Projet - Outils Statistique

Année 2020-2021

Algorithme EM

Introduction

Le document étudié est un article scientifique intitulé *SeqEM: an adaptive genotype-calling approach for next-generation sequencing studies* (Martin et al. 2010). Cet article de bio-informatique a pour but de proposer une nouvelle méthode afin de déterminer le génotype d'un individu à partir de lectures multiples d'une séquence de l'ADN.

Actuellement, il existe deux types de méthodes pour procéder à la détermination du génotype d'une personne. La première méthode est dite "filtrante". C'est une approche assez classique consistant à mettre en place un filtre (c'est-à-dire un pourcentage fixe) qui permet de dire si un individu est hétérozygote lorsque le nombre de nucléotide variants observés dépasse ce seuil. La deuxième méthode, quant à elle, consiste à utiliser le bayésien. On y retrouve plusieurs techniques dont notamment `MAQ` et `SOAPsnp`.

Ainsi, l'article que nous étudions propose une nouvelle méthode appelée `SeqEM`, qui utilise l'algorithme EM (Espérance-Maximisation). Selon ses auteurs, `SeqEM` serait plus performant que les méthodes évoquées ci-dessus, puisqu'elle aurait un taux d'erreur[1] plus faible.

Le contexte [2]

Lors du processus pour déterminer le génotype d'un individu (*genotype calling*), on observe les nucléotides présents dans l'ADN. Etant donné que nous ne sommes pas tous similaire les uns des autres, il est courant d'observer des variations de nucléotides au sein d'individus d'une même espèce. Une variation est appelée polymorphisme de nucléotide unique (*SNPs : Single nucleotide polymorphisms*).

En pratique, les nucléotides sont observés avec un taux d'erreur. Cependant, puisque les SNPs représentent environ 90‰ des variations dans le génome humain, l'intérêt de pouvoir observer un nucléotide avec un faible taux d'erreur a un enjeu conséquent. En effet, les SNPs sont à l'origine dont nous réagissons face à certains médicaments, et influencent également notre prédisposition face à certaines maladies génétiques.

Le modèle

Dans le modèle que nous allons étudier, on a : $\mathbf{X} = (X_1, ..., X_n)$ un échantillon tel que,

$$\begin{aligned} &\mathbb{P}(Z_i = k) = p_k, \qquad k \in \{VV, RV, RR\} \\ &X_i \mid Z_i = k \sim B(N_i, p_k), \qquad k \in \{VV, RV, RR\} \end{aligned}$$

Avec:

[1] Le fait d'observer un nucléotide variant alors que c'est un nucléotide de référence

[2] Afin de mieux comprendre le contexte, je m'appuye sur l'article *Techniques de recherche des polymorphismes génétiques* (Le Morvan et al. 2005)

X_i une variable alétoire correspondant au nombre de nucléotides variants (V) observés pour l'individu i.

Z_i une variable aléatoire correspondant vrai génotype de l'individu i, $Z_i = \{\text{VV}, \text{RR}, \text{RV}\}$

On va chercher à estimer le paramètre $\theta = (\alpha, p_{VV}, p_{RV})$.

Loi a posteriori $\mathbb{P}(Z \mid X; \theta)$

La loi a posteriori $\mathbb{P}(Z \mid X; \theta)$ correspond aux η.

De ce fait, on a :

$$\begin{aligned}
\eta_{VV}(i) = \mathbb{P}(Z_i = \text{VV} \mid X_i, N_i, \theta_{old}) &= \frac{\mathbb{P}(X_i, Z_i = \text{VV} \mid N_i, \theta_{old})}{\mathbb{P}(X_i \mid N_i, \theta_{old})} \\
&= \frac{\mathbb{P}(X_i, Z_i = \text{VV} \mid N_i, \theta_{old})}{\mathbb{P}(X_i \mid Z_i = \text{VV}, N_i, \theta_{old})\text{p}_{\text{VV}} + \mathbb{P}(X_i \mid Z_i = \text{RV}, N_i, \theta_{old})\text{p}_{\text{RV}} + \mathbb{P}(X_i \mid Z_i = \text{RR}, N_i, \theta_{old})(1 - \text{p}_{\text{VV}} - \text{p}_{\text{RV}})} \\
&= \frac{\binom{N_i}{X_i}(1-\alpha)^{X_i}\alpha^{N_i - X_i}\text{p}_{\text{VV}}}{\binom{N_i}{X_i}(1-\alpha)^{X_i}\alpha^{N_i - X_i}\text{p}_{\text{VV}} + \binom{N_i}{X_i}\left(\frac{1}{2}\right)^{N_i}\text{p}_{\text{RV}} + \binom{N_i}{X_i}(1-\alpha)^{N_i - X_i}\alpha^{X_i}(1 - \text{p}_{\text{VV}} - \text{p}_{\text{RV}})} \\
&= \frac{(1-\alpha)^{X_i}\alpha^{N_i - X_i}\text{p}_{\text{VV}}}{(1-\alpha)^{X_i}\alpha^{N_i - X_i}\text{p}_{\text{VV}} + \binom{N_i}{X_i}\left(\frac{1}{2}\right)^{N_i}\text{p}_{\text{RV}} + (1-\alpha)^{N_i - X_i}\alpha^{X_i}(1 - \text{p}_{\text{VV}} - \text{p}_{\text{RV}})} \\
&\Rightarrow \boxed{\eta_{VV}(i) \propto (1-\alpha)^{X_i}\alpha^{N_i - X_i}\text{p}_{\text{VV}}}
\end{aligned}$$

De même pour η_{RV} et η_{RR}, on a :

$$\begin{aligned}
\eta_{RV}(i) &= \frac{\mathbb{P}(X_i, Z_i = \text{RV} \mid N_i, \theta_{old})}{\mathbb{P}(X_i \mid N_i, \theta_{old})} \\
&= \frac{\left(\frac{1}{2}\right)^{N_i}\text{p}_{\text{RV}}}{(1-\alpha)^{X_i}\alpha^{N_i - X_i}\text{p}_{\text{VV}} + \binom{N_i}{X_i}\left(\frac{1}{2}\right)^{N_i}\text{p}_{\text{RV}} + (1-\alpha)^{N_i - X_i}\alpha^{X_i}(1 - \text{p}_{\text{VV}} - \text{p}_{\text{RV}})} \\
&\Rightarrow \boxed{\eta_{RV}(i) \propto \left(\frac{1}{2}\right)^{N_i}\text{p}_{\text{RV}}}
\end{aligned}$$

$$\begin{aligned}
\eta_{RR}(i) &= \frac{\mathbb{P}(X_i, Z_i = \text{RR} \mid N_i, \theta_{old})}{\mathbb{P}(X_i \mid N_i, \theta_{old})} \\
&= \frac{\alpha^{X_i}(1-\alpha)^{N_i - X_i}(1 - \text{p}_{\text{VV}} - \text{p}_{\text{RV}})}{(1-\alpha)^{X_i}\alpha^{N_i - X_i}\text{p}_{\text{VV}} + \binom{N_i}{X_i}\left(\frac{1}{2}\right)^{N_i}\text{p}_{\text{RV}} + (1-\alpha)^{N_i - X_i}\alpha^{X_i}(1 - \text{p}_{\text{VV}} - \text{p}_{\text{RV}})} \\
&\Rightarrow \boxed{\eta_{RR}(i) \propto \alpha^{X_i}(1-\alpha)^{N_i - X_i}(1 - \text{p}_{\text{VV}} - \text{p}_{\text{RV}})}
\end{aligned}$$

Etapes E/M de l'algorithme

Etape E:

$$
\begin{aligned}
Q(\theta \mid \theta_{old}) &= \int_Z \mathbb{P}(Z \mid X, N, \theta_{old}) \, \log(X, Z, N \mid \theta) \, \mathrm{d}Z \\
&= \sum_{i=1}^{n} \int_{Z_i} \mathbb{P}(Z_i \mid X_i, N_i, \theta_{old}) \, \log(X_i, Z_i, N_i \mid \theta) \, \mathrm{d}Z_i \\
&= \sum_{i=1}^{n} \sum\nolimits_{k \in \{\mathrm{RR,VV,RV}\}} \mathbb{P}(Z_i = k \mid X_i, N_i, \theta_{old}) \, \log(X_i, Z_i = k, N_i \mid \theta) \\
&= \sum_{i=1}^{n} \underbrace{\mathbb{P}(Z_i = \mathrm{VV} \mid X_i, N_i, \theta_{old})}_{\eta_{VV}(i)} \, \log(X_i, Z_i = \mathrm{VV}, N_i \mid \theta) \\
&+ \sum_{i=1}^{n} \underbrace{\mathbb{P}(Z_i = \mathrm{RV} \mid X_i, N_i, \theta_{old})}_{\eta_{RV}(i)} \, \log(X_i, Z_i = \mathrm{RV}, N_i \mid \theta) \\
&+ \sum_{i=1}^{n} \underbrace{\mathbb{P}(Z_i = \mathrm{RR} \mid X_i, N_i, \theta_{old})}_{\eta_{RR}(i)} \, \log(X_i, Z_i = \mathrm{RR}, N_i \mid \theta)
\end{aligned}
$$

$$
\begin{aligned}
Q(\theta \mid \theta_{old}) &= \sum_{i=1}^{n} \eta_{VV}(i) \, \log(X_i, Z_i = \mathrm{VV}, N_i \mid \theta) + \sum_{i=1}^{n} \eta_{RV}(i) \, \log(X_i, Z_i = \mathrm{RV}, N_i \mid \theta) \\
&+ \sum_{i=1}^{n} \eta_{RR}(i) \, \log(X_i, Z_i = \mathrm{RR}, N_i \mid \theta) \\
&= \sum_{i=1}^{n} \eta_{VV}(i) \, \log\left(\binom{N_i}{X_i} (1-\alpha)^{X_i} \alpha^{N_i - X_i} \mathrm{p_{VV}} \right) + \sum_{i=1}^{n} \eta_{RV}(i) \, \log\left(\binom{N_i}{X_i} \left(\frac{1}{2}\right)^{N_i} \mathrm{p_{RV}} \right) \\
&+ \sum_{i=1}^{n} \eta_{RR}(i) \, \log\left(\binom{N_i}{X_i} (1-\alpha)^{N_i - X_i} \alpha^{X_i} (1 - \mathrm{p_{VV}} - \mathrm{p_{RV}}) \right)
\end{aligned}
$$

$$
\begin{aligned}
Q(\theta \mid \theta_{old}) = \textit{constante} &+ \sum_{i=1}^{n} \eta_{VV}(i) \left((N_i - X_i) \, \log(\alpha) + X_i \, \log(1-\alpha) + \, \log(\mathrm{p_{VV}}) \right) + \sum_{i=1}^{n} \eta_{RV}(i) \left(\, \log(\mathrm{p_{RV}}) \right) \\
&+ \sum_{i=1}^{n} \eta_{RR}(i) \left(X_i \, \log(\alpha) + (N_i - X_i) \, \log(1-\alpha) + \, \log(1 - \mathrm{p_{VV}} - \mathrm{p_{RV}}) \right)
\end{aligned}
$$

Etape M:

On cherche maintenant à déterminer

$$\theta_{new} = \arg\max_{\theta} Q(\theta \mid \theta_{old})$$

où

$$\theta = (\alpha, \mathrm{p}_{VV}, \mathrm{p}_{RV})$$

$$\frac{\partial Q(\theta \mid \theta_{old})}{\partial \mathrm{p}_{VV}} = 0 \Leftrightarrow \sum_{i=1}^{n} \eta_{VV}(i) \frac{1}{\mathrm{p}_{VV}} - \sum_{i=1}^{n} \eta_{RR}(i) \frac{1}{1 - \mathrm{p}_{VV} - \mathrm{p}_{RV}} = 0$$
$$\Leftrightarrow \sum_{i=1}^{n} \eta_{VV}(i)(1 - \mathrm{p}_{VV} - \mathrm{p}_{RV}) = \sum_{i=1}^{n} \eta_{RR}(i)\mathrm{p}_{VV}$$
$$\Rightarrow \boxed{p_{VV}^{new} = \sum_{i=1}^{n} \frac{\eta_{VV}(i)(1 - \mathrm{p}_{\mathrm{RV}})}{\eta_{VV}(i) + \eta_{RR}(i)}}$$

$$\frac{\partial Q(\theta \mid \theta_{old})}{\partial \mathrm{p}_{RV}} = 0 \Leftrightarrow \sum_{i=1}^{n} \eta_{RV}(i) \frac{1}{\mathrm{p}_{RV}} - \sum_{i=1}^{n} \eta_{RR}(i) \frac{1}{1 - \mathrm{p}_{VV} - \mathrm{p}_{RV}} = 0$$
$$\Leftrightarrow \sum_{i=1}^{n} \eta_{RV}(i)(1 - \mathrm{p}_{VV} - \mathrm{p}_{RV}) = \sum_{i=1}^{n} \eta_{RR}(i)\mathrm{p}_{RV}$$
$$\Rightarrow \boxed{p_{RV}^{new} = \sum_{i=1}^{n} \frac{\eta_{RV}(i)(1 - \mathrm{p}_{\mathrm{VV}})}{\eta_{RV}(i) + \eta_{RR}(i)}}$$

$$\frac{\partial Q(\theta \mid \theta_{old})}{\partial \alpha} = 0 \Leftrightarrow \sum_{i=1}^{n} \eta_{VV}(i) \left(\frac{(N_i - X_i)}{\alpha} - \frac{X_i}{1 - \alpha} \right) + \sum_{i=1}^{n} \eta_{RR}(i) \frac{X_i}{\alpha} - \frac{(N_i - X_i)}{1 - \alpha} = 0$$
$$\Leftrightarrow \frac{1}{\alpha} \sum_{i=1}^{n} \eta_{VV}(i)(N_i - X_i) - \frac{1}{1 - \alpha} \sum_{i=1}^{n} \eta_{VV}(i) X_i + \frac{1}{\alpha} \sum_{i=1}^{n} \eta_{RR}(i) X_i - \frac{1}{1 - \alpha} \sum_{i=1}^{n} \eta_{VV}(i)(N_i - X_i) = 0$$
$$\Leftrightarrow (1 - \alpha) \sum_{i=1}^{n} \eta_{VV}(i)(N_i - X_i) - \alpha \sum_{i=1}^{n} \eta_{VV}(i) X_i + (1 - \alpha) \sum_{i=1}^{n} \eta_{RR}(i) X_i - \alpha \sum_{i=1}^{n} \eta_{VV}(i)(N_i - X_i) = 0$$
$$\Leftrightarrow \sum_{i=1}^{n} \eta_{VV}(i)(N_i - X_i) - \alpha \sum_{i=1}^{n} \eta_{VV}(i) N_i + \alpha \sum_{i=1}^{n} \eta_{VV}(i) X_i - \alpha \sum_{i=1}^{n} \eta_{VV}(i) X_i + \sum_{i=1}^{n} \eta_{RR}(i) X_i$$
$$- \alpha \sum_{i=1}^{n} \eta_{RR}(i) X_i - \alpha \sum_{i=1}^{n} \eta_{RR}(i) N_i + \alpha \sum_{i=1}^{n} \eta_{RR}(i) X_i = 0$$
$$\Leftrightarrow \sum_{i=1}^{n} \eta_{VV}(i)(N_i - X_i) + \sum_{i=1}^{n} \eta_{RR}(i) X_i = \alpha \sum_{i=1}^{n} \eta_{VV}(i) N_i + \alpha \sum_{i=1}^{n} \eta_{RR}(i) N_i$$
$$\Rightarrow \boxed{\alpha^{new} = \frac{\sum_{i=1}^{n} \eta_{VV}(i)(N_i - X_i) + \sum_{i=1}^{n} \eta_{RR}(i) X_i}{\sum_{i=1}^{n} N_i(\eta_{VV}(i) + \eta_{RR}(i))}}$$

Implémentation de l'algorithme EM

On va maintenant pouvoir implanter l'algorithme EM en R.

```
# Implementation de l'algorithme EM
algo_em=function(x,niter=500,start,N,trace=TRUE,verbose=FALSE) {

  n=length(x)
  alpha=start$alpha
  p_vv=start$p_vv
  p_rv=start$p_rv
  eta=matrix(NA,nrow=n,ncol=3);
  theta=NULL

  if (trace) {
```

```
    theta=matrix(NA,niter,3)
  }

  for (iter in 1:niter) {
    if (trace) {
      theta[iter,]=c(alpha,p_vv,p_rv)
    }

    # Mise à jour des eta
    # eta_vv
    eta[,1]=(1-alpha)^x * alpha^(N-x) * p_vv
    # eta_rv
    eta[,2]=(1/2)^N * p_rv
    # eta_rr
    eta[,3]=alpha^x * (1-alpha)^(N-x) * (1-p_vv-p_rv)

    eta=eta/apply(eta,1,sum)

    # Mise à jour des paramètres
    alpha=(sum(eta[,1]*(N-x))+sum(eta[,3]*x))/((sum(eta[,1]*N)+sum(eta[,3])*N))
    p_vv=(sum(eta[,1])*(1-p_rv))/(sum(eta[,1])+sum(eta[,3]))
    p_rv=(sum(eta[,2])*(1-p_vv))/(sum(eta[,2])+sum(eta[,3]))


    if (verbose) print(c(iter,alpha,p_vv,p_rv));
  }

  return(list(alpha=alpha,p_vv=p_vv,p_rv=p_rv,theta=theta,eta=eta))
}
```

Expérimentations

Maintenant que l'algorithme EM est implémenté, on va procéder aux expérimentations afin de vérifier que l'algorithme estime correctement les paramètres.

Pour cela, on crée tout d'abord une fonction \texttt{data_generation} permettant de générer un jeu de données. Ce jeu de données correspond en réalité à la variable X que l'on observe, puisque c'est le nombre de nucléotides variants (V).

Afin de simplifier l'expérimentation, on fixe le taux d'erreur α et la profondeur de lecture N. Cependant, nous allons les faire varier par la suite.

```
# Fonction générant des données pour essayer l'algorithme EM
data_generation <- function(p,N,proba_alpha,samplesize=500){

  #hidden state
  genotype=sample(c(1:3),samplesize,replace=TRUE,prob=p) # génotype généré via les
                                                         # probabilités p_vv, p_rv, p_rr

  #observation
  x=rbinom(samplesize, N, proba_alpha[genotype]) # nombre de nucléotides variants
  return(x)
}
```

```
# alpha fixé
alpha=0.15

# N fixé
N=10
```

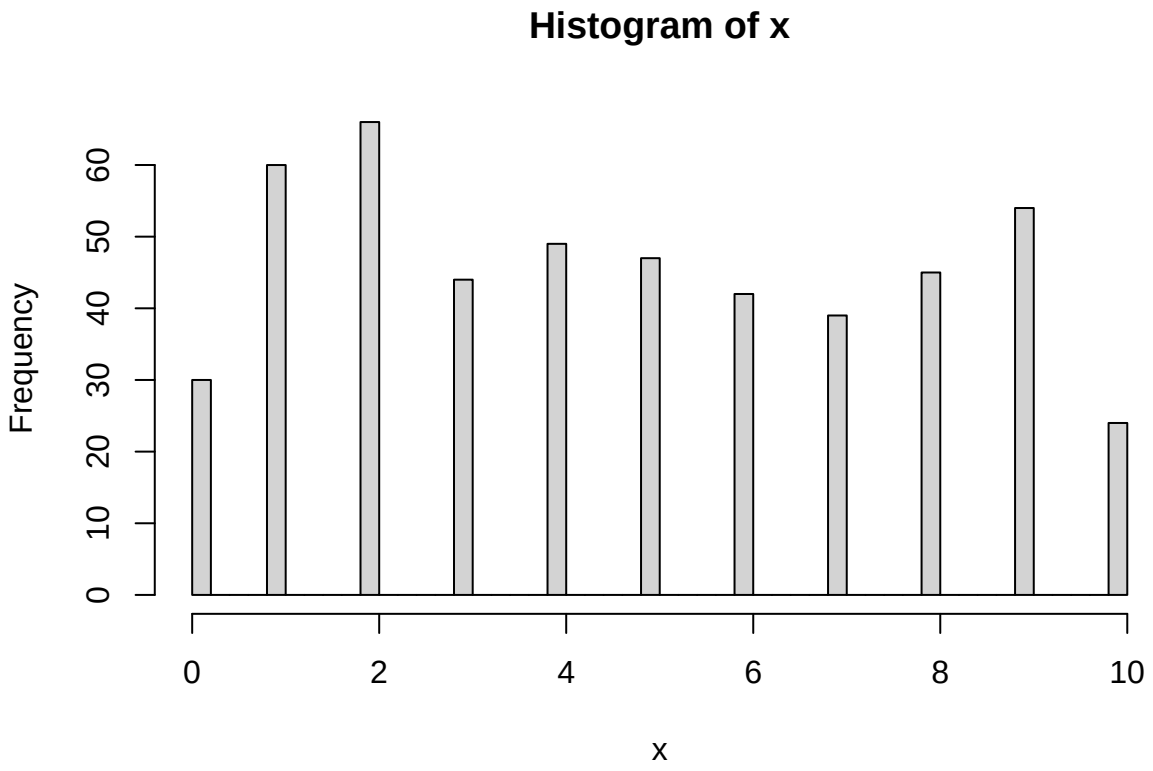
On crée à présent la variable X, en donnant les probabilités telles que : $p_{VV} = 0.3$, $p_{RV} = 0.4$ et $p_{RR} = 1 - p_{VV} - p_{VV} = 0.3$ et les taux d'erreurs lors de la lecture qui sont respectivement $1 - \alpha$, $\frac{1}{2}$ et α.

```
set.seed(12345) # pour fixer l'aléatoire

x <- data_generation(p=c(0.3,0.4,(1-0.3-0.4)), # probabilités p_vv, p_rv, p_rr
                     N=N, # profondeur de lecture
                     proba_alpha=c(1-alpha,1/2,alpha)) # erreur de lecture
```

De plus, on peut tracer un histogramme afin de vérifier la répartition de nos observations. Sur la figure ci-dessous, on constate que la répartition est plus ou moins uniforme, sauf sur les extrêmes.

Il est également important de noter que la partie gauche de l'histogramme représentent les individus ayant un génotype RR, puisqu'ils n'ont pas de nucléotides variants, au milieu nous avons les personnes avec un génotype RV, et enfin à droite on retrouve les individus VV.

```
# Histogramme de X
hist(x, breaks=50)
```



A présent, on procède à l'estimation des paramètres via l'algorithme EM que nous avons implanté précédemment. On donne des valeurs de départ à l'algorithme, qui sont choisies aléatoirement entre 0 et 1. Pour le paramètre α, on définit le maximum à 0.5 afin de s'assurer que le maximum global est unique (recommendation de l'article que l'on étudie).

```
# Valeurs de départ
start=list(alpha=runif(1,min=0.001,max=0.5),p_vv=runif(1,0.01,1),p_rv=runif(1,0.01,1))
```

```r
# Application de l'algorithme EM
emres <- algo_em(x,start=start,N=N)

emres$alpha
```

```
## [1] 0.1613974
```

```r
emres$p_vv
```

```
## [1] 0.2847214
```

```r
emres$p_rv
```

```
## [1] 0.3490597
```

On trouve α=0.1613974, p_{VV}=0.2847214, p_{RV}=0.3490597 et donc on en déduit que $p_{RR} = 0.3662188$.

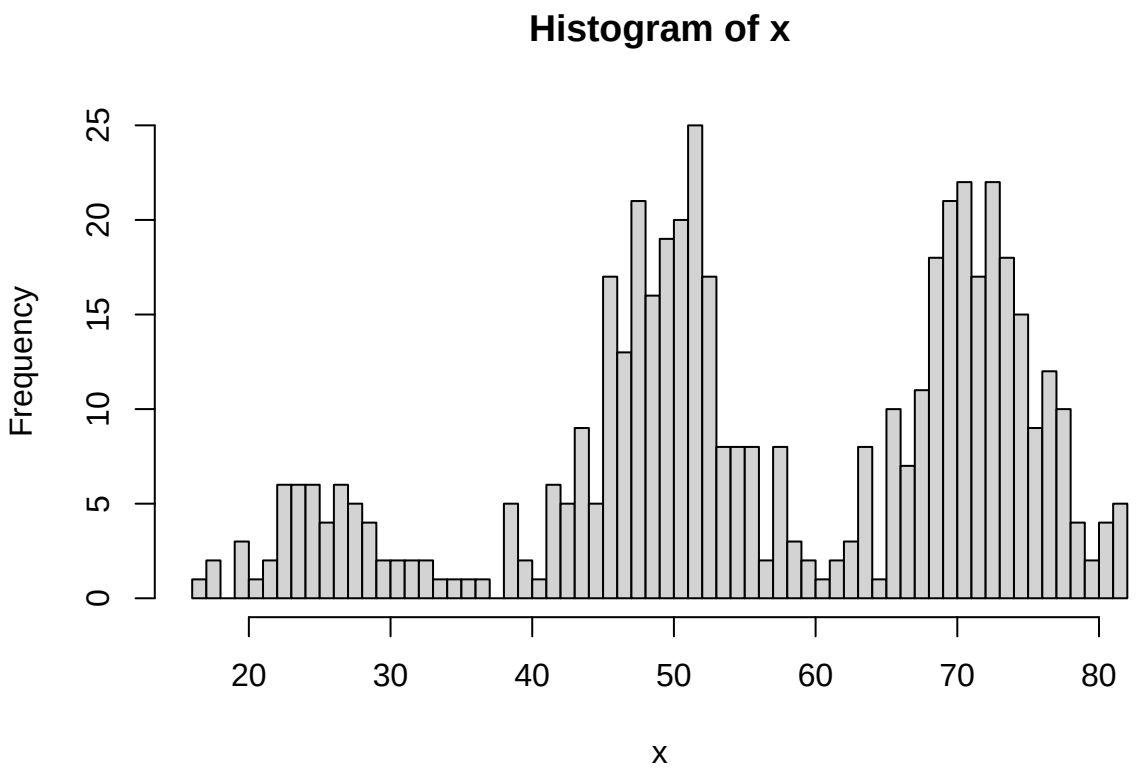
L'estimation est satisfaisante puisque les valeurs sont très proches de celles que nous avions fixées ($p_{VV} = 0.3$, $p_{RV} = 0.4$ et $p_{RR} = 1 - p_{VV} - p_{VV} = 0.3$ ainsi que $\alpha = 0.15$).

Maintenant, on réalise à un nouveau un test en changeant la valeur des paramètres et de N. On fixe $N = 100$, $\alpha = 0.27$, et $p_{VV} = 0.5$, $p_{RV} = 0.4$ et $p_{RR} = 1 - p_{VV} - p_{VV} = 0.1$

```r
set.seed(12345) # pour fixer l'aléatoire

# alpha fixé
alpha=0.27

# N fixé
N=100

x <- data_generation(p=c(0.4,0.5,(1-0.4-0.5)), # probabilités p_vv, p_rv, p_rr
                     N=N, # profondeur de lecture
                     proba_alpha=c(1-alpha,1/2,alpha)) # erreur de lecture
```

On trace une nouvelle fois un histogramme, et on peut, à présent, apercevoir de manière distincte qu'il y a bien 3 groupes. Cette fois-ci, il y a moins d'individus avec un génotype RR, ce qui est normal au vu de la probabilité $p_{RR} = 0.1$.

```r
# Histogramme
hist(x, breaks=50)
```



On estime les paramètres à l'aide de l'algorithme EM.

```
# Valeurs de départ
start=list(alpha=runif(1,min=0.001,max=0.5),p_vv=runif(1,0.01,1),p_rv=runif(1,0.01,1))

# Application de l'algorithme EM
emres <- algo_em(x,start=start,N=N)

emres$alpha
```

```
## [1] 0.2757478
```

```
emres$p_vv
```

```
## [1] 0.4394756
```

```
emres$p_rv
```

```
## [1] 0.4449636
```

A nouveau, l'algorithme fait une bonne estimation des paramètres. On trouve ici, α=0.2757478, p_{VV}=0.4394756, p_{RV}=0.4449636 et donc que $p_{RR} = 0.1155607$.

De ce fait, on peut affirmer que notre algorithme semble bien fonctionner puisque les paramètres à trouver étaient $p_{VV} = 0.4$, $p_{RV} = 0.5$ et $p_{RR} = 1 - p_{VV} - p_{VV} = 0.1$ ainsi que $\alpha = 0.27$).

Conclusion

Proposé pour la première fois dans *Maximum Likelihood from Incomplete Datavia the Em Algorithm* (Dempster, Laird, and Rubin 1977), l'algorithme EM se montre efficace dans divers problèmes et domaines.

Dans notre cas, la méthode`SeqEM`, qui est un algorithme EM pour déterminer le génotype d'invidus, semble être une bonne technique. En effet, cet algorithme est relativement simple à implémenter puisqu'il ne nécessite que quelques calculs au préalable avant de pouvoir être programmer.

De plus, comme nous avons pu le voir à travers nos expérimentations, l'algorithme EM a de bons résultats. Après l'avoir essayé avec différentes valeurs pour les paramètres, il a chaque fois réussi à en faire une bonne estimations.

Il serait intéressant de le tester sur un jeu de données réels afin de voir s'il serait tout aussi efficace. On pourrait également le comparer à des méthodes peut être plus poussées, qui n'auraient pas été mentionnée dans l'article ou bien de voir s'il existe d'autres algorithmes qui pourraient être implémentés afin d'être utilisé dans le domaine de la génétique.

Bibliographie

Dempster, A. P., N. M. Laird, and D. B. Rubin. 1977. "Maximum Likelihood from Incomplete Data via the Em Algorithm." *Journal of the Royal Statistical Society: Series B (Methodological)* 39 (1): 1–22. https://doi.org/https://doi.org/10.1111/j.2517-6161.1977.tb01600.x.

Le Morvan, V., J.-L. Formento, G. Milano, J. Bonnet, and J. Robert. 2005. "Techniques de Recherche Des Polymorphismes Génétiques." *Oncologie* 7 (1): 7–16. https://doi.org/10.1007/s10269-005-0146-8.

Martin, E., D. Kinnamon, M. Schmidt, E. Powell, S. Züchner, and R. W. Morris. 2010. "SeqEM: An Adaptive Genotype-Calling Approach for Next-Generation Sequencing Studies." *Bioinformatics* 26: 2803–10.