Exemple d'utilisation de l'algorithme EM dans le cadre de la génétique

Lucie Guillaumin, Johann Charpentier et Mehdi Chebli

04 novembre 2020

1 Introdution

Ce rapport porte sur l'article (Martin et al. 2010) : c'est un exemple d'utilisation de l'algorithme EM dans le cadre de la génétique.
Nous allons donc considérer un `génotype`, c'est un ensemble ou partie du matériel génétique (ADN) d'un individu. Parmi plusieurs séquences d'ADN, nous allons observer un locus à une position donnée que l'on aura choisie. Dans ce locus nous allons nous intéresser à l'ADN A,T,G ou C qui est le plus présent dans notre locus. On appellera cet ADN le nucléotide `R` pour le désigner comme le nucléotide de référence. En ce qui concerne l'ADN de notre locus, qui ne fait pas partie de nos nucléotides de référence `R`, nous l'appellerons le nucléotide `V` pour désigner comme un nucléotide variant.

On sélectionnera deux nucléotides dans notre locus et nous regardons à quelle appelation (calling) il appartient. Pour résumer nous pouvons avoir :

- `RV` : hétérozygote
- `RR` : homorozygote référent
- `VV` : homorozygote variant

Pour faciliter la compréhension de l'étude nous allons définir ce qu'est un génotype calling, NGS, et un SNP.

- Un `génotype calling` est un algorithme qui cible des allèles dans les génotypes qui permettent de ciblé les intensités de fluorescence et distingue les génotypes de la manière suivante :

  Homozygous allele 1
  Heterozygous
  Homozygous allele 2

- `NGS` signifie "séquencage de nouvelle génération" une expression désignant une variété de techniques de séquençage génétique, qui apportent des améliorations au processus initial de séquençage de Sanger. L'intérêt d'utilisé une telle méthode est de pouvoir réduire les coûts et la durée du séquençage de l'ADN et de l'ARN. Il permet également de réaliser le séquençage sur de plus petits échantillons. Le temps et l'effort requis pour préparer les échantillons destinés au séquençage sont donc également réduits en comparaison avec le séquençage de Sanger.
- Les `SNPs` correspondent à des variations mineures du génome au sein d'une population. Ils sont très courants, puisqu'entre deux personnes prises au hasard, on retrouve environ 3 millions de SNP. Si la plupart du temps ces variations sont silencieuses, ou à l'origine de nos différences morphologiques, elles peuvent néanmoins être à l'origine de maladies génétiques, ou de prédispositions à des maladies.

On appelle également `reads` la lecture de l'alignement des séquences de fragment de nucléotides d'un individu, permettant avec le nombre de nucléotides que l'on obtient de décider si on le classe comme `R` ou `V`.

Sur l'exemple ci-dessous avec la figure 1, nous avons 10 reads, avec une profondeur de lecture de nucléotides égal à 10 (N = 10). Sur ces 10 nucléotides nous en lisons 8 qui sont des nucléotides de référence `R`, et 2 sont des nucléotides variants `V`.

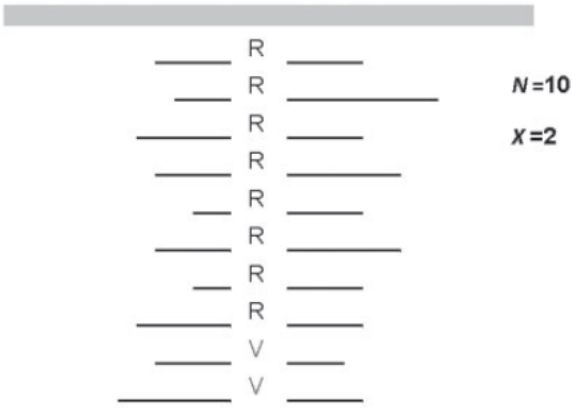


Figure 1: Schéma de 10 séquences de séquençage de nouvelle génération alignées (R = nucléotide de référence,V = nucléotide variant)

Dans cette étude, nous commencerons par expliquer le modèle étudié puis nous le simulerons, ensuite nous calculerons la loi à postériori de ce modèle, nous continuerons par l'explication de ce qu'est un algorithme EM et nous l'implémenterons pour notre modèle et nous finirons par une extension à l'algorithme EM.

2 Le modèle $\mathbb{P}(X, S|\theta)$ ainsi que sa simulation

2.1 Le modèle

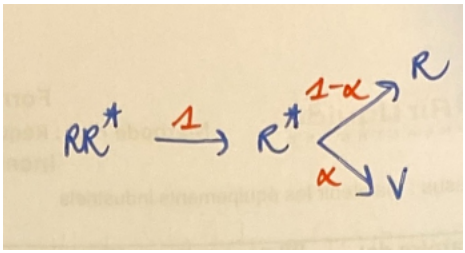
Dans le modèle étudié, nous avons les paramètres suivants :

- $X = (X_i)_{\text{i}\in\{1,\dots,N_i\}}$ représente le nombre de reads `V` au locus pour l'individu i : c'est la variable observée.
- $N_i \in \mathbb{N}^*$ correspond au nombre de reads au locus pour l'individu i.
  On suppose N_i donné.
- α représente la probabilité de se tromper sur un nucléotide.
- $S = (S_i)\text{i} \in \{1, ..., n\}$ avec $S_i \in \{RR, RV, VV\}$ représente le génotype de l'individu i au locus étudié : c'est la variable non observée.
  On remarquera que dans l'article que l'on étudie, S_i est noté G_i.
  De plus, on note $p_{rr} = \mathbb{P}(S_i = RR)$, $p_{rv} = \mathbb{P}(S_i = RV)$ et $p_{vv} = \mathbb{P}(S_i = VV)$.

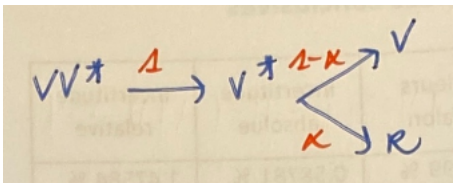
Voici la probabilité du nombre de reads "V" ainsi que le génotype de l'individus au locus choisit
$\mathbb{P}(X, S) = \prod_i \mathbb{P}(Xi, S_i) = \prod_i \mathbb{P}(S_i)\mathbb{P}(X_i|S_i, N_i)$

On a, d'après les informations de l'article, et grâce à l'équilibre de `Hardy Weinberg`, on en déduit que :
$\mathbb{P}(S_i = RR) = 1 - p_{VV} - p_{RV} = p_{RR} = (1 - f)^2$
$\mathbb{P}(S_i = RV) = p_{RV} = 2f(1 - f)$
$\mathbb{P}(S_i = VV) = p_{VV} = f^2$

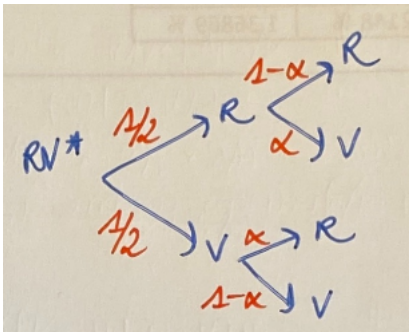
On définit `*` qui veut dire la probabilité d'obtenir réellement le génotype :



Donc $X_i|S_i = RR \sim Bin(N_i, \alpha)$



Donc $X_i|S_i = VV \sim Bin(N_i, 1 - \alpha)$



Donc $X_i|S_i = RV \sim Bin(N_i, \frac{1}{2})$

2.2 Simulation

On simule donc le génotype en choissant un $\alpha = 0.15$, on considère `f` la fréquence du génotype `VV` tel que $f_{VV} = 0.2$ ainsi $p_{VV} = (0.20)^2 = 0.04$

On compare nos estimations :

Table 1: Estimation et comparaison de la proportion des génotypes simulés avec les vraies proportions de génotypes.

	RR	RV	VV
counts	1303.0000	625.0000	72.000
estimate	0.6515	0.3125	0.036
true p	0.6400	0.3200	0.040

Sur les 2000 nucléotides que nous obtenons, on compte 1303 homozygotes référents, 625 hétérozygotes et 72 homozygotes variants. Les données que nous simulons donnent les proportions suivantes : 65.15% homozygotes référents, 31.25% d'hétérozygotes, et 3.6% d'homozygotes variants. Ces simulations sont très proches des vraies proportions : $p_{RR} = 0.64$, $p_{RV} = 0.32$ et $p_{VV} = 0.04$.
D'après le tableau ci-dessus, on peut dire que nos estimations sont proches des vraies proportions de chaque nucléotides dans notre échantillon.
D'après la figure 2 ci-dessous, on peut voir d'après la simulation de ce mélange gaussien qu'il y a une plus forte dispersion chez les hétérozygotes `RV` que chez les homozygotes `RR` et `VV`.
L'identification des classes ne sera pas aisée en conséquence de ces dispersions plus ou moins forte dans chacunes de ces classes.

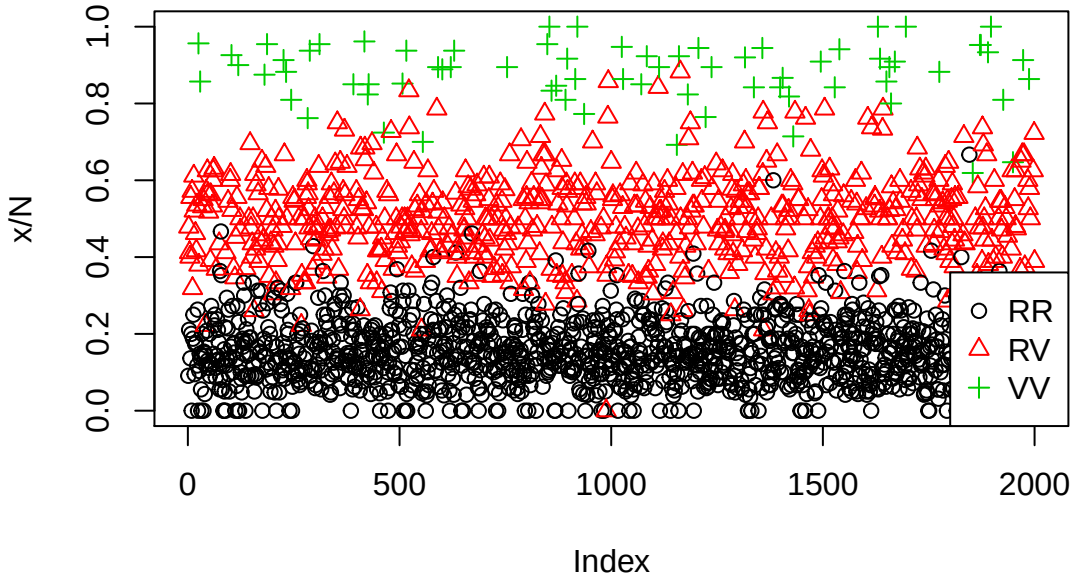


Figure 2 : le ratio Xi/N en ordonné pour la simulation avec 25 de profondeur de lecture et 15% de taux d'erreur. La couleur correspond au vrai genotype.

3 Loi a posteriori $\mathbb{P}(S|X;\theta^{old})$

On est dans le cas d'un mélange, en effet nous avons trois lois `binomiales` différentes.

On a : $\mathbb{P}(S_i = j|X_i, N_i, \theta^{old}) = \eta_i(j)$

Et donc les $\eta_i(j)$ se définissent par :

- $\eta_i(RR) \propto p_{RR} \times \text{dbinom}(X_i, N_i, \alpha)$
- $\eta_i(RV) \propto p_{RV} \times \text{dbinom}(X_i, N_i, \frac{1}{2})$
- $\eta_i(VV) \propto p_{VV} \times \text{dbinom}(X_i, N_i, 1-\alpha)$

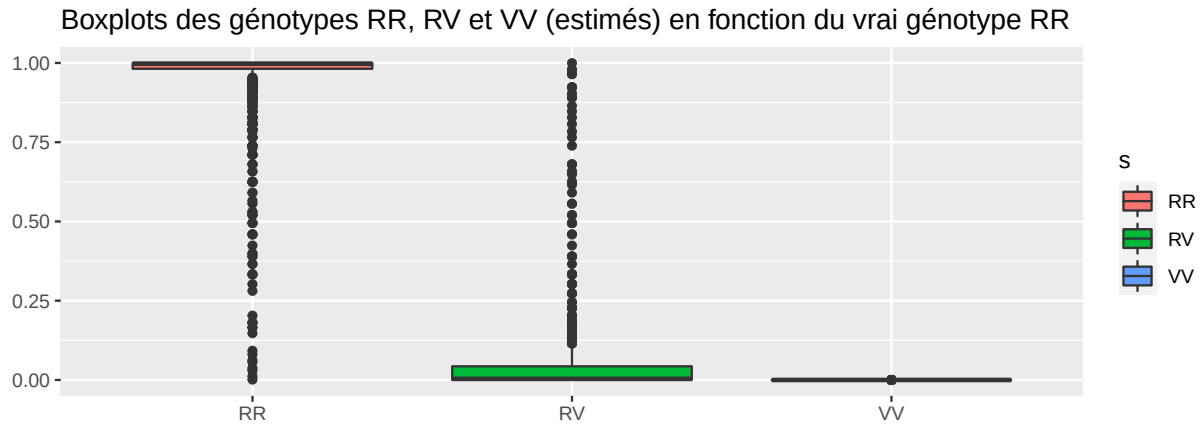
En utilisant les données génerées à la section précedente, nous simulons la loi à posteriori.

Table 2: Table de contingence représentant le nombre de génotypes observés (en lignes) en fonction des vrais génotypes (en colonne).

	RR	RV	VV
RR*	1268	35	0
RV*	34	578	13
VV*	0	6	66

A l'aide la table 2, nous affichons le nombre des génotypes à postériori en fonction du nombre de génotype à priori (avec *).
On remarque que sur 1303 `RR` à priori (d'après la table1), nous obtenons 1268 `RR` à postériori.
De la même façon, on voit que sur 625 `RV` à priori, nous obtenons 578 `RV` à postériori.
Et sur 72 `VV` à priori, nous obtenons 66 `VV` à postériori.
Il y a donc 3% de chance de se tromper pour le génotype `RR`, 7% pour le génotype `RV` et 8% de chance pour le génotype `VV` : ce qui paraît assez faible.

On trace maintenant trois boxplots qui représentent nos trois différents génotypes estimés (à postériori) en fonction du vrai génotype (à priori). Ils représentent tous les trois la table 2.



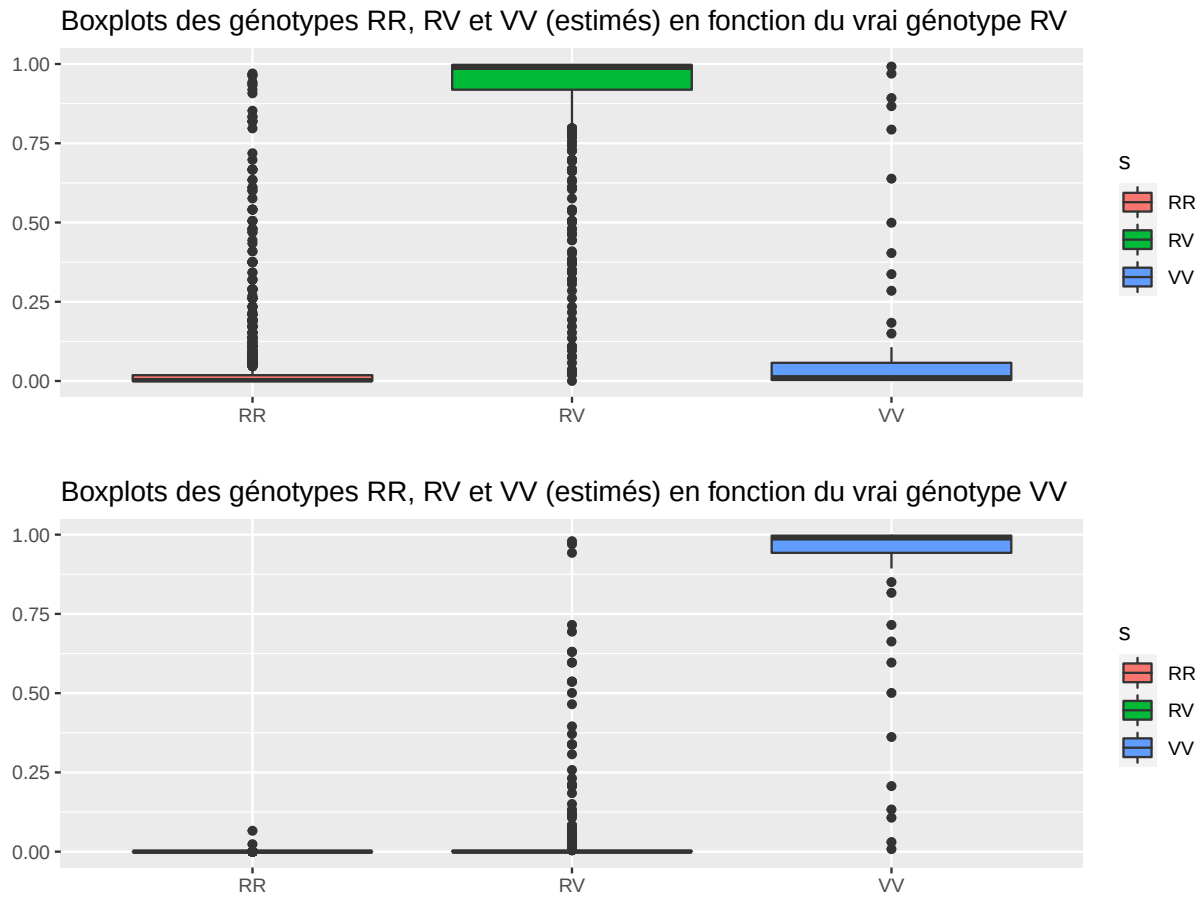


Figure 3: Boxplots des différents génotypes estimés (à postériori) en fonction du vrai génotype (à priori).

4 Algorithme EM et maximisation de $Q(\theta|\theta old)$

4.1 Algorithme EM

L'algorithme EM (espérence-maximisation) est une méthode d'estimation paramétrique qui s'inscrit dans le cadre général du maximum de vraisemblance. Il peut être utilisé pour la classification de données ou encore le machine learning.

Il consiste à :

- une étape d'évaluation de l'espérance (E), où l'on calcule l'espérance de la vraisemblance en tenant compte des dernières variables observées
- une étape de maximisation (M), où l'on estime le maximum de vraisemblance des paramètres en maximisant la vraisemblance trouvée à l'étape E

On utilise ensuite les paramètres trouvés en M comme point de départ d'une nouvelle phase d'évaluation de l'espérance, et l'on itère ainsi.

4.2 Maximisation de Q

On cherche à calculer la fonction `Q` de l'algorithme EM avec les notations suivantes :

$$SS_j = \sum_i \eta_i(j) \quad XX_j = \sum_i \eta_i(j)x_i \quad NN_j = \sum_i \eta_i(j)N_i$$

De plus, on décide de mettre dans une constante, tout ce qui ne concerne pas α et f.

$$\begin{aligned} Q(\theta|\theta^{old}) &= \sum_i \sum_j \mathbb{P}(S_i = j|X_i, N_i, \theta^{old}) \times log[\mathbb{P}(X_i, S_i = j|N_i, \theta)] \\ &= \sum_i \eta_i(RR) log[(1-f)^2 \binom{N_i}{X_i} \alpha^{X_i}(1-\alpha)^{N_i - X_i}] + \sum_i \eta_i(RV) log[2f(1-f)\binom{N_i}{X_i}(\frac{1}{2})^{X_i}(\frac{1}{2})^{N_i - X_i}] \\ &\quad + \sum_i \eta_i(VV) log[f^2 \binom{N_i}{X_i}(1-\alpha)^{X_i}\alpha^{N_i - X_i}] \\ &= \text{cst.} + (SS_{RV} + 2SS_{VV}) log(f) + (2SS_{RR} + SS_{RV}) log(1-f) + (XX_{RR} + NN_{VV} - XX_{VV}) log(\alpha) \\ &\quad + (XX_{VV} + NN_{RR} - XX_{RR}) log(1-\alpha) \end{aligned}$$

En posant :

$$A = SS_{RV} + 2SS_{VV} \quad B = 2SS_{RR} + SS_{RV} \quad C = XX_{RR} + NN_{VV} - XX_{VV} \quad D = XX_{VV} + NN_{RR} - XX_{RR}$$

On peut réécrire `Q` comme ce qui suit :

$$Q(\theta|\theta^{old}) = \text{cste} + log(f)A + log(1-f)B + log(\alpha)C + log(1-\alpha)D$$

Et donc, on obtient comme estimateur pour α et f :
$\hat{f} = \frac{A}{A+B}$
$\hat{\alpha} = \frac{C}{C+D}$

4.3 Implémentation de l'algorithme EM pour notre modèle

Nous implémentons maintenant l'algorithme EM pour notre modèle avec comme paramètre $\alpha = 0.10$ et $f = 0.10$.

Table 3: Résultat de l'algorithme EM à 50 itérations.

	alpha	prr	prv	pvv
thetastar	0.1500000	0.6400000	0.3200000	0.0400000
theta	0.1525076	0.6475663	0.3142975	0.0381362

D'après la table 3, on remarque que pour 50 itérations les valeurs des paramètres estimés θ approchent très convenablement les vraies valeurs de θ^*. Il est intéressant de noter que l'agorithme converge très vite vers les bonnes valeurs à estimer, qui indique que l'algorithme EM est efficace pour prédire les variable cachées. L'approche de manière pragmatique permet de voir dans la table 3 que les choix de départ pour $\theta^* = (f^*, \alpha^*)$ sont sensiblement proches du vrai $\theta = (f, \alpha)$ mais cela ne témoigne pas d'une généralité. Ici nous avons vu que l'algorithme EM était bon mais nous allons essayer de vérifier si les résultats obtenus sont effectivement interpratables.

5 Extension

Après avoir construit l'algorithme EM on a pu convenir qu'il fallait un peu moins d'une cinquantaine de valeur pour que nos estimations de θ^* convergent vers la bonne valeur de notre estimateur θ.
Dans cette partie on a construit à partir des Xi des échantillons bootsrap.

```
##   V1 V2 V3 V4 V5
## 1  3  5  7  8  3
## 2  1 10  4  4 11
## 3  0  3  3  2  9
## 4  4  1  9  7 20
## 5 10 19  7  5  2
## 6 11 14  3  3  1
```

```
## [1] 2000  500
```

On observe 500 échantillons bootstrap Yi pour $i = (1, .., n)$ contenant 2000 observations(chaque variables représentant un échantillon), on va visuellement s'intéresser à leurs comportements.

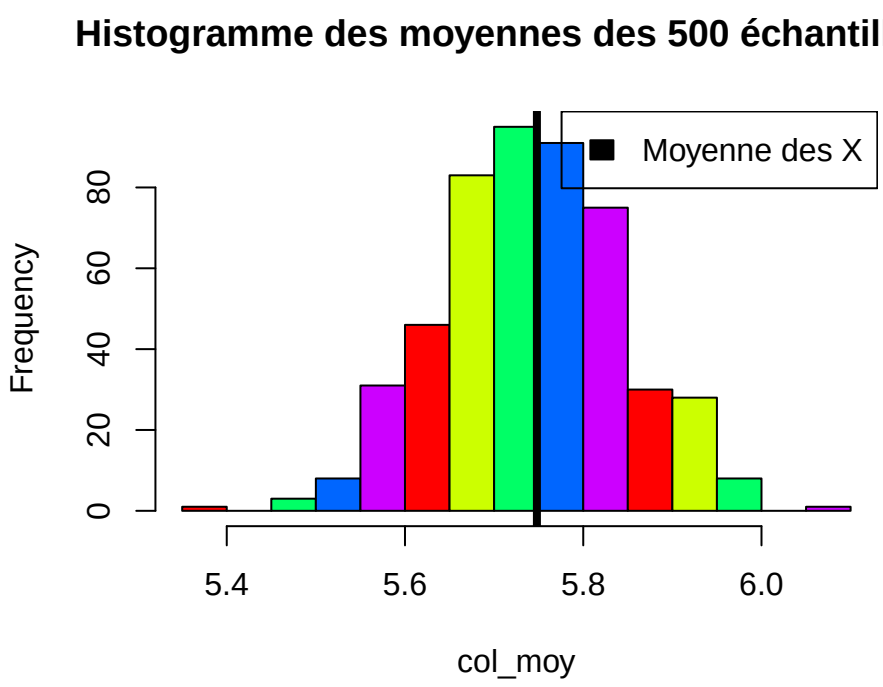


Figure 4: Histogramme des moyenes des 500 échantillons.

D'après la figure 4, l'allure de la courbe est gaussienne sur des valeurs centrées autour de $\overline{X}$. Les $\overline{Yi}$ étant vraiment semblables à $\overline{X}$ cela amène à construire un intervalle de confiance pour la moyenne des échantillons de niveau de confiance α=0.05%:

```
## [1] 5.5455 5.9655
```

```
## [1] 5.748
```

On lit sur la sortie un intervalle de confiance à 95%, $IC = [5.5610; 5.9425]$ ainsi que $\overline{X} = 5.748$ à l'intérieur de l'intervalle puisque nous avons logiquement vu que les valeurs étaient centrées sur $\overline{X}$.
Un premier pas dans la démarche d'interprétations des résultats.
Procédons donc à l'algorithme EM pour essayer de prédire les bonnes valeurs de $\theta = (f, \alpha)$ avec un Yi pour $i = 22$:

	alpha	prr	prv	pvv
thetastar	0.1500000	0.64	0.32	0.04
theta	0.3014779	0.81	0.18	0.01

```
## iter= 50 loglik= -6627.421 f= 0.1149433 alpha= 0.3014779
```

Après avoir effectué l'algorithme, on constate une nouvelle fois qu'il converge très rapidement vers $\theta = (f, \alpha)$ pour un même nombre d'itérations. Pour un $\alpha^* = 0.15$ et $f^* = 0.20$ choisi de la même manière que pour le 1er algorithme, on en a ressorti $\alpha^* = 0.30$ et $f^* = 0.11$.
Toutefois pour éviter les erreurs de calculs, les $\eta i < \epsilon$ avec $\epsilon = 10^-15$ ont étés retirés de l'experience.
On voit donc bien que l'algorithme est toujours aussi efficace et se revèle donc un bon moyen pour expliquer S. Cela convient donc à dire que les résultats fournis par l'algorithme EM sont interprétables et se révèle efficace pour déterminer sur chaque locus quels sont les SNPs variant à l'aide d'outils prédictibles bons.

6 Conclusion

Dans l'ensemble de cette atricle, le but principal étant après lecture biochimique (avec une profondeur de lecture $N = 10$) de déterminer les SNPs afin de prédire le nombre d'acides aminées variants qui determinera le genotype de l'individu i.
Cette problématique a necéssité la construction d'un algorithme EM dans la cadre d'un modèle de mélange afin d'estimer les paramètres $\theta = (f, \alpha)$ résultant du calcul de $Q(\theta|\theta^{old})$ à partir de $\theta^* = (f^*, \alpha^*)$.
Suite à la construction de l'algorithme nous avons lancé l'expérience pour prédire S et avions constaté qu'il fonctionnait bien et très vite pour estimer nos θ. Une fois cela fait on a procédé à un verification de nos résultats grâce à la méthode du bootstrap pour savoir si notre algorithme produisait un résultat interpretable et avons vu qu'il etait bon.
Dans une partie exploratoire, il serait vraiment intéressant de voir l'effet de X sur S en effectuant une validation croisée pour estimer le nombre d'itérations qui minimise le MSE (mean square error) tout en optimisant son nombre d'itérations.

7 Références

Martin, Eden R, D D Kinnamon, Michael A Schmidt, E H Powell, S Zuchner, and R W Morris. 2010. "SeqEM: an adaptive genotype-calling approach for next-generation sequencing studies." Bioinformatics 26 (22). Oxford University Press: 2803–10.