Université de Montpellier

Faculté des Sciences

M2 Statistiques et Sciences des Données

TP2 - A l'affût des chaînes de Markov cachées

HAX004X

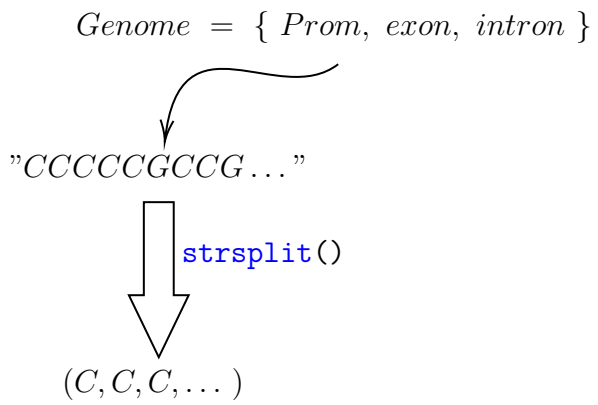
Auteurs:
Anas Zakroum

Enseigné par:
Pre. Alice Cleynen

21 février 2023

Partie 1 : Chaînes de Markov

Nous disposons du jeu de donnée *Genome* décrivant un gène connu pour être relié à un grand nombre de cancers. Précisement nous disposons des séquences de nucléotides du premier exon, le premier intron et la région promotrice.
Genome est sous la forme d'une liste de ces trois éléments exprimés sous forme de suite d'alphabet correspondants aux différents nucléotides ; A,C,G , T .
Nous pouvons séparer chaque suite d'alphabet de sorte à récupérer un vecteur à l'aide de la fonction `strsplit()`.



Ainsi nous retrouvons que le premier exon est de taille 179, la région promotrice est de taille 201 et le premier intron est de taille 101.
Regardons la composition de la région promotrice.

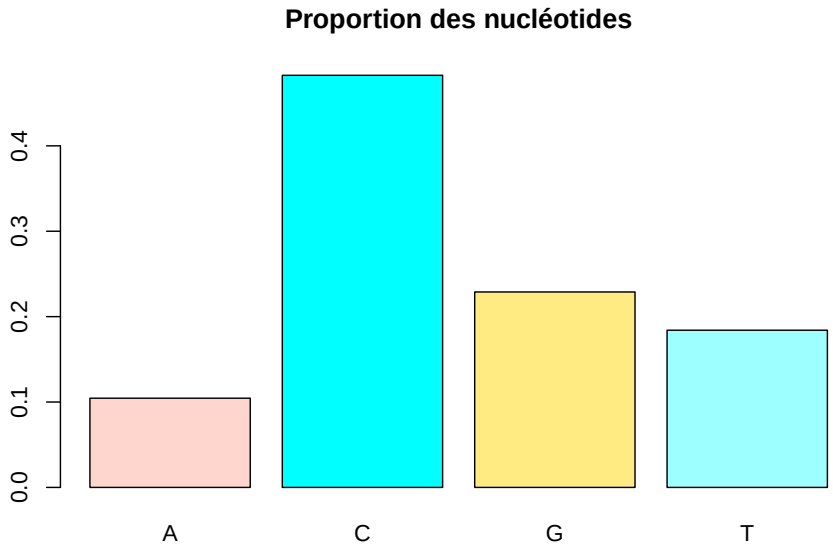


FIGURE 1 – Proportion de nucléotides dans la région promotrice.

En se positionnant dans un paradigme probabiliste, nous pouvons supposer que les transitions d'une lettre de l'alphabet $\mathcal{A} = \{A, C, G, T\}$ à une autre suivent un régime markovien discret. L'espace d'état étant l'alphabet $\mathcal{A}$, le concept de temps de la chaine de Markov

est remplacé par la "position dans la séquence". Notons Π la matrice de transition de la chaine et $\pi(a,b) = \mathbb{P}(X_n = b | X_{n-1} = a)$ la probabilité de transistion de la lettre a à la lettre b.

Ettant donné une séquence finie $S = (X_1, X_2, \ldots, X_T)$, sa vraisemblance à l'aide du théorème de Bayes et de la prorpiété de Markov s'ecrit comme

$$\nu(S|X_1) = \pi(X_1, X_2) \times \pi(X_2, X_3) \times \ldots \times \pi(X_{T-1}, X_T) = \prod_{a \in \mathcal{A}} \prod_{b \in \mathcal{A}} \pi(a,b)^{N(ab)}.$$

où $N(ab)$ le nombre de fois où le couple (ab) apparait dans la séquence.

L'estimateur qui maximise cette vraisemblance s'obtient par la résolution d'un prgramme de maximisation sous la contrainte de stochasticité de la matrice de transition de la chaine résultant en l'estimateur fréquentiste suivant :

$$\hat{\pi}(a,b) = \frac{N(ab)}{\sum_{b \in \mathcal{A}} N(ab)} = \frac{\text{Nombre de passages de a vers b}}{\text{Nombre total de passages de a vers une autre lettre}}.$$

Ainsi la matrice de transition se construit à partir de :

		Vers ...				
		a	c	g	t	Total
De ...	a	N(aa)	N(ab)	N(ag)	N(at)	N(a+)
	c	N(ca)	N(cc)	N(cg)	N(ct)	N(c+)
	g	N(ga)	N(gc)	N(gg)	N(gt)	N(g+)
	t	N(ta)	N(tc)	N(tg)	N(tt)	N(t+)

TABLE 1 – Construction de la matrice de transition de la chaîne de Markov pour une séquence donnée

La fonction `EstMatrice()` proposée dans l'implementation du caclul de la matrice de transition est basée sur l'utilisation de la fonction $f : \mathcal{A} \to \mathbb{N}$ suivante

$$f : \begin{cases} A & \longmapsto 1 \\ C & \longmapsto 2 \\ G & \longmapsto 3 \\ T & \longmapsto 4 \end{cases}$$

qui permet de réduire considérablement la compléxité algorithmique dans la tentative d'approcher la matrice de transition à travers la construction de la Table 1.
Nous détaillons ci dessous l'algorithme consistant à construire la matrice de transition à partir d'une séquence ADN.

Algorithm 1 Construction de la matrice de transition

Input: S : Séquence ADN , $|\mathcal{A}|$: Taille de l'espace d'état
Output: $\hat{\Pi}$: Estimation de la matrice de transition

$N \leftarrow$ Matrice de zéros de taille $|\mathcal{A}| \times |\mathcal{A}|$
for all $(S_i,\ S_{i+1}) \in S$ **do**
    $a, b \ \leftarrow \ f(S_i),\ f(S_{i+1})$ ▷ (ex : $a, b \leftarrow 1, 2$)
    $(N)_{a,b} \ \leftarrow \ (N)_{a,b} \ + \ 1$ ▷ (Construction incrémentale de la Table 1)
for all $a \in \mathcal{A}$ **do**
    $N(a+) \ \leftarrow \ \sum_{b\in\mathcal{A}}(N)_{a,b}$
    $\left(\hat{\Pi}\right)_{a,b} \ \leftarrow \ (N)_{a,b}/N(a+)$

return $\hat{\Pi}$

Il me semble naturel de penser que pour des séquences suffisament longues et donc en temp suffisament grand, la chaine "peut se passer" de sa distribution initiale sous résèrve de certaines hypothèses comme l'irreductibilité ou l'absence d'états absorbants.

Une chaine de Markov $\{X_n\}_{n\in\mathbb{N}}$ de loi initiale μ et de matrice de transition Πest dite stationnaire si la probabilité $\mu_j = \mathbb{P}(X_n = j)$ est indépendante de n pour tout $j \in \mathcal{A}$. Elle possède sous la contrainte $\sum_{j\in\mathcal{A}} \mu_j = 1$ la propriété suivante :

$$\mu\Pi = \mu.$$

En utilisant la fonction `Loi_Stationnaire()` proposée sur la région promotrice et le premier exon, nous trouvons

$$\hat{\Pi}_{prom} = \begin{bmatrix} 0.228 & 0.175 & 0.298 & 0.298 \\ 0.483 & 0.172 & 0.034 & 0.310 \\ 0.372 & 0.116 & 0.302 & 0.209 \\ 0.286 & 0.184 & 0.245 & 0.286 \end{bmatrix}.$$

Dont la loi stationnaire est :

$$\mu_{prom} = (0.105, 0.480, 0.230, 0.185).$$

C'est ce que nous avons trouvé dans la construction fréquentiste des nucléotides à la Figure 1.

Pour le premier exon, nous trouvons :

$$\hat{\Pi}_{exon} = \begin{bmatrix} 0.228 & 0.175 & 0.298 & 0.298 \\ 0.483 & 0.172 & 0.034 & 0.310 \\ 0.372 & 0.116 & 0.302 & 0.209 \\ 0.286 & 0.184 & 0.245 & 0.286 \end{bmatrix}$$

Dont la loi stationnaire est

$$\mu_{exon} = (0.3202247, 0.1629213, 0.2415730, 0.2752809).$$

Les lois stationnaires sont différentes, plus de C dans la région promotrice et plus de A dans le premier exon. Regardons les passages d'un nucléotide à un autre

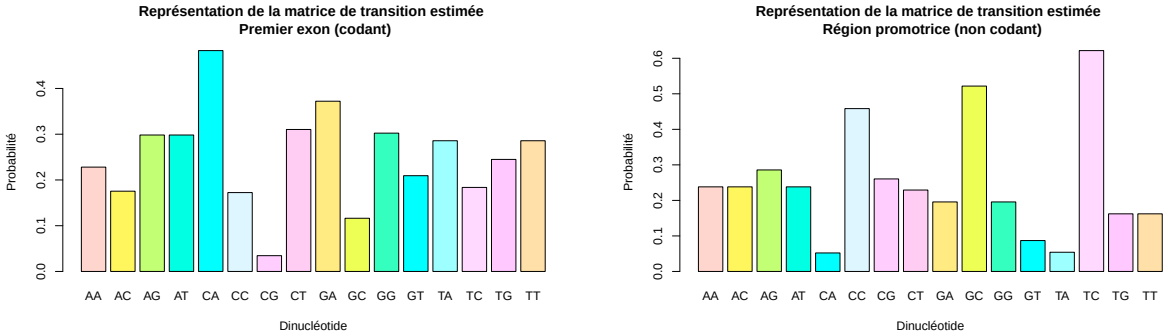


FIGURE 2 – Comparaison des probabilités de passage d'un nucléotide à un autre entre la région promotrice et le premier exon.

Commentaire

Au regard de la Figure 2, nous observons plus de passages vers C dans la région non promotrice à partir de C, G, T et quasiment pas de passsage de C vers A, ou de T vers A.

Tandis que pour le premier exon, nous observons plus d'activité autour du nucléotide A, comme si la chaine cherche à eviter les états G et C.

Partie 2 : Chaînes de Markov cachées

Nous avons pu voir à la partie 1 que les lois stationnaires diffèrent de manière significative pour les deux régions étudiées. Il se trouve qu'séquence d'ADN peut changer de "comportement" à certains endroits. On observe des parties de la séquence où il y a plus d'occurences des nucléotides C et G que de A et T, des fois l'inverse.
Nous pouvons supposer l'existence d'états qui soient résponsables de ces types d'occurences. En l'occurrence, deux états ont été identifiés : un état "codant" où les nucléotides A et T sont plus fréquents et un état non codant où les nucléotides C et G sont les plus fréquents.

Ce phénomène peut être modélisé par une chaine de Markov cachée. Les chaines de Markov cachées sont caractérisées par :

(I) **Une chaine de Markov non observable** $\{U_n\}_{n\in\mathbb{N}}$ décrivant les états codants et non codant de la séquence, prenant des valeurs dans un espace d'états **fini** à $J = 2$ états de matrice de transition A.

(II) **Un processus stochastique observable** $\{X_n\}_{n\in\mathbb{N}}$ d'espace d'états fini $\mathcal{A}$ à $K = 4$ états décrivant la séquence d'ADN.

Les processus $\{U_n\}_{n\in\mathbb{N}}$ et $\{X_n\}_{n\in\mathbb{N}}$ sont reliées par le conditionnement suivant appelé probabilité d'emission :

$$b_j(k) = \mathbb{P}(X_n = a_k | U_n = j).$$

(III) **Les observations X sont indépendantes sachant la chaine d'états** U :

$$\mathbb{P}(X_0 = a_0, \ldots, X_n = a_n | U_0 = j_0, \ldots, U_n = j_n, B) = \prod_{l=0}^{n} b_j(l).$$

Avec B la matrice de probabilités d'émissions formée par les $b_j(k)$ qui suivent une loi multinomiale de dimension 4.

Etant donné une suite d'observations, nous pouvons nous demander :

- Dans quel état caché se trouve la chaine ? c'est la probabilité de filtrage :

$$F^l(v) = \mathbb{P}(U_l = v | X_{1:l} = x_{1:l}).$$

- Dans quel état va se trouver la chaine au saut suivant, c'est la probabilité de prédiction :

$$P^{l+1}(v) = \mathbb{P}(U_{l+1} = v | X_{1:l} = x_{1:l}).$$

- Sachant toutes les observations, dans quel état se trouvait la chaine au temps l, c'est la probabilité de lissage :

$$L^l(v) = \mathbb{P}(U_l = v | X_{1:T} = x_{1:T}).$$

Pour $l = 1$,

$$P^1(v) = \mathbb{P}(U_1 = v)$$

est la loi initiale de U.

$$F^1(v) = \frac{f_v(X_1)P^1(v)}{\sum_u f_u(X_1)P^1(u)}.$$

Enfin

$$L^T(v) = F^T(v) \tag{1}$$

Question 9 :

Les valeurs calculées servent d'initialisations à l'algorithme Forward-Backward. Il calcule en premier lieu, de manière progressive les probabilités de prédiction et de filtrage grace aux équations de Viterbi.
A l'itération T, il utilise l'équation (1) pour calculer les probabilités de lissage de manière rétrograde. La fonction `ForBack()` proposée est une implémentation de cet algorithme.

Nous procédons dans la suite à une application.

Question 10 : Après concaténation des différentes régions, nous obtenons une séquence de taille 481. L'indice correspondant au passage de la région promotrice au premier exon est 202 et du premier exon à l'intron est 381.

Question 11 : On suppose qu'on dispose de la matrice de transition et lois d'émission suivantes :

$$\Pi = \begin{bmatrix} 0.65 & 0.35 \\ 0.55 & 0.45 \end{bmatrix}$$

$\mathrm{f}_0 = (0.4, 0.1, 0.1, 0.4)^t$ $\qquad$ $f_1 = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})^t$

.

En appliquant l'algorithme `ForBack()`, nous retrouvons :

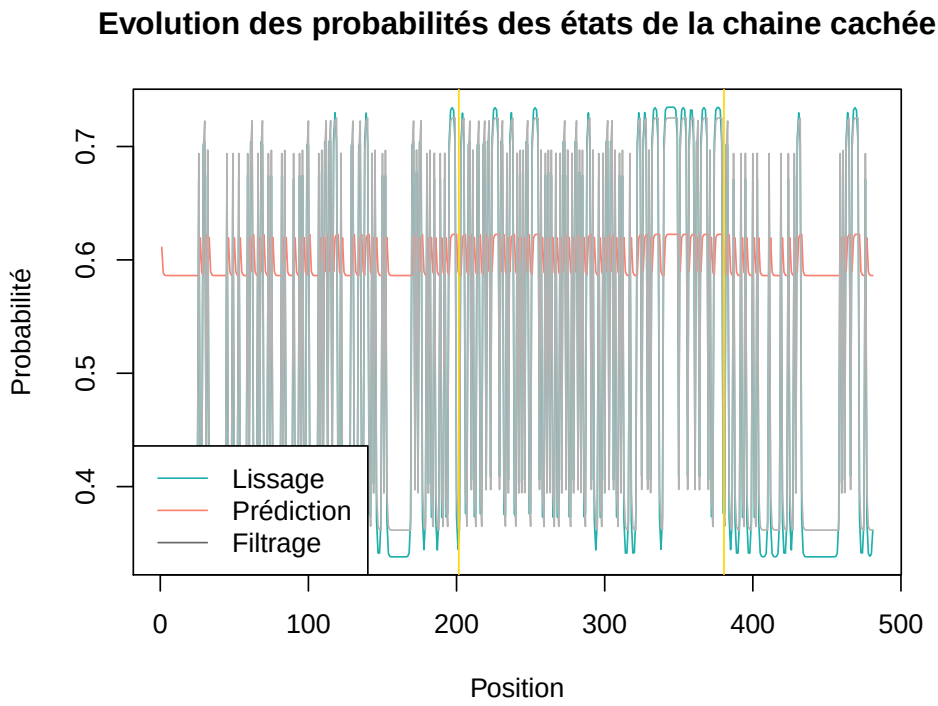


FIGURE 3 – Evolution des probabilités des états codants/non-codants de la chaine cachée.

Au regard de la Figure 3, la probabilité de lissage d'être dans l'état 0 qui varie entre 26% et 73%, mais le modèle prédit d'être dans l'état 0 avec une probabilité entre 37% et 62%. Ce modèle ne capture pas l'information qu'on connaît sur le gène, à savoir une transition de l'état non codant à l'état codant à l'instant 202, et un passage à l'état non codant à l'état 380.

Question 12 :

Dans la première partie, la séquence était coupée en deux régions non codantes et une région codante. Nous avons estimé les lois stationnaires de ces dernières. Nous les utilisons comme lois d'émissions des états cachés De plus, nous pouvons utiliser la Q10 pour construire un estimateur naïf de la matrice de transition en comptant le nombre de transitions d'un état à lui même et d'un état à un autre. Ainsi, nous trouvons

$$\hat{\Pi} = \begin{bmatrix} \frac{200+100}{301} & \frac{1}{301} \\ \frac{1}{179} & \frac{178}{179} \end{bmatrix}.$$

Cet estimateur a pu être construit car nous connaissons les endroits où le génome change d'état. En général ce n'est pas le cas, des méthodes de détections de points de rupture existent et sont utilisées.

Nous obtenons un bien meilleur modèle :

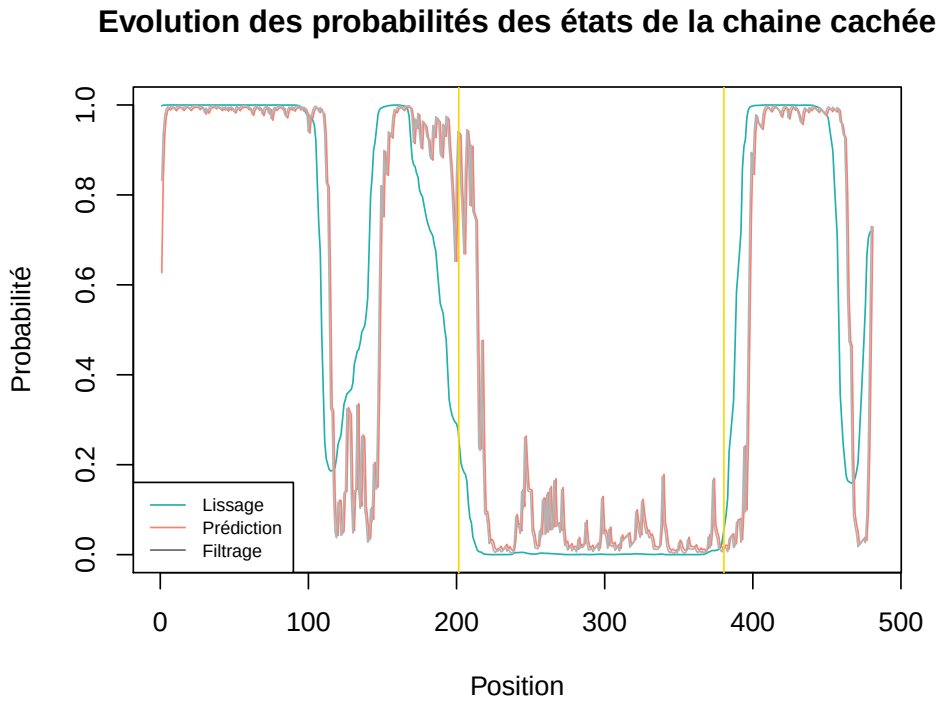


FIGURE 4 – Evolution des probabilités des états codants/non-codantes de la chaine cachée avec le nouveau modèle cité ci dessus.

Ces nouveaux paramètres permettent une meilleur modélisation des états cachés de la séquence, mais ce modèle semble sous estimer la probabilité d'être en région non codante quans la séquence dénombre plus de A et de T.
En revanche il semble bien avoir capturé les points de passage d'un état à un autre.

Paramètres du modèle : Baum-Welch

L'algorithme de Baum-Welch est un algorithme itératif de mise à jour des paramètres qui nécessite un critère d'arrêt.

Question 13 :

```
  NormTheta<-function(pi,f1,f2){
  d1<-sqrt(sum(diag(t(pi)%*%pi)))
  d2<-sum(abs(f1),abs(f2))/2
  return(d1+d2)
}
```

Question 14 :

L'algorithme de Baum-Welch est un algorithme itératif consistant à estimer les paramètres du modèle HMM quand les sont inconnus. Il fait intervenir l'algorithme Forwad-Backward, il a besoin des mêmes entrées. Ils vont servir de paramètre d'initialisation. Ces paramètres vont être mis à jour de manière itérative.
La vraisemblance globale du modèle augmente à chaque itération et donc l'algorithme converge vers un maximum local.

Les sorties de cet algorithme sont les paramètres de la chaine de Markov cachée dont la vraisemblance globale du modèle est maximum.

La fonction `Baum_Welch()` proposée implémente cet algorithme.

Question 15 : Application avec les paramètre précédents.
En appliquant l'algorithme avec comme paramètres initiaux, ceux proposés à la question 11 vs 5 et 12, nous retrouvons pour un critère d'arrêt $\epsilon = 10^{-5}$ des paramètres quasiment identiques.

La matrice de transition :

$$\hat{\Pi} = \begin{bmatrix} 0.99 & 0.01 \\ 0.02 & 0.98 \end{bmatrix}.$$

Lois d'émissions

$$f_0 = (0.298, 0.179, 0.274, 0.247) \qquad f_1 = (0.03, 0.56, 0.254, 0.154).$$

En relaxant le critère d'arrêt $\epsilon = 0.01$, nous avons observé des paramètres de sortie différents. Ceci est du au fait qu'il peut exister plusieurs maxima locaux qui vont dépendre

donc des paramètres initiaux.

Question 16 :

Voici l'implémentation de la vraisemblance

```
Vraisemblance<-function(ADN,pi,f1,f2)
{
  n <- length(ADN)
  NumADN<-as.numeric(sapply(ADN,LetterToNumber))
  mu <- Loi_stationnaire(pi)
  f <- matrix(c(f1, f2),  ncol=2 ,byrow=F)

  mat_vrais <- t(as.matrix(f[NumADN[1],] * mu))

  for(i in 2:n){
      mat_vrais <- rbind(mat_vrais,apply(t(as.matrix(mat_vrais[i-1,] * pi * f[NumADN[
  }

  res <- sum(mat_vrais[n,])
  return(res)
}
```

La vraisemblance dans cet exemple produit des valeurs très proche de 0. Nous utilisons plutôt la log vraisemblance qui va conserver les ordres du fait que le log est une fonction bijective croissante stricte.

Pour $\epsilon = 10^{-5}$, le rapport des log-vraisemblance des deux modèles est très proche de 1.

Pour $\epsilon = 10^{-2}$, le rapport des log-vraisemblances indique que le modèle construit à partir de nos estimations est meilleur.

Commentaire

Dans notre exemple, il semble que le control du critère d'arret résulte en une bonne estimation des paramètres de la chaine.

Question 17 :

Nous choisissons le modèle

$$\hat{\Pi} = \begin{bmatrix} 0.99 & 0.01 \\ 0.02 & 0.98 \end{bmatrix}.$$

avec

$$f_0 = (0.298, 0.179, 0.274, 0.247) \qquad f_1 = (0.03, 0.56, 0.254, 0.154).$$

Ce modèle est celui dans la vraisemblance est la plus grande. En faisant tourner l'algorithme FB, nous obtenons

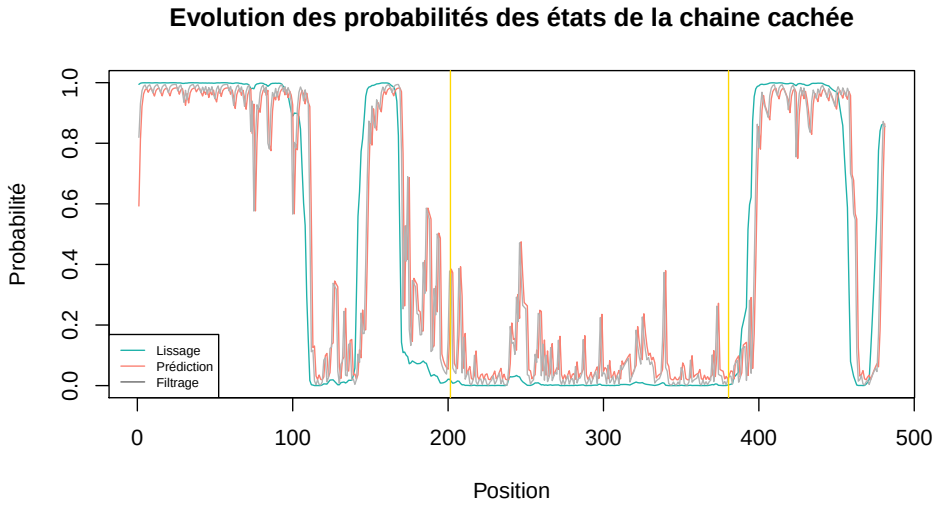


FIGURE 5 – Evolution des probabilités des probabilités de lissage, filtrage et prédiction pour le modèle retenu.

Nous obtenons un résultat similaire à celui de la Figure 4 à la partie 2. Une interprétation naive est qu'il y aurait une phase de transition de l'état non codant à l'état codant où l'on observe l'occurrence d'avantage de nucléotides G et C (position 110 à 147) un peu comme si la séquence se préparait à changer d'état.

References

Statistical Methods in Bioinformatics An Introduction - Warren J. Ewens, Gregory R. Grant

Biological Sequence Analysis Probabilistic Models of Proteins and Nucleic Acids - Richard Durbin, Sean R. Eddy, Anders Krogh, Graeme Mitchison

Hidden Markov models for bioinformatics - Timo Koski