Modèles probabilistes de séquences

ou comment écrire le Livre automatiquement

Gènes ARN

ARN de transfert

ARN ribosomale

d'autres (p.e., Ribonucléase P ARN)

acide nucléique à un brin — repliement par liaisons hydrogène

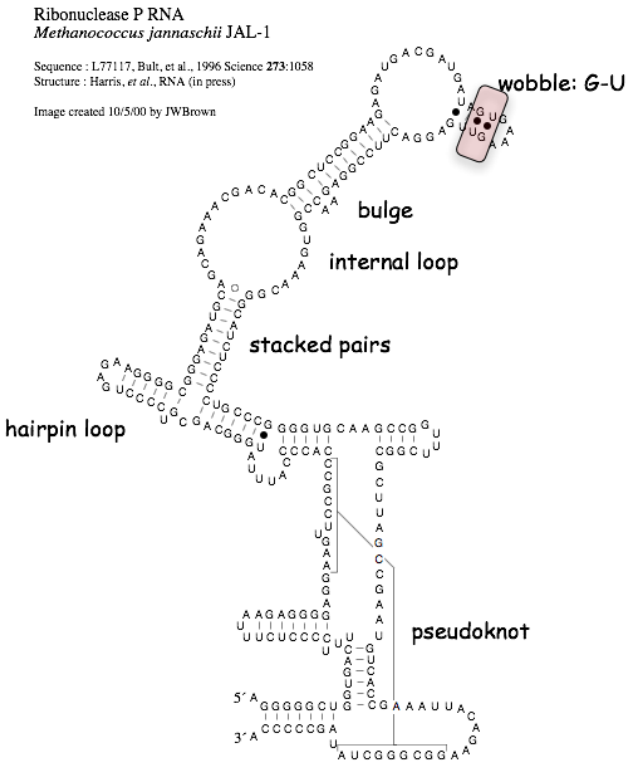
structure est importante

ARNt

(notez les hélices)

Wikipedia

Structure secondaire — RNase P RNA



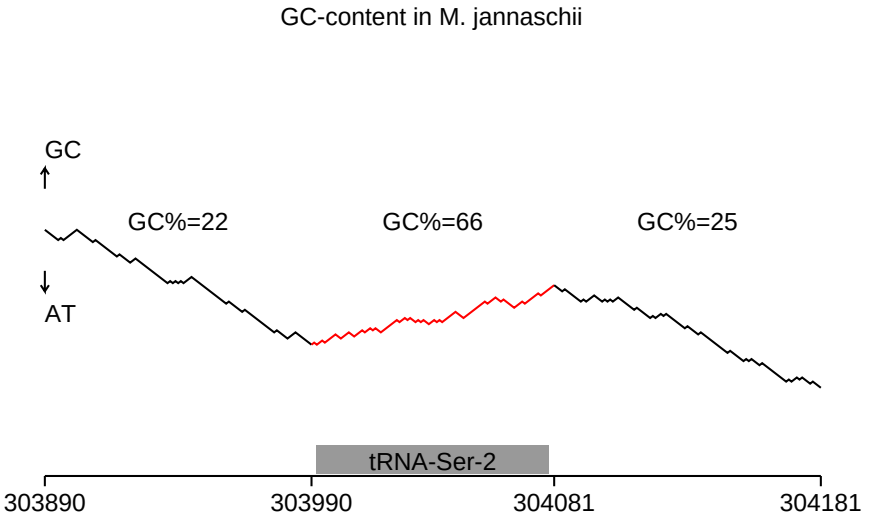
Recherche de gènes ARN

- modèle de structure secondaire : scanner la séquence du génome

- exemple : `tRNAScan`

Gènes ARN dans des thermophiles

en procaryotes thermophiliques on peut identifier les gènes ARN par contenu de GC



Segmentation

Modèle : séquence X de longueur n où $X[i] \in \{\mathtt{W}, \mathtt{S}\}$
($\mathtt{W} = \{\mathtt{A}, \mathtt{T}\}; \mathtt{S} = \{\mathtt{C}, \mathtt{G}\}$).

Classe de positions : séquence Z de longueur n où $Z[i] \in \{\mathtt{0}, \mathtt{1}\}$;
$\mathtt{0}$ = riche en $\mathtt{AT}$; $\mathtt{1}$ = riche en $\mathtt{GC}$.

Caractère en position i dépend de $Z[i]$ seulement :

$$\mathbb{P}\Big\{X[i] = x \;\Big|\; Z[i] = z\Big\} = p_z(x).$$

M. jannaschii : $p_{\mathtt{0}}(\mathtt{S}) = 0.22, p_{\mathtt{1}}(\mathtt{S}) = 0.66$

Segmentation : trouver Z à partir de X

Vraisemblance

Étant donné la séquence x :

- vraisemblance de z :

$$L(z) = \mathbb{P}\Big\{ X = x \;\Big|\; Z = z \Big\} = \prod_{i=1}^{n} p_{z[i]}(x[i]).$$

- log-vraisemblance

$$\ell(z) = \log L(z) = \sum_{i=1}^{n} \log p_{z[i]}(x[i])$$

$$= \underbrace{\sum_{i=1}^{n} \log p_0(x[i])}_{\text{vrais. de l'hypo } Z=0} + \underbrace{\sum_{i=1}^{n} \log \frac{p_{z[i]}(x[i])}{p_0(x[i])}}_{\text{LLR de l'hypo } z}.$$

Problème : le z qui maximise $\ell(z)$ n'est pas utile : $z[i] = \mathsf{argmax}_z p_z(x[i])$
— trop de segments ($z[i] = 0$ si $x[i] = \mathtt{W}$ et $z[i] = 1$ si $x[i] = \mathtt{S}$)

Longueur de déscription

Comment choisir z ?

Principe de longueur de déscription minimale : le meilleur hypothèse est celui qui est le plus court à encoder (Rissanen 1983)

Encodage : données et modèle en même temps

Ici : encoder x et z :

$$\underbrace{001010\cdots1}_{x \text{ encodé en binaire}} \ \# \ \underbrace{C(z)}_{z \text{ encodé}}$$

Encodage

Encoder x : codage optimale en utilisant z
$-\lg p_{z[i]}(x[i])$ bits pour encoder $x[i]$.

Comment ? Encodage Huffmann de blocs de taille m

Exemple de l'avantage de Huffmann : $p(\mathtt{W}) = 0.75, p(\mathtt{S}) = 0.25$; $m = 2$
$\mathtt{WW} \to \mathtt{0}$, $\mathtt{WS} \to \mathtt{10}$, $\mathtt{SW} \to \mathtt{110}$, $\mathtt{SS} \to \mathtt{111}$

Nombre de bits par bloc en moyenne : $\frac{9}{16} \cdot 1 + \frac{3}{16} \cdot 2 + \frac{3}{16} \cdot 3 + \frac{1}{16} \cdot 3 = \frac{27}{16}$,
donc 0.84 bits par caractère

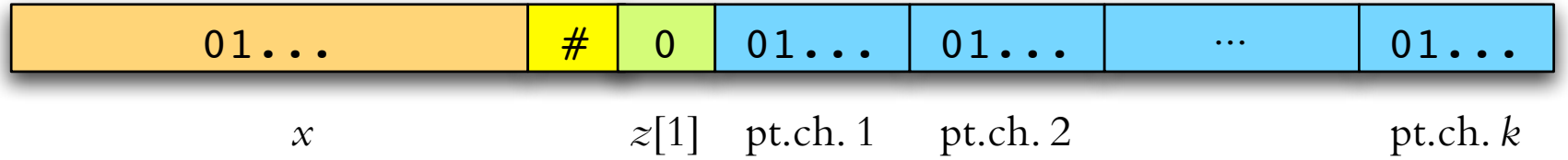
Encodage — segmentation

Segmentation z — comment l'encoder ?

Il y a beaucoup moins de changements $z[i] \neq z[i-1]$ que la longueur n de la séquence

Donc, on va juste donner la liste de *points de changement* ou $z[i] \neq z[i-1]$

1 bit pour encoder $z[0]$
chaque point de changement encodé en $\lg(n-1)$ bits (entier entre 2 et n)



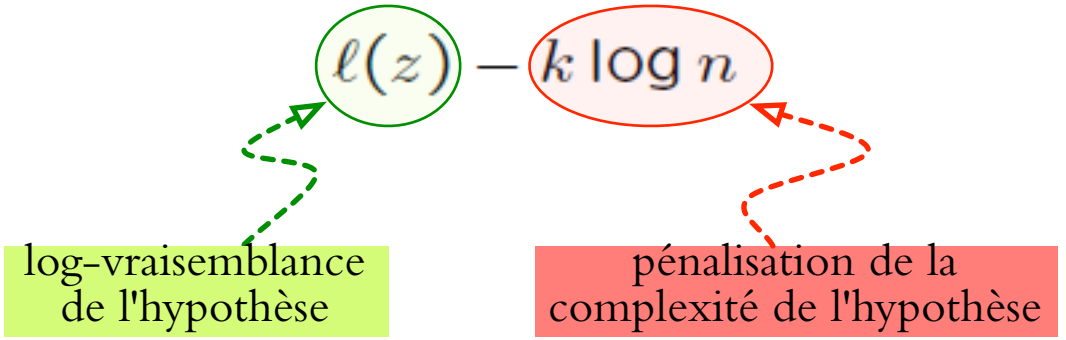
Encodage de z avec k points de changements : $1 + k \lg(n-1)$ bits

Encodage complète

Longueur de l'encodage complète x et z quand k points de changements :

$$\sum_{i=1}^{n} \Big(- \lg p_{z[i]}(x[i]) \Big) + k \lg n + O(1) = - \lg L(z) + k \lg n + O(1)$$

Le meilleur choix de z minimise cette longueur — c-à-d il maximise



Principe universel de sélection de modèles probabilistes :

balancer la complexité du modèle et son accord avec les données

Algorithme

But : trouver z qui maximise $\mathsf{LLR}(z) - kh$

(car $\ell(z) = \ell(0) + \mathsf{LLR}(z)$; $h = \log n$)

Notation : $w(i) = \log \frac{p_1(x[i])}{p_0(x[i])}$ donc $\mathsf{LLR}(z) = \sum_{i:\ z[i]=1} w(i)$.

PD : $V_a(i)$ est le score de la meilleure segmentation de $1..i$ où $z[i] = a$.

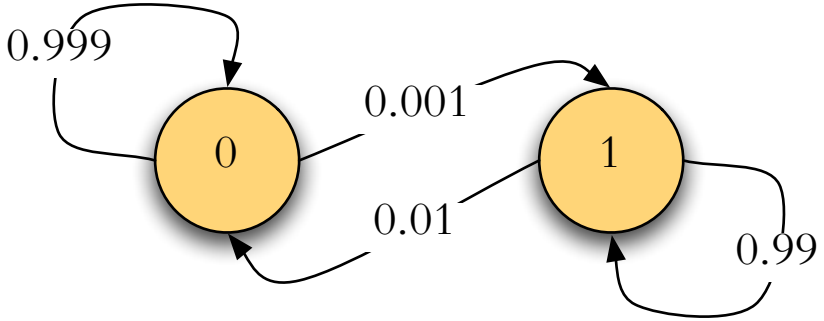
$$
\begin{aligned}
V_0(i) &= \max\Big\{V_0(i-1), V_1(i-1) - h\Big\} && i > 1 \\
V_1(i) &= \max\Big\{V_0(i-1) + w(i) - h, V_1(i-1) + w(i)\Big\} && i > 1 \\
V_0(1) &= 0; V_1(1) = w(1)
\end{aligned}
$$

+ traceback sur les $\max$

Csűrös *IEEE Trans Comput Biol Bioinform* 1 :139

Modèle de Markov caché

Un autre modèle : Z est une chaîne de Markov



Chaîne de Markov :

- ensemble d'états Q
- probabilité initiale $\mu_q = \mathbb{P}\{z[1] = q\}$
- probabilités de transition $t_{q \to q'} = \mathbb{P}\Big\{ z[i+1] = q' \;\Big|\; z[i] = q \Big\}$

Probabilité *a priori* de la segmentation $z \in Q^n$

$$\mathbb{P}\{Z = z\} = \mu_{z[1]} \prod_{i=2}^{n} t_{z[i-1] \to z[i]}.$$

Modèle de Markov caché 2

Trouver la meilleure segmentation

$$\begin{aligned}\mathbb{P}\Big\{Z = z \,\Big|\, X = x\Big\} &= \frac{\mathbb{P}\{Z = z; X = x\}}{\mathbb{P}\{X = x\}} \\ &= \frac{\mathbb{P}\Big\{X = x \,\Big|\, Z = z\Big\}\mathbb{P}\{Z = z\}}{\mathbb{P}\{X = x\}} \\ &\propto L(z) \cdot P(z)\end{aligned}$$

Maintenant,

$$\mathbb{P}\{Z = z; X = x\} = \mu_{z[1]} p_{z[1]}(x[1]) \prod_{i=2}^{n} p_{z[i]}(x[i]) t_{z[i-1] \to z[i]}.$$

Modèle de Markov caché 3

Avec les logarithmes

$$\begin{aligned}
&\log \mathbb{P}\{Z = z; X = x\} \\
&\qquad = \ell(0) + (n-1)\log t_{0\to 0} && //\text{constante} \\
&\qquad + \sum_{i:z[i]=1} \left(\log \frac{p_1(x[i])}{p_0(x[i])} + \log \frac{t_{1\to 1}}{t_{0\to 0}}\right) && //\sum_i w(i) \\
&\qquad + \sum_{i:z[i]=1;z[i-1]=0} \log \frac{t_{0\to 1}}{t_{1\to 1}} && //k_1 h_1 \\
&\qquad + \sum_{i:z[i]=0;z[i-1]=1} \log \frac{t_{1\to 0}}{t_{0\to 0}} && //k_0 h_0
\end{aligned}$$

Donc, il faut maximiser une expression de genre $\sum_{i:z[i]=1} w(i) - k_1 h_1 - k_0 h_0$ où k_j est le nombre de positions i avec $z[i] = j$ et $z[i-1] = 1-j$.
— presque le même problème qu'avant.

Segmentation postérieure

Au lieu de trouver z de probabilité maximale, trouver j avec
$\mathbb{P}\Big\{ Z[i] = j \;\Big|\; X = x \Big\}$ maximale en chaque position $i = 1, \ldots, n$.

Méthode :
prob. préfixe $\alpha_j(i) = \mathbb{P}\{Z[i] = j; X[1..i] = x[1..i]\}$
prob. suffixe $\beta_j(i) = \mathbb{P}\{Z[i] = j; X[i+1..n] = x[i+1..n]\}$.

Alors

$$\delta_j(i) = \mathbb{P}\Big\{ Z[i] = j \;\Big|\; X = x \Big\} = \frac{\alpha_j(i)\beta_j(i)}{\sum_{j \in Q} \alpha_j(i)\beta_j(i)}.$$

Segmentation postérieure 2

Probabilités préfixe

$$\alpha_j(1) = \mu_j p_j(x[1])$$
$$\alpha_j(i) = \left(\sum_{j' \in Q} \alpha_{j'}(i-1) t_{j' \to j} \right) p_j(x[i]) \qquad i > 1$$

Probabilités suffixe

$$\beta_j(n) = 1$$
$$\beta_j(i) = \sum_{j' \in Q} t_{j \to j'} p_{j'}(x[i+1]) \beta_{j'}(i+1) \qquad i < n$$

Aspects pratiques

Pbl : probabilités deviennent très petites, précision double (`double` de Java) ne permet que 10^{-307}

Solution : travailler explicitement avec des paires de mantisse-exponent (`double`, `int`)

Pbl : trop de mémoire (au moins $12n|Q|$ octets)

Solution : couper la séquence en blocs de taille $B = \sqrt{n}$ — *checkpoints*
1. de droite à gauche : calculer tous les $\beta_j(i)$ et les stocker pour $i = B+1, 2B+1, 3B+1, \ldots$
2. de gauche à droite : calculer tous les $\alpha_j(i)$; en arrivant au début de bloc k calculer d'abord $\beta_j(i)$ pour $i = kB+1..(k+1)B$

Usage de mémoire : $O(\sqrt{n}|Q|)$

Apprentissage de paramètres

Approche I. calculer la meilleure segmentation (Viterbi)

$$\widehat{t}_{j\to j'} = \frac{\sum_{i=2}^{n}\{z[i]=j'; z[i-1]=j\}}{\sum_{i=2}^{n}\{z[i-1]=j\}}$$
$$\widehat{p}_j(a) = \frac{\sum_{i=1}^{n}\{x[i]=a; z[i]=j\}}{\sum_{i=1}^{n}\{z[i]=j\}}.$$

Refaire la segmentation, recalculer les paramètres, etc.

Approche II. (Baum-Welch) calculer les probabilités postérieures

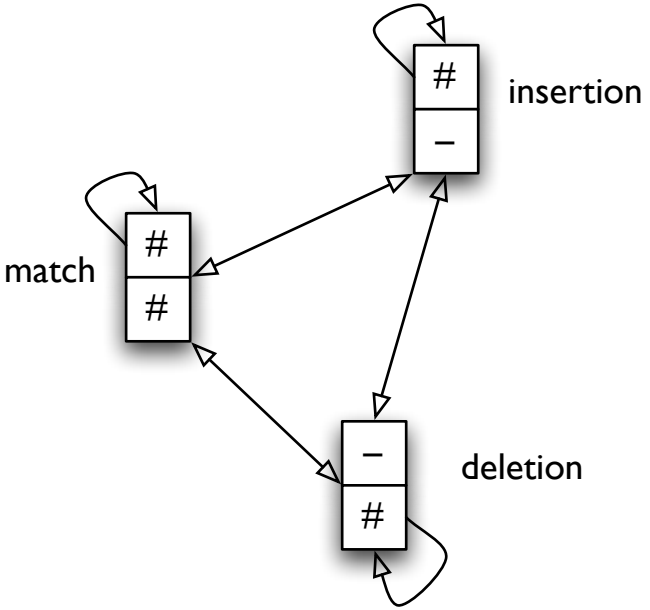
$$\widehat{t}_{j\to j'} = \frac{\sum_{i=2}^{n}\delta_{j'\to j}(i)}{\sum_{i=2}^{n}\sum_{j\in Q}\delta_{j'\to j}(i)}$$
$$\widehat{p}_j(a) = \frac{\sum_{i=1}^{n}\delta_j(i)\{x[i]=a\}}{\sum_{i=1}^{n}\{x[i]=n\}\sum_{j\in Q}\delta_j(i)}$$

où $\delta_{j'\to j}(i) = \alpha_{j'}(i-1)t_{j'\to j}p_j(x[i])\beta_j(i)$. Refaire la segmentation, recalculer les paramètres, etc.

Machine d'alignement

HMM : émission de caractères (a, b) où $a, b \in \{\mathtt{A}, \mathtt{G}, \mathtt{T}, \mathtt{C}, -\}$ (sauf $a = b = -$)

états : M («match»), I (insertion), D (suppression), transitions probabilistes t



Knudsen & Miyamoto *J. Mol. Biol.* 333 : 453 (2003)

Machine 2

Émission d'appariements en chaque état : une série d'émissions donne l'alignement : probabilités p_M, p_I, p_D.

Problème : trouver le meilleur alignement global de deux séquences S et T.

Sous-problèmes pour programmation dynamique : $M(i, j)$, $D(i, j)$, $I(i, j)$ score de l'alignement des préfixes si on finit en état M , I, ou D.

Récurrences

$$M(i,j) = p_M\begin{bmatrix} S[i-1] \\ T[j-1] \end{bmatrix} \cdot \max\Big\{ M(i-1,j-1)t(M \to M), \\ D(i-1,j-1)t(D \to M), I(i-1,j-1)t(I \to M) \Big\}$$

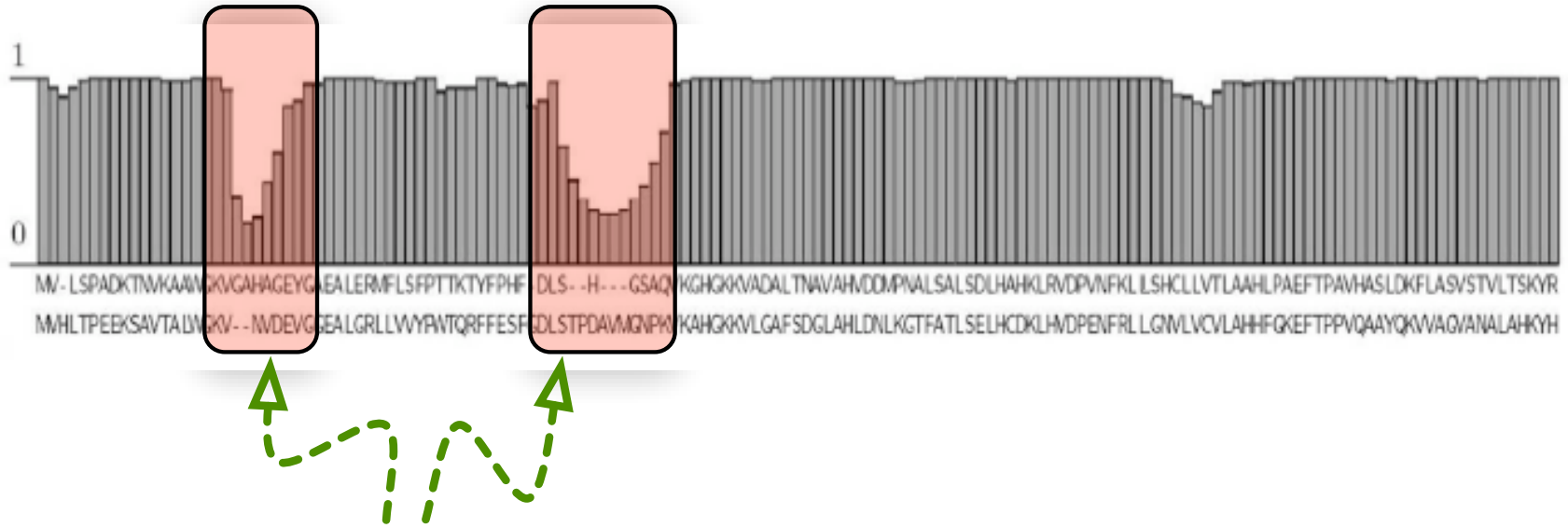
$$D(i,j) = p_D\begin{bmatrix} - \\ T[j-1] \end{bmatrix} \cdot \max\Big\{ M(i,j-1)t(M \to D), \\ D(i,j-1)t(D \to D), I(i,j-1)t(I \to D) \Big\}$$

$$I(i,j) = p_I\begin{bmatrix} S[i-1] \\ - \end{bmatrix} \cdot \max\Big\{ M(i-1,j)t(M \to I), \\ D(i-1,j)t(D \to I), I(i-1,j)t(I \to I) \Big\}.$$

+initialisation quand $i = 0$ ou $j = 0 \ldots$

Si on regarde les log de ces vraisemblances d'alignements, on arrive à une pondération de trous linéaire+pondération de substitutions linéaire +coûts de substitutions comme avant

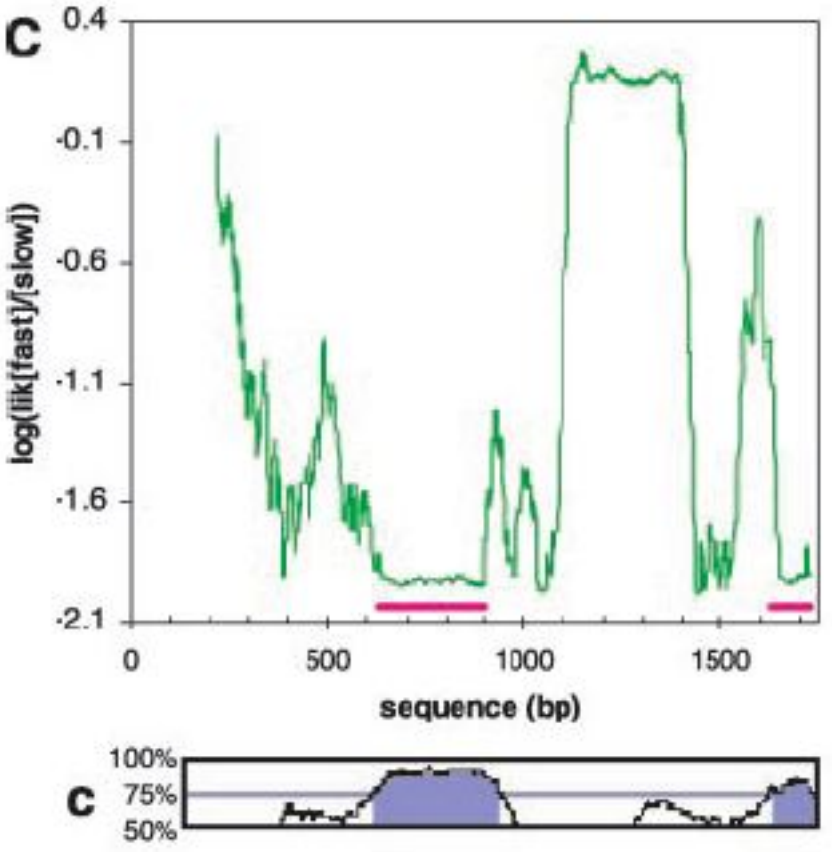
Probabilité postérieure



Lunter & al (2005)

Phylogenetic shadowing

Comparaison de séquences entre des espèces proches :
modèles d'évolution rapide/lente (HMM)



Boffelli & al. *Science* 299 :1391 (2003)