Rapport sur l'étude de l'algortihme SeqEM

Thomas DINH, Naila BOUTERFA et Wendy LEFEVRE

13 novembre, 2018

Contents

1 Introduction

Dans ce document vous trouverez un rapport de l'étude de l'article *SeqEM: an adaptive genotype-calling approach for next-generation sequencing studies* Martin et al. (2010). Un article qui consiste en la présentation d'une méthode statistique adaptative de génotypage pour les études de séquençage NGS ou *New Generation Sequencing*.

L'étude se place dans un contexte ou l'on souhaite déterminer le génotype d'une personne et ou on s'intéresse plus particulièrement à un *Single Nucleotide polymorphism* (SNP). Une fois qu'un certain nombre de séquences se chevauchant sont alignées, le nombre de nucléotides de référence et le nombre de nucléotides variants sur la séquence d'ADN sont comptés. C'est ce qui sera représenté dans le modèle. En ommettant les erreurs de séquencage, un individu homozygote pour un locus devrait fournir soit uniquement des référents soit uniquement des variants. Cependant, à cause de l'échantillonage aléatoire des paires de base homologues et d'erreurs de séquençage ou d'alignement, le comptage n'identifie pas directement le génotype en ce locus. Le manque de certitude à ce sujet pousse à recourir à l'usage du *genotype calling-algorithm* ou algorithme de détection de génotype pour déterminer le génotype d'un individu à partir de nombreuses séquences de reads. Ainsi appliqué au modèle il prend en compte ces degrés d'incertitude afin qu'ils soient minimisées par la suite.

L'algorithme SeqEM (Esperance Maximisation pour séquençage) met en application l'algo EM déjà connu Dempster, Laird, and Rubin (1977) à une vraisemblance adéquate pour un échantillon d'individus sans parenté et exploitant des infos à partir de l'échantillon pour estimer :
-Les probabilités de génotypes
-Le taux d'erreur nucléotide-read

Il est aussi démontré par les auteurs à travers des calculs analyiques et des simulations que l'algorithme Seq-EM donne un taux d'erreur autant ou plus faible dans la détection du génotype que les méthodes de

filtrage et la méthode MAQ. La méthode est donc améliorée, plus robuste et flexible. Cependant ce n'est pas l'aspect auquel nous allons nous intéresser dans ce rapport.

Nous allons dans un premier temps étudier le modèle pour comprendre son fonctionnement à travers plusieurs simulations puis nous procéderons à l'estimation des paramètres du modèle et à l'évaluation de ces estimateurs. Enfin, nous mettrons en application l'algorithme EM au modèle.

2 Présentation du modèle

2.1 Definition du modèle et des paramètres

Nous allons définir le système de notation pour le modèle étudié :

N_i	Nombre de reads pour l'individu i, avec $1 < i < n$
n	Nombre total d'individus
X_i	Données observées : Nombre de variants pour l'individu i avec $1 < i < n$
G_i	Données non observées : Génotype de l'individu i $G_i \in \{\text{RR}, \text{RV}, \text{VV}\}$ avec $1 < i < n$ La
α	probabilité que R soit appelé V et que V soit appelé R par erreur (supposées égales)
R	Nucléotide référent
V	Nucléotide variant
p_{RR}, p_{RV}, p_{VV}	Probabilité qu'ont respectivement les génotypes RR, RV, VV d'être représentés

Le modèle que nous allons voir est un modèle de mélange construit à partir d'une partie générative et d'une densité. $\theta = (\alpha, p_{RV}, p_{VV})$ est le paramètre du mélange que nous allons chercher à estimer. Le modèle est défini ainsi :

$$\mathbb{P}(X_i, G_i | N_i; \theta) = \mathbb{P}(G_i | N_i; \theta)\mathbb{P}(X_i | G_i; N_i; \theta)$$

Ce qui nous amène à écrire pour chaque génotype $G_i \in \{RR, RV, VV\}$ les densités suivantes :

$$\mathbb{P}(X_i | G_i = RR; N_i; \theta) = \texttt{dbinom}(X_i, \texttt{size} = N_i, \texttt{prob} = 1 - \alpha)$$

$$\mathbb{P}(X_i | G_i = RV; N_i; \theta) = \texttt{dbinom}(X_i, \texttt{size} = N_i, \texttt{prob} = \frac{1}{2})$$

$$\mathbb{P}(X_i | G_i = VV; N_i; \theta) = \texttt{dbinom}(X_i, \texttt{size} = N_i, \texttt{prob} = \alpha)$$

On sait également que $\mathbb{P}(G_i | N_i; \theta)$ se décompose comme :

$$\begin{aligned} \mathbb{P}(G_i = RR | N_i; \theta) = &\quad 1 - p_{VV} - p_{RV} \\ \mathbb{P}(G_i = RV | N_i; \theta) = &\quad p_{RV} \\ \mathbb{P}(G_i = VV | N_i; \theta) = &\quad p_{VV} \end{aligned}$$

Les hypothèses concernant ces quantités sont : $p_{RV} + p_{VV} < 1$ et $p_{RV}, p_{VV} > 0$

2.2 Simulations et variations des paramètres

Nous allons maintenant simuler $\theta = (\alpha, pRV, pVV)$ pour comprendre le rôle de certains paramètres dans la représentation des génotypes. Evidemment, dans la réalité, ce sont les paramètres α, p_{RV} et p_{VV} que nous souhaitons obtenir alors qu'ici nous fixons arbitrairement leurs valeurs que nous notons θ^*, nous les connaissons donc au préalable. Cela nous permet de faire varier les paramètres afin de comprendre le modèle avant de nous lancer dans des simulations et de rechercher ces paramètres selon les méthodes de vraisemblance et l'algorithme EM.

Après avoir paramétré sur R la simulation, nous sommes parvenus aux conclusions suivantes :

En figure 1, nous voyons que plus la valeur du taux d'erreur est petite, plus la séparation des différents génotypes possibles est nette. Aussi, en figure 2 lorsque le nombre de reads N est grand, on peut mieux observer la séparation et donc mieux classer les génotypes ce qui est logique puisqu'avec plus de "reads", on a plus d'informations.

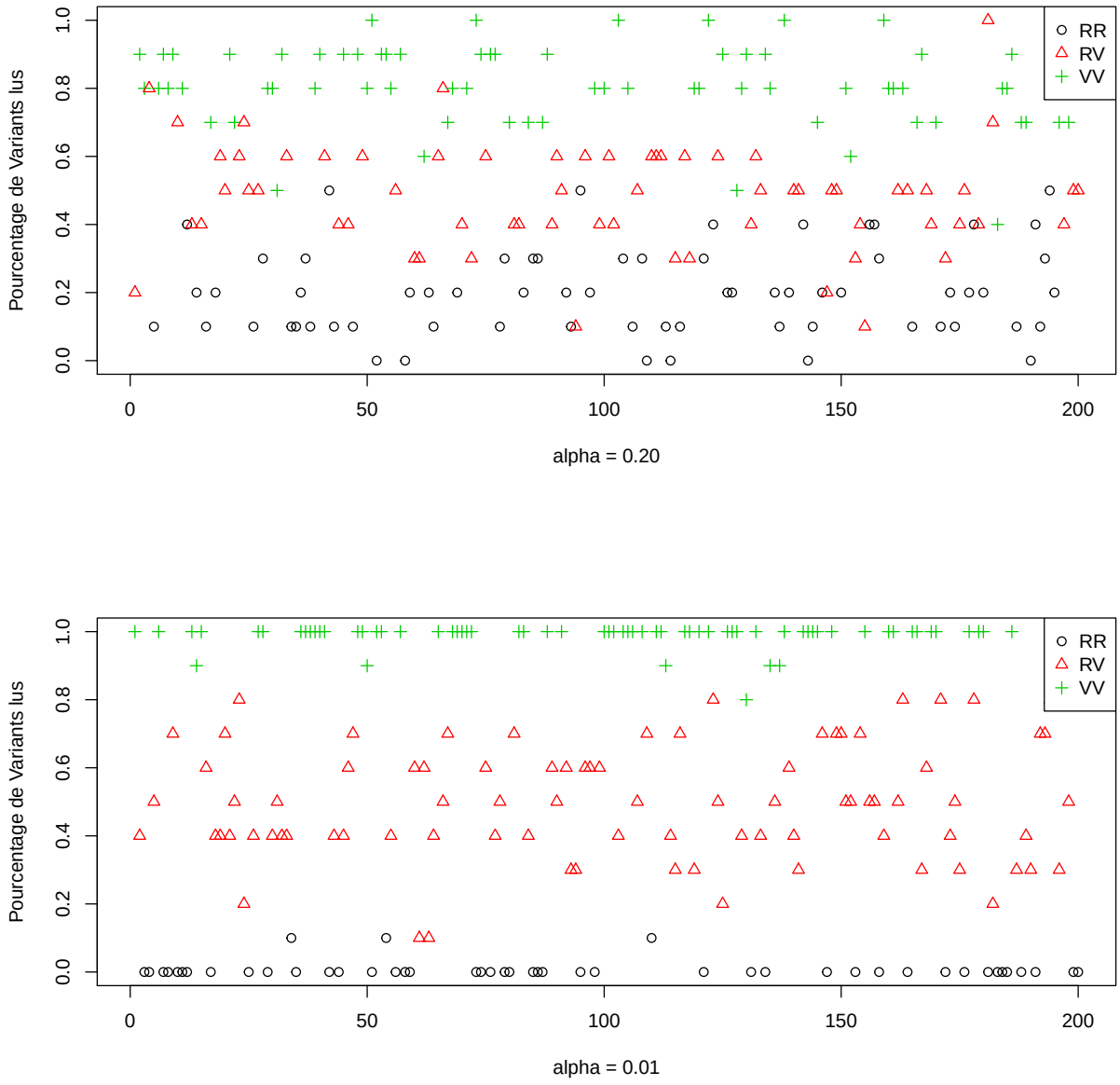

Figure 1: Influence du paramètre alpha pour la séparation des génotypes (n = 200, p = (0.3,0.4,0.3), N = 10)

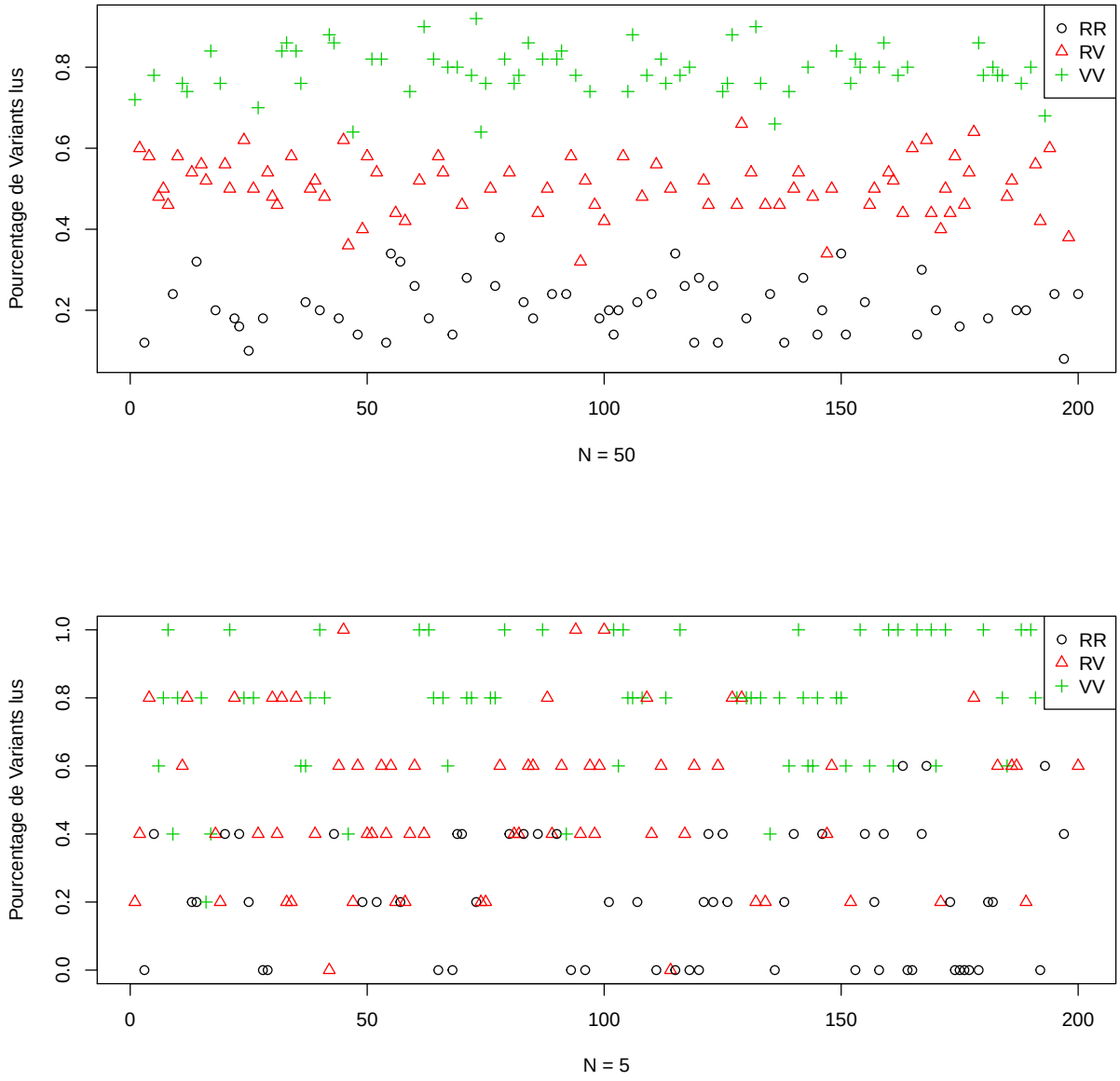

Figure 2: Influence du paramètre N pour la séparation des génotypes (n = 200, p = (0.3,0.4,0.3), alpha = 0.20)

Finalement, on peut voir que la caractérisation d'un génotype n'est pas toujours évidente, il faudrait dans l'idéal un α très petit ainsi qu'un grand nombre de reads N. Cependant les paramètres optimaux ne seraient pas facile à mettre en place car trop coûteux.

Table 1: Tableau comparatif entre les valeurs réelles et simulées

	pRV	pVV	alpha
Estimate	0.33	0.34	0.1962687
True	0.40	0.30	0.2000000

3 Estimation des paramètres du modèle par maximum de vraisemblance

3.1 La vraisemblance des données

Nous nous intéressons ici à la méthode d'estimation du maximum de vraisemblance pour rechercher les paramètres choisis dans la simulation créée précédemment.

$$\ell(\theta|x,g) = \log \mathbb{P}(X = x, G = g|\theta) \quad \tilde{\theta} = \arg\max_{\theta} \ell(\theta|x,g)$$

Après calcul de la dérivée du log-vraisemblance dont vous trouverez le développement en annexe 1, on obtient les résultats suivants :

$$p_{\tilde{RV}} = \frac{n_{RV}}{n_{RV} + n_{VV} + n_{RR}} = \frac{n_{RV}}{n}$$
$$p_{\tilde{VV}} = \frac{n_{VV}}{n_{RV} + n_{VV} + n_{RR}} = \frac{n_{VV}}{n}$$
$$\tilde{\alpha} = \frac{x_{RR} + N \times n_{VV} - x_{VV}}{N \times (n_{VV} + n_{RR})}$$

Avec : $n_{RR} = \sum_{i,g_i=RR} 1, n_{RV} = \sum_{i,g_i=RV} 1, n_{VV} = \sum_{i,g_i=VV} 1$
Et : $x_{RR} = \sum_{i,g_i=RR} x_i, x_{RV} = \sum_{i,g_i=RV} x_i, x_{VV} = \sum_{i,g_i=VV} x_i$

Par la suite, nous appelerons $\tilde{\theta}$ le paramètre estimé par cette méthode. Nous allons nous servir de ces calculs pour vérifier si les paramètres estimés de ce modèle sont plus ou moins proches des vraies valeurs (les valeurs que nous avons choisi dans la première partie). Evidemment, dans la réalité nous ne connaissons pas les vraies valeurs mais ici, le but est de comparer les méthodes.

Nous allons simuler $\tilde{\theta}$, nous voyons dans la table 1 qu'il se rapproche assez des vraies valeurs, mais qu'il reste assez imprécis.

3.2 Evaluation de la qualité de l'estimateur

Pour vérifier à quel point notre paramètre estimé $\widetilde{\theta}$ est bon, nous allons simuler 100 fois l'expérience et regarder l'intervalle de confiance des paramètres, le biais, la variance, et l'erreur quadratique dont les résultats se trouvent dans la table 2. En figure 3, nous avons également construit l'histogramme des 100 valeurs estimées pour chacune des variables auxquelles on s'intéresse :

Table 2: Mesures statistiques des estimateurs

	Alpha	pRV	pVV
IC inf	0.197327880622935	0.396262266313804	0.294665027818325
IC sup	0.201406746963854	0.408737733686196	0.307434972181675
Biais	-0.000632686206605476	0.0025	0.00105
Variance	0.000108273649436629	0.00101287878787879	0.0010612601010101
MSE	2.58426394534294e-06	2.58426394534294e-06	2.58426394534294e-06

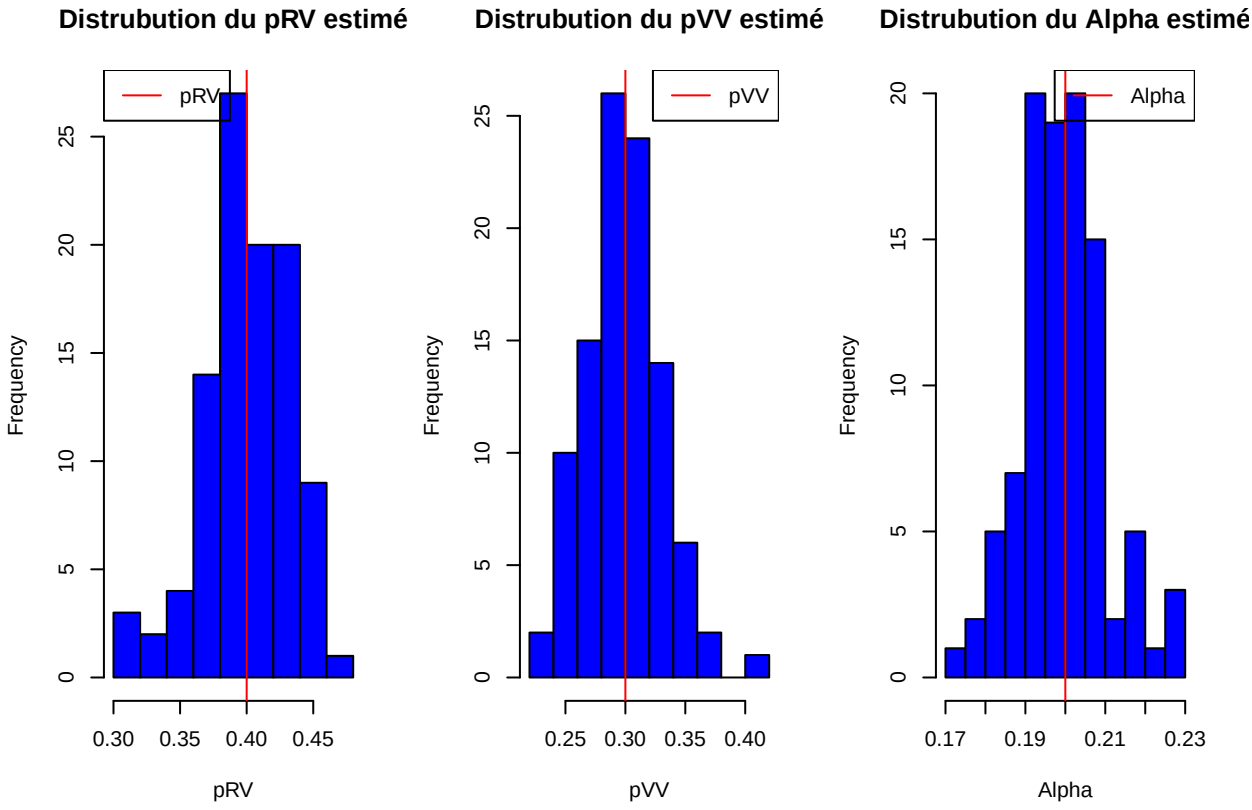

Figure 3: Distribution des probabilités estimées sur 100 simulations

Table 3: Pourcentage de représentation du génotype selon le nombre de nucléotides variants lus

	X=0	X=1	X=2	X=3	X=4	X=5	X=6	X=7	X=8	X=9	X=10
RR	98.8	95.4	83.7	56.2	24.0	6.9	1.5	0.2	0.0	0.0	0.0
RV	1.2	4.6	16.2	43.6	74.5	86.1	74.5	43.6	16.2	4.6	1.2
VV	0.0	0.0	0.0	0.2	1.5	6.9	24.0	56.2	83.7	95.4	98.8

3.2.1 Loi à posteriori

Dans ce paragraphe, nous définissons ce qu'est la loi à posteriori :

$\eta_i(g) = \mathbb{P}(G_i = g | X_i = x_i, N_i, \theta)$ avec $g \in$ {RR,RV,VV}

$\eta_i(g) = \frac{\mathbb{P}(X_i=x_i, N_i, \theta | G_i=g)\mathbb{P}(G_i=g)}{\mathbb{P}(X_i=x_i, N_i)}$

$\eta_i(g) = \frac{\texttt{dbinom}(X_i, \texttt{size}=N_i, \texttt{prob}=1-\alpha)\times p_g}{\mathbb{P}(X_i=x_i, N_i)}$

Nous calculons ainsi les η définis.

La table 3 nous fait remarquer que :

- Plus de nucléotides variants sont lus, plus il y a des chances que l'individu soit de génotype VV,
- Moins il y en a, plus il y a des chances que l'individu soit RR,
- Entre les deux, c'est à dire quand il y a presque autant de variants que de référents, il y a plus de chance qu'il soit RV (on n'approche pas autant les 99% que pour les deux autres courbes). Nous pouvons voir la représentation associée à ses résultats dans la figure 4.

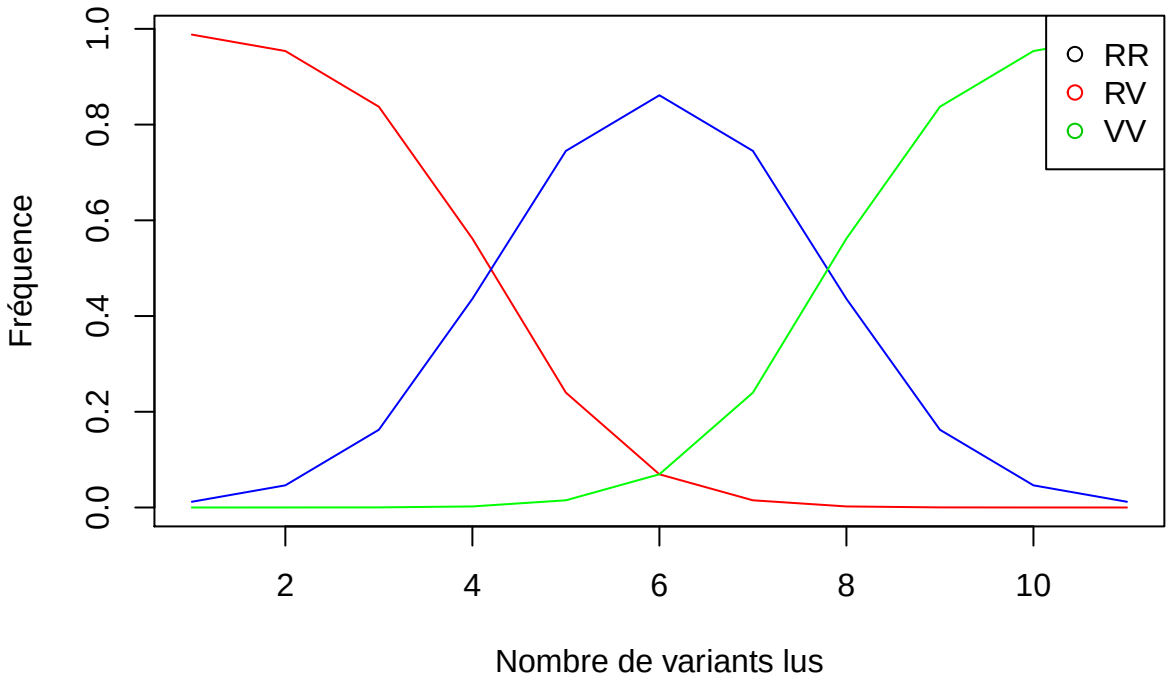

Figure 4: Probabilité d'appartenance à un génotype selon le nombre de variants lus

4 Estimation par l'Algorithme EM

On procède à une maximisation de la vraisemblance des données observées en prenant en compte les fréquences génotypiques et le taux d'erreur de nucléotide-reads. En utilisant une estimation du maximum de vraisemblance de ces paramètres, on simule numériquement les probas "à posteriori" de chaque génotype en utilisant les données reads et en attribuant à chaque individu le génotype avec la plus grande probabilité à postériori. C'est une procédure de classification bayésienne qui va nous permettre de minimiser le taux de classification erroné du génotype ou *genotype call error*.

4.0.1 Calcul de Q

La définition de la fonction Q est la suivante :

$$Q(\theta|\theta^{\text{old}}) = \sum_{G} \mathbb{P}(G|X;\theta^{\text{old}}) \log \mathbb{P}(X,S|\theta)$$

dont le calcul détaillé en annexe 2 donne :

$$Q(\theta|\theta^{\text{old}}) = constante + A\log(\alpha) + B\log(1-\alpha) + C\log(1-p_{VV}-p_{RV}) + D\log(p_{RV}) + E\log(p_{VV})$$

Avec :

$A = \sum_{i,g_i=RR} \eta_i(RR)x_i + \sum_{i,g_i=VV} \eta_i(VV)N_i - \sum_{i,g_i=VV} \eta_i(VV)x_i$

$B = \sum_{i,g_i=RR} \eta_i(RR)N_i - \sum_{i,g_i=RR} \eta_i(RR)x_i + \sum_{i,g_i=VV} \eta_i(VV)x_i$

$C = \sum_{i,gi=RR} \eta_i(RR)$

$D = \sum_{i,gi=RV} \eta_i(RV)$

$E = \sum_{i,gi=VV} \eta_i(VV)$

On cherche à estimer :

$$\hat{\theta} = \theta^{\text{new}} = \arg\max_{\theta} Q(\theta|\theta^{\text{old}})$$

On prend un θ arbitraire (ici = 0.4 et $(p_{RR}, p_{RV}, p_{VV}) = (0.6, 0.1, 0.3)$), puis on calcule le θ^{new} avec la fonction Q, ce θ^{new} obtenu sera utilisé dans la fonction Q pour en calculer un nouveau et ainsi de suite jusqu'à convergence de θ . Nous avons choisi d'itérer l'algorithme EM 200 fois. Par souci de présentation, on a décidé de ne montrer qu'une partie des itérations

```
## iter= 1 alpha= 0.3005657 p= 0.5393732 0.08404927 0.3765775
## iter= 2 alpha= 0.2575763 p= 0.4896654 0.07229719 0.4380374
## iter= 3 alpha= 0.2508799 p= 0.4722598 0.07644652 0.4512937
## iter= 4 alpha= 0.2489685 p= 0.4651777 0.08391835 0.450904
## iter= 5 alpha= 0.2473871 p= 0.4601139 0.09233978 0.4475463
## iter= 15 alpha= 0.2236773 p= 0.3977007 0.2105593 0.39174
## iter= 20 alpha= 0.2086705 p= 0.3630717 0.2760509 0.3608774
## iter= 30 alpha= 0.1896094 p= 0.3227445 0.3523155 0.32494
## iter= 50 alpha= 0.1838145 p= 0.3110607 0.3744888 0.3144505
## iter= 75 alpha= 0.1836442 p= 0.3107204 0.3751356 0.3141439
## iter= 100 alpha= 0.1836424 p= 0.3107168 0.3751425 0.3141407
## iter= 150 alpha= 0.1836424 p= 0.3107168 0.3751426 0.3141407
## iter= 200 alpha= 0.1836424 p= 0.3107168 0.3751426 0.3141407
```

Table 4: Comparaison des différents estimateurs avec le paramètre de base défini

	alpha	pRR	pRV	pVV
theta_star	0.2000000	0.3000000	0.4000000	0.3000000
theta_tilde	0.1962687	0.3300000	0.3300000	0.3400000
theta_hat	0.1836424	0.3107168	0.3751426	0.3141407

Table 5: Résultats d'estimations avec différentes valeurs de départ

	alpha	pRR	pRV	pVV
alpha=0.01, p=c(0.6,0.1,0.3)	0.183642380617334	0.310716788706429	0.375142552699797	0.314140658593774
alpha=0.99, p=c(0.6,0.1,0.3)	0.816357619382666	0.314140658593774	0.375142552699797	0.310716788706429
alpha=0.4, p=c(0.05,0.15,0.8)	0.183642380617334	0.310716788706429	0.375142552699796	0.314140658593774

Au vu de nos simulations présentées en table 5, il nous apparait que quels que soient les paramètres p choisis au début de l'algorithme et pour < 0.5, nous sommes proches du vrai θ^* sauf pour alpha supérieur à 0.5, dans ce cas, alpha semble se rapprocher d'une autre valeur qui semble être un autre point de convergence symétrique au premier. L'algorithme SeqEM converge assez rapidement, si on cherche à avoir une précision de 3 chiffres après la virgule, on peut voir qu'après 20 itérations les paramètres ne changent plus et qu'après 75 itérations il y a une convergence complète. (On voit dans la table 4 le résultat de l'algorithme)

5 Conclusion

L'algorithme EM est un très bon outil vu le peu d'hypothèses dont il nécessite la vérification, particulièrement quand on considère qu'il donne des résultats aussi bon ou meilleurs que ceux obtenus par maximum de vraisemblance , qui nécessite plus d'hypothèses dont nous n'avons pas accès dans la situation que nous modélisons. Basé sur des notions de statistiques bayésiennes, l'algorithme permet d'estimer des paramètres sous-jacents aux probabilités à postériori et donc de façon adaptative au lieu de les spécifier arbitrairement à priori. il ne dépend donc pas de l'information préspécifiée ou connue qu'est la fréquence allèlique. C'est là d'ailleurs qu'il apporte un avantage par rapport aux anciennes méthodes, les méthodes précédemment développées ex MAQ, SOAPsnp ou la méthode classifiant l'individu comme hétérozygote si plus d'un pourcentage spécifié de reads ont des nucléotides variants détectés ne sont pas aussi flexibles.

6 Annexes

6.1 Annexe 1

Vous trouverez ici le développement du calcul vu dans la deuxième partie. Après avoir dérivé la formule de définition, nous obtenons le resultat suivant :

$$\begin{aligned}
\ell(\theta|x,g) &= \log\prod_{i=1}^{n}\mathbb{P}(X_i=x_i,G_i=g_i|N_i;\theta)\\
&= \sum_{i=1}^{n}\log\mathbb{P}(X_i=x_i,G_i=g_i|N_i;\theta)\\
&= \sum_{i=1}^{n}\log\mathbb{P}(X_i=x_i,G_i=RR|N_i;\theta)+\sum_{i=1}^{n}\log\mathbb{P}(X_i=x_i,G_i=RV|N_i;\theta)+\sum_{i=1}^{n}\log\mathbb{P}(X_i=x_i,G_i=VV|N_i;\theta)\\
\ell(\theta|x,g) &= constante + A\log(\alpha)+B\log(1-\alpha)+C\log(1-p_{VV}-p_{RV})+D\log(p_{RV})+E\log(p_{VV})
\end{aligned}$$

On appelle :

$$A=\sum_{i,g_i=RR}x_i+\sum_{i,g_i=VV}N_i-\sum_{i,g_i=VV}x_i$$

$$B=\sum_{i,g_i=RR}N_i-\sum_{i,g_i=RR}x_i+\sum_{i,g_i=VV}x_i$$

$$C=\sum_{i,gi=RR}1$$

$$D=\sum_{i,gi=RV}1$$

$$E=\sum_{i,gi=VV}1$$

On determine l'estimateur de α :

$$\frac{\partial\ell(\theta|x,g)}{\partial\alpha}=0\Leftrightarrow\alpha=\frac{A}{A+B}\Leftrightarrow\alpha=\frac{x_{RR}+N\times n_{VV}-x_{VV}}{N\times(n_{VV}+n_{RR})}$$

On détermine les estimateurs de p_{RV} et p_{RV} :

$$\begin{cases}\dfrac{\partial\ell(\theta|x,g)}{\partial p_{RV}} = \dfrac{-C}{1-p_{VV}-p_{RV}}+\dfrac{D}{p_{RV}} = 0\\[2ex] \dfrac{\partial\ell(\theta|x,g)}{\partial p_{RV}} = \dfrac{-C}{1-p_{VV}-p_{RV}}+\dfrac{E}{p_{VV}} = 0\end{cases}\Longleftrightarrow\begin{cases}p_{RV}=\dfrac{D}{C+D+E}\\[2ex] p_{VV}=\dfrac{E}{C+D+E}\end{cases}\tag{1}$$

6.2 Annexe 2

La définition de la fonction Q est la suivante :

$$Q(\theta|\theta^{old}) = \sum_{G} \mathbb{P}(G|X;\theta^{old}) \log \mathbb{P}(X,G|\theta)$$

$$Q(\theta|\theta^{old}) = \sum_{i=1} (\sum_{g \in \{RR,RV,VV\}} \eta_i(g) \log \mathbb{P}(X,G|\theta))$$

$$= \sum_{i=1} \{\eta_i(RR) \log(\binom{N_i}{x_i} \alpha^{x_i}(1-\alpha)^{N_i-x_i}) + \eta_i(RV) \log(\binom{N_i}{x_i} \alpha^{x_i}(1-\alpha)^{N_i-x_i}) + \eta_i(RR) \log(\binom{N_i}{x_i} \alpha^{x_i}(1-\alpha)^{N_i-x_i})\}$$

$$= \sum_{i=1} \{\eta_i(RR) \log(\binom{N_i}{x_i} \alpha^{x_i}(1-\alpha)^{N_i-x_i})\} + \sum_{i=1} \{\eta_i(RV) \log(\binom{N_i}{x_i} \alpha^{x_i}(1-\alpha)^{N_i-x_i})\} + \sum_{i=1} \{\eta_i(RR) \log(\binom{N_i}{x_i} \alpha^{x_i}(1-\alpha)^{N_i-x_i})\}$$

$$= constante + A \log(\alpha) + B \log(1-\alpha) + C \log(1 - p_{VV} - p_{RV}) + D \log(p_{RV}) + E \log(p_{VV})$$

$$A = \sum_{i,g_i=RR} \eta_i(RR) x_i + \sum_{i,g_i=VV} \eta_i(VV) N_i - \sum_{i,g_i=VV} \eta_i(VV) x_i$$

$$B = \sum_{i,g_i=RR} \eta_i(RR) N_i - \sum_{i,g_i=RR} \eta_i(RR) x_i + \sum_{i,g_i=VV} \eta_i(VV) x_i$$

$$C = \sum_{i,gi=RR} \eta_i(RR)$$

$$D = \sum_{i,gi=RV} \eta_i(RV)$$

$$E = \sum_{i,gi=VV} \eta_i(VV)$$

Références

Dempster, Arthur P, Nan M Laird, and Donald B Rubin. 1977. "Maximum Likelihood from Incomplete Data via the Em Algorithm." *Journal of the Royal Statistical Society. Series B (Methodological)*. JSTOR, 1–38.

Martin, Eden R, DD Kinnamon, Michael A Schmidt, EH Powell, S Zuchner, and RW Morris. 2010. "SeqEM: An Adaptive Genotype-Calling Approach for Next-Generation Sequencing Studies." *Bioinformatics* 26 (22). Oxford University Press: 2803–10.