Régression logistiquemultiple et prédiction des facteurs de risque concernant la survie en soins intensifs.

changer le titre

Julia Guerra [1], Maxime Jaunatre [2], Ellie Tideswell [3] | Master 2 BEE Grenoble
Mail [1], Mail [2] Mail [3]| 30 novembre 2019

Todo list

L'avancée des techniques de séquençage a permis d'obtenir de grandes quantités d'informations génomiques. Durant ces dernières années, la génétique a embrassé les outils mathématiques de modélisation, parvenant à une caractérisation statistique des données de séquençage. Cette approche a permis de décrire avec précision les motifs observés dans des régions étudiées , et de prédire leurs présences dans des séquences encore inconnues (Wu *et al.* , 2010). Une des méthodes les plus connues dans ce domaine est l'utilisation des chaînes de Markov, introduites premièrement par Churchill (1992) pour l'analyse de séquences génomiques puis par Durbin *et al.* (1998) pour la détection de régions CGI.

Dans le génome des deutérostomiens, la fréquence du dinucléotide C-G est moins importante qu'attendu sous une distribution aléatoire indépendante des quatre bases azotées. Ceci est une conséquence des mécanismes de protection contre la mutation spontanée du génome. Cependant, dans certaines régions de l'ADN nommées îlots CpG (ou CGI) ce processus de mutation est évolutivement reprimé et donc la fréquence des dinucléotides C-G est donc plus élevée ; par exemple aux alentours de certains promoteurs (Haque *et al.* , 2011, Saxonov *et al.* , 2006, Wu *et al.* , 2010).

La grande variabilité dans la taille, la composition et l'emplacement de ces CGI rend difficile leurs définitions et donc l'établissement d'un algorithme unique permettant leurs détections indubitable (Wu *et al.* , 2010). Ainsi, les modèles de Markov permettent de modéliser les fréquences des nucléotides en fonction de séquences déjà connues ; ces séquences contenant ou non des CGI. En supplément des chaînes de Markov simples, il existe aussi les chaînes de Markov cachées (HMM, Churchill (1992)) : ces dernières décrivent de nombreux processus réels qui suivent un modèle de Markov, mais qui ne sont pas observables. Une chaîne de Markov cachée permettrait donc l'utilisation d'un seul modèle pour identifier un nucléotide (l'observation) et si ce dernier est à l'intérieur d'un îlot CpG ou non (aussi appelé l'état de la région). Les HMM permettent ainsi d'augmenter la résolution de l'analyse, c'est-à-dire de détecter l'emplacement des régions CGI à l'intérieur des séquences.

Matériel et méthodes

0.1 Modèles de Markov simples

Les modèles de Markov réalisés dans cette étude ont été construits à partir de deux jeux de séquences de souris (Mus musculus). Ces jeux de séquences avaient été caractérisés en avance comme contenant des îlots CpG (on notera "CpG+"), ou ne contenant pas d'îlots CpG ("CpG-"). Les deux jeux de séquences "app", pour la construction des modèles CpG+ et CpG- ,contenaient 1160 et 5755 séquences respectivement. Des jeux supplémentaires "test" également caractérisés comme CpG+ ou CpG- (1163 et 5137 séquences) ont servi à évaluer la performance des modèles.

Dans un premier temps, les fréquences relatives d'observation des bases A, C, G, T ont été calculées pour la totalité des séquences de chaque jeu de données "app" (R, fonction `count` du package `seqinr` ; Charif & Lobry (2007)). Ces données ont permis de construire la matrices de probabilité A du modèle d'ordre 0 (M0). Le terme "ordre" fait référence au nombre de bases précédentes conditionnant la probabilité de présence de la base étudiée. De cette manière, le M0 considère la probabilité d'occurrence de chaque base comme une variable aléatoire (équation 1) dont les probabilités d'occurrence (équation 2)) sont différentes. En plus, des résultats différents sont attendus en fonction de la nature CpG+ ou CpG- des séquences ; c'est pourquoi deux matrices de probabilité A+ et A- ont été construites, provenant respectivements des comptages du jeu CpG+ et CpG-.

Le modèle de Markov d'ordre 1 (M1) rassemble les occurrences de chaque base en fonction de la base précédente. Les matrices de transition q1+ et q1- sont matrices 4x4 qui ont été donc construites à partir des comptages de chaque couple de bases. Vu qu'elle ne peut pas dépendre d'une base précédente, la probabilité d'occurrence de chaque base initiale a été considérée comme une variable aléatoire (équation 3) à probabilités équivalentes (équation 4). Ce protocole de construction de modèle a été refait pour l'ordre 2, obtenant une matrice 16x4. Pareil pour l'ordre 3 (matrice 64 x 4), l'ordre 4 . . .jusqu'à l'ordre 5. Pour les modèles d'ordre supérieur 0, les lignes de la matrice ont été rangées de sorte que la somme de chaque ligne soit égal à 1, à cause de la nature conditionnelle des probabilités, comme dans l'exemple suivant (table 1.

$$Y \in B; B = a, c, g, t \tag{1}$$

$$P(Y_i = k) \forall k \in B \tag{2}$$

$$X \in B; B = a, c, g, t \tag{3}$$

$$P(A) = P(C) = P(G) = P(T) = P(X \in B) = \frac{1}{4} \tag{4}$$

A partir des matrices de transition, on peut calculer la log-Vraisemblance d'une séquence sous un modèle MX correspondant comme la somme du log de la probabilité de premières bases (région de taille égale à l'ordre) avec la somme du produit de la matrice de transition par la matrice d'occurence des mots dans la séquence (voir équation 5).

	a	c	g	t
a	0.29	0.21	0.30	0.20
c	0.26	0.30	0.17	0.27
g	0.24	0.27	0.30	0.20
t	0.18	0.26	0.28	0.29

TABLE 1 – Matrice de transition du modèle CpG+ d'ordre 1

$$P_+(Sequence) = log\,[P_{initial}(mot_{initial})] + \sum_{i=1}^{n=4^{ordre+1}} log\,[P_i(mot_i) \cdot N_i(mot_i)] \tag{5}$$

Choix du meilleur modele

La performance du M1 a été testée sur les deux jeux de séquences de test. La log-vraisemblance de chaque séquence a été calculé pour chaque modèle (CpG+ et CpG-) et la séquence est donc associée à l'état pour lequel la log-vraisemblance est la plus grande. Pour le jeu de données CpG+, les séquences caractérisées comme CpG+ sont considérées comme vrais positifs (VP) et les séquences caractérisées CpG- comme faux négatifs (FN). Pour le jeu de données CpG-, les séquences caractérisées comme CpG+ sont considérées comme faux positifs (FP) et celles caractérisées comme CpG- comme vrais négatifs (VN). Le même protocole a été suivi pour

tester la performance des modèles 1 à 6.

La spécificité et la sensibilité de chaque modèle ont été calculées à partir de ces résultats, selon les équations illustrées en 6 et 7.

$$Sensibility = \frac{VP}{VP+FN} \tag{6}$$

$$Specificity = \frac{VN}{VN+FP} \tag{7}$$

Ce processus, répété pour toutes les combinaisons de Mi+/Mj- (avec i et j allant de 0 à 5), a permis de connaître la meilleure combinaison de modèle. Pour l'obtenir, les données de sensibilité et spécificité pour les modèles ont été sommés entre elles. La combinaison d'ordres portant la valeur maximale étant la valeur (5,4) de la matrice ; les calculs de la chaîne de Markov cachée ont été réalisés sur un modèle d'ordre 5 pour les séquences CpG+ et un modèle d'ordre 4 pour les séquences CpG-. La figure 1 montre les résultats de sensibilité et spécificité mentionnées ici.

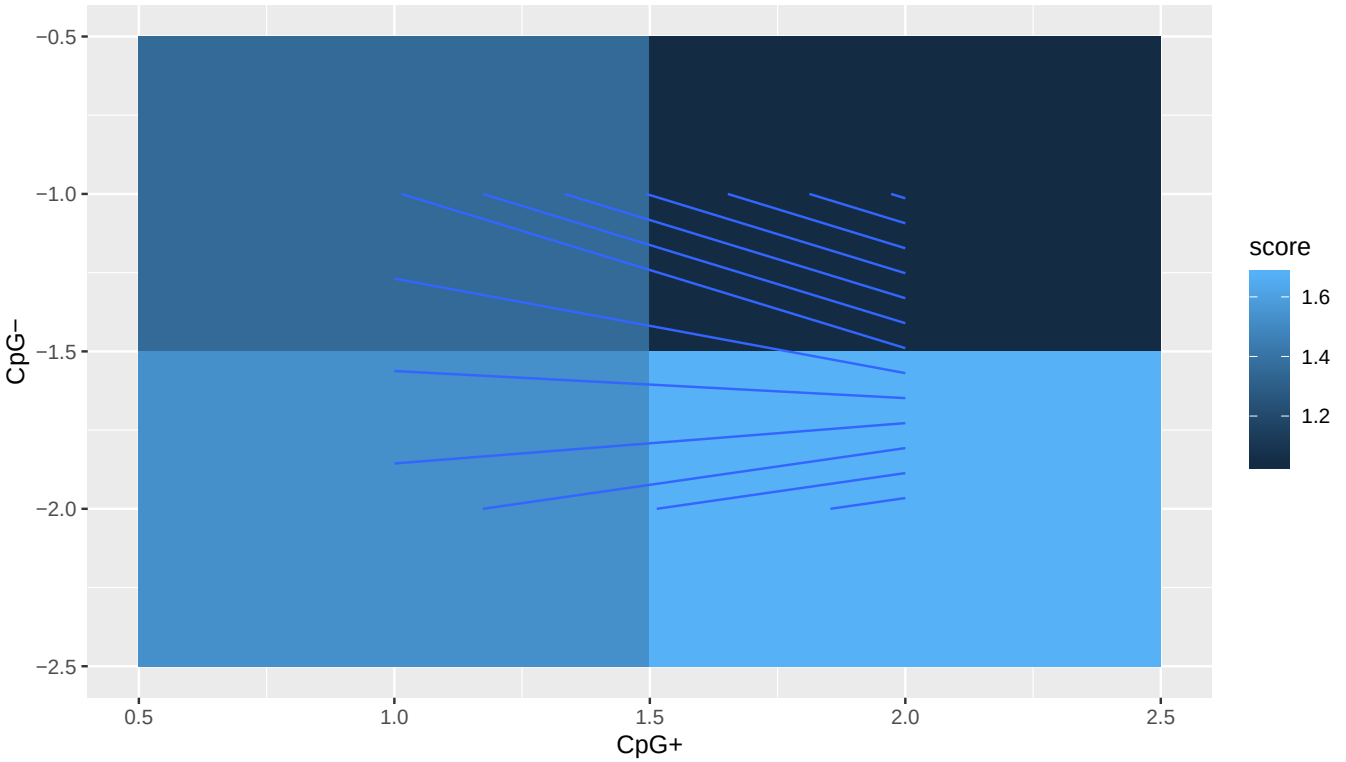

FIGURE 1 – Evolution du score de sensibilité + spécificité selon les ordre de modèles

Modèles de Markov cachés Les bases mathématiques de l'analyse des îlots CGI parmi des HMM dans cet étude suit le protocole décrit dans Churchill (1992). Pour la construction des HMM, il a été nécessaire de calculer la probabilité de transition entre état CpG+ et état CpG- à l'intérieur d'une séquence. Les valeurs de cette matrice de transition 2x2 contenant d'états ont été obtenues à partir de la bibliographie, en prenant compte de la longueur moyenne des îlots CpG. Cette matrice (2) s'ajoute donc à la matrice des probabilités d'occurrence de chaque base (ou combinaison de bases) et à la matrice d'occurrence des bases initiales.

0.2 L'algorithme de Viterbi

Afin de trouver la séquence optimale d'états qui correspond à une séquence donnée d'observations, il est possible d'utiliser une fenêtre glissante (un algorithme naïf), dans laquelle les log vraisemblances sont calculés pour des segments de bases

	M+	M-
M+	-0.00	-6.91
M-	-11.74	-0.00

TABLE 2 – Matrice de transition du modèle CpG+ d'ordre 1

d'une longueur donnée. Bien que facile à implémenter, les résultats (en terme des prédictions des CGI) dépendent de la taille de la fenêtre choisi, ceci peut représenter un biais de cette méthode. Une façon alternative peut être l'algorithme de Viterbi, un exemple de la programmation dynamique, qui permet d'identifier la séquence qui maximise la probabilité de générer les observations

Pardoux

. Le chemin le plus probable étant donné un modèle est déterminé via une procédure récursive. L'algorithme de Viterbi est décrit comme suit :

0.3 Smoothing

La technique de "Smoothing" représente une technique mathématique qui enlève la variabilité parmi les données, impliquant souvent la redistribution du poids entre des régions de haute probabilité, et des régions de "zéro probabilité"

Boodidi 2007

. Dans le cadre de cette étude, le "Smoothing" revient donc à lisser la caractérisation des différentes régions en les ré-assignant selon 2 procédés successifs.

Dans un premier temps, les régions de longueur inférieur à un certain seuil (S) sont assignée à une nouvelle catégorie "Ambiguous", en vert dans la figure 2. Cette première étape comporte également un algorithme qui compile ces nouvelles régions en une seule quand elles se suivent dans la séquence (voir bases 9 à 13), afin de mesurer la longueur de cette nouvelle région dont la catégorie est devenue unique. Le second procédé vérifie la longueur de ces nouvelles régions ambiguës et leurs situations sur la séquence. En effet, il arrive qu'une région de petite taille soit considérée comme ambiguë entre deux régions d'une même catégorie (voir base 5). On peut donc supposer qu'il s'agit de bruit et que cette région est probablement de la même catégorie que celles qui l'entoure. Ainsi, le second procédé de smoothing va ré-assigner des régions ambiguës si leurs tailles sont inférieures à un seuil et que les régions bordantes sont de même nature. On note que l'algorithme de 'Smoothing' ne peut être utilisé qu'avec le second procédé, car en l'absence de régions ambiguës aucune ré-assignation vers CpG+ ou CpG- n'est possible.

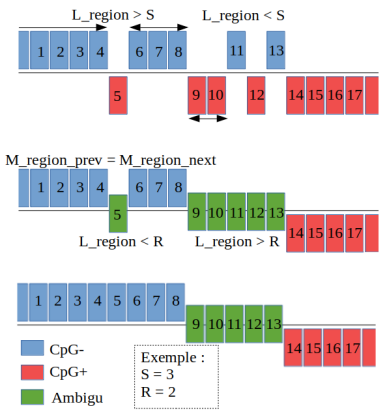

FIGURE 2 – Evolution du score de sensibilité + spécificité selon les ordre de modèles

Résultats

mus1

table tronquée figure tronquée description du chromosome (nombre d'îlots cpg, taille des cpg, fenetre de smooth)

mus2

description

mus3

description

Discussion

nos resultats sont bien badass!!!!!

L'apprentissage de nos modèles est ici relativement rapide car le jeu de donnée d'entraînement est limité et nos modèles encore simplifiés. Cependant, il est important de noter que des modèles plus complexes entraînent des temps de calculs d'autant plus important. Dans ce contexte, il est donc important de souligner que tout n'a pas été fait pour optimiser l'apprentissage de nos modèles. Il reste encore possible de paralléliser différentes étapes de l'apprentissages ou du test des modèles différents. Une solution encore plus avancée serait de changer de langage de programmation afin de produire un algorithme plus efficace que celui proposé ici sour `R`.

Afin d'améliorer la performance de notre modèle, il serait raisonnable de raffiner la sélection des CGI identifiés, selon des propriétés connus des CGIs, tels que le contenu de GC, la fraction de CpG et un seuil de longueur

ref to do(Lan et al 2009)

. De nombreux études ont relevé la fausse identification dans les CGI de petites quantités des nucléotides CpG+, identifiées comme CpG-. Un seuil minimal de longueur entre des nucléotides CpG+ en voisinage pourrait résoudre ce problème, donc un 'smoothing' plus dynamique ou les paramètres changent en fonction de l'état dans lequel la nucléotide se situe (CpG+ ou CpG-). Les CGI contiennent également une ratio élevé de G/C, ce qui est normalement de 60% au minimum. L'application d'un tel seuil aux CGI identifiés pourrait représenter une amélioration supplémentaire du modèle. Les CGI sont souvent définis comme des régions d'un longueur de 140 paires de base, cependant certains auteurs indiquent qu'il existe une certaine variabilité qui rend cette définition éronée. Un seuil minimum de longueur a donc été suggéré par certains auteurs et il pourrait s'avérer intéressant de comparer les prédictions sans et avec son application, à faire avec caution

ref to do(Lan et al 2009)

.

Une des limites des modèles de Markov cachés en général est la supposition que les distributions des paramètres d'observation suivent une loi géométrique.

ref to doBerg (2013)

a identifié d'autres limitations de l'utilisation des modèles de Markov cachés, dans un context des prédictions des CGI. Parmi ces limitations se trouve le constat que les résultats d'un tel modèle dépendent fortement des probabilités initiales, ainsi que des itérations d'entraînement du modèle. Berg a donc suggéré l'entraînement du modèle, et la ré-estimation subséquente des probabilités initiales afin de mieux représenter les états cachés, et ceci pourrait également représenter une amélioration possible de notre modèle.

Le détection précise des îlots CPG reste un sujet important dans un contexte médical, avec, parmi d'autres, de plus en plus d'associations identifiées entre le méthylation modifié et le cancer. Les régions se situant à moins de 20000 paires de base des frontières CGI, pour exemple, ont été identifiés comme des bons prédicteurs pour la location des régions qui subissent un méthylation modifié, spécifique aux cancers ("cancer-specific differentially methylated regions")

ref to do(Irizarry et al 2009)

. L'amélioration des modèles qui nous permettent donc de prédire avec précision ces régions représentent un défi important dans la détection des cancers.

Ressources

Ce document est disponible en ligne sous format ".Rnw", contenant tout le code néccessaire à la reproduction de l'analyse, réalisée avec un script en langage R (R.Team, 2017), ainsi que le jeu de données de départ. L'ensemble est situé sur Github : `https://github.com/gowachin/BeeMarkov` et peut être installé sur R via les commandes suivantes.

```
> # NOT RUN
> library(devtools)
> install_github("gowachin/BeeMarkov")
> library(BeeMarkov)
```

Bibliographie

Charif, Delphine, & Lobry, Jean R. 2007. SeqinR 1.0-2 : A Contributed Package to the R Project for Statistical Computing Devoted to Biological Sequences Retrieval and Analysis.

Churchill, Gary A. 1992. Hidden Markov chains and the analysis of genome structure. *Computers and Chemistry*, **16**(2), 107–115.

Durbin, Richard, Eddy, Sean R., Krogh, Anders, & Mitchison, Graeme. 1998. Biological sequence analysis. *Biological sequence analysis*.

Haque, A. N.A., Hossain, M. E., Haque, M. E., Hasan, M. M., Malek, M. A., Rafii, M. Y., & Shamsuzzaman, S. M. 2011. CpG islands and the regulation of

transcription. *GENES & DEVELOPMENT*, **25**(1), 1010–1022.

R.Team. 2017. R : A language and environment for statistical computing (Version 3.4. 2)[Computer software]. *Vienna, Austria : R Foundation for Statistical Computing*.

Saxonov, Serge, Berg, Paul, & Brutlag, Douglas L. 2006. A genome-wide analysis of CpG dinucleotides in the human genome distinguishes two distinct classes of promoters. *Proceedings of the National Academy of Sciences of the United States of America*, **103**(5), 1412–1417.

Wu, Hao, Caffo, Brian, Jaffee, Harris A., Irizarry, Rafael A., & Feinberg, Andrew P. 2010. Redefining CpG islands using hidden Markov models. *Biostatistics*, **11**(3), 499–514.

Annexes

mus1

table figure

mus2

table figure

mus3

table figure

scripts

Les analyses ont été effectuées avec le code ci-dessous, sous Rstudio (Version 1.1.456) :

put script inside

```
library(BeeMarkov)
mus1 <- viterbi(file = "raw_data/mus1.fa",
  l_word_pos = 5,
  l_word_neg = 4
)
```