Prédiction génomique

Timothée Flutre

12/02/2016

Abstract

Ce document a pour but d'explorer par simulation l'intérêt de la prédiction génomique en sélection artificielle pour la création variétale.

Contents

1 Contexte

Ce document fait partie de l'atelier "Prédiction Génomique" organisé et animé par Jacques David et Timothée Flutre en 2016 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).

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(MASS))
suppressPackageStartupMessages(library(QTLRel))
suppressPackageStartupMessages(library(rrBLUP))
suppressPackageStartupMessages(library(BGLR))
```

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 pour distinguer deux organismes biologiques, animaux ou végétaux, ayant des génomes "suffisamment" différents (pas des clones).

Pour chaque individu i, on écrit:

$$y_i = g_i + \epsilon_i \tag{1}$$

où:

- i: indice du $i$ème individu parmi les N qui composent l'échantillon ($i \in \{1, \ldots, N\}$);
- y_i: valeur phénotypique de l'individu i, considérée comme continue;
- g_i: valeur génotypique de l'individu i (peut être interprétée comme la valeur phénotypique moyenne de l'individu s'il est cloné dans tous les environnements possibles);
- ϵ_i: composante non-génétique pour l'individu i ("déviation environnementale").

Le but de la génétique quantitative comme discipline scientifique est de quantifier la part de la variation phénotypique au sein de la population expliquée par la composante génétique. Ceci passe par la quantification de la valeur génotypique de chaque individu, celle-ci pouvant être interprétée comme étant le phénotype de l'individu moyenné sur tous les environnements possibles. La valeur génotypique est donc d'intérêt pour le sélectionneur qui peut s'en servir comme critère pour trier des individus. Au cours d'un programme de sélection, cycle après cycle, l'augmentation de la valeur génétique moyenne est communément appelée "progrès génétique".

Si l'on suppose que les valeur génotypique et composante non-génétique ne sont pas corrélées, alors la variance phénotypique est égale à $\sigma_p^2 = \sigma_g^2 + \sigma_\epsilon^2$. A ce stade, l'héritabilité au sens large est définit 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: $g_i = a_i + d_i + \zeta_i$. 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 par $h^2 = \frac{\sigma_a^2}{\sigma_g^2+\sigma_\epsilon^2}$.

Le même modèle, mais en notation matricielle:

$$\boldsymbol{y} = \boldsymbol{g} + \boldsymbol{\epsilon} \tag{2}$$

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

La matrice G, dite "d'apparentement", se décompose aussi en relations additives, de dominance et d'épistasie, même si les premières sont généralement les seules utilisées en pratique. 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. De plus, 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.

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) 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'échantillonage 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 l'on dispose des génotypes $\{\boldsymbol{x}_i\}$ à un ensemble de P marqueurs génétiques, le modèle devient:

$$y_i = g(\boldsymbol{x}_i, \boldsymbol{\beta}) + \epsilon_i \tag{3}$$

où l'erreur est différente du modèle précédent, mais gardons la même notation par simplicité.

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

Avec le toujours plus grand débit des technologies de séquençage, il est très fréquent qu'il y ait beaucoup plus de marqueurs que d'individus: $N << P$. Dans de tels cas, la méthode traditionnelle du maximum de vraisemblance présentée dans le document "premiers-pas.pdf" ne donne plus de bonnes estimations des effets des allèles. On dit que la vraisemblance doit être "pénalisée" (on dit aussi "régularisée"). On parle encore de "rétrécir" les estimations des effets (shrinkage en anglais).

Explicitement incorporer les effets de dominance, et surtout d'épistasie, semble infaisable étant donné l'explosion combinatoire qui en résulte. Certains auteurs ont donc proposé des modèles semi-paramétriques (espace de Hilbert à noyau reproduisant, RKHS en anglais; réseaux neuronaux).

Quoi qu'il en soit, une abondance d'articles existe sur ces sujets (voir les revues 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.pdf":

- N: nombre d'individus (diploïdes, plus ou moins apparentés)
- i: indice indiquant le i-ème individu, donc $i \in \{1, \ldots, N\}$
- y_i: phénotype de l'individu i pour la 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; les génotypes de l'individu i à tous les SNPs sont réunis dans le vecteur $\boldsymbol{x}_i^T$ et les génotypes du SNP p pour tous les individus sont réunis dans le vecteur $\boldsymbol{x}_p$;
- β_p: effet additif de chaque copie de l'allèle minoritaire du SNP p en unité du phénotype; tous ces effets sont réunis dans le vecteur $\boldsymbol{\beta}$
- ϵ_i: erreur pour l'individu i
- σ^2: variance des erreurs

Données: $\mathcal{D} = \{(y_1|\boldsymbol{x}_1), \ldots, (y_N|\boldsymbol{x}_N)\}$

Paramètres: $\Theta = \{\mu, \boldsymbol{\beta}, \sigma\}$

3.2 Vraisemblances d'extrêmes d'architecture génétique

L'architecture génétique se définit comme étant la fonction reliant les génotypes des individus de la population à leurs phénotypes (genotype-phenotype map en anglais).

A l'échelle de l'individu, son étude vise à découvrir quel est le gène ou quels sont les gènes impliqué(s) directement dans la construction d'un caractère donné et, surtout, à décrypter les mécanismes sous-jacents.

A l'échelle de la population, son étude vise à quantifier la part de variation phénotypique contribuée par la variation génotypique au(x) gène(s) impliqué(s) directement 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 aux cas "simples" (effets additifs, un seul caractère continu, etc) et en considérant deux cas extrêmes d'architecture génétique.

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 mono-génique*. Donc, si l'on teste chaque SNP un par un à la manière du document "premier-pas.pdf", on devrait pouvoir identifier ce SNP particulier:

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

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 à la covariance peut se faire en ajoutant un effet dit *aléatoire*. Alors que, jusqu'à maintenant, seule la moyenne des phénotypes était modélisée, maintenant sa variance-covariance l'est aussi, et on écrit le *modèle mixte* suivant:

$$\forall p,\ \boldsymbol{y} = \mathbf{1}\mu + \boldsymbol{x}_i\beta_p + \boldsymbol{u} + \boldsymbol{\epsilon} \text{ avec } \boldsymbol{\epsilon} \sim \mathcal{N}_N(\mathbf{0}, \sigma^2 I) \text{ et } \boldsymbol{u} \sim \mathcal{N}_N(\mathbf{0}, \sigma_u^2 K) \tag{5}$$

où K dénote cette fameuse matrice d'apparentement génétique, et (σ_u^2, σ^2) sont appelés *composants de la variance*.

En supposant $Cov(\boldsymbol{u}, \boldsymbol{\epsilon}) = 0$, on obtient: $Var(\boldsymbol{y}) = Var(\boldsymbol{u}) + Var(\boldsymbol{\epsilon}) = \sigma_u^2 K + \sigma^2 I$.

Si l'on connaît le pédigree reliant tous les individus, il est possible de calculer les apparentements deux-à-deux attendus. Sinon, il faut utiliser les marqueurs. Notons X_0 la matrice contenant les génotypes codés en $\{-1, 0, 1\}$ pour faciliter les calculs (la différence entre utiliser X ou X_0 est capturée par la moyenne globale μ).

Voici un exemple avec 3 individus et 4 SNPs:

```
## X =
```

```
##      snp1 snp2 snp3 snp4
## ind1    0    1    0    1
## ind2    0    1    1    0
## ind3    2    0    0    0
```

```
## X0 =
```

```
##      snp1 snp2 snp3 snp4
## ind1   -1    0   -1    0
## ind2   -1    0    0   -1
## ind3    1   -1   -1   -1
```

La matrice $X_0 X_0^T$ est alors symmétrique de dimension $N \times N$. Sur la diagonale, elle contient le nombre de locus homozygotes pour chaque individu; hors de la diagonale, elle contient le nombre d'allèles partagés par chaque paire d'individus apparentés:

```
## X0 X0^T =
```

```
##      ind1 ind2 ind3
## ind1    2    1    0
## ind2    1    2    0
## ind3    0    0    4
```

Ce modèle (5) a cependant le désavantage d'utiliser les marqueurs pour estimer l'apparentement, tout en testant leurs effets par ailleurs, 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 explicitemenent tous les marqueurs dans la régression.

Dans le deuxième cas extrême d'architecture génétique, tous les SNPs ont un effet non-nul. On parle alors de *caractère polygénique*. Comme il y a vraiment beaucoup de SNPs ($P >> N$), l'hypothèse habituelle est qu'ils ont tous des effets très faibles. Donc chercher à les estimer individuellement n'est pas une stratégie

pertinente. Il vaut mieux viser à estimer leur effet global, par exemple en supposant qu'ils s'additionnent tous: $\sum_{p=1}^{P} \boldsymbol{x}_p \beta_p = X\boldsymbol{\beta}$. On parle alors d'architecture génétique additive infinitésimale. De plus, sans connaissance plus précise a priori de l'architecture génétique du caractère en question, il est habituel de supposer que les effets sont tous indépendants (attention, les génotypes, eux, ne sont généralement pas indépendants à cause du déséquilibre de liaison). Au final, le modèle mixte s'écrit:

$$\boldsymbol{y} = \mathbf{1}\mu + X\boldsymbol{\beta} + \boldsymbol{\epsilon} \text{ avec } \boldsymbol{\epsilon} \sim \mathcal{N}_N(\mathbf{0}, \sigma^2 I) \text{ et } \boldsymbol{\beta} \sim \mathcal{N}(\mathbf{0}, \sigma^2_\beta I) \tag{6}$$

En modélisation statistique, ce modèle est connu sous le nom de régression d'arête (ridge regression en anglais).

En supposant $Cov(X\boldsymbol{\beta}, \boldsymbol{\epsilon}) = 0$, on obtient: $Var(\boldsymbol{y}) = Var(X\boldsymbol{\beta}) + Var(\boldsymbol{\epsilon}) = \sigma^2_\beta XX^T + \sigma^2 I$; où nous avons utilisé la formule mathématique $Var(A\boldsymbol{x}) = A\,Var(\boldsymbol{x})\,A^T$ car c'est l'équivalent matriciel de $Var(a\,x) = a^2\,Var(x)$ lorsque a (A) est un coefficient (matrice) connu(e) et x ($\boldsymbol{x}$) est une variable (vecteur) aléatoire.

Mais surtout, remarquez que XX^T apparaît ici aussi! Il s'avère qu'en considérant les génotypes dans X comme étant aléatoires, il est possible de prouver que l'espérance $E(XX^T)$ tend vers $A \times 2\sum_p f_p(1-f_p)$ à une constante prêt, où A est la matrice d'apparentement calculée à partir du pédigree et les f_p's sont les fréquences alléliques. Un estimateur simple de l'apparentement génétique deux-à-deux à partir des génotypes aux SNPs est donc:

$$\hat{K} = \frac{XX^T}{2\sum_p f_p(1-f_p)} \tag{7}$$

Le modèle (6) s'avère être équivalent au modèle suivant avec l'apparentement estimé via (7):

$$\boldsymbol{y} = \mathbf{1}\mu + \boldsymbol{u} + \boldsymbol{\epsilon} \text{ avec } \boldsymbol{\epsilon} \sim \mathcal{N}_N(\mathbf{0}, \sigma^2 I) \text{ et } \boldsymbol{u} \sim \mathcal{N}_N(\mathbf{0}, \sigma^2_u \hat{K}) \tag{8}$$

Une estimation de l'héritabilité au sens strict peut s'obtenir via: $h^2 = \frac{\sigma^2_u}{\sigma^2_u + \sigma^2}$.

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){
  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))
X0 <- X - 1
```

La matrice d'apparentement peut s'estimer avec (7):

```
K <- X0 %*% t(X0)
```

4.2 Effets des allèles, erreurs, puis phénotypes

Dans tous les cas, calculons les phénotypes, $\boldsymbol{y}$, à partir de la formule (6). Seul le vecteur d'effets aux marqueurs, $\boldsymbol{\beta}$, sera différent.

- Caractère monogénique:

Commençons par choisir le SNP causal, avec une fréquence ni trop faible ni trop élevée:

```
afs <- colMeans(X) / 2 # fréquences alléliques
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 à une valeur élevée, les autres SNPs ayant un effet nul:

```
beta.mono <- setNames(rep(0, P), snps.id)
beta.mono[snp.qtl] <- 4
```

Enfin, fixons la moyenne globale, simulons les erreurs et calculons les phénotypes:

```
mu <- 36
sigma.epsilon2 <- 1
epsilon <- matrix(rnorm(n=N, mean=0, sd=sqrt(sigma.epsilon2)))
y.mono <- matrix(1, nrow=N) * mu + X0 %*% beta.mono + epsilon
```

- Caractère polygénique:

Commençons par simuler les effets de tous les marqueurs:

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

Enfin, calculons les phénotypes:

```
y.poly <- matrix(1, nrow=N) * mu + X0 %*% beta.poly + epsilon
```

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

```r
sigma.u2 <- sigma.beta2.poly * 2 * sum(afs * (1 - afs))
(h2 <- sigma.u2 / (sigma.u2 + sigma.epsilon2))
```

```
## [1] 0.677
```

```r
(var(X0 %*% beta.poly) / (var(X0 %*% beta.poly) + var(epsilon)))
```

```
##       [,1]
## [1,] 0.712
```

Notez qu'on aurait aussi pu simuler les valeurs génotypiques via la distribution Normal multivariée $\boldsymbol{u} \sim \mathcal{N}_N(\mathbf{0}, \sigma_u^2 K)$. En R, en utilisant la fonction `mvrnorm` du paquet MASS, cela aurait donné:

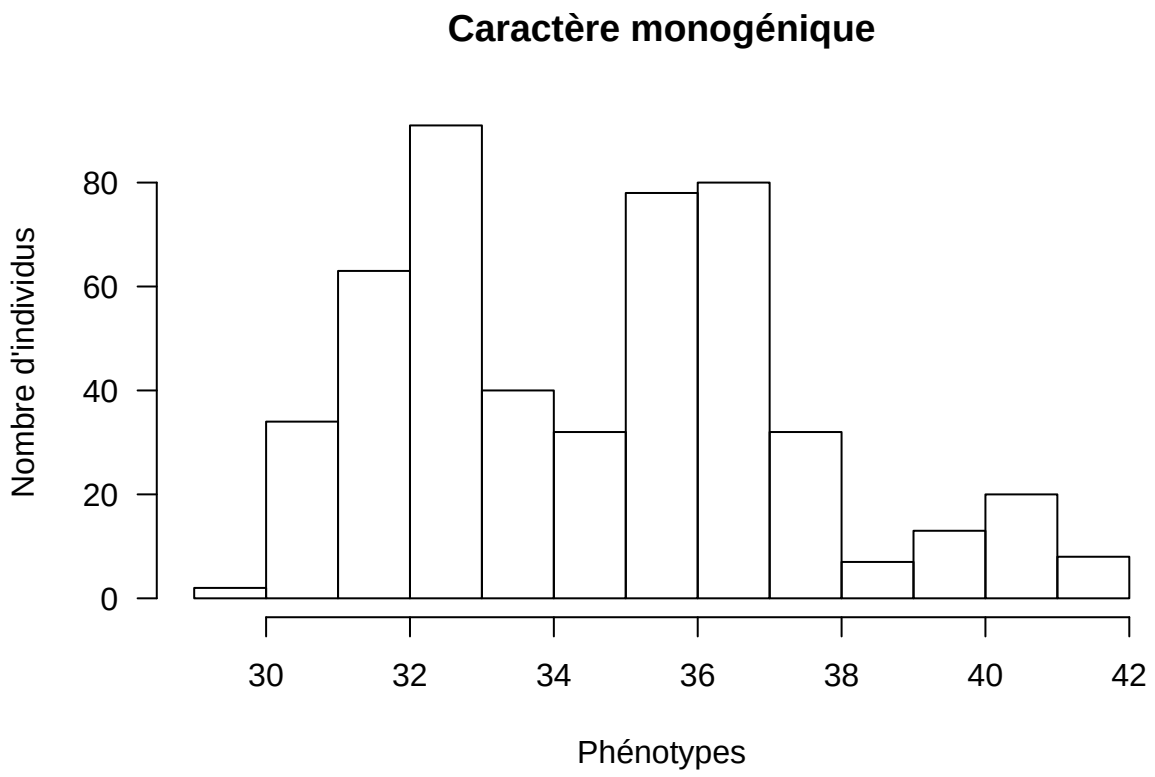
```r
u <- mvrnorm(n=1, mu=rep(0, N), Sigma=sigma.u2 * K)
y.poly <- matrix(1, nrow=N) * mu + u + epsilon
```

5 Réaliser l'inférence

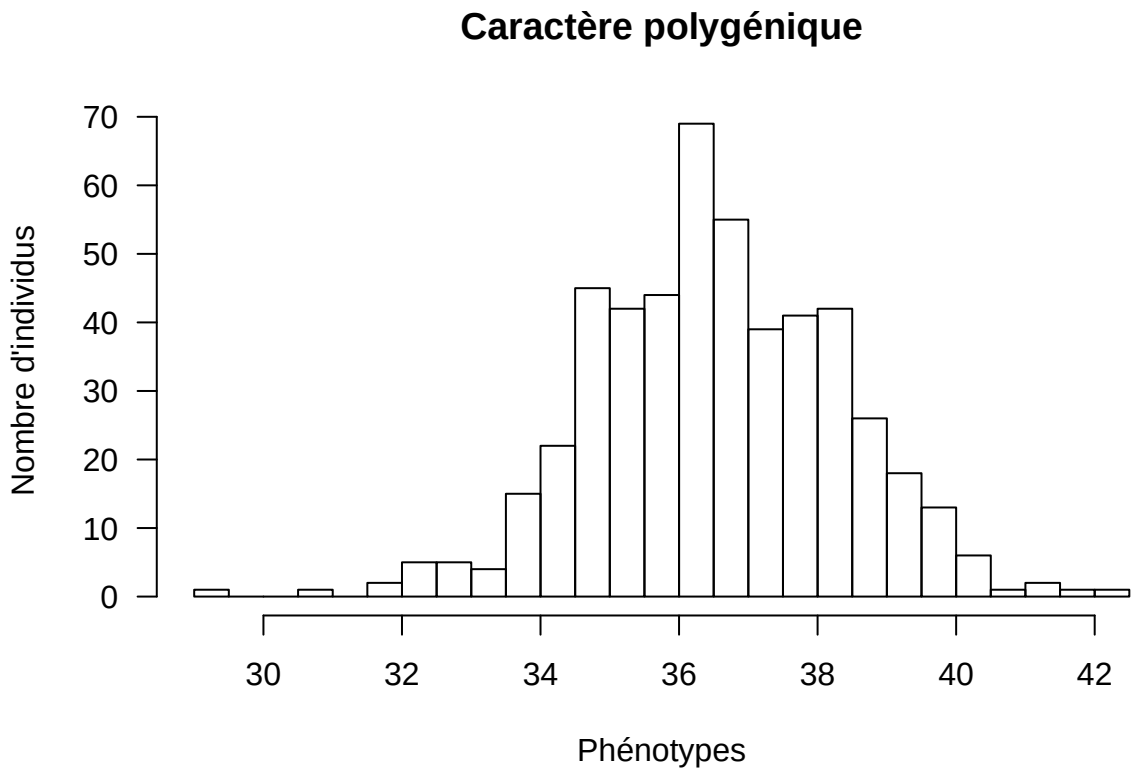
5.1 Représentation graphique

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

```r
hist(y.mono, breaks="FD", las=1, main="Caractère monogénique",
     xlab="Phénotypes", ylab="Nombre d'individus")
```



```r
hist(y.poly, breaks="FD", las=1, main="Caractère polygénique",
     xlab="Phénotypes", ylab="Nombre d'individus")
```



5.2 SNP à SNP ("GWAS")

Le paquet QTLRel implémente une méthode permettant de tester l'effet allélique SNP par SNP tout en contrôlant l'apparentement entre individus. Sur le plan statistique, il existe de meilleurs méthodes, mais celle-ci suffit aux besoins de ce document. Sur le plan informatique, comme elle est relativement lente, nous allons tester seulement un sous-ensemble des $P = 5000$ SNPs pour aller plus vite.

Echantillonnons donc uniformément un sous-ensemble de SNPs à tester (mais incluant le causal):

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

Ajustons le modèle SNP à SNP (5) sur les données monogéniques:

```r
res.mono.gwas <- list()
res.mono.gwas$vc <- estVC(y=y.mono, v=list(AA=K, DD=NULL, HH=NULL, AD=NULL,
                                           MH=NULL, EE=diag(N)))
res.mono.gwas$scan <- scanOne(y=y.mono, gdat=X0[,subset.snps],
                              vc=res.mono.gwas$vc, test="F", numGeno=TRUE)
```

Ajustons ce même modèle sur les données polygéniques:

```
res.poly.gwas <- list()
res.poly.gwas$vc <- estVC(y=y.poly, v=list(AA=K, DD=NULL, HH=NULL, AD=NULL,
                                           MH=NULL, EE=diag(N)))
res.poly.gwas$scan <- scanOne(y=y.poly, gdat=X0[,subset.snps],
                              vc=res.poly.gwas$vc, test="F", numGeno=TRUE)
```

5.3 Tous les SNPs conjointement ("ridge")

Le paquet rrBLUP implémente la régression d'arête permettant d'estimer tous les effets alléliques conjointement (6).

Ajustons le modèle conjoint (6) sur les données monogéniques:

```
res.mono.ridge <- mixed.solve(y=y.mono, Z=X0)
```

Ajustons ce même modèle sur les données polygéniques:

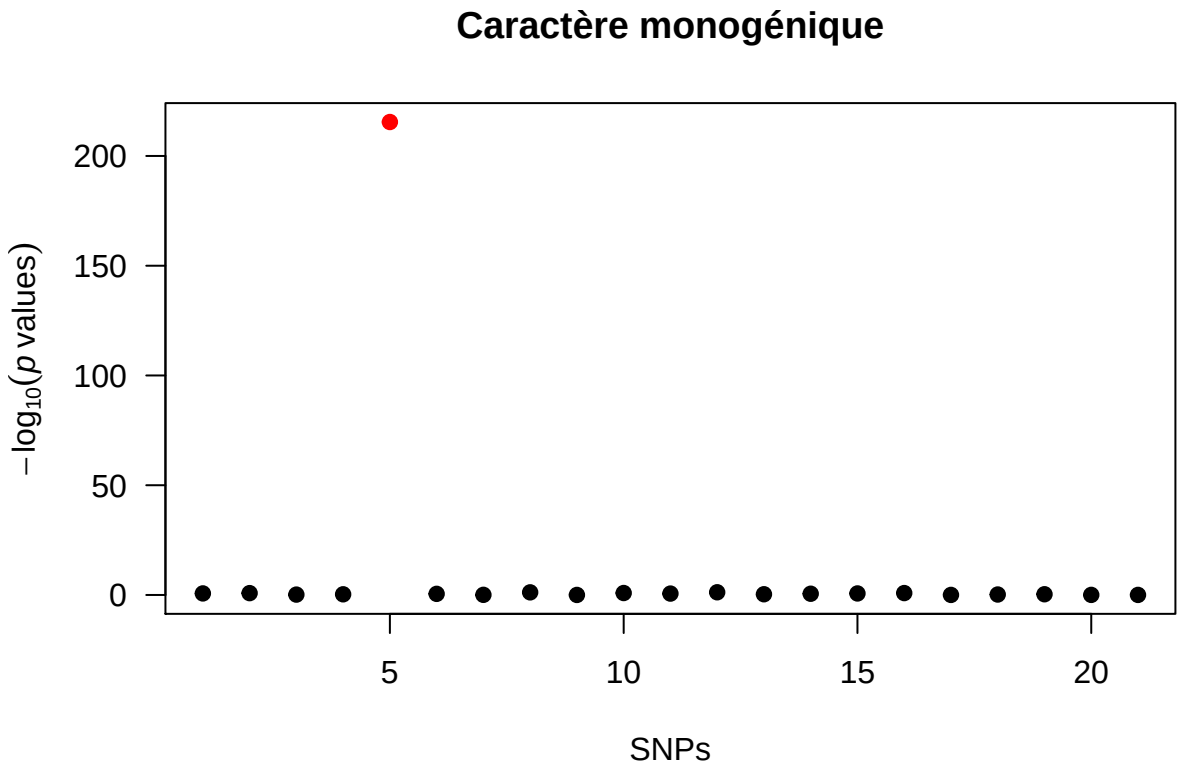
```
res.poly.ridge <- mixed.solve(y=y.poly, Z=X0)
```

6 Evaluer les résultats

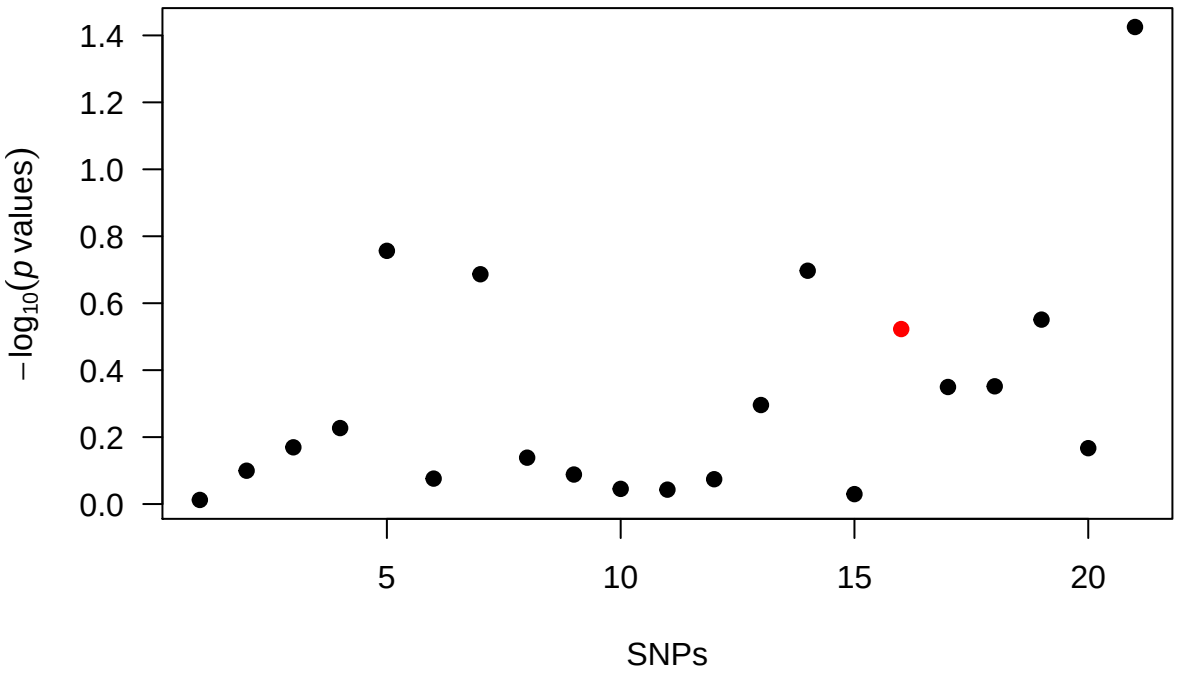
La manière habituelle de regarder les résultats des tests SNP à SNP est de tracer un "Manhattan plot".

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-dessous.

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




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



Le modèle d'inférence SNP à SNP parvient bien à détecter le SNP causal dans le cas du caractère monogénique, mais le signal est impossible à distinguer du bruit dans le cas du caractère polygénique.

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

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

```
## [1] 1.00 1.03
```

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

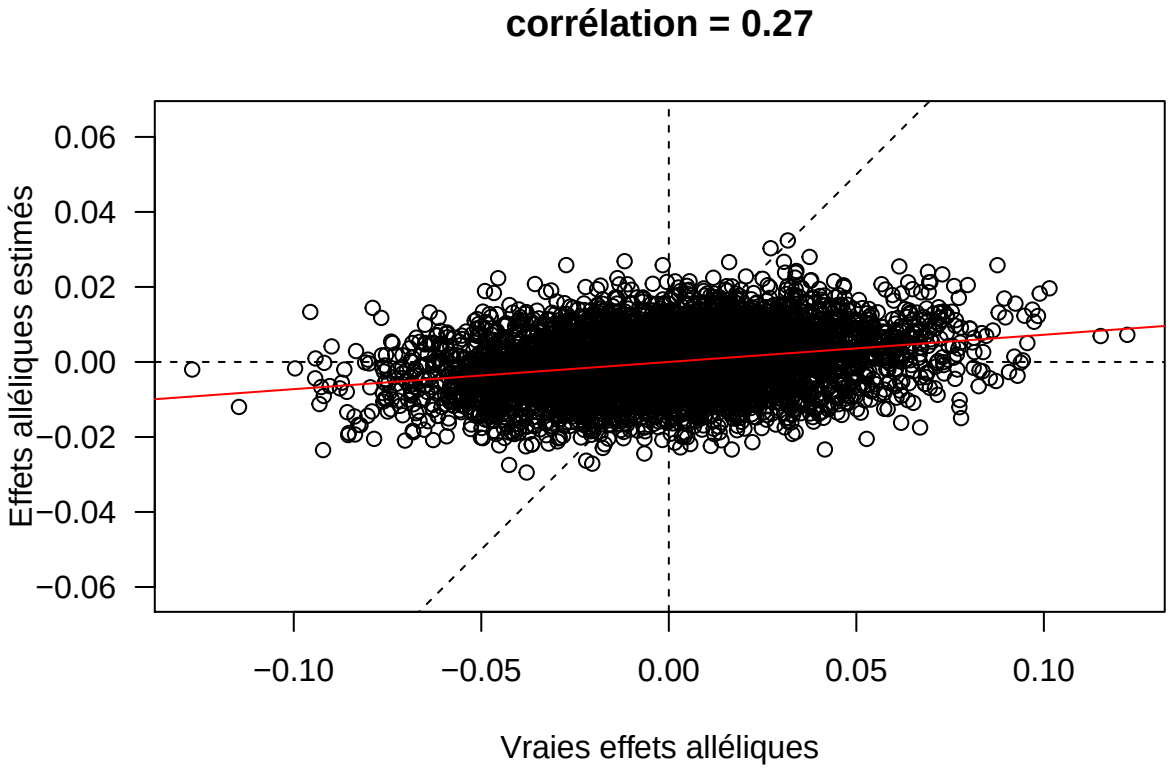
```
## [1] 0.00100 0.00106
```

Il est ensuite intéressant de remarquer que les effets aux marqueurs sont mal estimés individuellement:

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

```
## [1] 0.274
```

```
plot(beta.poly, res.poly.ridge$u, xlab="Vraies effets alléliques",
     ylab="Effets alléliques estimés", las=1, asp=1,
     main=paste0("corrélation = ", format(cor(beta.poly, res.poly.ridge$u),
                                          digits=2)))
abline(v=0, lty=2); abline(h=0, lty=2); abline(a=0, b=1, lty=2)
abline(lm(res.poly.ridge$u ~ beta.poly), col="red")
```



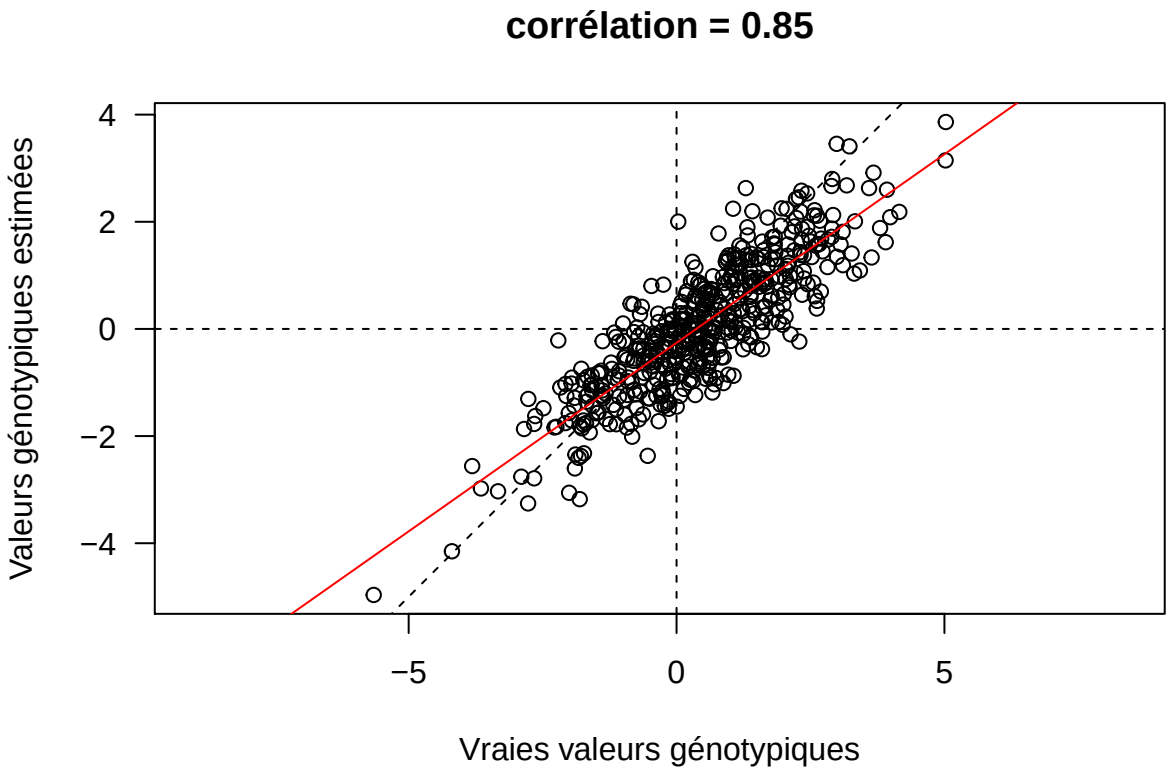
On voit bien cependant clairement l'effet du "rétrécissement" des estimations vers 0.

Par contre, les valeurs génotypiques (breeding values), elles, sont bien mieux estimées:

```
cor(X0 %*% beta.poly, X0 %*% res.poly.ridge$u)
```

```
##       [,1]
## [1,] 0.853
```

```
plot(X0 %*% beta.poly, X0 %*% res.poly.ridge$u, las=1, asp=1,
     xlab="Vraies valeurs génotypiques", ylab="Valeurs génotypiques estimées",
     main=paste0("corrélation = ", format(cor(X0 %*% beta.poly,
                                              X0 %*% res.poly.ridge$u),
                                          digits=2)))
abline(v=0, lty=2); abline(h=0, lty=2); abline(a=0, b=1, lty=2)
abline(lm(X0 %*% res.poly.ridge$u ~ X0 %*% beta.poly), col="red")
```



C'est en cela qu'analyser conjointement tous les marqueurs est pertinent. Pour les caractères polygéniques, les tests SNP à SNP ne sont pas efficaces car les effets, pris individuellement, sont trop faibles. En estimant tous les effets des SNPs conjointement, même si chacun d'eux est sous-estimé, leur somme, elle, le sera bien plus précisément.

Notez que je parle de valeurs génotypiques "estimées". En effet, ce sont des variables non-observées et non-observables, donc des paramètres qui sont à estimer. Mais dans la littérature scientifique utilisant l'interprétation fréquentiste des probabilités, on parle de valeurs génotypiques "prédites". Les inconnues $\boldsymbol{u}$ sont les *breeding values* et les résultats $\hat{\boldsymbol{u}}$ sont les *Best Linear Unbiased Predictions* (BLUPs). Dans cette littérature, les effets fixes sont *estimés* et les effets aléatoires sont *prédits*.

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

7 Intermédiaires d'architecture génétique

Nous avons vu ci-dessus comment différents modèles d'inférence sont plus ou moins performants selon l'architecture génétique d'un caractère. Mais ne pourrait-on pas avoir un seul modèle s'adaptant à toutes les architectures ?

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:

$$\boldsymbol{y} = \mathbf{1}\mu + X\boldsymbol{\beta} + \boldsymbol{\epsilon} \text{ avec } \boldsymbol{\epsilon} \sim \mathcal{N}_N(\mathbf{0}, \sigma^2 I) \text{ et } \forall p\ \beta_p \sim \pi\,\mathcal{N}(\mathbf{0}, \sigma^2_\beta) + (1-\pi)\,\delta_0 \tag{9}$$

où δ_0 est la "distribution de Dirac" qui prend la valeur 0 avec une probabilité de 1.

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 + X0 %*% beta.sparse + epsilon
```

Dans ce modèle, π représente la proportion de SNPs ayant un effet non-nul, ce paramètre visent donc à s'adapter à différentes architectures génétiques.

Le paquet BGLR implémente un tel modèle, sous le nom de "BayesB", ainsi que d'autres modèles dont notamment la régression d'arête présentée ci-dessus.

```
res.BGLR <- BGLR(y=y.sparse, ETA=list(list(X=X0, model="BayesB")),
                 verbose=FALSE, nIter=10^4, burnIn=2*10^3, thin=5)
```

Ce paquet utilise une méthode d'inférence bayésienne, l'échantillonneur de Gibbs. Mais il est trop long de détaillé cela ici.

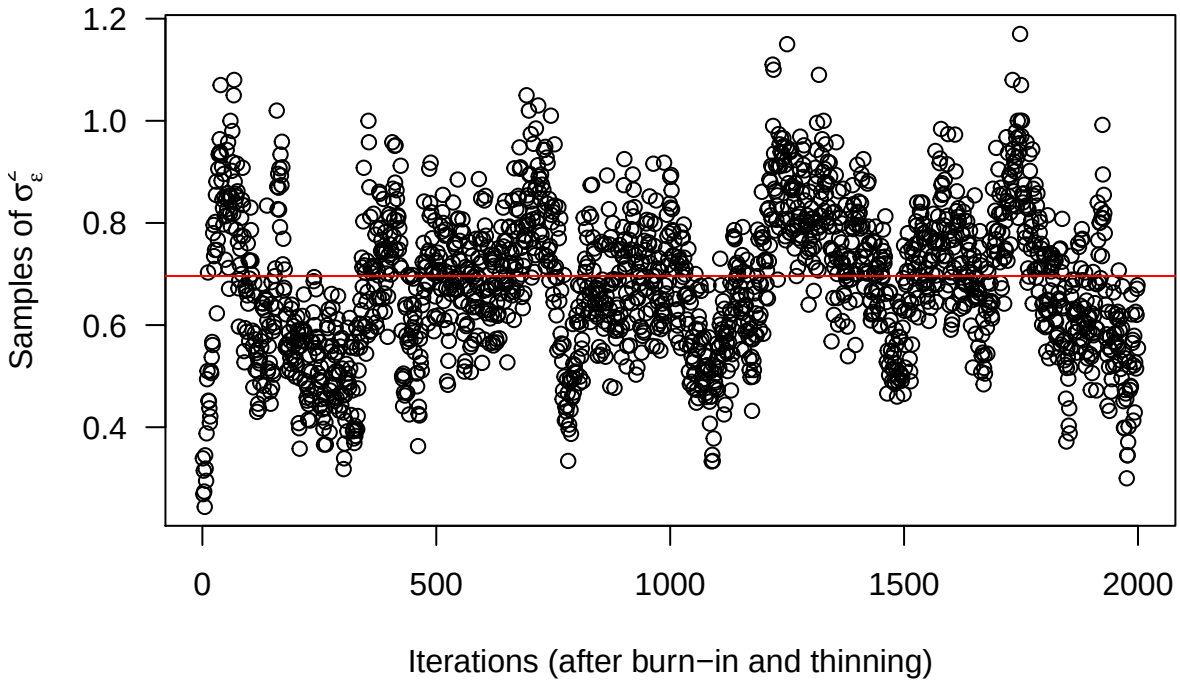
Lorsque l'on utilise un algorithme MCMC, il est toujours important de vérifier qu'il a convergé, au moins visuellement (renseignez-vous plus en détails sur cette étape importante !):

```
print(str(res.BGLR))
```

```
## List of 19
##  $ y            : num [1:500] 34.9 36.3 34.2 36.4 38 ...
##  $ whichNa      : int(0)
##  $ saveAt       : chr "/home/tflutre/work/atelier-prediction-genomique/"
##  $ nIter        : num 10000
##  $ burnIn       : num 2000
##  $ thin         : num 5
##  $ weights      : num [1:500] 1 1 1 1 1 1 1 1 1 1 ...
##  $ verbose      : logi FALSE
##  $ response_type: chr "gaussian"
##  $ df0          : num 5
##  $ S0           : num 4.31
##  $ yHat         : num [1:500] 35.4 36.2 35.1 36.1 36.9 ...
##  $ SD.yHat      : num [1:500] 0.546 0.539 0.579 0.556 0.585 ...
##  $ mu           : num 36
##  $ SD.mu        : num 0.473
##  $ varE         : num 0.696
##  $ SD.varE      : num 0.133
##  $ fit          :List of 4
##   ..$ logLikAtPostMean: num -507
##   ..$ postMeanLogLik  : num -613
##   ..$ pD              : num 213
##   ..$ DIC             : num 1440
##  $ ETA          :List of 1
##   ..$ :List of 23
##   .. ..$ model      : chr "BayesB"
##   .. ..$ Name       : chr "ETA_1"
##   .. ..$ p          : int 5000
##   .. ..$ colNames   : chr [1:5000] "snp0001" "snp0002" "snp0003" "snp0004" ...
```

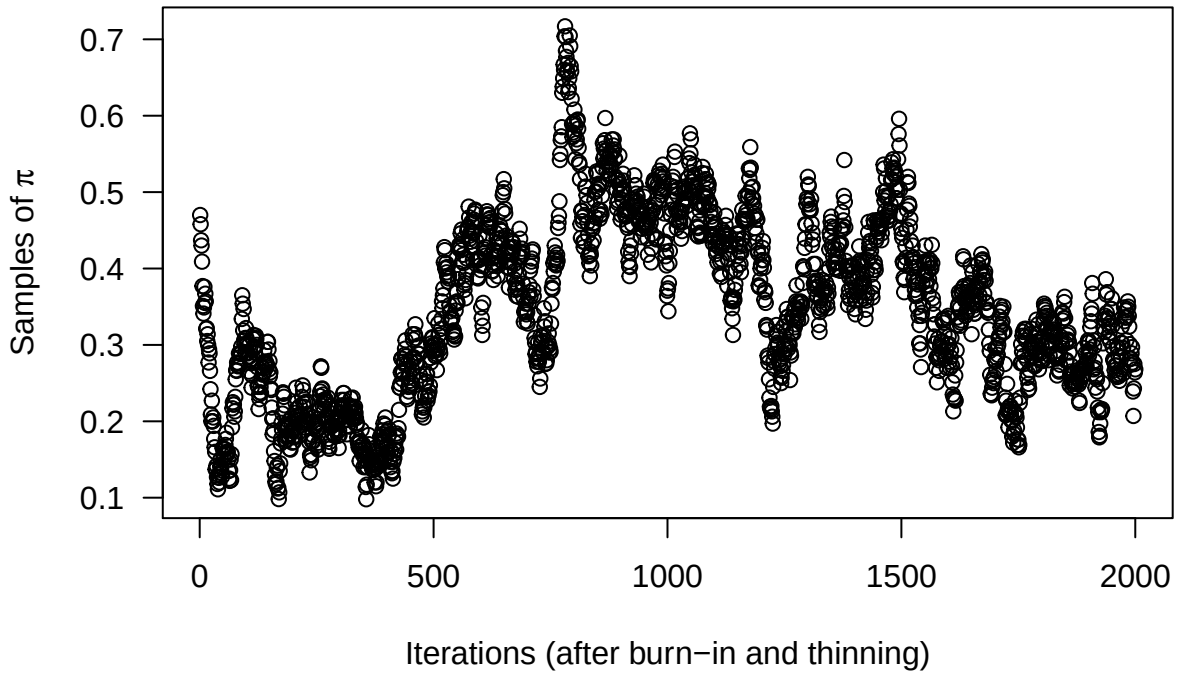
```
##   .. ..$ MSx         : num 2096
##   .. ..$ R2          : num 0.5
##   .. ..$ df0         : num 5
##   .. ..$ probIn      : num 0.381
##   .. ..$ counts      : num 10
##   .. ..$ countsIn    : num 5
##   .. ..$ countsOut   : num 5
##   .. ..$ S0          : num 0.00412
##   .. ..$ b           : Named num [1:5000] 0.008428 -0.003095 -0.001905 0.009812 -0.000218 ...
##   .. .. ..- attr(*, "names")= chr [1:5000] "snp0001" "snp0002" "snp0003" "snp0004" ...
##   .. ..$ d           : num [1:5000] 0.422 0.381 0.376 0.418 0.371 ...
##   .. ..$ shape0      : num 1.1
##   .. ..$ rate0       : num 24.3
##   .. ..$ S           : num 0.00233
##   .. ..$ varB        : num [1:5000] 0.000894 0.00072 0.000751 0.000906 0.000789 ...
##   .. ..$ NamefileOut: chr "/home/tflutre/work/atelier-prediction-genomique/ETA_1_parBayesB.dat"
##   .. ..$ SD.b        : Named num [1:5000] 0.0336 0.026 0.027 0.0342 0.0271 ...
##   .. .. ..- attr(*, "names")= chr [1:5000] "snp0001" "snp0002" "snp0003" "snp0004" ...
##   .. ..$ SD.varB     : num [1:5000] 0.001544 0.000696 0.000827 0.001348 0.001083 ...
##   .. ..$ SD.probIn   : num 0.1
##   .. ..$ SD.S        : num 0.000752
##   - attr(*, "class")= chr "BGLR"
## NULL
```

```
plot(scan("varE.dat"), las=1,
     xlab="Iterations (after burn-in and thinning)",
     ylab=expression(paste("Samples of ", sigma[epsilon]^2)))
abline(h=res.BGLR$varE, col="red")
```




```
samples.pi <- read.table("ETA_1_parBayesB.dat", header=TRUE)[, "probIn"]
plot(samples.pi, las=1,
```

```
     xlab="Iterations (after burn-in and thinning)",
     ylab=expression(paste("Samples of ", pi)))
```




```
## abline(h=res.BGLR$ETA[[1]]$varB, col="red")
```

Si l'on considère que l'algorithme a convergé, nous pouvons alors réaliser l'inférence en regardant la distribution a posteriori des paramètres:

```
c(mu, res.BGLR$mu)
c(sigma.epsilon2, res.BGLR$varE)
c(pi, res.BGLR$ETA[[1]]$varB)
cor(beta, fit.BGLR$ETA[[1]]$b)
```

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 du déséquilibre de liaison entre SNPs ?
- quel est l'impact de l'apparentement entre individus du jeu d'entraînement et individus du jeu de test ?
- etc

C'est à vous !

9 Explorer 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)

10 Perspectives

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

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: créer de nouvelles variétés combinant plusieurs caractères d'intérêt et adaptées à l'itinéraire technique, à la filière économique, à l'agriculteur et au consommateur, etc.

Mais ce sera pour le cours suivant !

11 Références

- 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

12 Annexe

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

```
## R version 3.2.2 (2015-08-14)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 14.04.3 LTS
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] BGLR_1.0.4      rrBLUP_4.4      QTLRel_0.2-15   MASS_7.3-45
## [5] knitr_1.12.3    rmarkdown_0.9.2
##
## loaded via a namespace (and not attached):
##  [1] lattice_0.20-33 gtools_3.5.0    digest_0.6.9    grid_3.2.2
##  [5] formatR_1.2.1   magrittr_1.5    evaluate_0.8    stringi_1.0-1
##  [9] gdata_2.17.0    tools_3.2.2     stringr_1.0.0   yaml_2.1.13
## [13] htmltools_0.3
```