TP 2 - Statistique bayésienne

RAMDÉ Ismaïl

Master 2 - SSD

Présentation des données

On dispose de deux fichiers de données d'expression de gènes.

- Le fichier `geneDecode` contient deux variables : le niveau d'expression des gènes le long du génome a été mesuré et est stocké dans la variable `Y`. Un biologiste a séquencé le génome en des parties sur-exprimées (correspondant à `X=2`) et des parties sous-exprimées (correspondant à `X=1`).
- Le fichier `gene` ne contient que la variable `Y` du niveau d'expression des gènes d'un autre individu. Vous représenterez également l'expression de ces gènes.

Objectif

L'objectif du projet est d'estimer un modèle expliquant le niveau d'expression des gènes. Le modèle proposé est le suivant. On note $(X_1, Y_1), \ldots, (X_n, Y_n)$ les n couples de variables aléatoires décrivant l'échantillon. La suite des valeurs prises par X est décrite par une chaine de Markov selon la loi de transition pour $i = 2, \ldots, n$:

$$X_i | X_{i-1} \sim Ber(\beta_{X_{i-1}}) + 1$$

avec deux paramètres inconnus β_1 et β_2.

La loi du niveau Y d'expression des gènes dépend de la valeur de la classe X et est décrite suivante cette loi pour $i = 1, \ldots, n$:

$$Y_i | X_i \sim \mathcal{N}(\alpha_{X_i}, \sigma^2)$$

avec trois paramètres inconnus α_1, α_2 et σ^2.

1. Estimation pour le fichier `geneDecode`

Dans cette partie, vous utiliserez le fichier `geneDecode`.

- Vous représenterez l'expression des gènes en distinguant les deux classes par des couleurs.

```
data <- load('project.Rdata')
```

- Vous proposerez un algorithme de Gibbs qui estime les lois a posteriori des paramètres $(\alpha_1, \alpha_2, \sigma^2, \beta_1, \beta_2)$. Pour cela, vous choisirez des lois a priori pertinentes et détaillerez les étapes de l'algorithme de Gibbs.
- Vous pourrez étudier la convergence de votre algorithme.
- Vous pourrez étudier l'influence des lois a priori.

2. Estimation pour le fichier gene

Dans cette partie, vous utiliserez le fichier `gene`.

- Vous représenterez l'expression des gènes.
- Vous proposerez un algorithme de Gibbs qui estime les lois a posteriori des paramètres $(\alpha_1, \alpha_2, \sigma^2, \beta_1, \beta_2)$ en simulant les variables latentes non observées X. Pour cela, vous choisirez des lois a priori pertinentes et détaillerez les étapes de l'algorithme de Gibbs.
- Vous pourrez étudier la convergence de votre algorithme.
- Vous pourrez étudier l'influence des lois a priori.
- Vous pourrez vous comparer aux résultats obtenus avec un algorithme de mélange de deux gaussiennes. Quelle est la différence entre les deux modèles en terme d'hypothèse ? en terme d'estimation obtenue ?

3. Simulation

Dans cette partie, on n'utilise aucun des deux fichiers.

- Vous proposerez une simulation d'un nouveau fichier de données avec le modèle proposé.
- Vous étudierez la sensibilité de l'algorithme de Gibbs proposé à la section 2 en fonction du choix des paramètres $(\alpha_1, \alpha_2, \sigma^2, \beta_1, \beta_2)$.

```
require(invgamma)
```

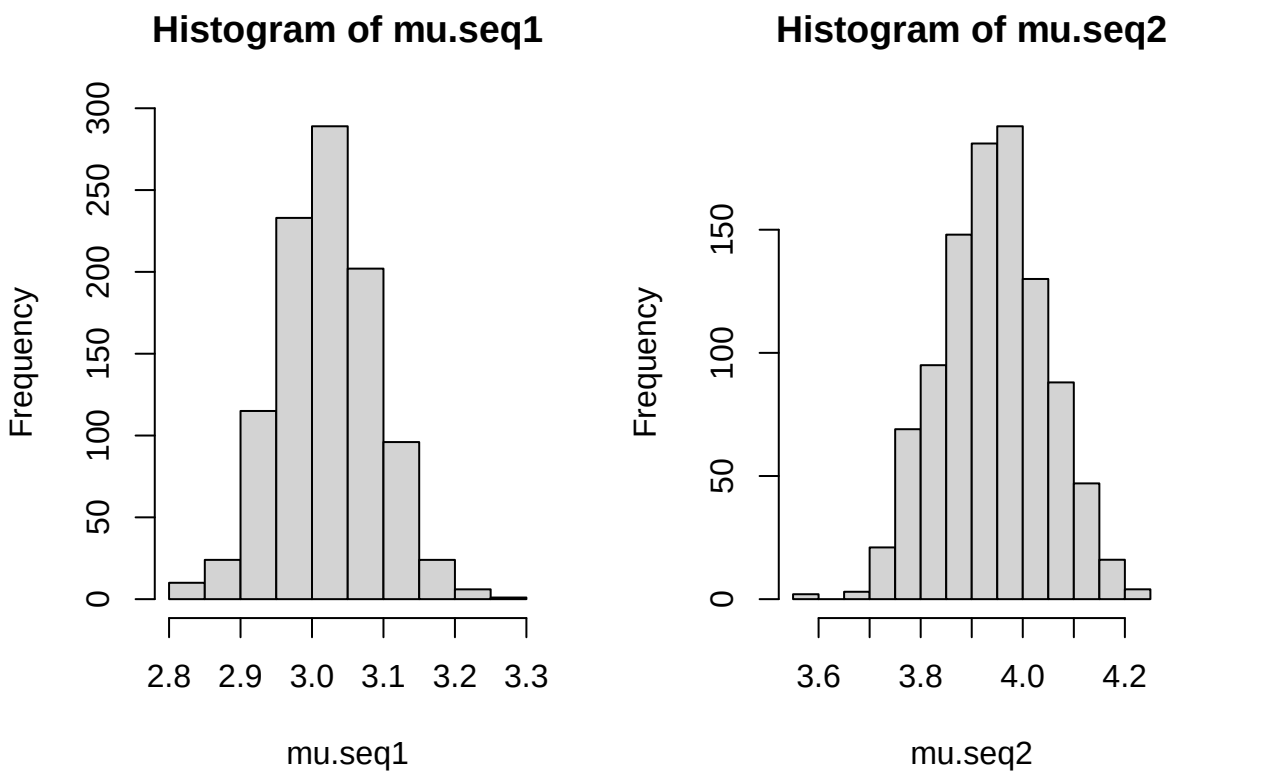
```
## Loading required package: invgamma
```

```
K=1000
alpha1 = 3
alpha2 = 4
sigma2 = 2
beta1 = .6
beta2 = .4

# Simulation des deux echantillons
n = 200
ech1 = rnorm(n,alpha1,sqrt(sigma2))
ech2 = rnorm(n,alpha2,sqrt(sigma2))

alpha0 = 0
sigma2_0 = 10
```

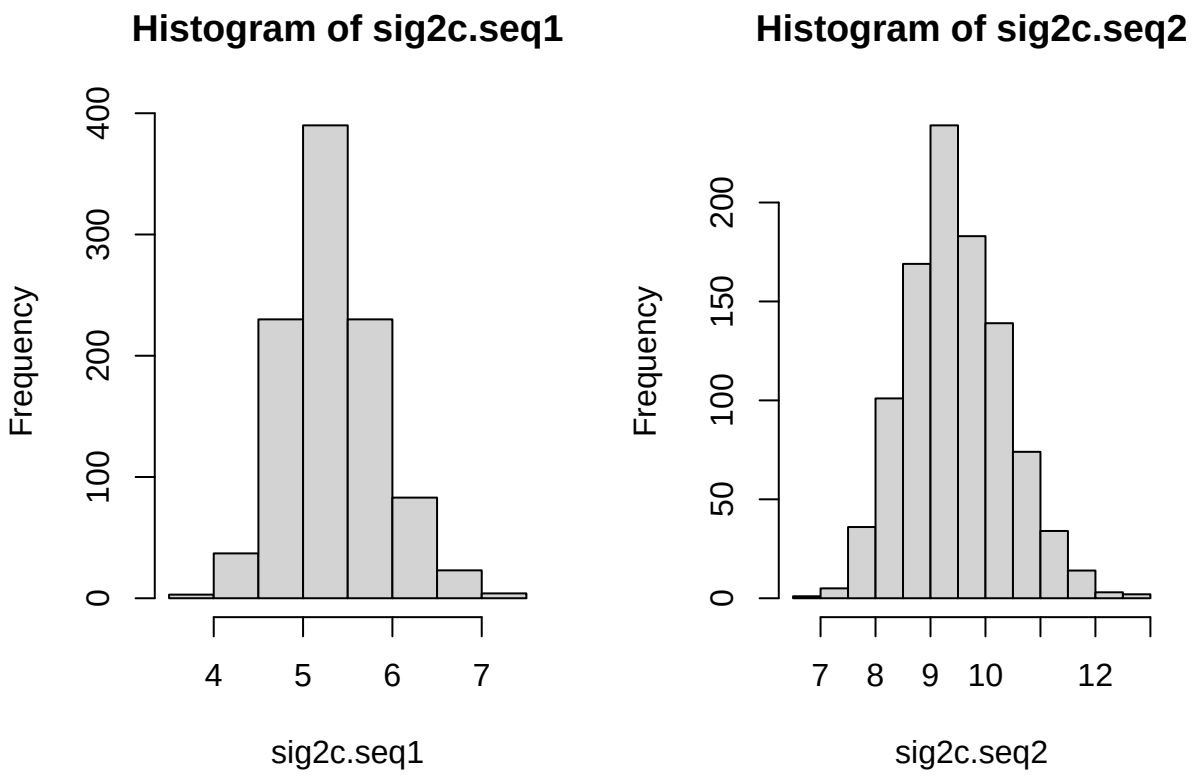
```
sig2.c1 <- rinvgamma(1,beta1,beta2)
sig2.c2 <- rinvgamma(1,beta1,beta2)

mu.c1 <- rnorm(1,alpha0,sqrt(sigma2_0))
mu.c2 <- rnorm(1,alpha0,sqrt(sigma2_0))

mu.seq1 = rep(0,K)
sig2c.seq1 = rep(0,K)
mu.seq2 = rep(0,K)
sig2c.seq2 = rep(0,K)

muc1 = rnorm(1,alpha0, sqrt(sigma2_0))
sig2c1 = rinvgamma(1,beta1,beta2)
muc2 = rnorm(1,alpha0, sqrt(sigma2_0))
```

```r
sig2c2 = rinvgamma(1,beta1,beta2)

# Algo de Gibbs
for (k in 1:K) {
  muc1 = rnorm(1,(sigma2_0*sum(ech1)+sig2.c1*alpha0)/(n*sigma2_0+sig2.c1),
               sqrt(sig2.c1*sigma2_0/(n*sigma2_0+sig2.c1)))
  mu.seq1[k] = muc1
  muc2 = rnorm(1,(sigma2_0*sum(ech2)+sig2.c2*alpha0)/(n*sigma2_0+sig2.c2),
               sqrt(sig2.c2*sigma2_0/(n*sigma2_0+sig2.c2)))
  mu.seq2[k] = muc2


  sig2c1 = rinvgamma(1,beta1+n/2,beta2+1/2*sum((ech1-mu.c1)^2))
  sig2c.seq1[k] = sig2c1
  sig2c2 = rinvgamma(1,beta1+n/2,beta2+1/2*sum((ech2-mu.c1)^2))
  sig2c.seq2[k] = sig2c2
}
```

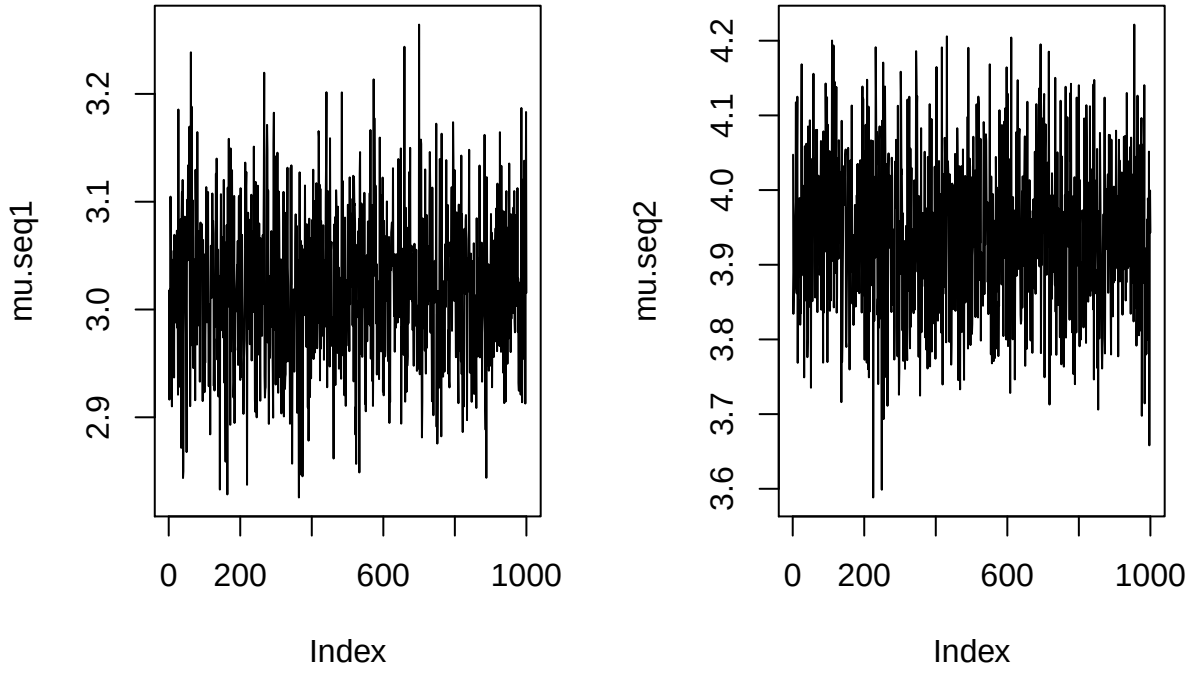
```r
par(mfrow=c(1,2))
hist(mu.seq1)
hist(mu.seq2)
```



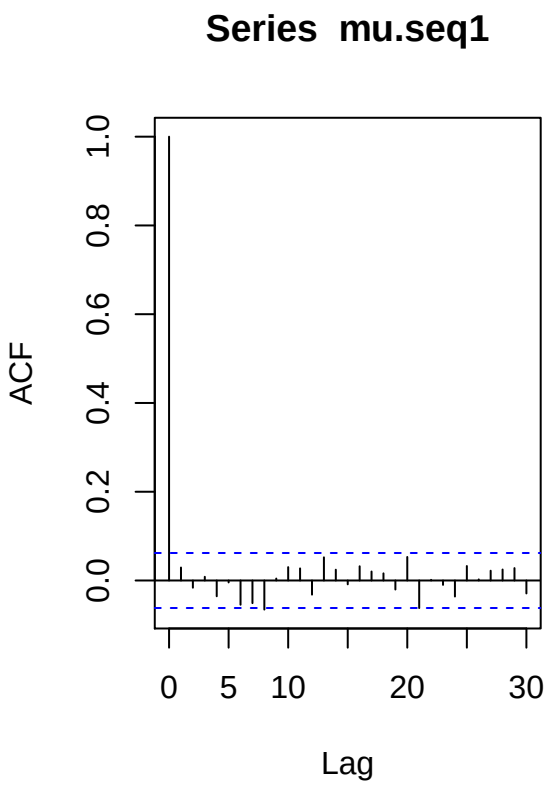

```r
par(mfrow=c(1,2))
hist(sig2c.seq1)
hist(sig2c.seq2)
```

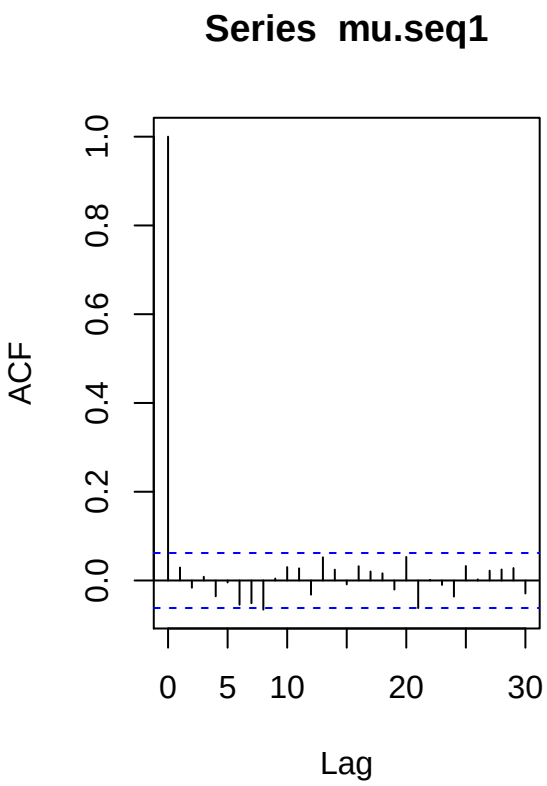



```
par(mfrow=c(1,2))
plot(mu.seq1, type = 'l')
plot(mu.seq2, type = 'l')
```




```
par(mfrow=c(1,2))
plot(acf(mu.seq1))
```






```
plot(acf(mu.seq2))
```