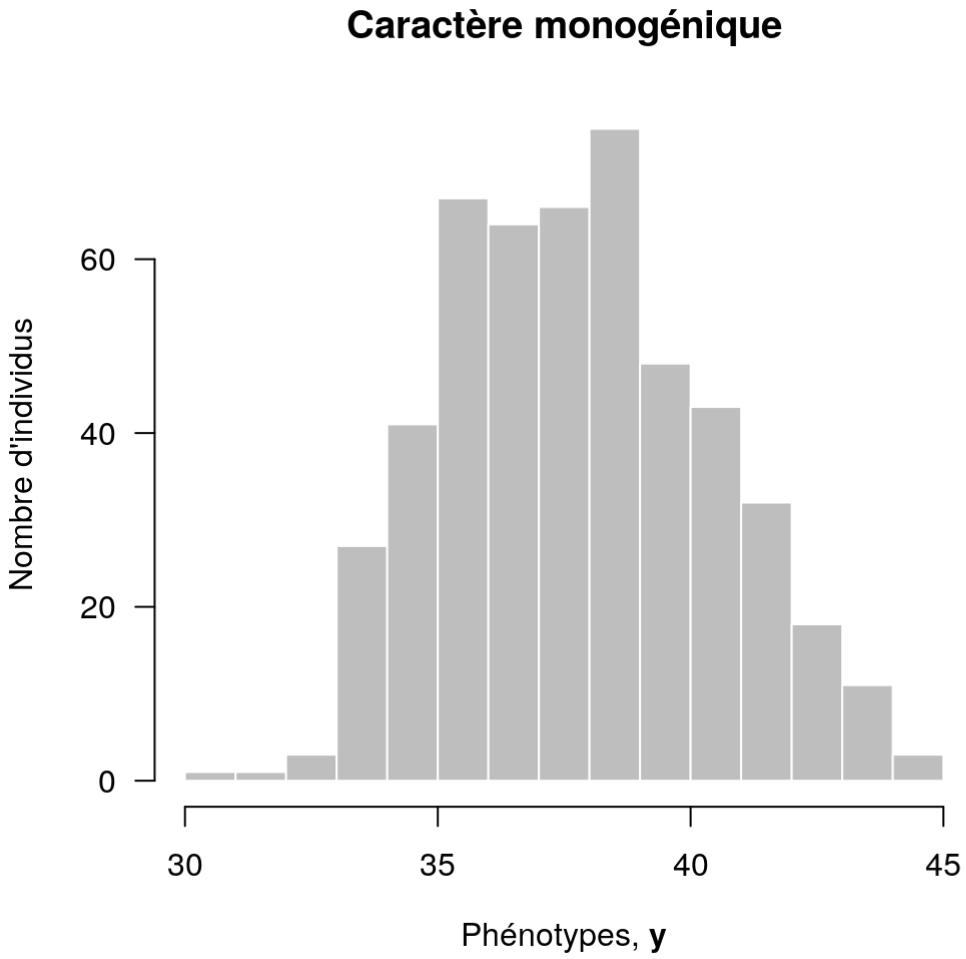
```
if(requireNamespace("MASS", quietly=TRUE)){
  a <- MASS::mvrnorm(n=1, mu=rep(0, N), Sigma=sigma.a2 * A.mark)
  g <- a
  y.poly <- matrix(1, nrow=N) * mu + g + epsilon
}
```

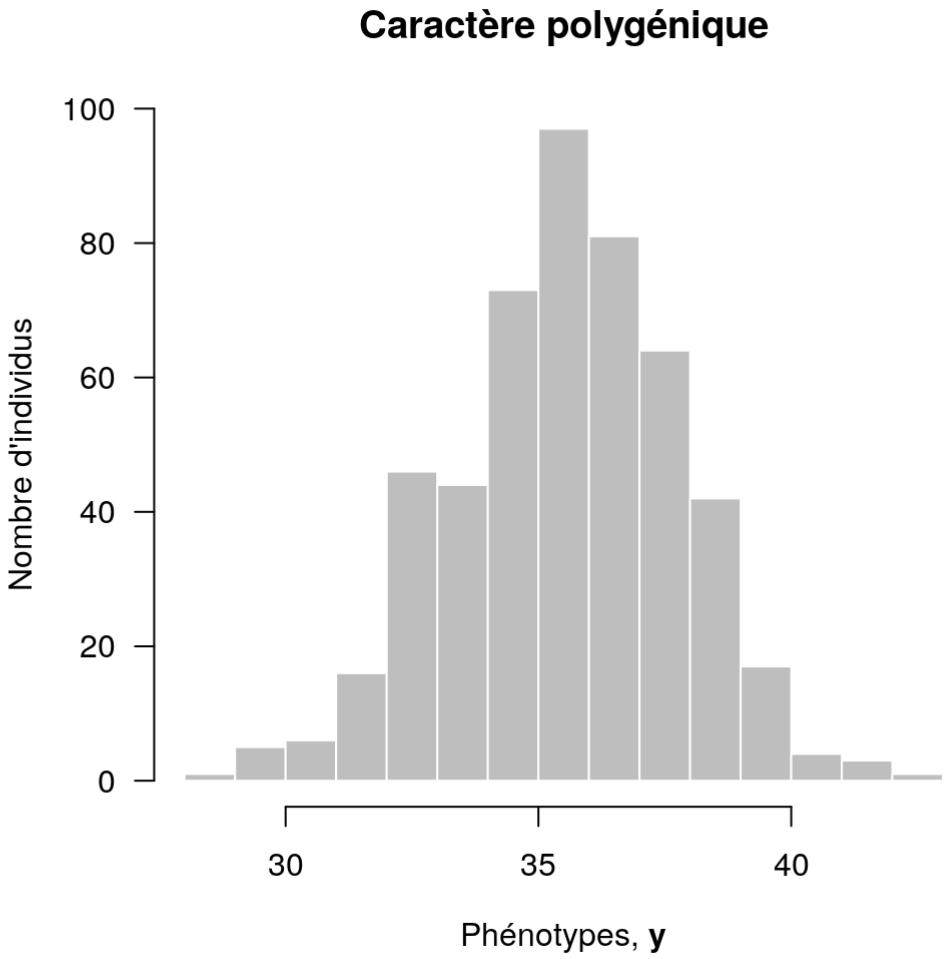
Réaliser l’inférence

Visualisation graphique

Phénotypes

```
par(mfrow = c(1, 2))
hist(y.mono, breaks="FD", las=1, col="grey", border="white",
     main="Caractère monogénique", ylab="Nombre d'individus",
     xlab=expression(paste("Phénotypes, ", bold(y))))
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 monogénique
Nombre d'individus
0
20
40
60
30
35
40
45
Phénotypes, y


Caractère polygénique
Nombre d'individus
0
20
40
60
80
100
30
35
40
Phénotypes, y

SNP à SNP (“GWAS”)

- On utilise le package MM4LMM qui implémente un algorithme MM pour ajuster le modèle par ML ou ReML
- Caractère monogénique

```
out.mmest <- MMEst(Y=y.mono[,1], X=X,
                   VarList=list(Additive=A.mark, Error=diag(N)))
out.anovatest <- AnovaTest(out.mmest, Type="TypeI")
res.mono.gwas <- sapply(out.anovatest, function(x){x["Xeffect", "pval"]})
```

- Caractère polygénique

```
out.mmest <- MMEst(Y=y.poly[,1], X=X,
                   VarList=list(Additive=A.mark, Error=diag(N)))
out.anovatest <- AnovaTest(out.mmest, Type="TypeI")
res.poly.gwas <- sapply(out.anovatest, function(x){x["Xeffect", "pval"]})
```

Tous les SNPs conjointement (“ridge”)

- On utilise le package rrBLUP
- Caractère monogénique

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

- Caractère polygénique

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

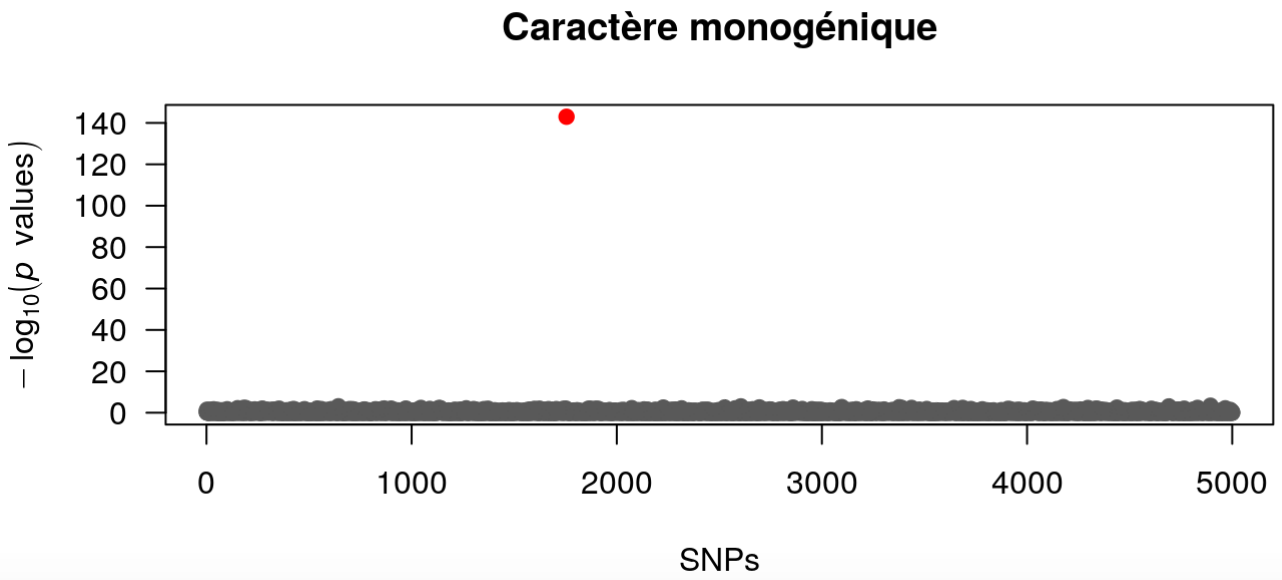
Evaluer les résultats

Manhattan plot

- La manière habituelle de regarder les résultats des tests du modèle d'inférence SNP à SNP (GWAS) est de tracer un *Manhattan plot*
- 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-après

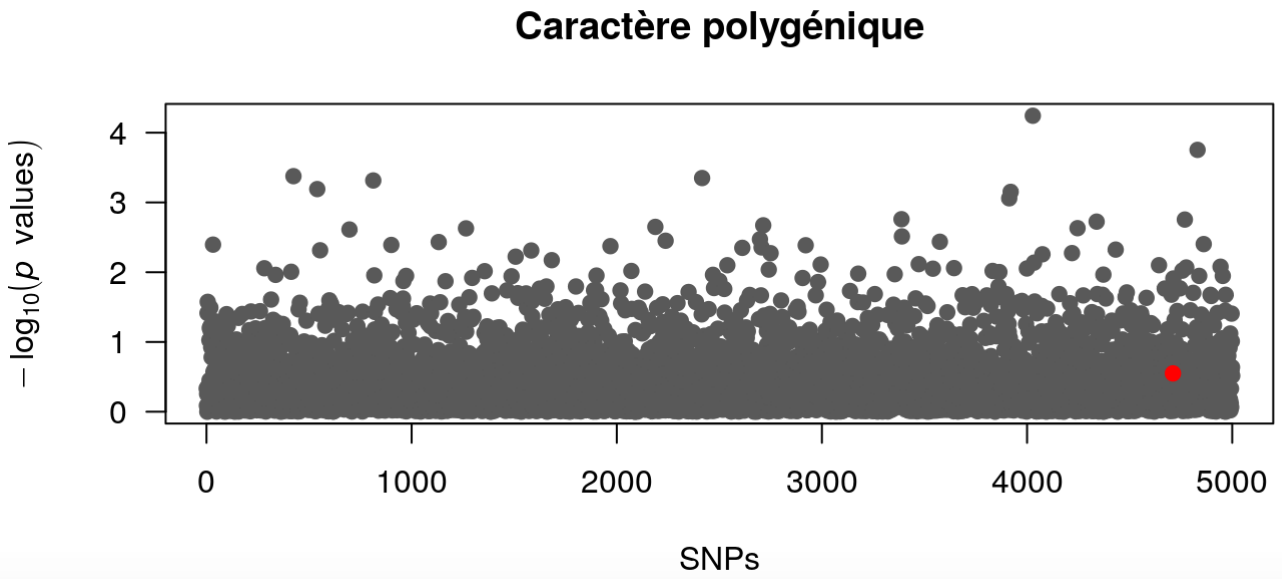
- Caractère monogénique

```
plot(x=1:P, y=-log10(res.mono.gwas),
     main="Caractère monogénique", las=1, type="n",
     xlab="SNPs", ylab=expression(-log[10](italic(p)~values)))
idx <- which(names(res.mono.gwas) == snp.qtl)
points(x=which(names(res.mono.gwas) != snp.qtl),
       y=-log10(res.mono.gwas[-idx]), col="grey35", pch=19)
points(x=idx, y=-log10(res.mono.gwas[idx]), col="red", pch=19)
```



- Caractère polygénique

```
plot(x=1:P, y=-log10(res.poly.gwas),
     main="Caractère polygénique", las=1, type="n",
     xlab="SNPs", ylab=expression(-log[10](italic(p)~values)))
idx <- which(names(res.poly.gwas) == names(which.max(beta.poly)))
points(x=which(names(res.poly.gwas) != names(which.max(beta.poly))),
       y=-log10(res.poly.gwas[-idx]), col="grey35", pch=19)
points(x=idx, y=-log10(res.poly.gwas[idx]), col="red", pch=19)
```



- Le modèle d'inférence SNP à SNP parvient bien à détecter le **SNP causal** dans le cas du **caractère monogénique**
- C'est beaucoup moins clair dans le cas du **caractère polygénique** : aucun SNP ne ressort vraiment et celui avec le plus grand β n'a pas le plus grand $\hat{\beta}$…

Composantes de la variances et coefficients

- 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)
```

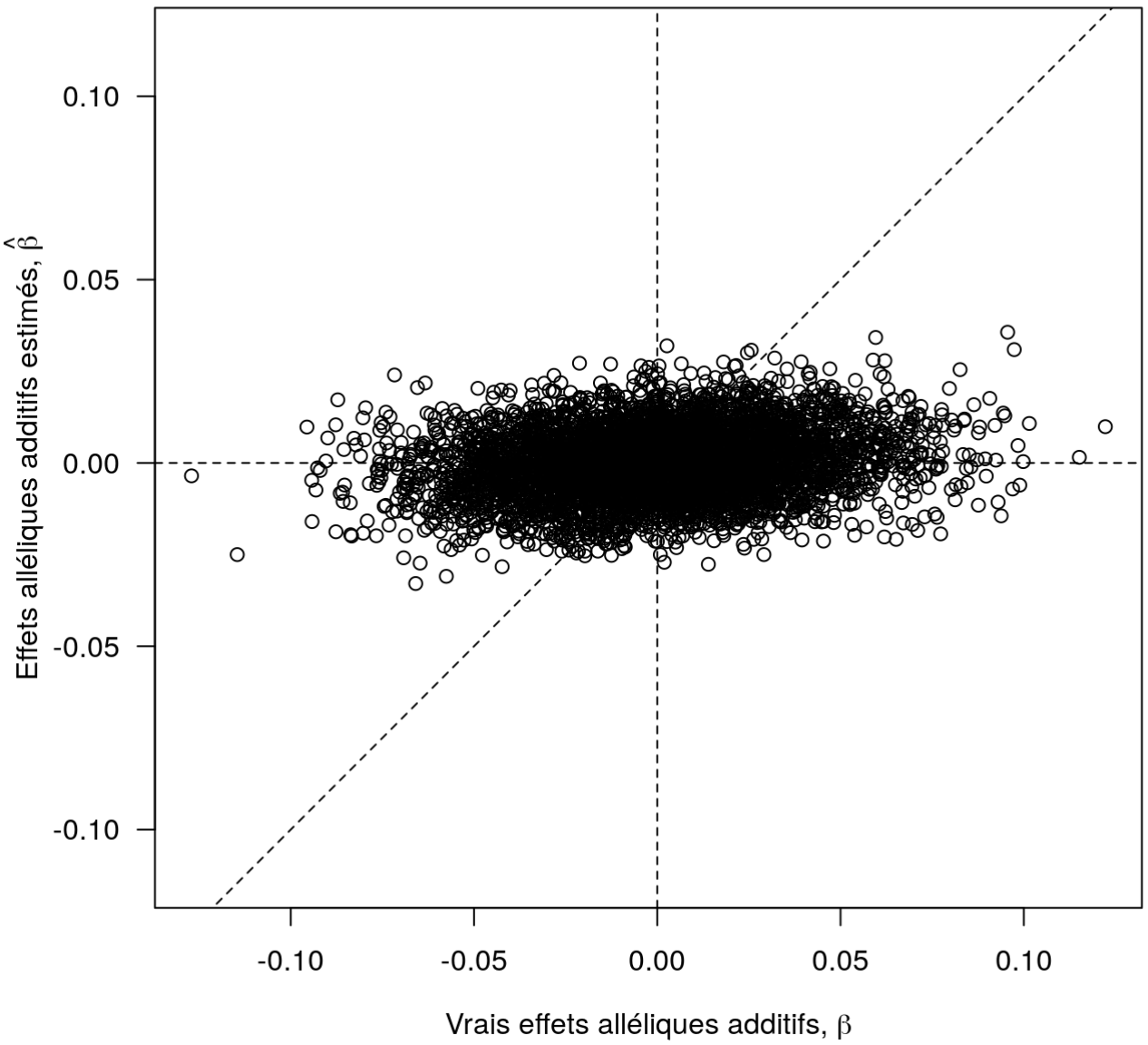
```
## [1] 0.00100 0.00144
```

- 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)
```

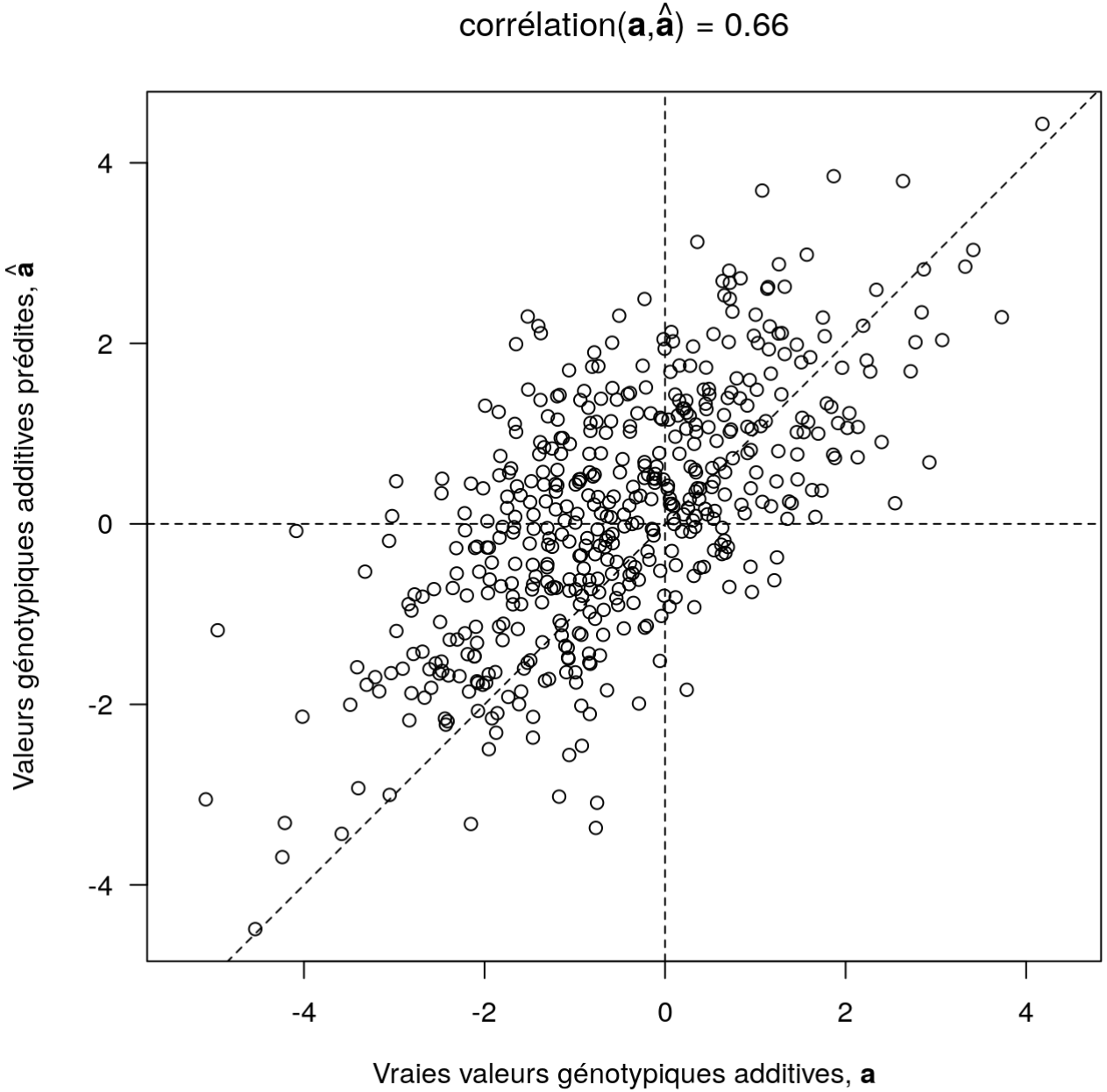

corrélation(β,β̂) = 0.19
Effets alléliques additifs estimés, β̂
Vrais effets alléliques additifs, β
0.10
0.05
0.00
-0.05
-0.10
-0.10
-0.05
0.00
0.05
0.10

- 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)
```


corrélation(**a**,$\hat{\mathbf{a}}$) = 0.66
Valeurs génotypiques additives prédites, $\hat{\mathbf{a}}$
Vraies valeurs génotypiques additives, **a**
4
2
0
-2
-4
-4
-2
0
2
4

Bilan

- Pour les caractères **polygéniques**, le modèle d'inférence SNP à SNP (GWAS) n'est pas efficace car les effets alléliques, pris individuellement, sont trop faibles
- En estimant tous les effets conjointement avec le modèle "ridge", même si chacun d'eux est biaisé, leur somme, elle, est estimée bien plus précisément
- On parle d'**effets alléliques "estimés"** et de **valeurs génotypiques "prédites"**, même si les deux sont des effets aléatoires dans les modèles mixtes
- L'une des raisons vient du fait que dans le modèle $\boldsymbol{y} = \mathbf{1}\mu + \boldsymbol{a} + \boldsymbol{\epsilon}$, les inconnues $\boldsymbol{a}$ sont les *breeding values* et les résultats $\hat{\boldsymbol{a}}$ sont les *BLUPs* des *breeding values*
- C'est la raison pour laquelle on parle de **prédiction génomique**, qui mène ensuite tout naturellement à la **sélection génomique**

Autres points importants

Eviter le sur-ajustement

- 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
- 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 validation croisée

- La variante fréquemment utilisée de cette procédure consiste à répartir aléatoirement les individus en k sous-ensembles (“folds”) de taille égale
- Pour chaque sous-ensemble k, les $k-1$ autres sont utilisés pour estimer les paramètres
- Au final, pour chaque marqueur, on dispose de k estimations de son effet allélique et on peut alors en faire la moyenne pour prédire de nouveaux individus non-phénotypés

- La validation croisée peut-être aussi utilisée pour :
  - **Sélectionner** le meilleur modèle sur le jeu d'entraînement (modèle monogénique versus modèle additif infinitésimal versus … voir ci-après)
  - Estimer une **précision** de prédiction
- Chaque individu est en fait prédit une fois (lorsqu'il ne se trouve pas dans le set d'apprentissage), on peut donc calculer à partir des phénotypes observés et prédits une **erreur quadratique moyenne de validation-croisée**
- Concernant la **précision** de prédiction, on utilise plutôt la **corrélation** (coefficient de Pearson) entre les valeurs génotypiques additives prédites ($\hat{\boldsymbol{a}}_k$) et les phénotypes corrigés pour les facteurs environnementaux ($\boldsymbol{y}_k$)
- En anglais on parle d'**accuracy** (et de *reliability* pour le carré de la corrélation)
- Il est recommandé de regarder également les estimations des moyenne globale et pente de la régression linéaire simple $\boldsymbol{y}_k = a + b\,\hat{\boldsymbol{a}}_k$

- Pour faire de la validation croisée sous R on peut utiliser le package cvTools
- Mais il requiert une méthode `predict`, qui n'est pas fournie par rrBLUP
- Il faut donc d'abord encapsuler la fonction `mixed.solve` dans une autre fonction pour qu'elle renvoie un objet de classe `"rr"`, puis 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

```
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))
}
```

- Cf. "Exemple de simulation pour explorer la prédiction génomique" pour plus de détails

Intermédiaires d'architecture génétique additive

- On a vu 2 extrêmes d'architecture génétique : monogénique *vs.* polygénique
- Mais avec de "vraies" données on ne connaît généralement pas *a priori* 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
- C'est le cas par exemple du **Lasso** qui utilise une autre norme, L^1, pour pénaliser la vraisemblance ou de l'**Elastic Net** qui combine les normes L^1 et L^2.

Au-delà de l'architecture génétique additive

- La matrice A_{mark} est proportionnelle à XX^T
- L'apparentement génétique additif entre deux individus i et j est donc $A_{ij} \propto \sum_p X_{ip} X^T_{pj} = \boldsymbol{x}_i^T \cdot \boldsymbol{x}_j$, 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
  $\boldsymbol{y} = \mathbf{1}\mu + \boldsymbol{a} + \boldsymbol{\epsilon}$ avec $\boldsymbol{\epsilon} \sim \mathcal{N}_N(\mathbf{0}, \sigma^2 I)$ et $\boldsymbol{a} \sim \mathcal{N}_N(\mathbf{0}, \sigma_a^2\, A_{\text{mark}})$
  avec A_{mark} calculée via un noyau défini dans un **espace de Hilbert à noyau reproduisant** (*Reproducing Kernel Hilbert Space*, RKHS)

- Ce terme compliqué peut en fait simplement correspondre un noyau gaussien tel que $A_{ij} = \exp\left(-(D_{ij} \,/\, \theta)^2\right)$ où D_{ij} est la distance euclidienne entre $\boldsymbol{x}_i$ et $\boldsymbol{x}_j$ normalisée dans l'intervalle $[0, 1]$ et θ est un paramètre d'échelle (qui doit être estimé par validation croisée).

- Le package rrBLUP permet d'utiliser ce noyau.

Perspectives

Explorer les simulations possibles

- Voici quelques exemples de 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'apparentement entre individus du jeu d'entraînement et individus du jeu de test ?

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 dérégréssés)
  - cleveland *et al* (g3, 2012): porc (3534 animaux, pédigrée, 5 caractères, 53000 marqueurs snp)

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

Annexe

```
print(sessionInfo(), locale=FALSE)
```

```
## R version 3.6.2 (2019-12-12)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 19.10
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.8.0
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.8.0
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] rrBLUP_4.6.1 MM4LMM_2.0.1
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.3      lattice_0.20-38 digest_0.6.24   MASS_7.3-51.5
##  [5] grid_3.6.2      magrittr_1.5    evaluate_0.14   rlang_0.4.4
##  [9] stringi_1.4.5   Matrix_1.2-18   rmarkdown_2.1   tools_3.6.2
## [13] stringr_1.4.0   xfun_0.12       yaml_2.2.1      parallel_3.6.2
## [17] compiler_3.6.2  htmltools_0.4.0 knitr_1.28
```