

Modélisation probabiliste en biologie

Chaînes de Markov et chaînes de Markov cachées

Christelle Gonindard

Christelle.gonindard@univ-grenoble-alpes.fr

4- Chaîne de Markov cachées

Un modèle de CMC permet de modéliser une séquence par un ensemble fini de modèles qui s'alternent le long de la séquence



4- Chaîne de Markov cachées

Un modèle de CMC permet de modéliser une séquence par un ensemble fini de modèles qui s'alternent le long de la séquence



Il y a donc deux processus sous-jacents :

- Le processus non observable (caché) $S_1 S_2 S_3 \dots S_n$ qui modélisera la suite des états le long de la séquence.

Ce processus est une chaîne de Markov d'ordre 1.

⇒ “Chaîne de Markov Cachée”.

4- Chaîne de Markov cachées

Un modèle de CMC permet de modéliser une séquence par un ensemble fini de modèles qui s'alternent le long de la séquence



Il y a donc deux processus sous-jacents :

- Le processus non observable (caché) $S_1S_2S_3 \dots S_n$ qui modélisera la suite des états le long de la séquence.

Ce processus est une chaîne de Markov d'ordre 1.

⇒ “Chaîne de Markov Cachée”.

- Le processus observable $X_1X_2X_3 \dots X_n$ qui modélisera la succession des lettres.

Le modèle pour générer X_i dépend de l'état S_i .

4- Chaîne de Markov cachées

Dans le schéma ci-dessous, on peut distinguer trois états (**rouge**, **vert**, **bleu**) : les premières lettres suivent le modèle rouge, etc. les dernières le modèle vert.

X	at	tagg	cagatac	ga	gg	t gattact	cgct	tagtct
S								

4- Chaîne de Markov cachées

Dans le schéma ci-dessous, on peut distinguer trois états (**rouge**, **vert**, **bleu**) : les premières lettres suivent le modèle rouge, etc. les dernières le modèle vert.

X	at	tagg	cagata	c g	gt	gattac	tgc	ctag	tct
S									

Les régions rouges, vertes et bleues sont caractérisées par des lois d'apparition des bases différentes (par ex. les régions rouges sont riches en **g**, etc.).

L'alternance des couleurs (états) est régie par une chaîne de Markov d'ordre 1.

4- Chaîne de Markov cachées

Processus caché

Une chaîne de Markov est une suite de variables aléatoires **dépendantes**

$$S_1 S_2 S_3 \cdots S_n \cdots$$

Ici S_i peut prendre un nombre fini de valeurs \mathcal{S} (par ex. $\{\textcolor{red}{r}, \textcolor{green}{v}, \textcolor{blue}{b}\}$).

Une dépendance d'**ordre 1** signifie :

$$\mathbb{P}(S_i = b \mid S_1, S_2, \dots, S_{i-1}) = \mathbb{P}(S_i = b \mid S_{i-1}) ;$$

la valeur de S_{i-1} suffit pour connaître avec quelle probabilité S_i prend la valeur b .

4- Chaîne de Markov cachées

Processus caché

Les S_i sont donc générées successivement selon les probabilités de transition :

$$\pi(u, v) = \mathbb{P}(S_i = v \mid S_{i-1} = u) ;$$

celles-ci sont rangées dans une matrice de transition Π .

Par exemple, $\mathcal{S} = \{\text{r, v, b}\}$ et

$$\Pi = \begin{pmatrix} \textcolor{red}{0.6} & 0.4 & 0 \\ 0.5 & \textcolor{green}{0} & 0.5 \\ 0.3 & 0.5 & \textcolor{blue}{0.2} \end{pmatrix}$$

$$\mathbb{P}(S_i = \textcolor{green}{v} \mid S_{i-1} = \textcolor{green}{v}) = 0$$

$$\mathbb{P}(S_i = \textcolor{green}{v} \mid S_{i-1} = \textcolor{red}{r}) = 0.4 \quad \text{etc.}$$

Propriété : les sommes en ligne de la matrice de transition font 1.

4- Chaîne de Markov cachées

Processus caché

Pour démarrer la chaîne, il faut se donner une loi de probabilité pour la première couleur appelée **loi initiale** :

$$\mathbb{P}(S_1 = u), \quad \forall u \in \mathcal{S}$$

Une chaîne de Markov d'ordre 1 est donc définie par une loi initiale et une matrice de transition.

4- Chaîne de Markov cachées

Processus caché

Pour démarrer la chaîne, il faut se donner une loi de probabilité pour la première couleur appelée **loi initiale** :

$$\mathbb{P}(S_1 = u), \quad \forall u \in \mathcal{S}$$

Une chaîne de Markov d'ordre 1 est donc définie par une loi initiale et une matrice de transition.

En pratique, la loi initiale est choisie comme étant la **loi stationnaire** $\mu(\cdot)$, c'est-à-dire vérifiant $\mu = \mu\Pi$. Ceci garantit que les variables S_i ont la même loi μ :

$$\mathbb{P}(S_i = u) = \mu(u), \quad \forall i, \quad \forall u \in \mathcal{S}.$$

La chaîne est alors dite stationnaire.

4- Chaîne de Markov cachées

Processus observé

- **Processus caché** $S = (S_1, S_2, S_3, \dots, S_n), S_i \in \mathcal{S}$ (M1)

$$\mu_e(u) = \mathbb{P}(S_1 = u), \quad \forall u \in \mathcal{S}$$

$$\pi_e(u, v) = \mathbb{P}(S_i = v \mid S_{i-1} = u), \quad \forall u, v \in \mathcal{S}$$

4- Chaîne de Markov cachées

Processus observé

- **Processus caché** $S = (S_1, S_2, S_3, \dots, S_n), S_i \in \mathcal{S}$ (M1)

$$\begin{aligned}\mu_e(u) &= \mathbb{P}(S_1 = u), \quad \forall u \in \mathcal{S} \\ \pi_e(u, v) &= \mathbb{P}(S_i = v \mid S_{i-1} = u), \quad \forall u, v \in \mathcal{S}\end{aligned}$$

- **Processus observé** $X = (X_1, X_2, X_3, \dots, X_n), X_i \in \mathcal{A}$. Conditionnellement à S , le processus X peut suivre le modèle M0 ou M1 ou Mm ou un autre. Les modèles peuvent être de différentes natures selon les états.

4- Chaîne de Markov cachées

Processus observé

- **Paramètres pour les X_i** : les X_i sont indépendants, conditionnellement à S , et générés selon une loi μ_o

$$\mu_o(u, a) = \mathbb{P}(X_i = a \mid S_i = u), \quad a \in \mathcal{A}, \quad u \in \mathcal{S}$$

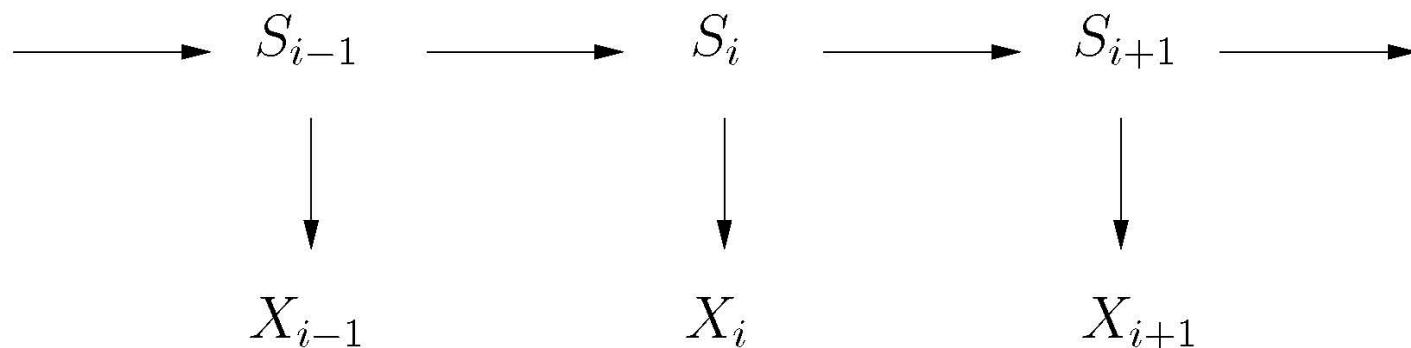
4- Chaîne de Markov cachées

Processus observé

- **Paramètres pour les X_i** : les X_i sont indépendants, conditionnellement à S , et générés selon une loi μ_o

$$\mu_o(u, a) = \mathbb{P}(X_i = a \mid S_i = u), \quad a \in \mathcal{A}, \quad u \in \mathcal{S}$$

- **Schéma de dépendances :**



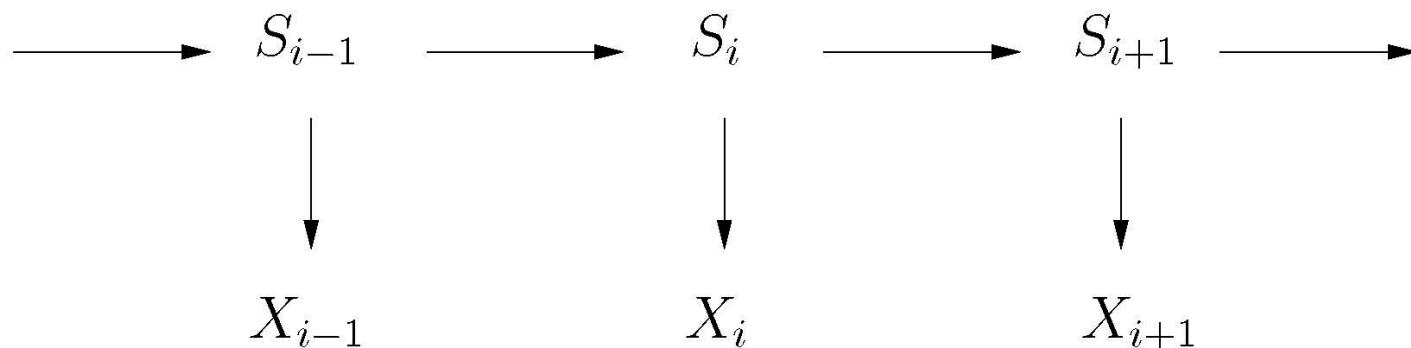
4- Chaîne de Markov cachées

Processus observé

- **Paramètres pour les X_i** : les X_i sont indépendants, conditionnellement à S , et générés selon une loi μ_o

$$\mu_o(u, a) = \mathbb{P}(X_i = a \mid S_i = u), \quad a \in \mathcal{A}, \quad u \in \mathcal{S}$$

- **Schéma de dépendances** :



- **Simulation** : on simule d'abord $S = (S_1, S_2, S_3, \dots, S_n)$ selon une CM d'ordre 1, puis on simule les X_i indépendamment des autres : X_i suit la loi d'émission de l'état s_i .

4- Chaîne de Markov cachées

- **Paramètres pour les X_i :**

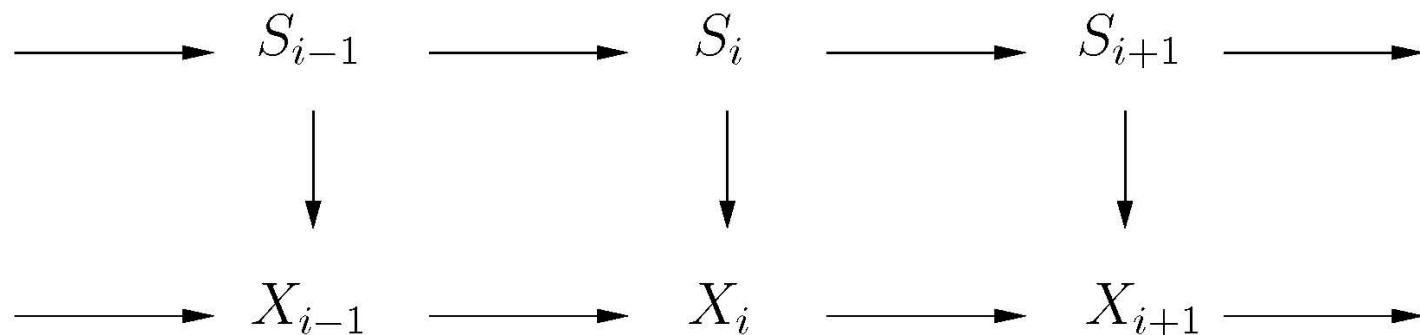
les X_i forment une chaîne de Markov stationnaire d'ordre 1, conditionnellement à S , de loi initiale μ_o

$$\mu_o(u, a) = \mathbb{P}(X_1 = a \mid S_1 = u), \quad a \in \mathcal{A}, \quad u \in \mathcal{S}$$

et de matrice de transition

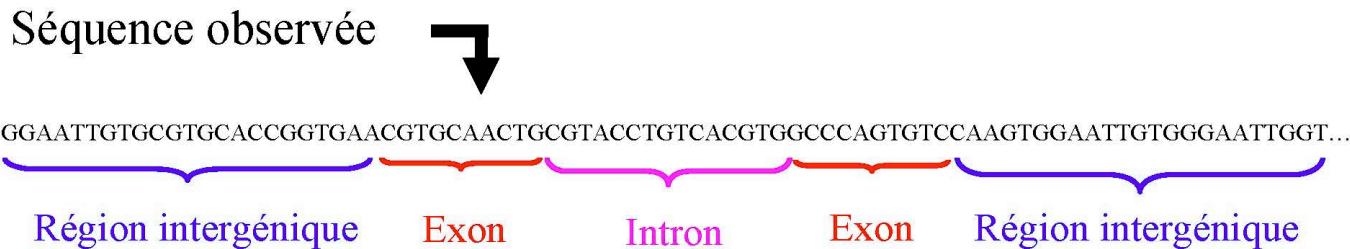
$$\pi_o(u, a, b) = \mathbb{P}(X_i = b \mid X_{i-1} = a, S_1 = u), \quad a, b \in \mathcal{A}, \quad u \in \mathcal{S}$$

- **Schéma de dépendances :**

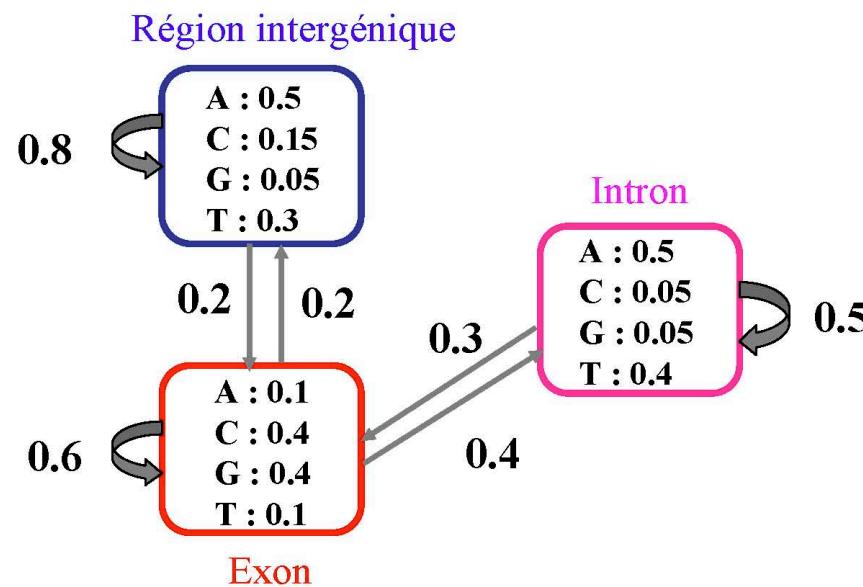


4- Chaîne de Markov cachées

Exemple 1 :



Suite des états cachés ↪



4- Chaîne de Markov cachées

Exemple 2 : les îlots CpG

Attention : CpG désigne c puis g sur le même brin, et non pas une paire complémentaire c-g en un locus donné des deux brins.

Principe biologique : la cytosine c des CpG a tendance à être méthylée, souvent en thymine t. Donc les dinucléotides cg sont plus rares que le produit des fréquences de c et de g.

Sauf autour des promoteurs de certains gènes, où la méthylation est réprimée.

Fait d'expérience : plus de cg et de c et g autour des régions promotrices qu'ailleurs: on parle d'îlots CpG.

4- Chaîne de Markov cachées

Exemple 2 : les îlots CpG

Objectifs : trouver les îlots CpG.

Remarque : problème de dinucléotides donc M1 naturel

Référence : Durbin, Eddy, Krogh, Mitchison (1998)

- Ensemble d'entraînement de 60kb, 48 îlots GpG
- Deux modèles M1 (estimation par maximum de vraisemblance : comptage), notés + pour les îlots GpG et - pour le reste.

$$q_+ = \begin{pmatrix} .180 & .274 & .426 & .120 \\ .171 & .368 & .274 & .188 \\ .161 & .339 & .375 & .125 \\ .079 & .355 & .384 & .182 \end{pmatrix}.$$

$$q_- = \begin{pmatrix} .300 & .205 & .285 & .210 \\ .322 & .298 & .078 & .302 \\ .248 & .246 & .298 & .208 \\ .177 & .239 & .292 & .293 \end{pmatrix}.$$

4- Chaîne de Markov cachées

Exemple 2 : les îlots CpG

$$q_+ = \begin{pmatrix} .180 & .274 & .426 & .120 \\ .171 & .368 & .274 & .188 \\ .161 & .339 & .375 & .125 \\ .079 & .355 & .384 & .182 \end{pmatrix},$$
$$q_- = \begin{pmatrix} .300 & .205 & .285 & .210 \\ .322 & .298 & .078 & .302 \\ .248 & .246 & .298 & .208 \\ .177 & .239 & .292 & .293 \end{pmatrix}.$$

Premier problème :

Identifier une séquence x comme étant un îlot CpG ou non.

Calculs de vraisemblance : le log(score) de x est :

$$l(x) = \log\left(\frac{P_+(x)}{P_-(x)}\right) = \sum_{x,x' \in A} N(x,x') \log\left(\frac{q_+(x,x')}{q_-(x,x')}\right)$$

4- Chaîne de Markov cachées

Exemple 2 : les îlots CpG

Deuxième problème :

Trouver la place des îlot CpG dans une séquence donnée.

Approche naïve : utiliser des fenêtres glissantes et calculer le (log)score de chaque fenêtre.

Inconvénient : quelle longueur de fenêtre choisir?

Utiliser les HMM :

Option de Durbin et al. En passant de + à - ou vice versa, on saute vers une des 4 lettres choisies avec la même probabilité.

Chemin +/- le plus probable estimé par Viterbi.

Donc prédiction des îlots CpG d'une nouvelle séquence

Exemple :

a c g a t c g c g c c a c g g t t t a t a t a a g c a a
-----+ + + + + + +-----

La suite de + est une île prédictive.

4- Chaîne de Markov cachées

Historique

- Reconnaissance de la parole (année 70)
- Génomique (1989)
- Modélisation de la croissance des plantes
- Courbes de consommation électrique
- Fiabilité des logiciels
- Etc...

4- Chaîne de Markov cachées

Estimation / Segmentation

Il y a deux écoles.

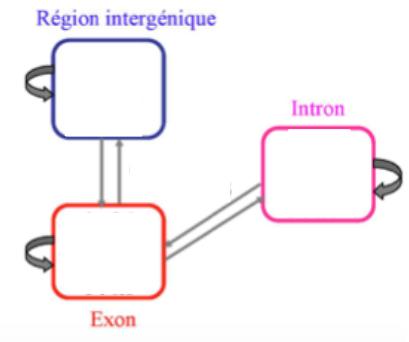
- L'approche **supervisée** consiste
 - à estimer les paramètres du modèle sur des séquences déjà segmentées
= maximum de vraisemblance sachant (X, S) .
→ implique de connaître la signification des états.
 - puis à segmenter la séquence d'intérêt avec ces paramètres.
= algorithme de Viterbi, ou algorithme “forward-backward”.
- L'approche **non supervisée** consiste à itérer les deux étapes d'estimation/segmentation directement sur la séquence
= algorithme EM.

4- Chaîne de Markov cachées

Apprentissage supervisé : séquence connue

Si la segmentation S est connue, alors la vraisemblance des données est simple et on peut la maximiser analytiquement :

$$\begin{aligned}\mathbb{P}(X \mid \theta, S) &\stackrel{M1=M0}{=} \mu_e(S_1)\pi_e(S_1, S_2)\dots\pi_e(S_{n-1}, S_n) \\ &\quad \times \mu_o(S_1, X_1)\dots\mu_o(S_n, X_n) \\ &= \mu_e(S_1) \prod_{u,v=1}^q \pi_e(u, v)^{N(uv)} \times \prod_{u=1}^q \prod_{a \in \mathcal{A}} \mu_o(u, a)^{N(u,a)}\end{aligned}$$



où $N(uv)$ est le nombre d'états u suivis de l'état v , et $N(u, a)$ est le nombre de lettres a dans l'état u .

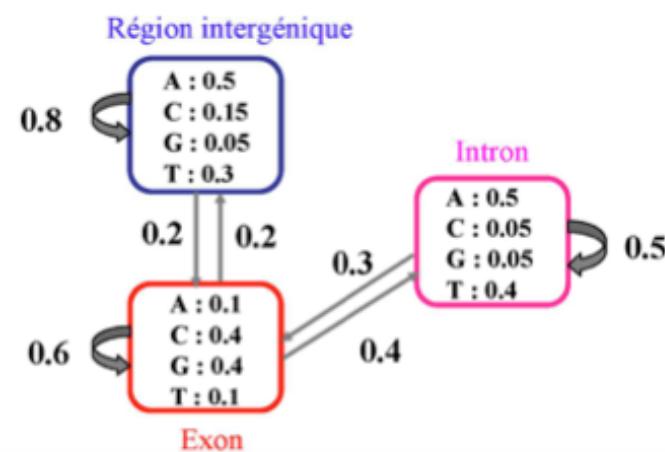
4- Chaîne de Markov cachées

Apprentissage supervisé : séquence connue

Pour maximiser la vraisemblance, on annule simultanément les dérivées partielles par rapport aux $\pi_e(u, v)$ et $\mu_o(u, a)$ et on obtient les estimateurs naturels :

$$\hat{\pi}_e(u, v) = \frac{N(uv)}{N(u)}$$

$$\hat{\mu}_o(u, a) = \frac{N(u, a)}{N(u)}$$



4- Chaîne de Markov cachées

Segmentation à paramètres connus

C'est typiquement le cas quand on estime les paramètres sur un jeu de test déjà segmenté (cf. paragraphe précédent), et que l'on veut segmenter une nouvelle séquence en gardant ces valeurs de paramètres θ .

Etant donné $X = (X_1, X_2, X_3, \dots, X_n)$ et θ , on cherche la suite d'états $(s_1^*, s_2^*, \dots, s_n)$ la plus probable, c'est-à-dire celle qui maximise

$$\mathbb{P}(S_1 = s_1, \dots, S_n = s_n \mid X, \theta)$$

ou encore (formule de Bayes)

$$\mathbb{P}(X_1, \dots, X_n, S_1 = s_1, \dots, S_n = s_n \mid \theta)$$



Algorithme de Viterbi

4- Chaîne de Markov cachées

Segmentation à paramètres connus : algorithme de Viterbi

Objectif trouver le meilleur chemin (succession d'états) :

- calculer tous les chemins : peu réaliste
- optimisation : programmation dynamique

Programmation dynamique :

- résoudre un problème en le décomposant en sous-problèmes
- puis à résoudre les sous-problèmes
- des plus petits aux plus grands en stockant les résultats intermédiaires
- puis combinent leurs solutions pour résoudre le problème initial



principe de diviser pour mieux régner

4- Chaîne de Markov cachées

Segmentation à paramètres connus : algorithme de Viterbi

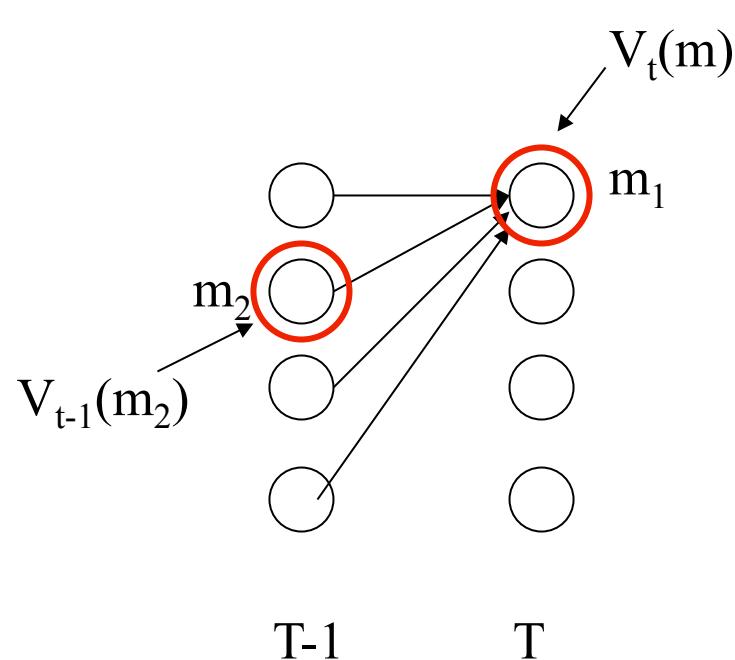
Programmation dynamique : 4 étapes

- 1- Caractériser la structure d'une solution optimale.
- 2- Définir (souvent de manière récursive) la valeur d'une solution optimale.
- 3- Calculer la valeur d'une solution optimale de manière ascendante.
- 4- Construire une solution optimale à partir des informations calculées.

4- Chaîne de Markov cachées

Segmentation à paramètres connus : algorithme de Viterbi

Réurrence :

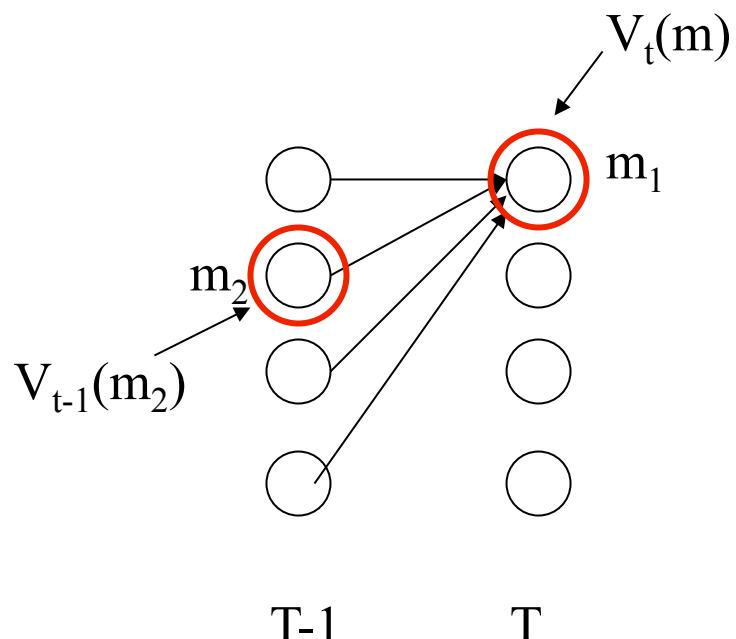


Sélection d' un chemin dans
le treillis entre les instants $t-1$ et t

4- Chaîne de Markov cachées

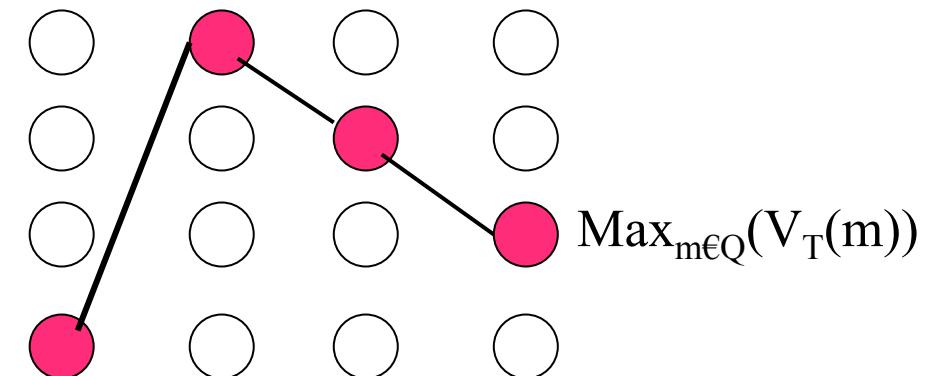
Segmentation à paramètres connus : algorithme de Viterbi

Récurrence :



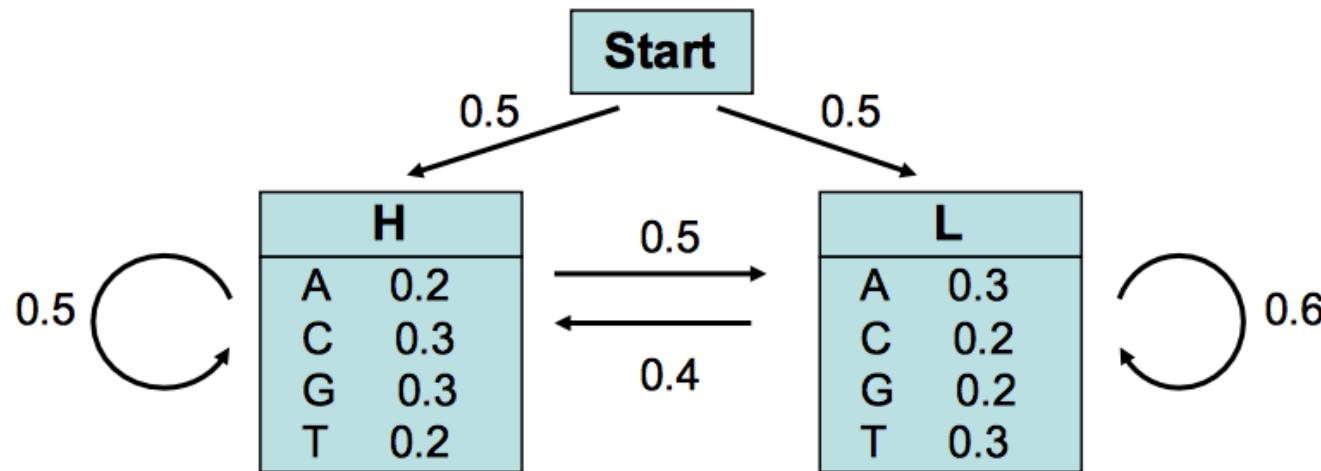
Sélection d' un chemin dans
le treillis entre les instants $t-1$ et t

Reconstitution :



Reconstruction du chemin
correspondant à la séquence d' état
optimale

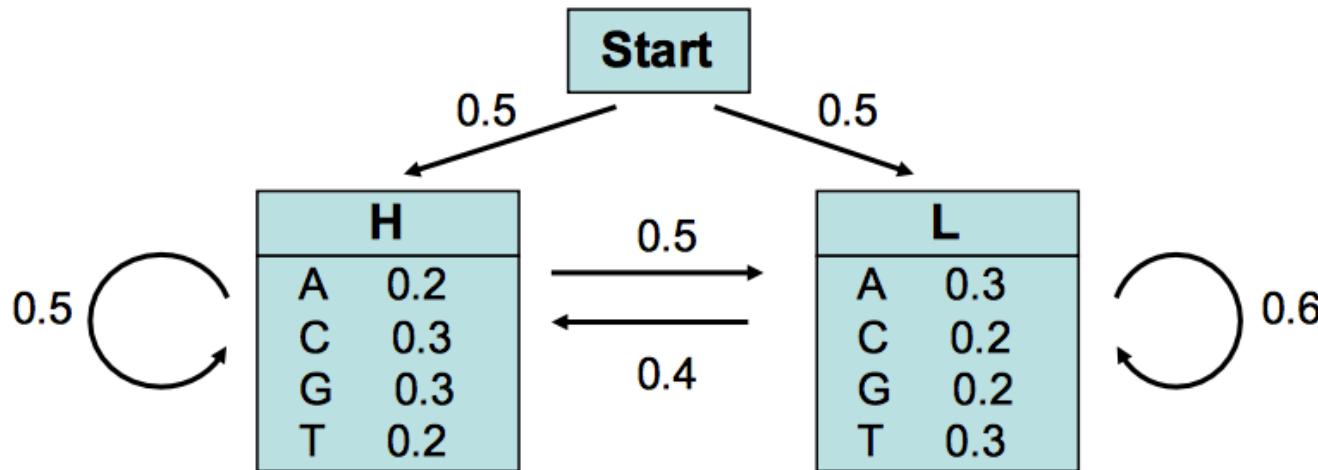
Algorithme de Viterbi : application à un exemple simple



Ce CMC a 2 états :

- H (high GC : région codante) et L (low GC : région non codante)
- Objectif : prédire les régions codantes d'une séquence d'ADN donnée.

Algorithme de Viterbi : application à un exemple simple



Soit la séquence S= **GGCACTGAA**

Plusieurs chemins cachés peuvent donner la séquence

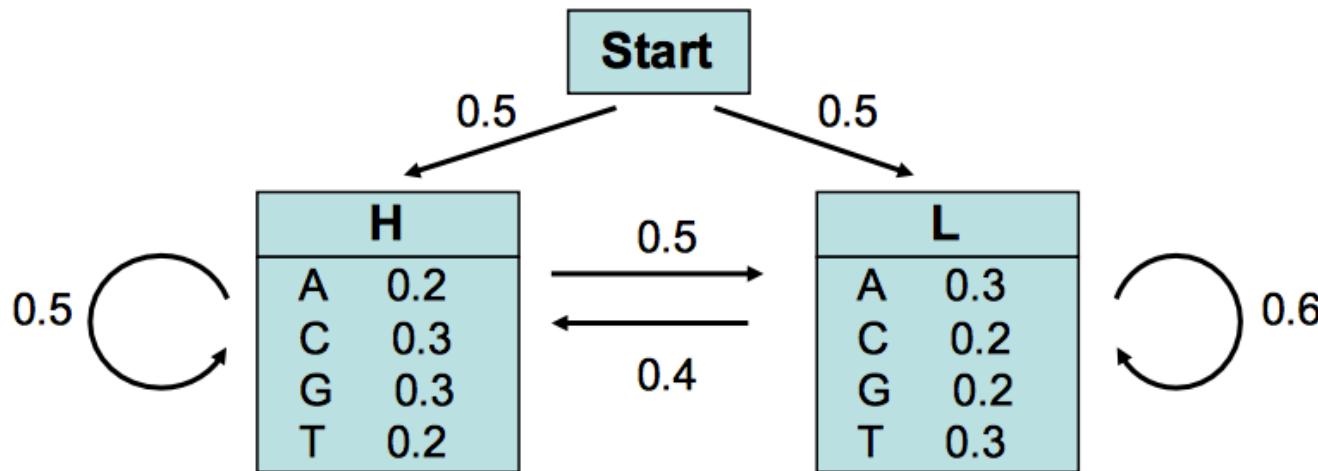
Exemple: P = **LLHHHHHLLL**

La probabilité de produire la séquence S à partir du CMC à partir du chemin

P est :

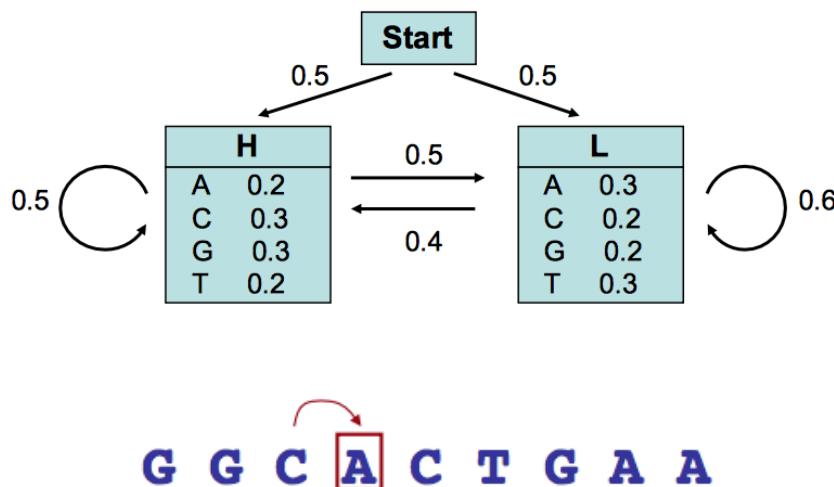
$$\begin{aligned} p &= pL(0) * pL(G) * pL(L) * pL(G) * pL(H) * pH(C) * \dots \\ &= 0.5 * 0.2 * 0.6 * 0.2 * 0.4 * 0.3 * \dots = \dots \end{aligned}$$

Algorithme de Viterbi : application à un exemple simple



Plusieurs chemins à partir des états cachés (**H** et **L**) peuvent produire la séquence observée mais qui ont des probabilités différentes.
L'algorithme de Viterbi permet de calculer le chemin le plus probable

Algorithme de Viterbi : application à un exemple simple



Séquence :

Position : $x-1$ x

Lettre : *i* *i*

—

états précédents possibles : tous état /

Principe : Probabilité du chemin le plus probable finissant dans l'état k avec l'observation i est

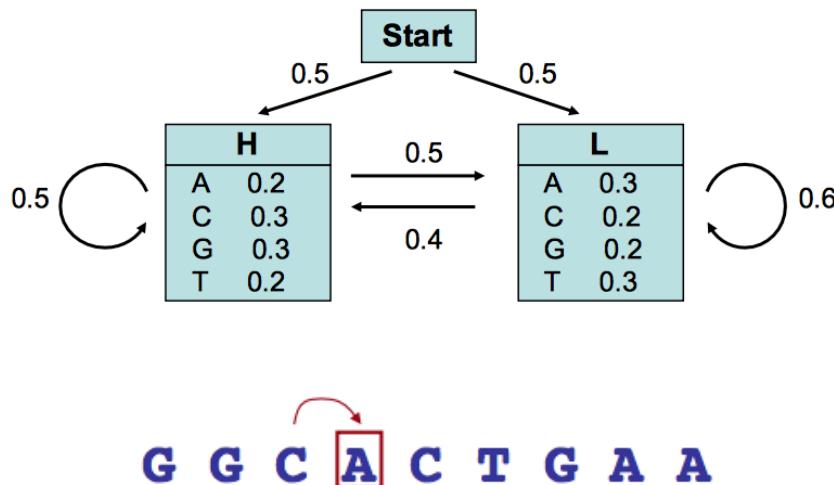
$$p_l(i, x) = e_l(i) \max_k (p_k(j, x-1)).$$

Probabilité d'observer l'élément i dans l'état I

Probabilité du chemin le plus probable finissant en position $x-1$ dans l'état k avec l'élément j

Probabilité de
transition de l'état I à
l'état k

Algorithme de Viterbi : application à un exemple simple



Séquence :

Position : $x-1$ x

Lettre : *i* *i*

états précédents possibles : tous état /

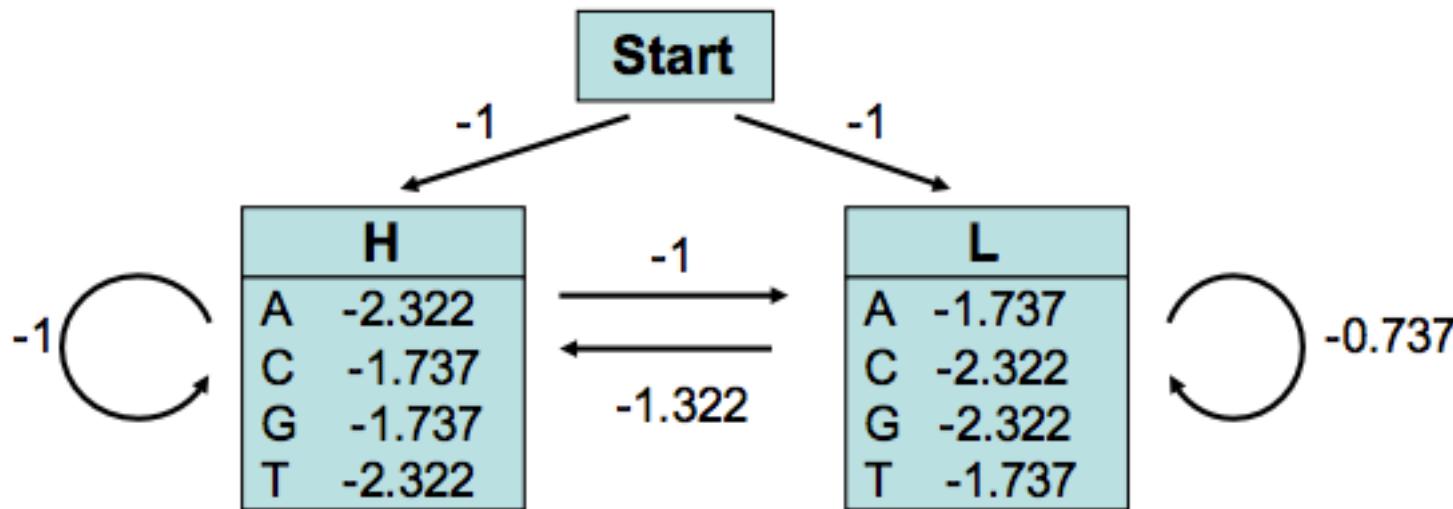
Principe : Probabilité du chemin le plus probable finissant dans l'état k avec l'observation i est $p_l(i, x) = e_l(i) \max_k (p_k(j, x-1) \cdot p_{kl})$

Dans notre exemple : la probabilité du chemin le plus probable finissant en état H avec l'observation « A » en 4^{ème} position est :

$$p_H(A,4) = e_H(A) \max(p_L(C,3).p_{LH}, p_H(C,3).p_{HH})$$

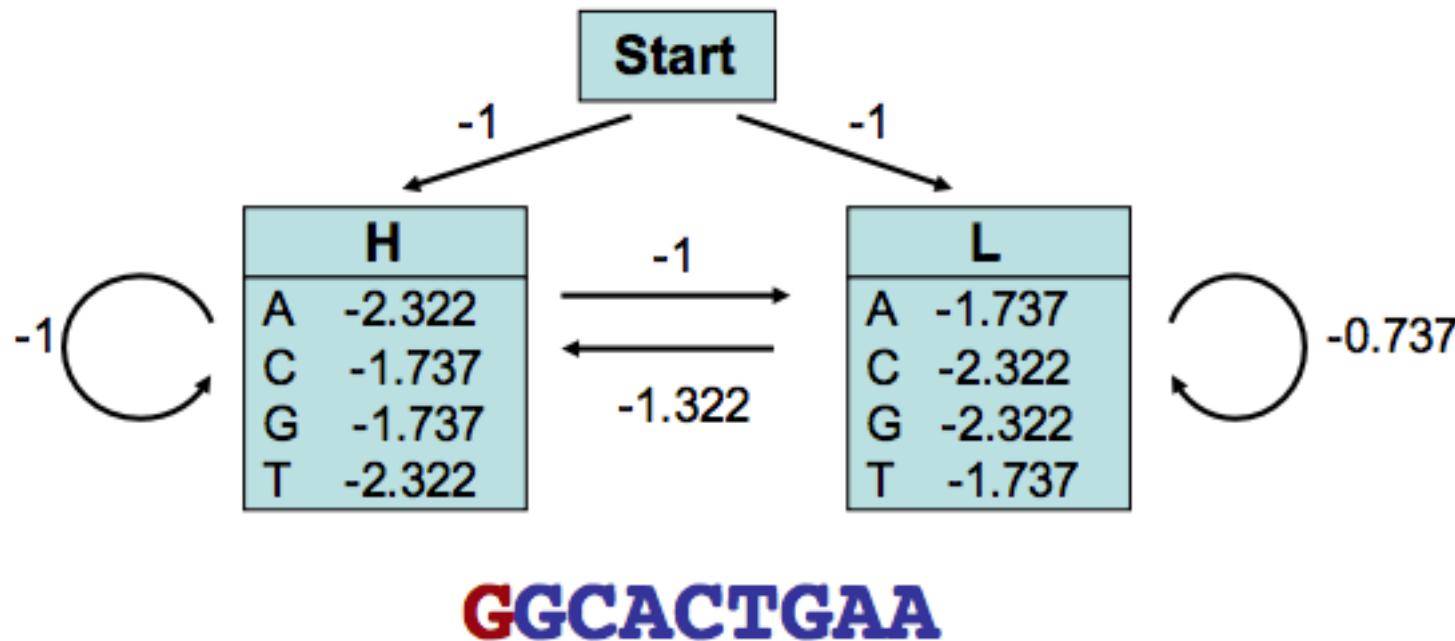
Nous pouvons calculer récursivement (du premier au dernier élément de notre séquence), le chemin le plus probable

Algorithme de Viterbi : application à un exemple simple



Remarque : pour les calculs, il est préférable d'utiliser les log des probabilités, ainsi on peut faire des sommes à la place des produits ce qui est plus efficace et plus précis pour les calculs.
Nous utiliserons ici le $\log_2(p)$.

Algorithme de Viterbi : application à un exemple simple



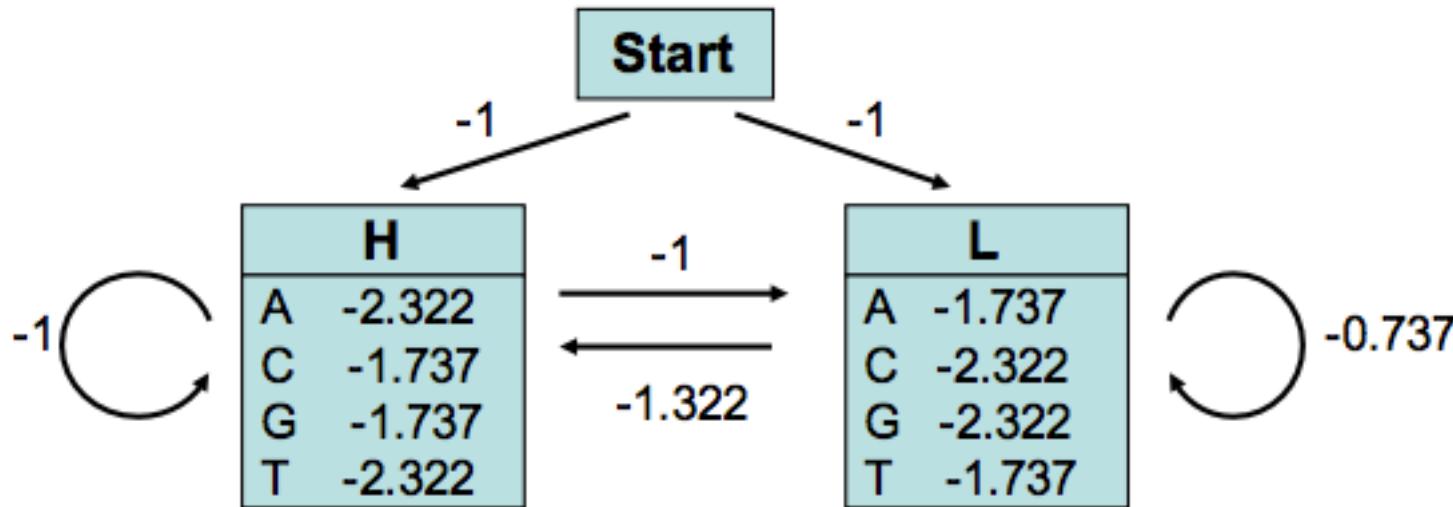
Probabilité (log2) que le G à la première position soit émis par l état H

$$p_H(G, 1) = -1 - 1.737 = -2.737$$

Probabilité (log2) que le G à la première position soit émis par l état L

$$p_L(G, 1) = -1 - 2.322 = -3.322$$

Algorithme de Viterbi : application à un exemple simple



GGCACTGAA

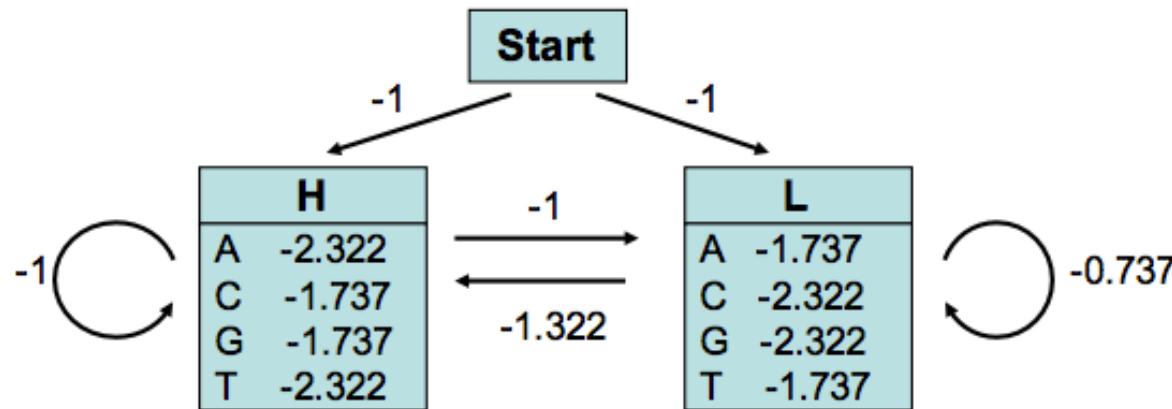
Probabilité (log2) que G à la deuxième position soit émis par l'état H

$$\begin{aligned}
 p_H(G,2) &= -1.737 + \max(p_H(G,1) + p_{HH}, p_L(G,1) + p_{LH}) \\
 &= -1.737 + \max(-2.737 - 1, -3.322 - 1.322) \\
 &= -5.474
 \end{aligned}$$

Probabilité (log2) que G à la deuxième position soit émis par l'état L

$$\begin{aligned}
 p_L(G,2) &= -2.322 + \max(p_H(G,1) + p_{HL}, p_L(G,1) + p_{LL}) \\
 &= -2.322 + \max(-2.737 - 1.322, -3.322 - 0.737) \\
 &= -6.059
 \end{aligned}$$

Algorithme de Viterbi : application à un exemple simple



GGCACTGAA

	G	G	C	A	C	T	G	A	A
H	-2.73	-5.47	-8.21	-11.53	-14.01	...			-25.65
L	-3.32	-6.06	-8.79	-10.94	-14.01	...			-24.49

Trouver le chemin qui possède la plus forte probabilité :

Le chemin le plus probable est : **HHHLLLLL**

La probabilité est $2^{-24.49}$

4- Chaîne de Markov cachées

Apprentissage non supervisé

Si la segmentation S n'est pas connue, la vraisemblance $\mathbb{P}(X \mid \theta)$ n'est pas manipulable. Pour la maximiser, on utilise des *algorithmes itératifs* qui permettent d'approcher l'estimateur $\hat{\theta}$ du maximum de vraisemblance.

L'algorithme EM (*Expectation-Maximization*) est le plus populaire.
Clé : à chaque étape, la vraisemblance croît.

- point de départ $\theta^{(0)}$
- itération k alterne une étape E et une étape M
- critère d'arrêt :
$$|\log \mathbb{P}(X = x \mid \theta^{(k+1)}) - \log \mathbb{P}(X = x \mid \theta^{(k)})| < \varepsilon \text{ ou } k > M$$

Attention : pb. des maxima locaux \Rightarrow plusieurs points de départ.

4- Chaîne de Markov cachées

Apprentissage non supervisé

Une itération de l'algorithme :

- Etape E : on calcule $\mathbb{P}(S_i = u \mid X = x, \theta^{(k)})$, $i = 1, \dots, n$,
 $u \in \mathcal{S}$ (algorithme *Forward-Backward*)
- Etape M : on calcule $\theta^{(k+1)}$ en utilisant la segmentation obtenue

$$\pi_e^{(k+1)}(u, v) = \frac{\sum_i \mathbb{P}(S_i = u, S_{i+1} = v \mid X = x, \theta^{(k)})}{\sum_i \mathbb{P}(S_i = u \mid X = x, \theta^{(k)})}$$

$$\mu_o^{(k+1)}(u, a) = \frac{\sum_i \mathbf{1}\{X_i = a\} \mathbb{P}(S_i = u \mid X = x, \theta^{(k)})}{\sum_i \mathbb{P}(S_i = u \mid X = x, \theta^{(k)})}$$

5- Application à l' analyse de séquences

Quelques applications des HMMs à la génomique

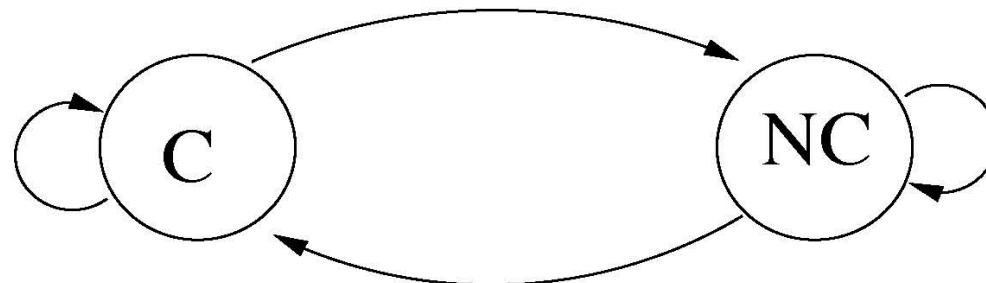
- Hétérogénéité des séquences sans renseignements a priori
- Transfert horizontaux de gènes
- Recherche de motifs
- Prédiction et annotation de gènes
- Alignements de séquences
- Reconstruction d' arbres phylogénétiques
- Prédiction de structures secondaires
- etc.

5- Application à l' analyse de séquences

Détection de gènes

De nombreux “déTECTEURS de gènes”.

Principe de base : alternance de codant/intergénique (procaryotes) ou intergénique/exons/introns (eucaryotes), prise en compte de la phase pour le codant.

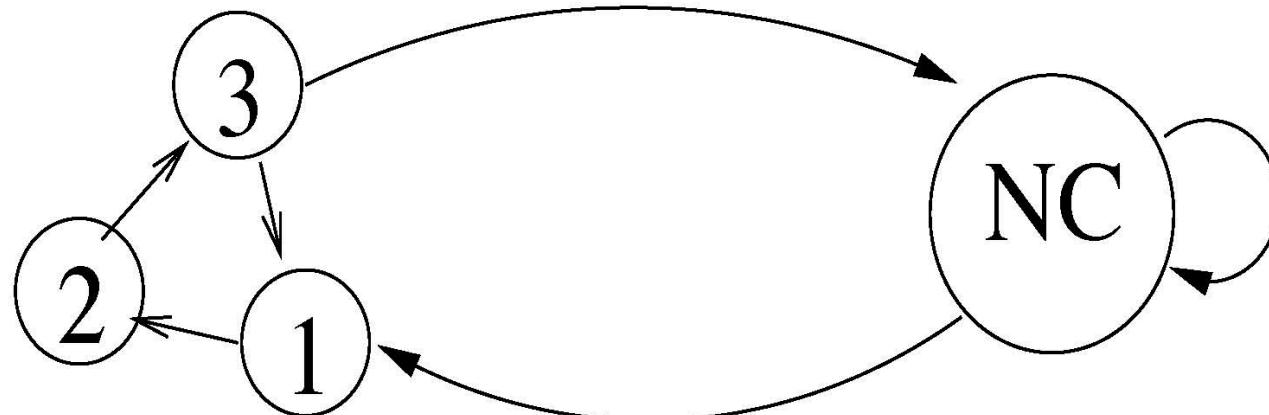


5- Application à l' analyse de séquences

Détection de gènes

De nombreux “déTECTEURS de gènes”.

Principe de base : alternance de codant/intergénique (procaryotes) ou intergénique/exons/introns (eucaryotes), prise en compte de la phase pour le codant.

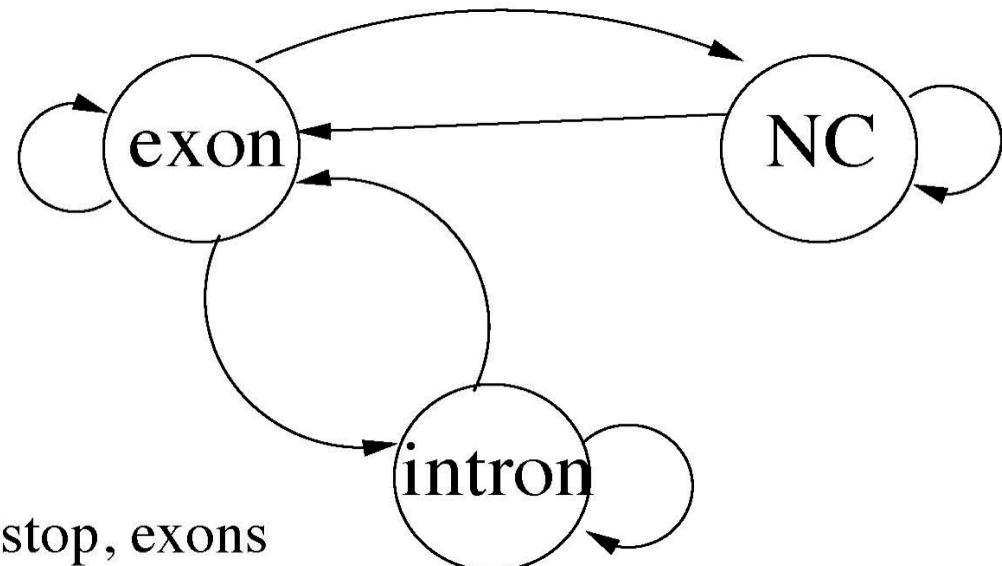


5- Application à l' analyse de séquences

Détection de gènes

De nombreux “déTECTEURS de gènes”.

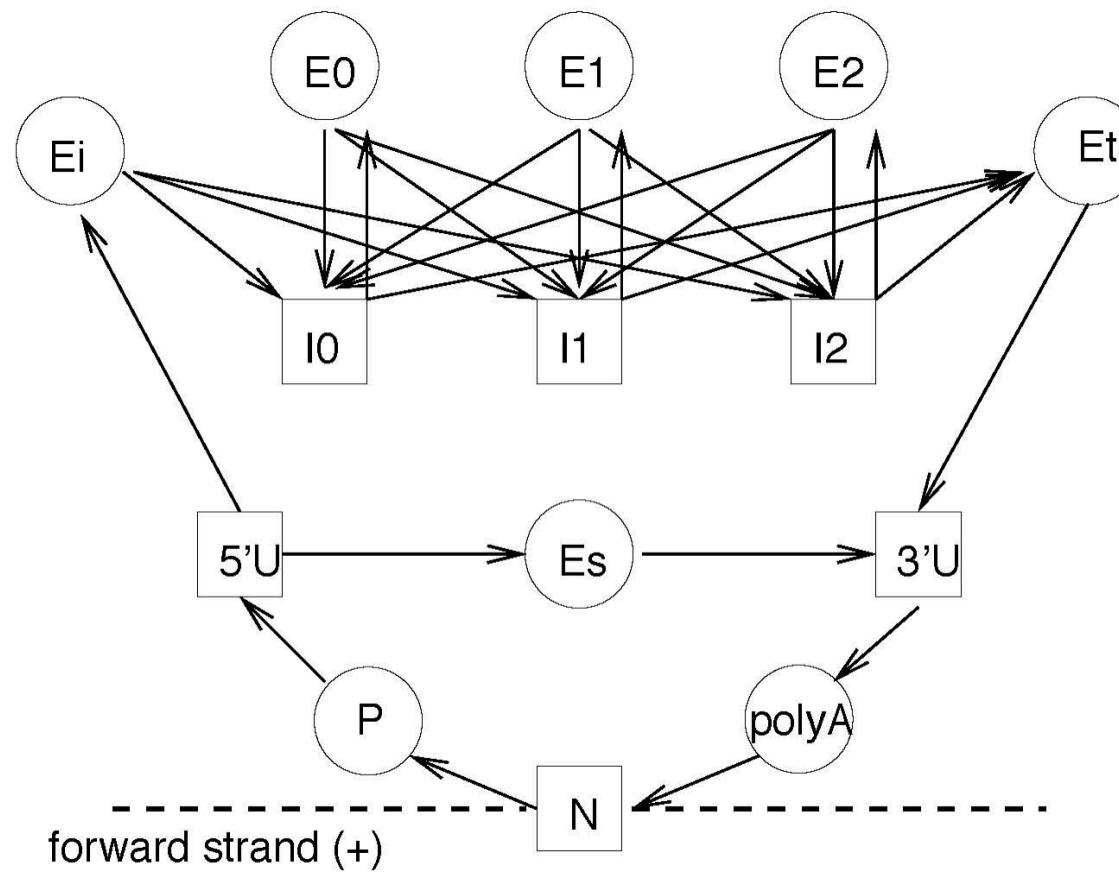
Principe de base : alternance de codant/intergénique (procaryotes) ou intergénique/exons/introns (eucaryotes), prise en compte de la phase pour le codant.



Complexifications : codons start/stop, exons initial/centraux/terminal, etc.

5- Application à l' analyse de séquences

Détection de gènes



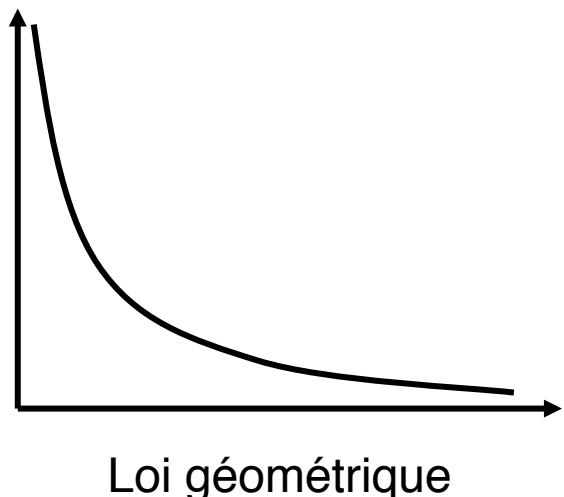
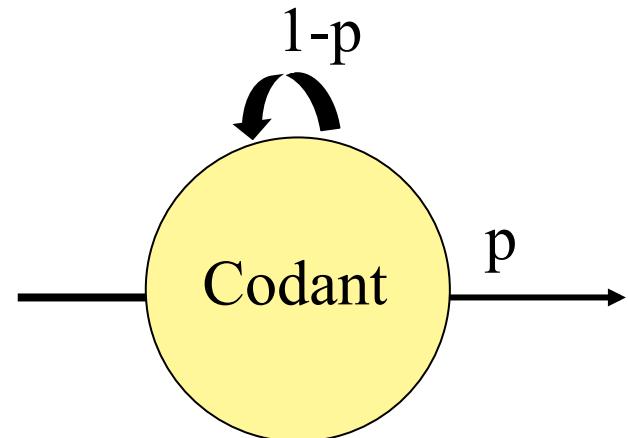
(Genscan)

5- Application à l' analyse de séquences

Détection de gènes

Distribution de longueur d'un état caché :

T : temps de séjour dans un état donné
T suit une loi géométrique

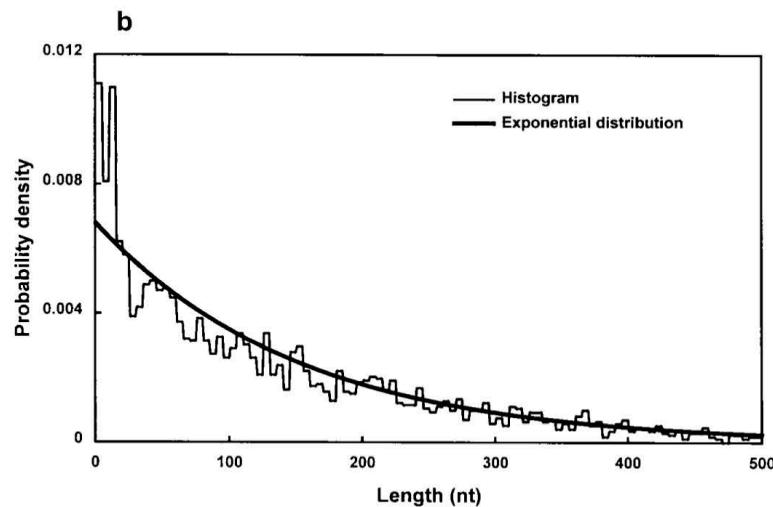
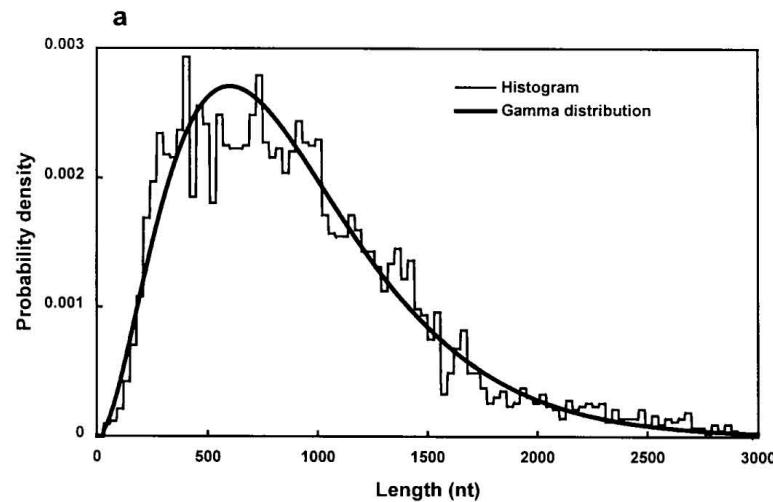


$$P[X=k] = (1-p)^{k-1}p$$

$$E[X] = 1/p$$

5- Application à l' analyse de séquences

Détection de gènes



(GeneMark.hmm, Genscan)