Rapport Projet

Bodian Elion Louis

18/11/2020

Table of Contents

Introduction

Un sujet est en situation de consanguinité si pour un locus donné, il possède deux allèles identiques, par copie d'un seul et même gène ancêtre (voir pédigré ci-dessous). Le coefficient de consanguinité (cc ou f) est la probabilité pour que les deux allèles que possède un individu en un locus donné soient identiques par descendance. Le vrai coefficient de consanguinité d'un individu est souvent inconnu. Dans cet article de Leutenegger et al. (2003), on s'intéresse à l'estimation du coefficient de consanguinité grâce à l'utilisation de données génomique. Afin de mieux étudier le modèle des chaines de Markov cachées (HMM) pour le Processus IBD de l'individu, nous allons d'abord faire la présentation du modèle et la simulation, ensuite nous allons faire un Forward/Backward pour ce modèle et enfin nous allons appliquer HMM pour estimer ses coefficients.

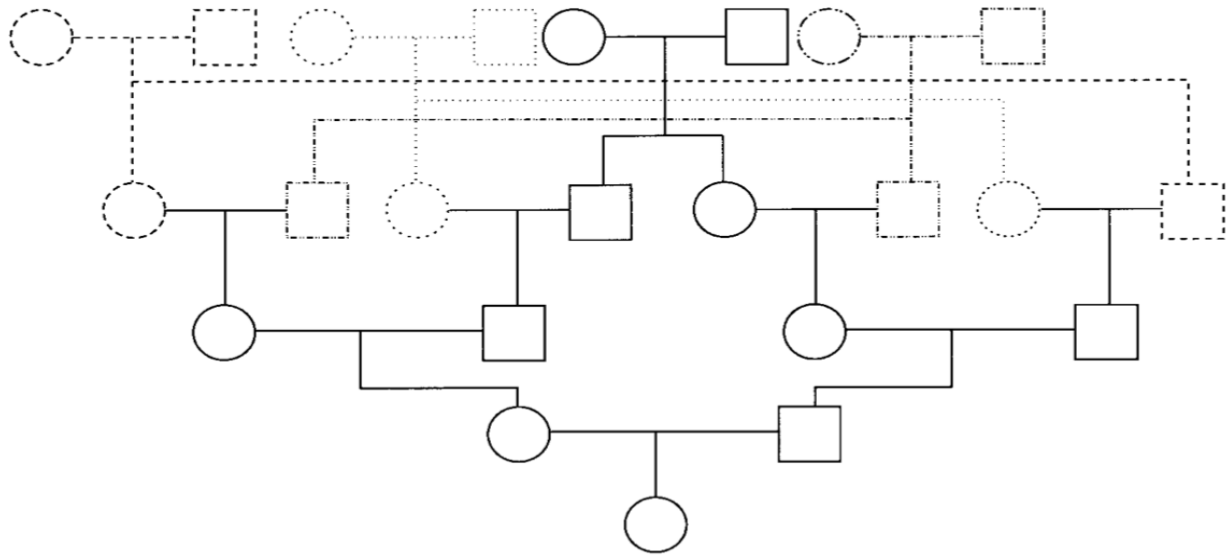


Figure 1: pédigré complexe tiré de (Leutenneger et al, 2003).

Présentation du modèle et simulation

Présentation du modèle

On définit les variables qui interviennent dans le modèle de mélange

NB: on se limite au cas de loci bi-allélique pour simplifier et sans perte de généralité.

Pour $k = 1, \dots, n$:

- $X_k \in \{0,1\}$, X_k est le statut IBD au locus k
- $Y_k \in \{00,01,11\}$, Y_k est le génotype au locus k (1 pour l'allele rare, 0 pour l'allèle de référence).

On note $X = (X_k)_{k=1,\dots,n}$ et $Y = (Y_k)_{k=1,\dots,n}$ et on suppose que :

$$\mathbb{P}(X,Y) = \mathbb{P}(X_1)\prod_{k=2}\mathbb{P}(X_k|X_{k-1}) \times \prod_{k=1}\mathbb{P}(Y_k|X_k) \qquad (1)$$

$$\begin{aligned}
P(X_k = 1|X_{k-1} = 1) &= (1 - e^{-at_k})f + e^{-at_k}\,, \\
P(X_k = 0|X_{k-1} = 1) &= (1 - e^{-at_k})(1 - f)\,, \\
P(X_k = 1|X_{k-1} = 0) &= (1 - e^{-at_k})f\,, \text{ and} \\
P(X_k = 0|X_{k-1} = 0) &= (1 - e^{-at_k})(1 - f) + e^{-at_k}\,, \qquad (2)
\end{aligned}$$

Figure 2: Matrice de transition du modèle (Eq. 2 de Leutenegger, 2003).

En plus de la définition de la matrice de transition en fig. 2 on pose :

$$L_{\mathbf{Y}_c}(f,a) = P(\mathbf{Y}_c|f,a) = \sum_{\mathbf{x}} P(\mathbf{Y}_c|\mathbf{X}=\mathbf{x})L_{\mathbf{x}}(f,a)$$

$$= \sum_{\mathbf{x}} P(\mathbf{Y}_c|\mathbf{X}=\mathbf{x})P(\mathbf{X}=\mathbf{x}|f,a)$$

$$= \sum_{\mathbf{x}} \left[\prod_{k=1}^{M_c} P(Y_k|X_k=x_k)\right]$$

$$\times \left[\prod_{k-2}^{M_c} P(X_k=x_k|X_{k-1}=x_{k-1},f,a)\right]P(X_1=x_1|f) \ .$$

Où :

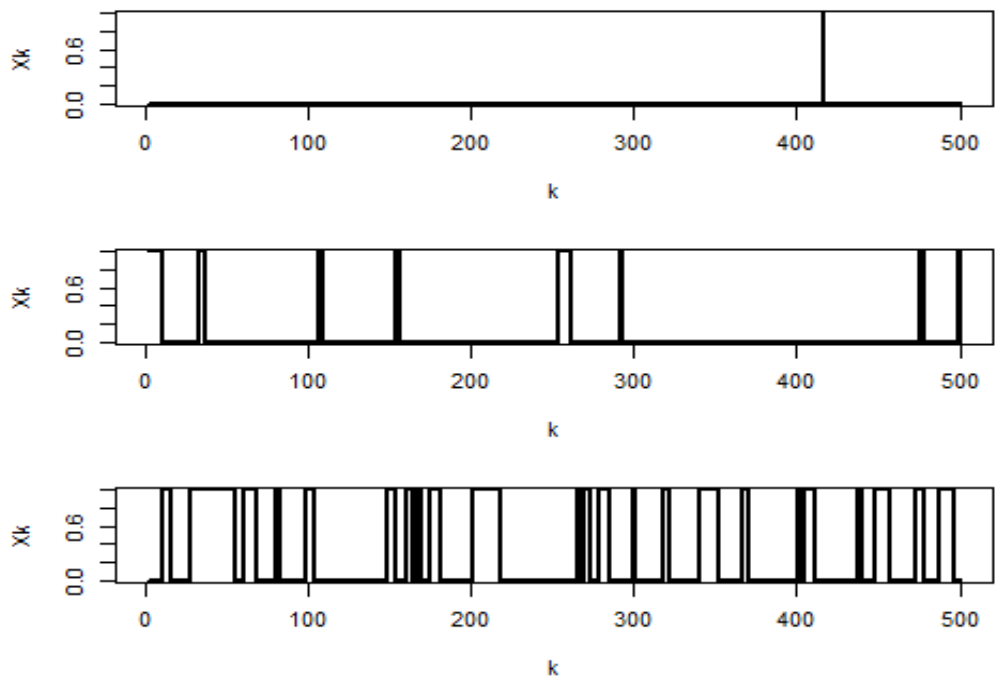
- $a > 0$ Est un paramètre du modèle
- $f \in]0,1[$ Est le coefficient de consanguinité
- $t = (t_k)_{k=1,\dots,n}$ Les distances entre loci (donnée du problème)
- $\varepsilon \in]0,1[$ Le taux d'erreur
- $p = (p_k)_{k=1,\dots,n}$ Les fréquences alléliques de l'allèle rare au loci.
- $\theta = (a, f)$ est le paramètre du modèle à estimer.

Simulations

Dans cette partie on va faire une simulation du modèle en observant l'influence des paramètres f, a et ϵ.

-Pour le paramètre f:

On fixe le paramètre $a = 0.063$ et $\epsilon = 0.05$ et on fait varier le coeficient f.



Dans la fig. 3 on représente la variable X simulée avec trois taux de consanguinité f. On constate que le nombre de plages d'IBD=1 augmente lorsque le coefficient de consanguinité f augmente. Par ailleurs, pour la simulation avec $f = 1/64 = 0.0156$ on trouve une proportion de région IBD=1 de 0.002; pour la simulation avec $f = 1/16 = 0.0625$ on trouve une proportion de région IBD=1 de 0.062; pour la simulation avec $f = 1/4 = 0.25$ on trouve une proportion de région IBD=1 de 0.3.

Table1 : fréquence observée des positions IBD=1 dans différentes

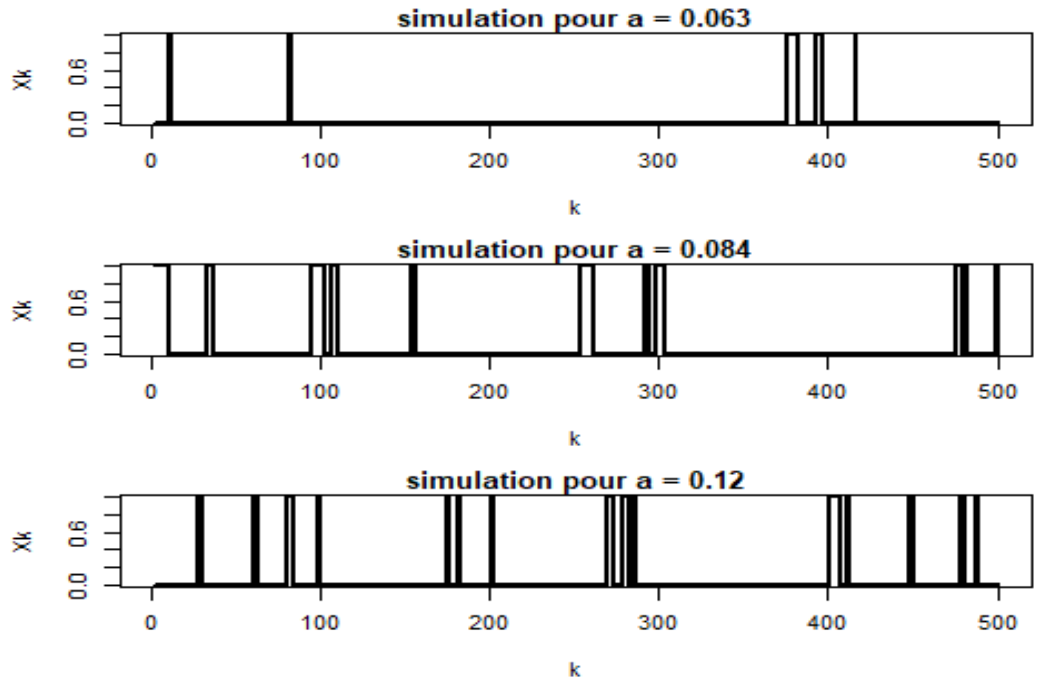
Simulations pour $f = 1/16$ *et* $f = 1/4$, ...

	n=500	n=1000
rep1	0.076	0.286
rep2	0.048	0.269
rep3	0.082	0.245

En Table 1 on voit que la fréquence observée des positions IBD=1 est proche de la valeur de $f = 1/16$ lorsque $n = 500$ et de la valeur $f = 1/4$ lorsque $n = 1000$.

-Pour le paramètre a:

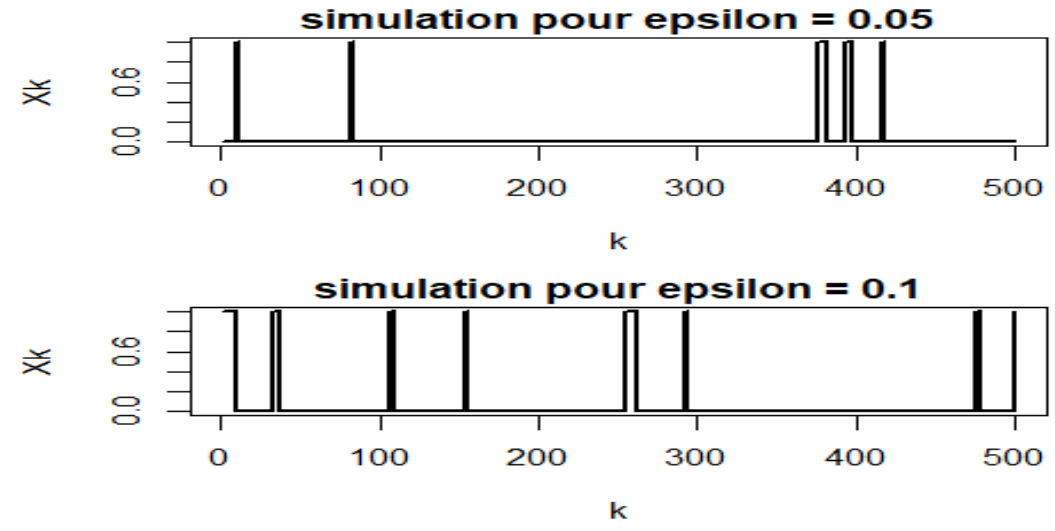
On fixe le paramètre $f = 1/16$ et $\epsilon = 0.05$ et on fait varier le paramètre du modèle a.



Dans la fig. 4 on représente la variable X simulée avec le même taux de consanguinité f. On constate que le nombre de plages d'IBD = 1 augmente lorsque le paramètre du modèle a augmente légèrement pas comme l'expérience de la fig 3. Par ailleurs, pour la simulation avec $a = 0.063$ on trouve une proportion de région IBD=1 de 0.028; pour la simulation avec $a = 0.084$ on trouve une proportion de région IBD=1 de 0.102; pour la simulation avec $a = 0.12$ on trouve une proportion de région IBD=1 de 0.076.

-Pour le paramètre ϵ:

On fixe le paramètre $f = 1/16$ et $a = 0.063$ et on fait varier le paramètre ϵ.



Dans la fig. 5 on représente la variable X simulée avec le même taux de consanguinité f. On constate que le nombre de plages d'IBD = 1 augmente lorsque le paramètre du modèle ϵ

augmente légèrement aussi. Par ailleurs, pour la simulation avec $\epsilon = 0.05$ on trouve une proportion de région IBD=1 de 0.028; pour la simulation avec $a = 0.1$ on trouve une proportion de région IBD=1 de 0.062.

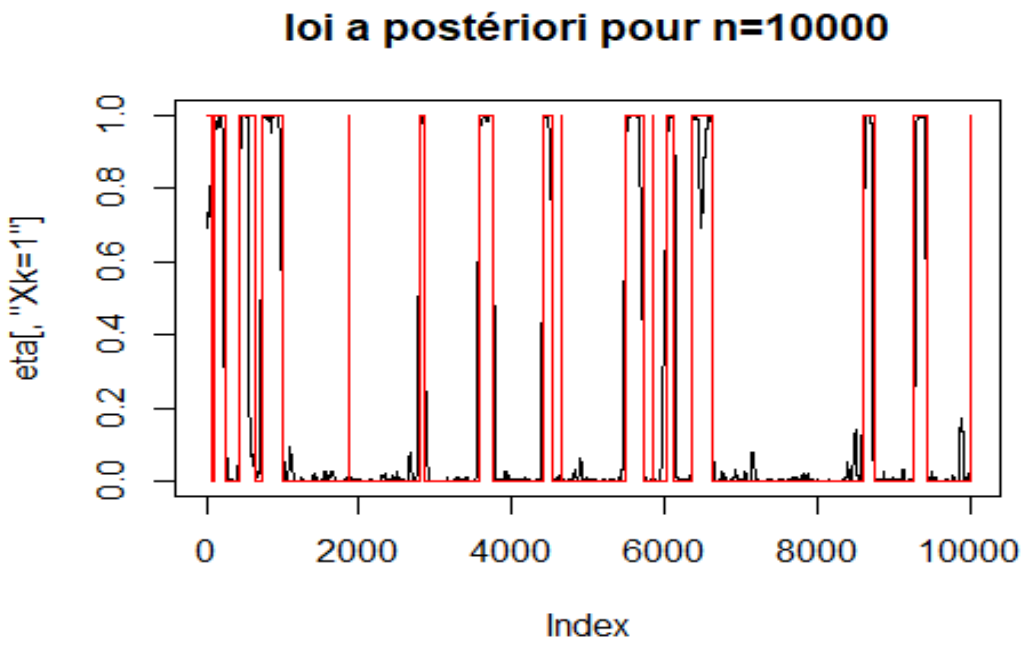
En résumé nous pouvons conclure qu'il y a une influence vis à vis des paramètres du modèle. En effet le nombre de plages d'IBD = 1 augmente lorsque le coefficient de consanguinité f, le paramètre a et le taux d'erreur ϵ augmente de maniére légère.

Forward/Backward

Loi à postériori

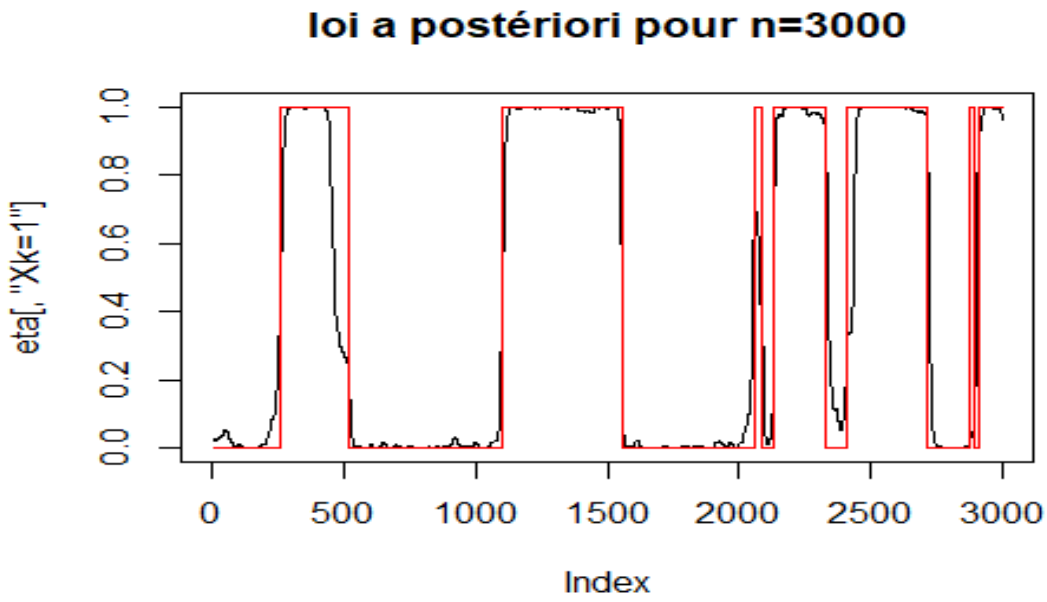
On se sert de Forward/Backward pour faire une simulation de la loi à postériori. C'est à dire pour n quelconque, comment est le comportement du modèle prédictive par rapport à la loi postériori. On fait une Simulation à l'aide du modèle de chaîne de Markov cachée pour différentes valeurs de n.

Pour $n = 10000$:



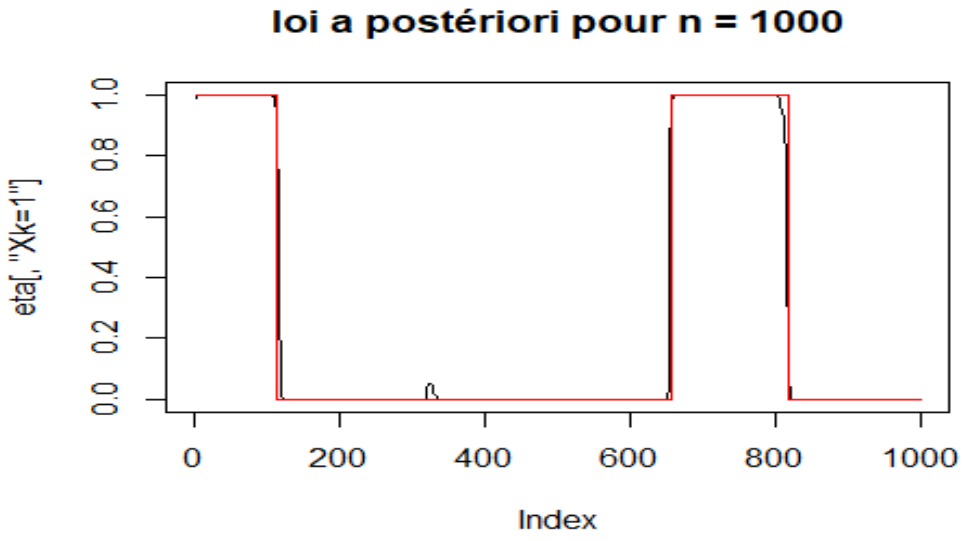
Dans ce graphe on voit les vraies valeurs et puis les probabilités. On n'y voit pas grand-chose sur cette simulation parce que en particulier n est trop grand. On va prendre n plus petit afin de voir plus clair.

-Pour $n = 3000$



On voit une simulation qui a l'air de montrer que la loi à postériori semble être correcte. La courbe en noir qui représente ici ce qu'on a prédit par le modèle, il est à peu près similaire à la courbe rouge qui représente la loi à postériori.
On prend à présent un n plus petit :

-Pour $n = 1000$



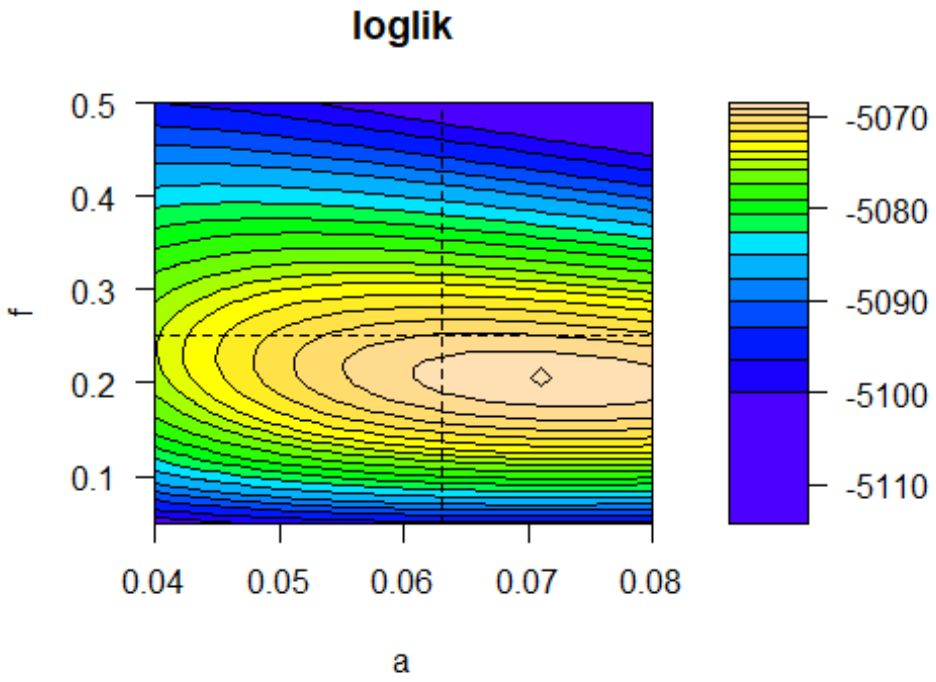
Pour $n = 1000$, on voit que la simulation montre une loi à postèriori correcte. On peut dire que le modèle prédit est similaire à la courbe rouge qui est un modèle parfait. Donc on a une bonne loi à postériori.

Log vraisemblance

On effectue une maximisation numérique afin d'obtenir les estimations du maximum de vraisemblance de f et a . Pour cela on se donne deux simulations de fréquence d'allèle différents et de marqueurs unique. Pour chacune de ces deux simulations, on calcul la valeur maximale en déterminant les estimations du log de vraisemblance.

Simulation S1

Le coefficient de consanguinité $f = 1/4$ va prendre une séquence d'allèles entre 0.05 et 0.50, le paramétre $a = 0.063$ lui aussi va prendre une séquence entre 0.04 et 0.08 avec 50 marqueurs chacun.



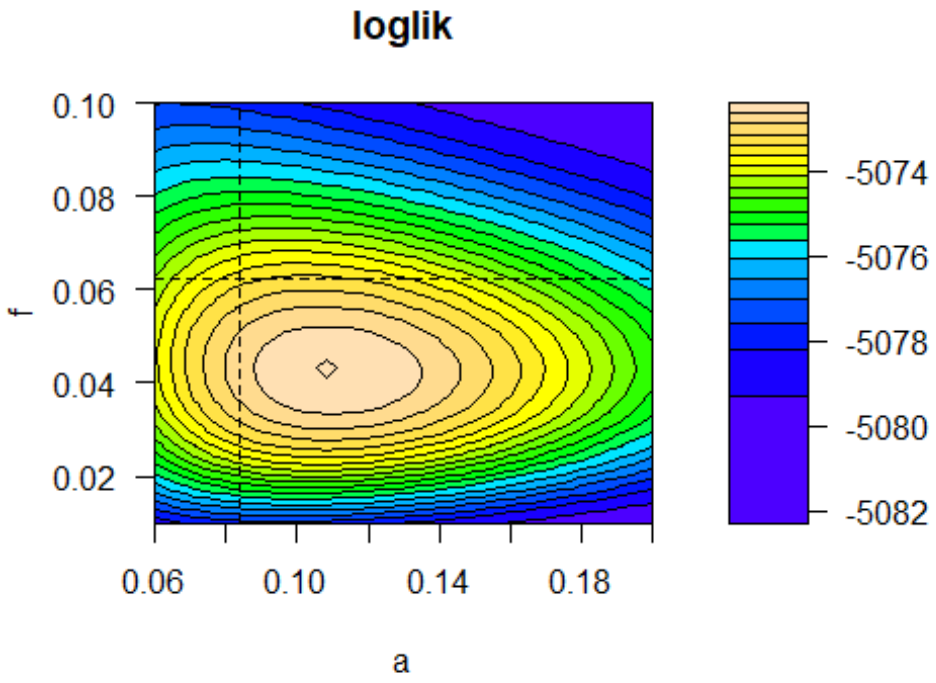
Le graphe ci-dessus atteint son maximum au point sommet, ce point converge. Seul le point qui est au sommet représente le maximum déterminant les estimations de la log vraisemblance du coefficient de consanguinité f et du paramètre a.

```
##            [,1]      [,2]      [,3]
## [1,] 0.07102041 0.2061224 -5068.559
## [2,] 0.07143847 0.2034941 -5068.555
```

On trouve le maximum de vraisemblance avec comme point maximum $z = -5068.559$ on a aussi les valeurs estimées de f et a avec $\hat{f} = 0.2061224$ et $\hat{a} = 0.07102041$. On voit que les valeurs des paramètres f et a sont égales aux estimations qui seraient fournies par le processus IBD

Simulation S2:

On change le coefficient de consanguinité $f = 1/16$, on prend une séquence d'alléles entre 0.01 et 0.10, le paramétre $a = 0.084$ lui aussi va prendre une séquence entre 0.06 et 0.2 avec 50 marqueurs chacun.



Le graphe ci-dessus atteint son maximum au voisinage de certains points. Ce point représente le maximum de vraisemblance du coefficient de consanguinité f et de a.

```
##           [,1]       [,2]      [,3]
## [1,] 0.1085714 0.04306122 -5072.404
## [2,] 0.1100505 0.04224869 -5072.401
```

On trouve le maximum de vraisemblance avec comme point maximum $z = -5072.404$ on a aussi les valeurs estimées de f et a avec $\hat{f} = 0.04306122$ et $\hat{a} = 0.1085714$. Les valeurs des paramètres estimés de f et a sont proches des vraies valeurs.

Dans les deux simulations nous constatons que notre méthode estime avec précision f et a. Nous pouvons dire que la méthode du maximum de vraisemblance qui prend en compte les dépendances des marqueurs via un modèle de Markov caché permet de déduire la distribution de probabilité complète de l'état d'identité par IBD des deux allèles d'un individu à chaque marqueur le long du génome et fournit les estimations du coefficient de consanguinité f et du paramètre a.

Estimation du modèle grâce à l'utilisation du HMM

Nous simulons une analyse complète du génome imitant le véritable génome afin d'évaluer la méthode et valider nos estimations. Nous effectuons 10 répliques, pour chaque réplique on fait plusieurs simulations, estime le coefficient f et le paramétre a en présentant les valeurs médianes sur toutes les répliques, ainsi que l'IC à 95% observé.

```
## rep= 1
## rep= 2
## rep= 3
## rep= 4
## rep= 5
## rep= 6
## rep= 7
## rep= 8
## rep= 9
## rep= 10
```

Par exemple le $rep = 1$ fait une simulation et donne une estimation du coeficient de consanguinité f et du paramétre a.

```
##                a          f
##  [1,] 0.11005045 0.04224872
##  [2,] 0.10574692 0.09502721
##  [3,] 0.12791981 0.05254959
##  [4,] 0.08672247 0.05351231
##  [5,] 0.05381901 0.09839443
##  [6,] 0.12509396 0.05014677
##  [7,] 0.09742080 0.06422573
##  [8,] 0.09642472 0.07442024
##  [9,] 0.10001599 0.06720216
## [10,] 0.08948401 0.06310808
```

On remarque les valeurs obtenues des estimations de $\hat{a}$ et $\hat{f}$ sont proche de leurs vrais valeurs $a = 0.084$ et $f = 1/16 = 0.0625$. Donc les valeurs fournies par les données observées du génome sur les paramètres f et a sont égales aux estimations qui seraient fournies par le processus IBD.

```
##        a                 f
##  Min.   :0.05382   Min.   :0.04225
##  1st Qu.:0.09122   1st Qu.:0.05279
##  Median :0.09872   Median :0.06367
##  Mean   :0.09927   Mean   :0.06608
##  3rd Qu.:0.10897   3rd Qu.:0.07262
##  Max.   :0.12792   Max.   :0.09839
```

La valeur médiane des estimations de f et a sont très proches de la proportion dans les conditions de simulation pour $f = 1/16$ et $a = 0.084$. Donc c'est exactement les valeurs qu'on a espéré obtenir. La valeur moyenne des estimations fournit une surestimation de la valeur attendue de $\hat{a}$. On voit aussi que les estimations faites par la médiane sont beaucoup

proches des vraies valeurs que les estimations faites par la moyenne. Donc on peut dire que la médiane marche mieux que la moyenne, car la médiane est robuste aux valeurs aberrantes alors que la moyenne ne l'est pas.

```
##        50%         5%        95%
## 0.09871840 0.06862556 0.12664818

## [1] 0.09926981

## [1] 0.02102096
```

Pour la variable $\hat{a}$ on voit sur l'intervalle de confiance que la médiane est proche de 0.084. Donc la proportion de marqueurs IBD = 1 de $\hat{a}_{vrai}$, dont la valeur attendue est a=0.0992698. On remarque la valeur de l'estimateur $\hat{a}$= 0.09872 est légèrement plus petit que le marqueur IBD ($\hat{a}_{vrai}$) = 0.0992698.

```
##        50%         5%        95%
## 0.06366691 0.04580284 0.09687918

## [1] 0.06608353

## [1] 0.01864305
```

Pour la variable $\hat{f}$ on voit sur l'intervalle de confiance que la médiane est proche de 0.0625. La proportion de marqueurs IBD = 1 de $\hat{f}_{vrai}$, dont la valeur attendue est f=0.0660835. L'estimateur $\hat{f}$= 0.06367 est plus proche de f et plus petit que le marqueur IBD $\hat{f}_{vrai}$ = 0.0660835.

Conclusion

En résumé on peut dire que l'estimation du modèle grâce à l'utilisation du HMM est une méthode efficace qui donne des estimations de meilleures qualités. On remarque aussi que les estimateurs $\hat{f}$ et $\hat{a}$ sont plus proche des vraies valeurs f et plus petit que le marqueur IBD $\hat{f}_{vrai}$ et $\hat{a}_{vrai}$. Cela nous permet d'affirmer que les génotypes de marqueurs fournissent une bonne information sur le statut des IBD.

Annexes

Code pour simuler le modèle

```
simul=function(n,f,a,epsilon) {
  x=y=rep(NA,n)
  t=runif(n,min=1,max=10)
  p=runif(n,min=0.05,max=0.45)
  # simulation
  x[1]=sample(0:1,size=1,prob=c(1-f,f))
  for (k in 2:n) {
    pi=matrix(NA,2,2)
```

```
    rownames(pi)=c("Xkm1=0","Xkm1=1")
    colnames(pi)=c("Xk=0","Xk=1")
    pi["Xkm1=0","Xk=0"]=(1-exp(-a*t[k]))*(1-f)+exp(-a*t[k])
    pi["Xkm1=0","Xk=1"]=(1-exp(-a*t[k]))*f
    pi["Xkm1=1","Xk=0"]=(1-exp(-a*t[k]))*(1-f)
    pi["Xkm1=1","Xk=1"]=(1-exp(-a*t[k]))*f+exp(-a*t[k])
    # verification apply(pi,1,sum)
    x[k]=sample(0:1,size=1,prob=pi[paste0("Xkm1=",x[k-1]),])
  }
  # verifier x correctement simulé
  # plot(x,t='s',lwd=2,xlab="k",ylab="Xk")
  e=array(NA,dim=c(n,2,3),dimnames=list(paste0("k=",1:n),
                                        paste0("Xk=",c(0,1)),
                                        paste0("Yk=",c("00","01","11"))))
  e[,"Xk=0","Yk=00"]=(1-p)^2
  e[,"Xk=0","Yk=01"]=2*p*(1-p)
  e[,"Xk=0","Yk=11"]=p^2
  e[,"Xk=1","Yk=00"]=(1-epsilon)*(1-p)+epsilon*(1-p)^2
  e[,"Xk=1","Yk=01"]=epsilon*2*p*(1-p)
  e[,"Xk=1","Yk=11"]=(1-epsilon)*p+epsilon* p^2
  # verification apply(e,c(1,2),sum)
  for (k in 1:n)
    y[k]=sample(c("00","01","11"),size=1,prob=e[paste0("k=",k),paste0("Xk=",x
[k]),])
  # verification table(y[x==0]) table(y[x==1])
  return(list(n=n,f=f,a=a,epsilon=epsilon,t=t,p=p,x=x,y=y))
}
```

Code pour simuler la loi à postériori

```
sim=simul(n=3000,f=1/4,a=0.0063,epsilon=0.5,pmin=0.35,pmax= 0.5,tmin=0.5,tmax
=2.0)
n= sim$n
a=sim$a
f=sim$f
epsilon=sim$epsilon
n=sim$n
t=sim$t
p=sim$p
y=sim$y

pi=array(NA,dim=c(n,2,2),dimnames=list(paste0("k=",1:n),
                                       paste0("Xkm1=",0:1),
                                       paste0("Xk=",0:1)))
pi[,"Xkm1=0","Xk=0"]=(1-exp(-a*t))*(1-f)+exp(-a*t)
pi[,"Xkm1=0","Xk=1"]=(1-exp(-a*t))*f
pi[,"Xkm1=1","Xk=0"]=(1-exp(-a*t))*(1-f)
pi[,"Xkm1=1","Xk=1"]=(1-exp(-a*t))*f+exp(-a*t)

#voici la matrice de transition ayant la même loi stationnaire $μ_{\infty} =
(0.4 0.6)$
```

```
p.emission=array(NA,dim=c(n,2,3),dimnames=list(paste0("k=",1:n),
                                               paste0("Xk=",c(0,1)),
                                               paste0("Yk=",c("00","01","11")
)))
p.emission[,"Xk=0","Yk=00"]=(1-p)^2
p.emission[,"Xk=0","Yk=01"]=2*p*(1-p)
p.emission[,"Xk=0","Yk=11"]=p^2
p.emission[,"Xk=1","Yk=00"]=(1-epsilon)*(1-p)+epsilon*(1-p)^2
p.emission[,"Xk=1","Yk=01"]=epsilon*2*p*(1-p)
p.emission[,"Xk=1","Yk=11"]=(1-epsilon)*p+epsilon* p^2
save(file="loi_sta.Rdata", p.emission)

mu1=c((1-f),f)

Fw=matrix(NA,n,2)
rownames(Fw)=paste0("k=",1:n)
colnames(Fw)=paste0("Xk=",0:1)
L=rep(NA,n)
Fw[1,]=mu1[]*p.emission[k,,paste0("Yk=",y[1])]
tmp=max(Fw[1,]); L[1]=log(tmp); Fw[1,]=Fw[1,]/tmp
for (k in 2:n) {
  for (j in 0:1) Fw[k,paste0("Xk=",j)]=sum(Fw[k-1,]*pi[k,,paste0("Xk=",j)]*p.
emission[k,paste0("Xk=",j),paste0("Yk=",y[k])])
  tmp=max(Fw[k,]); L[k]=log(tmp)+L[k-1]; Fw[k,]=Fw[k,]/tmp
}
save(file="forward.Rdata", Fw)
load("forward.Rdata")

loglik = L[n] + log(sum(Fw[n,]))

load("loi_sta.Rdata")
Bk=matrix(NA,n,2)
rownames(Bk)=paste0("k=",1:n)
colnames(Bk)=paste0("Xk=",0:1)
M=rep(NA,n)
Bk[n,]=1; M[n]=0;
for (k in n:2) {
  for (i in 0:1) Bk[k-1,paste0("Xk=",i)]=sum(pi[k,paste0("Xkm1=",i),]*p.emiss
ion[k,,paste0("Yk=",y[k])]*Bk[k,])
  tmp=max(Bk[k-1,]); M[k-1]=log(tmp)+M[k]; Bk[k-1,]=Bk[k-1,]/tmp
}
save(file= "backward.Rdata", Bk)
eta = Fw*Bk
eta = eta/apply(eta,1,sum)
eta.mixture = eta
spred.mixture = apply(eta,1,which.max)

#plot(eta[,"Xk=1"],t='l', main = "loi a postériori pour n=3000")
```

```
#points(sim$x,t='s',col="red")
#legend(1, 95, legend=c("modèle de mélange", " chaine de Markov cachée pour n
= 3000"),
 #      col=c("blue", "red"), lty=1:2, cex=0.8)
```

Code pour estimer les coeficients

```
set.seed(42)
nrep=20
res=matrix(NA,nrep,2)
colnames(res)=c("a","f")
for (rep in 1:nrep) {
  #cat("rep=",rep,"\n")
  sim=simul(n=5000,f=1/16,a=0.084,epsilon=0.05,pmin=0.35,pmax=0.45,tmin=0.5,t
max=2.0)
  a=sim$a
  f=sim$f
  epsilon=sim$epsilon
  n=sim$n
  t=sim$t
  p=sim$p
  y=sim$y

  pi=array(NA,dim=c(n,2,2),dimnames=list(paste0("k=",1:n),
                                         paste0("Xkm1=",0:1),
                                         paste0("Xk=",0:1)))
  pi[,"Xkm1=0","Xk=0"]=(1-exp(-a*t))*(1-f)+exp(-a*t)
  pi[,"Xkm1=0","Xk=1"]=(1-exp(-a*t))*f
  pi[,"Xkm1=1","Xk=0"]=(1-exp(-a*t))*(1-f)
  pi[,"Xkm1=1","Xk=1"]=(1-exp(-a*t))*f+exp(-a*t)

  opt=optim(par=c(0.05,0.4),fn=function(par)-loglik(par[1],par[2]),method="L-
BFGS-B",
            lower=c(0.01,0.001),upper=c(2.0,0.90))
  res[rep,]=opt$par
}
```

Bibliographie

Leutenegger, Anne-Louise, Bernard Prum, Emmanuelle Génin, Christophe Verny, Arnaud Lemainque, Françoise Clerget-Darpoux, and Elizabeth A Thompson. 2003. “Estimation of the Inbreeding Coefficient Through Use of Genomic Data.” *The American Journal of Human Genetics* 73 (3): 516–23.