Chaînes de Markov et génome

Etienne Pardoux

Leçon 3

10 Chaînes de Markov non homogènes

Plusieurs types de non homogénéités sont pertinents dans la modélisation de l'ADN.

10.1 Chaînes de Markov phasées

Dans une "plage codante", on peut penser que la probabilité de transition n'est pas indépendante du site, mais périodique de période 3. Comme la notion de "chaîne de Markov périodique" désigne tout autre chose (à savoir une chaîne qui n'est pas "apériodique" au sens de la définition 5.3), nous utiliserons, à la suite du livre de Robin, Rodolphe, Schbath, la terminologie "chaîne Markov phasée" pour désigner une chaîne de Markov $(X_n,\, 1 \leq n \leq N)$ telle que pour tous $x, y \in E$, l'application $n \to P(X_{n+1} = y | X_n = x)$ est périodique. Dans le cas qui nous occupe, on peut y compris songer à une chaîne de Markov d'ordre 2, telle que pour tout $y \in E$, la quantité $P(X_{n+1} = y | X_n = x, X_{n-1} = x')$ ne dépende pas de x, x' pour $n = 3k$, que de x pour $n = 3k+1$ et dépende de x, x' pour $n = 3k+2$, k entier. Cela veut dire en particulier que les codons successifs sont i.i.d. On pourraît aussi supposer que les codons successifs forment une chaîne de Markov d'ordre 1.

10.2 Chaînes de Markov localement homogènes

Si l'on regarde plus globalement la séquence génomique, on s'attend à ce que la chaîne de Markov qui décrit celle–ci soit homogène dans la réunion des régions non codantes, dans celle des régions codantes à l'endroit, celle des régions non codantes, la réunion des introns et celle des exons, mais pas globalement homogène, et c'est d'ailleurs cette inhomogénéité qui doit nous permettre de réaliser l'annotation. Le principal problène est bien de détecter ce que l'on appelle les "ruptures de modèle".

Il existe une importante littérature statistique sur ces problèmes de rupture de modèle, mais il n'est pas clair que les algorithmes correspondant sont adaptables à la situation qui est la nôtre ici, où il est essentiel d'exploiter l'homogénéité de la chaîne sur la réunion des plages de même type (non codant, codant à l'endroit,...), et pas seulement sur chacune de ces plages prise isolément.

Cependant, Audic et Claverie ont mis au point un algorithme pour l'annotation des génomes prokaryotes, que nous allons maintenant décrire. On suppose que notre modèle (qui peut être $M0$, $M1$, $M2$,...) est décrit par un paramètre θ (qui est une probabilité sur E dans le cas $M0$, une probabilité de transition dans le cas $M1$, ...), lequel prend trois valeurs distinctes (toutes trois inconnues !) $(\theta_0, \theta_1, \theta_2)$, suivant que l'on est dans une plage non codante, codante à l'endroit ou codante à l'envers.

- •*Étape d'initialisation* On découpe la séquence en plages de longueur 100 (éventuellement, la dernière plage est de longueur > 100). On décide au hasard de placer chaque plage dans l'une des trois "boîtes" 0, 1, et 2. Sur la base de tous les X_n se trouvant dans la boîte 0, on estime une valeur du paramètre θ, soit $\theta_0^{(1)}$. On estime de même les valeurs $\theta_1^{(1)}$ et $\theta_2^{(1)}$.
- •*Étape de mise à jour* Supposons que nos trois "boîtes" 0, 1, et 2 contiennent chacune des plages distinctes de longueur ≥ 100, sur la base desquelles on a estimé les valeurs $\theta_0^{(n)}$, $\theta_1^{(n)}$ et $\theta_2^{(n)}$. On commence par vider ces boîtes, et on reprend la séquence complète $\{X_n,\ 1 \leq n \leq N\}$. On extrait la sous–suite $\{X_n,\ 1 \leq n \leq 100\}$. On estime le paramètre θ sur la base de cette sous–suite, et on choisit laquelle des trois valeurs $\theta_0^{(n)}$, $\theta_1^{(n)}$ et $\theta_2^{(n)}$ est la plus proche de cette nouvelle valeur estimée. Puis on se pose le même problème avec la suite $\{X_n,\ 10 \leq n \leq 110\}$, avec la suite $\{X_n,\ 20 \leq n \leq 120\}$,.. jusqu'à ce que la valeur estimée devienne plus proche d'une autre des trois valeurs de l'étape précédente. Alors on revient en arrière de 50 nucléotides, et on place l'intervalle ainsi sélectionné depuis le début de la séquence dans la boîte 0, 1, ou 2, suivant le cas. On recommence, en prenant une plage de longueur 100, adjacente à l'intervalle que l'on vient de placer dans une des boîtes, et on répète les opérations précédentes. Lorsque l'on a épuisé la séquence, on se retrouve avec trois boîtes contenant chacune (du moins il faut l'espérer) des plages de longueur ≥ 100. On estime alors les trois nouvelles valeurs $\theta_0^{(n+1)}$, $\theta_1^{(n+1)}$ et $\theta_2^{(n+1)}$, sur la base du contenu des boîtes 0, 1, et 2 respectivement.

Si la séquence initiale est effectivement constituée de plages dont les compositions statistiques sont de trois types différents, l'algorithme converge rapidement, et quand on s'arrête, on a un découpage de la séquence initiale en sous–séquences de trois types différents. Il ne reste plus qu'à décider "qui est qui", ce qui requiert des connaissances a priori, acquises en observant des séquences qui ont déjà été annotées.

11 Chaînes de Markov cachées

Le point de vue Bayésien consiste à se donner une loi de probabilité a priori sur les paramètres inconnu θ_i, et leur évolution. Plus précisément on va maintenant se donner une

nouvelle chaîne de Markov $(Y_1, \ldots, Y_N)$, dite "cachée" parce que non observée. Dans le cas des génomes prokaryotes, la chaîne (Y_n) prend par exemple ses valeurs dans l'ensemble à trois éléments $F = \{0, 1, 2\}$, et dans le cas eukaryote il faut différencier les états 1 et 2 entre les parties *intron* et *exon*. En réalité c'est encore un peu plus compliqué, car il faudrait prendre en compte les codons START et STOP, mais on verra cela un peu plus loin. L'avantage de cette approche est que l'on dispose d'algorithmes pour répondre aux questions que nous nous posons. On note F l'espace dans lequel la chaîne cachée prend ses valeurs, et $d = \text{card}(F)$. Rappelons que $d \geq 3$.

Pour simplifier la présentation succincte de ces algorithmes, on va supposer que $(Y_1, \ldots, Y_N)$ est une chaîne de Markov (μ, P) à valeurs dans F, et que, connaissant les (Y_n), la suite des nucléotides $(X_1, \ldots, X_N)$ est indépendante, la loi de chaque X_n dépendant uniquement du Y_n correspondant, i.e. pour tout $1 \leq n \leq N$,

$$\begin{aligned}\mathbb{P}(X_1 = x_1, \ldots, X_n = x_n | Y_1 = y_1, \ldots, Y_n = y_n) &= \prod_{k=1}^{n} \mathbb{P}(X_k = x_k | Y_k = y_k) \\ &= \prod_{k=1}^{n} Q_{y_k x_k}.\end{aligned}$$

Le problème que l'on cherche à résoudre est le suivant : ayant observé la suite des nucléotides $(x_1, \ldots, x_N)$, quelle est la suite des états cachés $(y_1^*, \ldots, y_N^*)$ qui "explique le mieux" ces observations ? Autrement dit, il s'agit de calculer la suite qui maximise la vraisemblance de la loi a posteriori sachant les observations, i.e.

$$(y_1^*, \ldots, y_n^*) = \text{argmax}_{y_1, \ldots, y_n} \mathbb{P}(Y_1 = y_1, \ldots, Y_n = y | X_1 = x_1, \ldots, X_n = x_n).$$

Notons que, dans ce modèle, on a comme paramètres inconnus le triplet (μ, P, Q). Pour résoudre le problème ci–dessus, on est obligé d'estimer d'abord les paramètres (mais nous discuterons ce problème à la fin). Si l'on admet que l'on connaît les paramètres, notre problème est résolu par :

11.1 L'algorithme de Viterbi

Définissons la suite de vecteurs ligne $\delta(n)$ par :

$$\delta_y(n) = \max_{y_1, y_2, \ldots, y_{n-1}} \mathbb{P}_\theta(Y_1 = y_1, \ldots, Y_{n-1} = y_{n-1}, Y_n = y, X_1 = x_1, \ldots, X_n = x_n)$$

$\delta_y(n)$ est en quelque sorte la plus forte probabilité d'une trajectoire des $\{Y_k, 1 \leq k \leq n-1\}$, qui se termine par $Y_n = y$, et correspondant à la suite des nucléotides observés $x_1, \ldots, x_n$. On a la formule de récurrence suivante entre les vecteurs $\delta(n)$:

$$\delta_y(n+1) = (\delta(n) * P)_y Q_{y x_{n+1}}$$

où l'opération $*$ qui à un vecteur ligne de dimension d et une matrice $d \times d$ associe un vecteur ligne de dimension d est définie comme suit :

$$(\delta * P)_y = \sup_{z \in F} \delta_z P_{zy}.$$

L'algorithme de Viterbi consiste à calculer les $\delta(n)$ de $n = 0$ à $n = N$, puis à retrouver la trajectoire optimale en cheminant pas à pas dans le sens "rétrograde" : connaissant y_n^*, on en déduit y_{n-1}^* par la formule :

$$y_{n-1}^* = \psi_{y_n^*}(n),$$

avec

$$\psi_y(n) = \text{argmax}_{z \in F} \delta_z(n-1) P_{zy}.$$

L'algorithme de Viterbi est décrit comme suit :

1. Initialisation :

$$\begin{aligned} \delta_y(1) &= \mu_y Q_{yx_1}, \ \ y \in F; \\ \psi(1) &= 0. \end{aligned}$$

2. Récurrence : pour $1 < n \leq N$,

$$\begin{aligned} \delta_y(n) &= (\delta(n-1) * P)_y Q_{yx_n}, \\ \psi_y(n) &= \text{argmax}_{z \in F} \delta_z(n-1) P_{zy}, \ \ y \in F. \end{aligned}$$

3. Etape finale :

$$\begin{aligned} \delta^* &= \max_{y \in F} \delta_y(N) \\ y_N^* &= \arg\max_{y \in F} \delta_y(N). \end{aligned}$$

4. Récurrence rétrograde

$$y_n^* = \psi_{y_{n+1}^*}(n+1), \ \ 0 \leq n < N.$$

11.2 Estimation des paramètres

Il y a deux stratégies possibles. L'une consiste à estimer les paramètres sur une séquence d'apprentissage déjà annotée. Dans ce cas, on estime les paramètres d'un modèle où toute la suite $\{(X_n, Y_n),\ 1 \leq n \leq N\}$ est observée. On utilise les algorithmes d'estimation bien connus que nous avons présentés dans les sections précédentes.

L'autre stratégie consiste à estimer les paramètres sur la base des seules observations de la suite des nucléotides. L'avantage est de faire l'estimation à partir du génome étudié, et non pas à partir d'un génome différent. L'inconvénient est bien sûr que l'on estime un modèle avec des observations très partielles. Il existe cependant des algorithmes maintenant classiques (l'algorithme EM, et sa variante SEM), qui permettent de résoudre ce problème.