Profils d'alignement et HMM

HLIN608 Algorithmique du texte

sylvain.daude@umontpellier.fr
annie.chateau@umontpellier.fr

Alignements multiples

- Comment évaluer l'alignement de plusieurs séquences de même longueur n ?

Alignements multiples

- Comment évaluer l'alignement de plusieurs séquences de même longueur n ?
- En général : problème NP-complet $\longrightarrow$ nombreuses approches

Alignements multiples

- Comment évaluer l'alignement de plusieurs séquences de même longueur n ?
- En général : problème NP-complet $\longrightarrow$ nombreuses approches
- Approche des profils d'alignement

Alignements multiples

- Comment évaluer l'alignement de plusieurs séquences de même longueur n ?
- En général : problème NP-complet $\longrightarrow$ nombreuses approches
- Approche des profils d'alignement
  - on dispose d'un panel de référence de séquences similaires

Alignements multiples

- Comment évaluer l'alignement de plusieurs séquences de même longueur n ?
- En général : problème NP-complet $\longrightarrow$ nombreuses approches
- Approche des profils d'alignement
  - on dispose d'un panel de référence de séquences similaires
    - ex : protéines codant une même fonction biologique

Alignements multiples

- Comment évaluer l'alignement de plusieurs séquences de même longueur n ?
- En général : problème NP-complet $\longrightarrow$ nombreuses approches
- Approche des profils d'alignement
  - on dispose d'un panel de référence de séquences similaires
    - ex : protéines codant une même fonction biologique
  - on évalue la ressemblance de nouvelles séquences avec ce panel

Alignements multiples

- Comment évaluer l'alignement de plusieurs séquences de même longueur n ?
- En général : problème NP-complet $\longrightarrow$ nombreuses approches
- Approche des profils d'alignement
  - on dispose d'un panel de référence de séquences similaires
    - ex : protéines codant une même fonction biologique
  - on évalue la ressemblance de nouvelles séquences avec ce panel
    - objectif : ces nouvelles séquences codent-elles la même fonction ?

Alignements multiples

- Comment évaluer l'alignement de plusieurs séquences de même longueur n ?
- En général : problème NP-complet $\longrightarrow$ nombreuses approches
- Approche des profils d'alignement
  - on dispose d'un panel de référence de séquences similaires
    - ex : protéines codant une même fonction biologique
  - on évalue la ressemblance de nouvelles séquences avec ce panel
    - objectif : ces nouvelles séquences codent-elles la même fonction ?
  - étape 1 : calcul du "profil d'alignement" du panel de référence

Alignements multiples

- Comment évaluer l'alignement de plusieurs séquences de même longueur n ?
- En général : problème NP-complet $\longrightarrow$ nombreuses approches
- Approche des profils d'alignement
  - on dispose d'un panel de référence de séquences similaires
    - ex : protéines codant une même fonction biologique
  - on évalue la ressemblance de nouvelles séquences avec ce panel
    - objectif : ces nouvelles séquences codent-elles la même fonction ?
  - étape 1 : calcul du "profil d'alignement" du panel de référence
  - étape 2 : calcul du score d'alignement de la séquence candidate sur le profil

Calcul du profil d'alignement

- Exemple : calculer le profil l'alignement du panel

$$
\begin{array}{cccccc}
G & A & T & T & C & A \\
G & - & C & T & - & A \\
G & A & T & T & - & T \\
G & - & - & T & C & -
\end{array}
$$

Calcul du profil d'alignement

- Exemple : calculer le profil l'alignement du panel

  $G \quad A \quad T \quad T \quad C \quad A$
  $G \quad - \quad C \quad T \quad - \quad A$
  $G \quad A \quad T \quad T \quad - \quad T$
  $G \quad - \quad - \quad T \quad C \quad -$

- alphabet de l'alignement : $\Sigma = \{G\ A\ T\ C\ -\}$

Calcul du profil d'alignement

- Exemple : calculer le profil l'alignement du panel

  $G \quad A \quad T \quad T \quad C \quad A$
  $G \quad - \quad C \quad T \quad - \quad A$
  $G \quad A \quad T \quad T \quad - \quad T$
  $G \quad - \quad - \quad T \quad C \quad -$

- alphabet de l'alignement : $\Sigma = \{G\ A\ T\ C\ -\}$
- profil d'alignement = matrice $|\Sigma| \times n$

Calcul du profil d'alignement

- Exemple : calculer le profil l'alignement du panel

  $G \quad A \quad T \quad T \quad C \quad A$
  $G \quad - \quad C \quad T \quad - \quad A$
  $G \quad A \quad T \quad T \quad - \quad T$
  $G \quad - \quad - \quad T \quad C \quad -$

- alphabet de l'alignement : $\Sigma = \{G\ A\ T\ C\ -\}$
- profil d'alignement = matrice $|\Sigma| \times n$
  - les lignes correspondent aux symboles de l'alphabet

Calcul du profil d'alignement

- Exemple : calculer le profil l'alignement du panel

  $G\ A\ T\ T\ C\ A$
  $G\ -\ C\ T\ -\ A$
  $G\ A\ T\ T\ -\ T$
  $G\ -\ -\ T\ C\ -$

- alphabet de l'alignement : $\Sigma = \{G\ A\ T\ C\ -\}$
- profil d'alignement = matrice $|\Sigma| \times n$
  - les lignes correspondent aux symboles de l'alphabet
  - chaque case correspond à un symbole et à une colonne d'alignement

Calcul du profil d'alignement

- Exemple : calculer le profil l'alignement du panel

  $G\ A\ T\ T\ C\ A$
  $G\ -\ C\ T\ -\ A$
  $G\ A\ T\ T\ -\ T$
  $G\ -\ -\ T\ C\ -$

- alphabet de l'alignement : $\Sigma = \{G\ A\ T\ C\ -\}$
- profil d'alignement = matrice $|\Sigma| \times n$
  - les lignes correspondent aux symboles de l'alphabet
  - chaque case correspond à un symbole et à une colonne d'alignement
  - elle contient le taux d'apparition (entre 0 et 1) du symbole dans la colonne

Résultat du calcul

- Profil d'alignement :

	G	A	T	T	C	A
	G	-	C	T	-	A
	G	A	T	T	-	T
	G	-	-	T	C	-
G	1	0	0	0	0	0
A	0	0,5	0	0	0	0,5
T	0	0	0,5	1	0	0,25
C	0	0	0,25	0	0,5	0
-	0	0,5	0,25	0	0,5	0,25

Calcul du score d'alignement sur le profil

- alignement d'une séquence S sur un profil M (de même longueur) ?

Calcul du score d'alignement sur le profil

- alignement d'une séquence S sur un profil M (de même longueur) ?
  - le symbole $S[i]$ obtient le score $M[S[i], i]$

Calcul du score d'alignement sur le profil

- alignement d'une séquence S sur un profil M (de même longueur) ?
  - le symbole $S[i]$ obtient le score $M[S[i], i]$
  - la somme des scores obtenus donne le score de S

Calcul du score d'alignement sur le profil

- alignement d'une séquence S sur un profil M (de même longueur) ?
  - le symbole $S[i]$ obtient le score $M[S[i], i]$
  - la somme des scores obtenus donne le score de S
- ex : CGTTCG, GACCAT

G	1	0	0	0	0	0
A	0	0,5	0	0	0	0,5
T	0	0	0,5	1	0	0,25
C	0	0	0,25	0	0,5	0
-	0	0,5	0,25	0	0,5	0,25
	C	G	T	T	C	G
	0	0	0,5	1	0,5	0
	G	A	C	C	A	A
	1	0,5	0,25	0	0	0,5

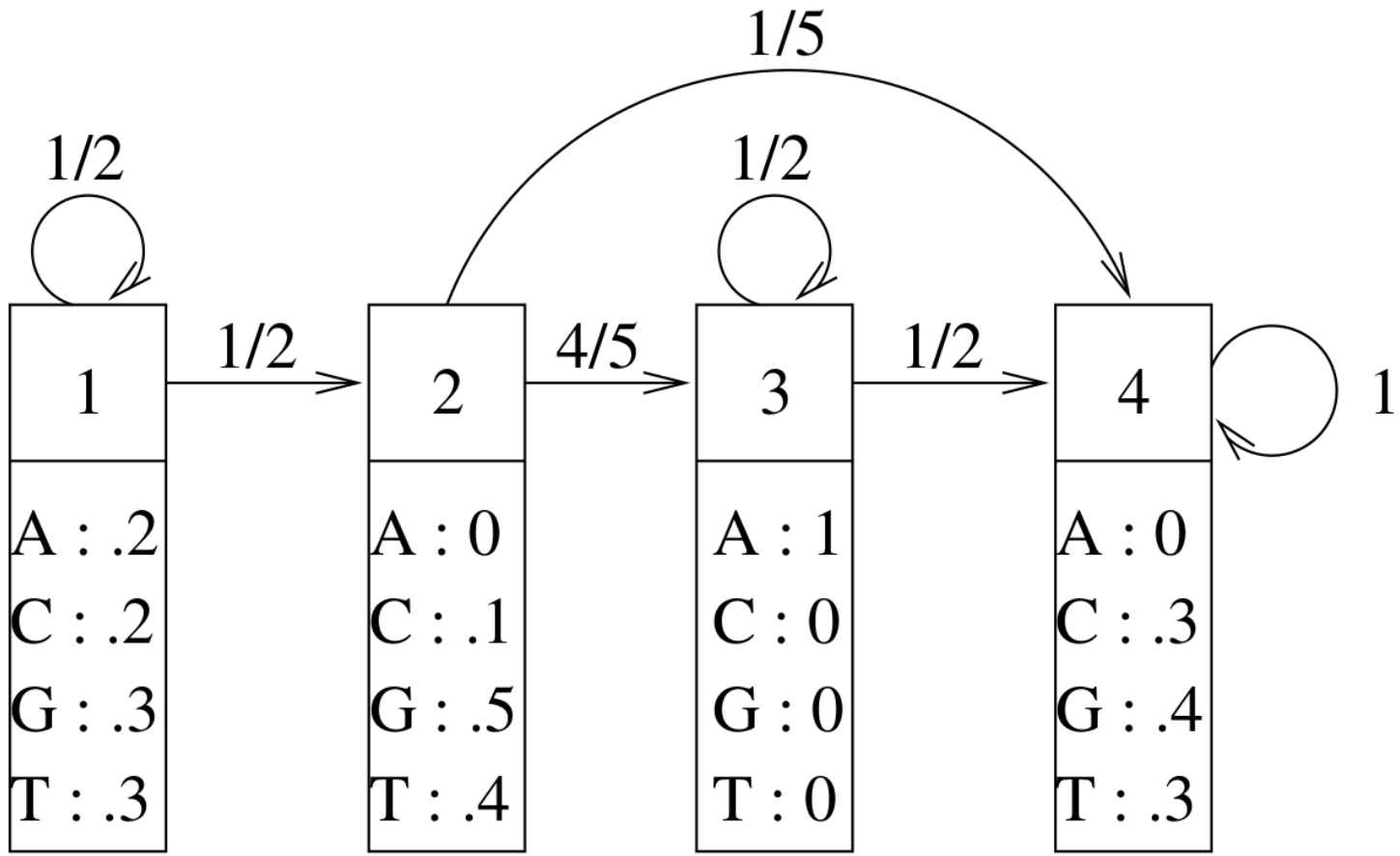
Total : 2

Total : 2,25

Une autre approche, probabiliste : Les chaînes de Markov cachées

Objectif : Représenter et modéliser une famille de séquences.

H =



Structure d'un HMM

HMM : Hidden Markov Model

Une séquence émise par ce modèle probabiliste
W = CGAAAC

Éléments de ce modèle :

- $\mathcal{A}$: alphabet $= \{A, C, G, T\}$; $|\mathcal{A}| = m$
- $\mathcal{S}$: états (sommets) $= \{1, 2, 3, 4\}$; $|\mathcal{S}| = n$
- T : matrice des probabilités de transition
- E : matrice des probabilités d'émission
- Π : vecteur des probabilités de départ

Structure d'un HMM

Matrices de transition, émission, initialisation

	1	2	3	4
1	1/2	1/2	0	0
2	0	0	4/5	1/5
3	0	0	1/2	1/2
4	0	0	0	1

$T = (t_{i,j})_{n \times m}$

	A	C	G	T
1	.2	.2	.3	.3
2	0	.1	.5	.4
3	1	0	0	0
4	0	.3	.4	.3

$E = (e_{i,j})_{n \times m}$

1	1
2	0
3	0
4	0

$\Pi = (\pi_i)_n$

Structure d'un HMM

$$\forall i, \sum_{j=1}^{n} t_{i,j} = 1 : \text{on bouge à coup sûr}$$

$$\forall i, \sum_{s \in \mathcal{A}} e_{i,s} = 1 : \text{on émet à coup sûr}$$

Émission et transition indépendantes du chemin parcouru

Marche : déplacement dans le graphe avec émission d?un symbole à chaque sommet
d'où : une marche engendre un mot sur $\mathcal{A}$

mais il y a plusieurs marches possibles pour un mot :

w		A	C	C	A	C	C
M_1	1	1	1	2	3	4	4
M_2	2	1	1	1	1	2	4

Caché : l'observateur ne voit que la séquence et non la marche

Structure d'un HMM

Vraisemblance d'une marche : $Prob(w, M|H)$

w		A	C	C	A	C	C
M_1	$\boxed{1}$	1	1	2	3	4	4
M_2	$\boxed{2}$	1	1	1	1	2	4

$$\boxed{1} = 1/2 \cdot 1/2 \cdot 4/5 \cdot 1/2 \cdot 1 \cdot (0.2 \cdot 0.2 \cdot 0.2 \cdot 1 \cdot 0.3 \cdot 0.4) = 4/5 \cdot 0.3$$

$$\boxed{2} = 1/2 \cdot 1/2 \cdot 1/2 \cdot 1/2 \cdot 1/5 \cdot (0.2 \cdot 0.2 \cdot 0.2 \cdot 0.2 \cdot 1 \cdot 0.4) = 1/10 \cdot 0.02$$

Problèmes autour des HMM

HMM = modèle probabiliste "capturant" les propriétés d'une famille de séquences ET outil de production (émission) de séquences (toute marche M produit une séquence w)

$\Rightarrow$ Informations importantes pour M et/ou w

$Prob(w, M|H)$: vraisemblance que M engendre w vis-à-vis du modèle H

$Prob(w|H)$: vraisemblance de la séquence w vis-à-vis du modèle H

Problèmes autour des HMM

- Évaluation : étant donnés H et w , calculer $Prob(w|H)$
- Décodage : étant donnés H et w, calculer M tel que $Prob(w, M|H)$ est maximale
- Apprentissage : étant donnés H et une famille $\mathcal{F}_0$ de séquences, ajuster les paramètres E , T , et ? de H pour maximiser la vraisemblance des séquences de $\mathcal{F}_0$

Évaluation : l'algorithme FORWARD

$$
\begin{aligned}
Prob(w|H) &= \sum_{M=q_1,\ldots,q_l} Prob(w, M|H) \\
&= \sum_{M=q_1,\ldots,q_l} \left\{ \prod_{i=1}^{l-1} t_{q_i,q_{i+1}} \times \prod_{i=1}^{l} e_{q_i,w_i} \right\}
\end{aligned}
$$

$\Rightarrow$ nombre exponentiel de marches !

Le tableau des $\alpha_i(j)$: $\alpha_i(j)$ = probabilité qu'une marche se terminant en l'état j produise le préfixe $w_1 \ldots w_i$

$\Rightarrow P(w|H = \sum_{j \in \mathcal{S}} \alpha_l(j))$

Évaluation : l'algorithme FORWARD

Calcul des $\alpha_i(j)$:

$$\alpha_1(j) = \pi_j \times e_{j,w_1}$$

$$\alpha_{i+1}(j) = e_{j,w_{i+1}} \times \sum_{k \in \mathcal{S}} \alpha_i(k) \times t_{k,j}$$

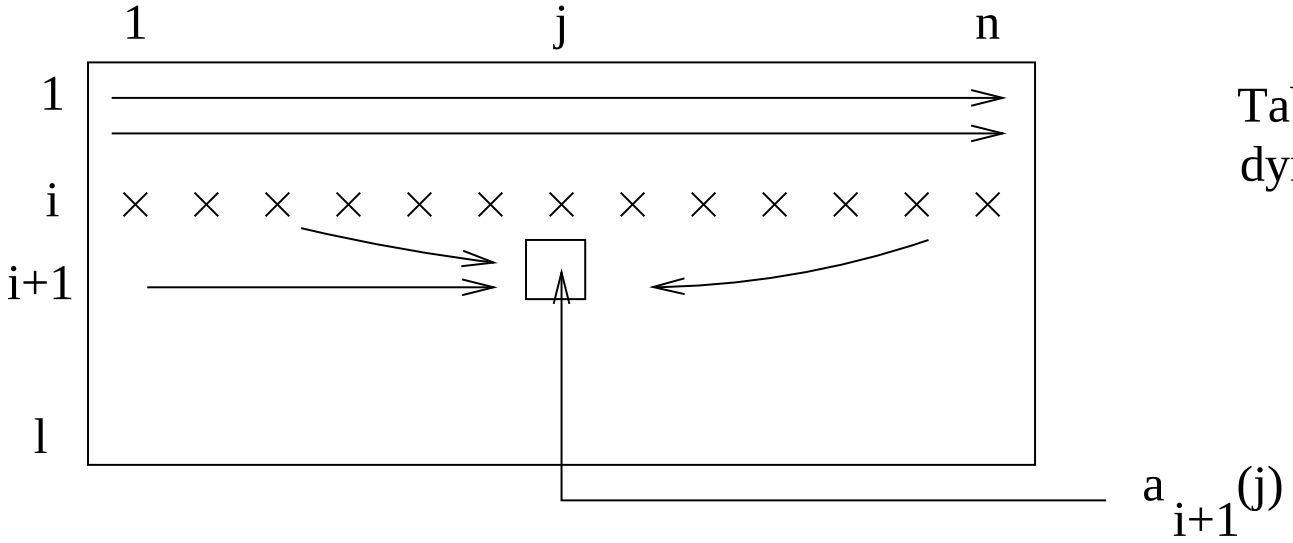
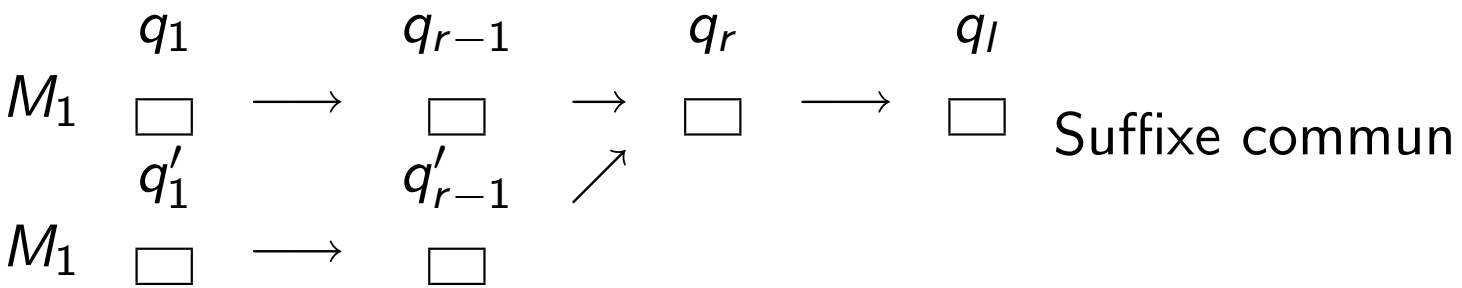


Tableau de programmation dynamique classique

Évaluation : l'algorithme FORWARD

Principe permettant d'utiliser la programmation dynamique



Évaluation : l'algorithme FORWARD

$$Prob(w, M_1|H) = \left\{ \prod_{i=1}^{r-1} t_{q_i,q_{i+1}} \times e_{q_i,w_i} \right\} \times \{e_{q_{r+1},w_{r-1}}\} \times$$

$$\left\{ \prod_{i=r}^{l-1} t_{q_i,q_{i+1}} \times e_{q_i,w_i} \right\} \times \{e_{q_l,w_l}\}$$

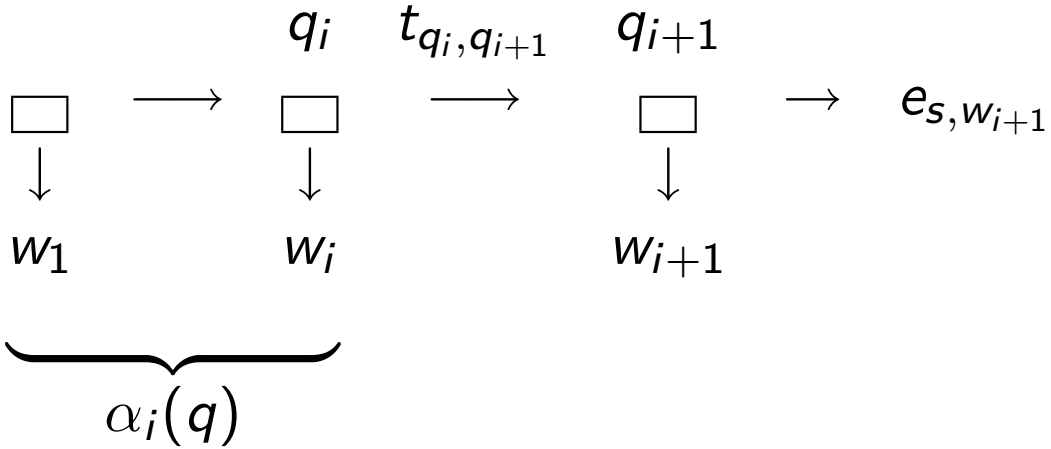
$$Prob(w, M_2|H) = \left\{ \prod_{i=1}^{r-1} t_{q'_i,q'_{i+1}} \times e_{q'_i,w_i} \right\} \times \{e_{q'_{r+1},w_{r-1}}\} \times$$

$$\left\{ \prod_{i=r}^{l-1} t_{q_i,q_{i+1}} \times e_{q_i,w_i} \right\} \times \{e_{q_l,w_l}\}$$

$\Rightarrow$ la partie du calcul correspondant au « coût » du suffixe $(q_r \ldots q_l)$ n'a besoin d'être calculée qu'une seule fois pour toutes les marches ayant ce suffixe.

Évaluation : l'algorithme FORWARD

1 Principe :



2 Complexité : calcul du tableau des α_i
espace : $l \times n$
temps : $\mathcal{O}(l \times n^2)$

Évaluation : l'algorithme FORWARD

3 Algorithme BACKWARD : $\beta_i(s)$ = vraisemblance de $w_1 \ldots w_i$ pour une marche débutant en s

Calcul similaire

Décodage

Décodage

Décodage : l'algorithme de Viterbi

- Algorithme forward : $P(w|H) = \sum_{M} Prob(w, M|H)$

  Décodage : vraisemblance max. d'une marche

  $\max_{M}\{Prob(w, M|H)\}$

Décodage : l'algorithme de Viterbi

- $\delta_i(j) = \max_{\text{marches } q_1 \ldots q_i = j} \{Prob(w_1 \ldots w_i, q_1 \ldots q_i | H)\}$

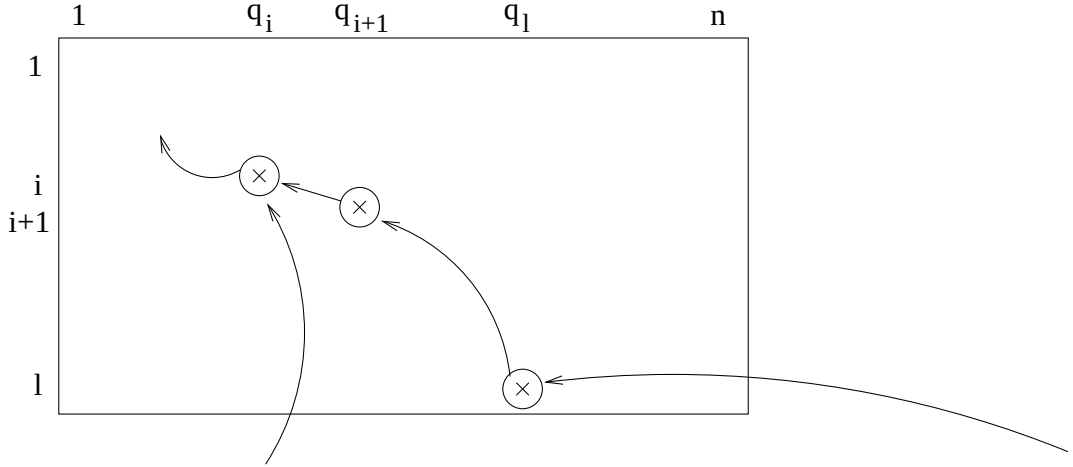
$\Rightarrow \max_M \{Prob(w, M|H)\} = \max_{j=1\ldots n} \{\delta_I(j)\}$ (*)

$\delta_1(j) = \pi_j \times e_{j,w_1}$

$\delta_{i+1}(j) = e_{j,w_{i+1}} \times \max_{k \in \mathcal{S}} \{\delta_i(k) t_{k,j}\}$

Décodage : l'algorithme de Viterbi

- (*) = vraisemblance d'une marche optimale : pour retrouver la marche, backtracking



$\delta_{i+1}(q_{i+1}) = e_{q_{i+1},w_{i+1}} \times \delta_i(q_i) \times t_{q_i,q_{i+1}}$ $\delta_l(q_l)$ max sur la ligne l

critère de choix de q_i connaissant $q_i + 1$

Décodage : l'algorithme de Viterbi - Commentaires

1. On prend l'algorithme Forward (tableau δ) et on remplace $\sum$ par max
2. Backtracking : comme pour l'alignement de séquences :
   - calcul tableau $\Rightarrow$ score
   - backtracking $\Rightarrow$ alignement
3. Complexité :
   - calcul de δ : $\mathcal{O}(l \times n^2)$
   - backtracking : $\mathcal{O}(l \times n)$

Apprentissage

Apprentissage

Apprentissage : Algorithme Baum-Welch

- Données : H et w

  But : optimiser $\sum_M Prob(M|w, H)$ (*) en modifiant les paramètres de H

- $\gamma_{i,k}$ = probabilité que l'état i émette w_k parmi toutes les marches engendrant w

  $\gamma_{i,j,k}$ = probabilité que i émette w_k et j émette w_{k+1} parmi toutes les marches engendrant w

Apprentissage : Algorithme Baum-Welch

- Algorithme : Expectation-Maximization (EM)

Répéter

Calculer les $\gamma_{i,k}$ et $\gamma_{i,j,k}$

En déduire Π', T', E' :

$$\Pi'_i = \gamma_{i,1} \qquad t'_{i,j} = \frac{\sum_{k=1}^{l-1} \gamma_{i,j,k}}{\sum_{k=1}^{l-1} \gamma_{i,k}} \qquad e'_{i,c} = \frac{\sum_{k=1,w_k=c}^{l} \gamma_{i,k}}{\sum_{k=1}^{l} \gamma_{i,k}}$$

Tant que la différence entre $Prob(w|H')$ et $Prob(w|H)$ est $\geq \epsilon$

Apprentissage : Algorithme Baum-Welch

(*) $\Rightarrow$ on a besoin d'informations sur toutes les marches engendrant w

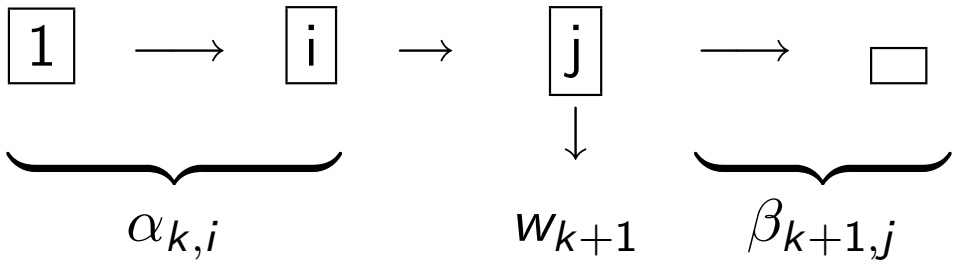
Une fois connues $\gamma_{i,k}$ et $\gamma_{i,j,k}$, on sait quelles transitions et émissions on doit optimiser pour augmenter la vraisemblance de w

Calcul des $\gamma_{i,k}$ et $\gamma_{i,j,k}$:

$$\gamma_{i,j,k} = \frac{\alpha_{k,i} \times t_{i,j} \times \beta_{k+1,j} \times e_{j,w_{k+1}}}{Prob(w|H) \sum_{s_1,s_2 \in \mathcal{S}} \alpha_{k,s_1} \times t_{s_1,s_2} \times \beta_{k+1,s_2} \times e_{s_2,w_{k+1}}}$$

$\gamma_{i,k} = \sum_{j \in \mathcal{S}} \gamma_{i,j,k} \Rightarrow$ calculable avec les α et β

Apprentissage : Algorithme Baum-Welch



Variante : Viterbi. Utiliser δ au lieu de $\alpha \Rightarrow$ optimiser sur la meilleure marche

Remarque : c'est une approximation (l'optimal n'est pas calculable) mais en temps polynomial. Maximum local

$\overrightarrow{w}$: multiplier les vraisemblances

Résumé

F : famille de séquences $\longrightarrow$ HMM H_F capturant les propriétés de F
Apprentissage + design initial

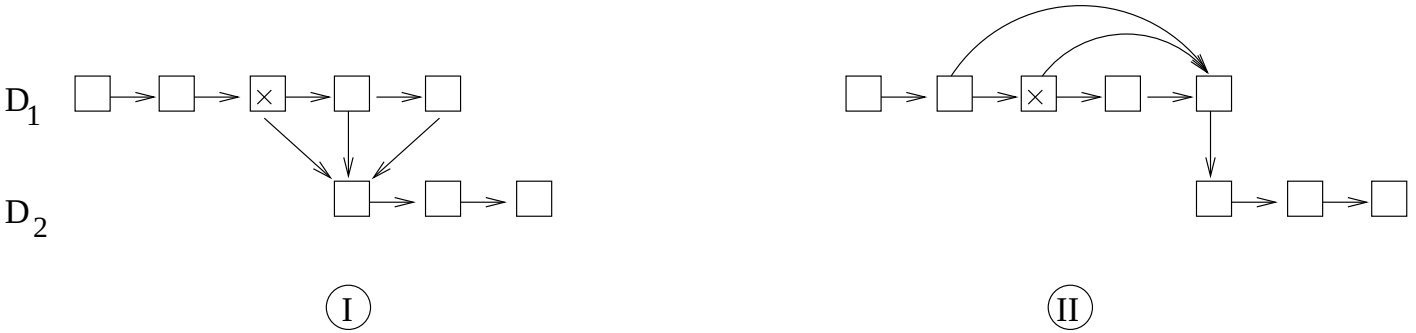
$w + H_F$

Forward Vraisemblance que le modèle H_F ait engendré w = « w ressemble-t-elle aux séquences de F ? »

Viterbi Marche la plus vraisemblable engendrant w = structure de w par rapport à ce qu'on sait de F
$\Rightarrow$ annotation automatique de domaines de protéines

Compléments techniques

1 Importance de l'architecture du modèle (forme du graphe)



Dans les deux cas : D_1 est de longueur 3 à 5, et D_2 est de longueur 3

Données pour entraîner le modèle : pas de A en position 3

$$\Rightarrow \begin{cases} A : 0 & \text{en I.X} \\ A :\neq 0 & \text{en II.X} \end{cases} \quad \text{à coup sûr}$$

Compléments techniques

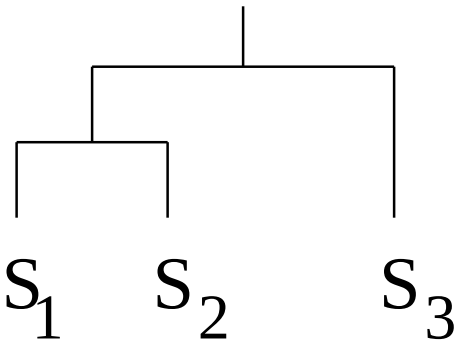
2 D'où l'importance des paramètres de départ (Π, T, E avant la phase d'apprentissage) : on a 3 choix

- basé sur la connaissance de F
- aléatoire
- distribution fixée (Dirichlet, Gaussienne, ...)

3 et l'importance dans la constitution du jeu d'apprentissage : éviter un biais vers une sous-famille de F dû à une mauvaise constitution de ce jeu

Compléments techniques

4 Pondération des séquences du jeu d'apprentissage



$$pds(S_1) = pds(S_2) = 1/2 pds(S_3)$$

$\Rightarrow$ permet de conserver de nombreuses séquences dans le jeu d'apprentissage sans privilégier une sous-famille

Compléments techniques

5 Scorer une séquence w par rapport à H : log-likelihood ratio-test

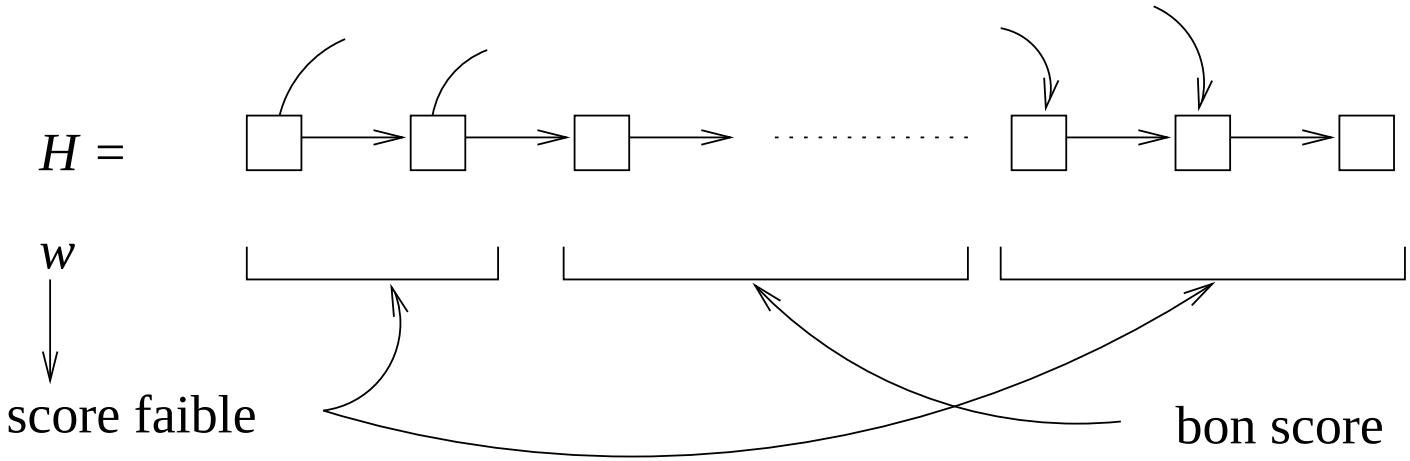
$$\frac{\log(Prob(w|H))}{\log(\text{score modèle nul})}$$

: $\log\left(\frac{1}{|\mathcal{A}|^l}\right) = l \times \log\left(\frac{1}{|\mathcal{A}|}\right)$

$l =$ longueur de w

Compléments techniques

6 Marche globale vs. marche locale



$\Rightarrow$ rechercher la sous-séquence de w maximisant son score local (algo « à la Viterbi »)

Compléments techniques

7 Pseudo-counts

Jeu d'entraînement : N séquences

A ...
A ...
C ...
C ...
G ...

$\Rightarrow$pas de T en position 1, mais ce fait est juste dû à la composition du jeu de séquences et non à la nature biologique de ces séquences

Toute séquence T... aura un score 0 : problème !

Compléments techniques

Idée : pour chaque état i et $c \in \mathcal{A}$:

si $e_{i,c} \neq 0$: $e'_{i,c} = \frac{N}{N+1} e_{i,c}$

si $e_{i,c} = 0$: $e'_{i,c} = \frac{1}{k(N+1)}$

où $K =$ nombre de symboles c tels que $e_{i,c} = 0$

Principe des « pseudo-counts »

Compléments techniques

8 Classification : choix du score d'acceptation

w est reconnue comme appartenant à F si

$$-\log(Prob(w|H)) \geq -\log(\sigma) + \log(N)$$

$N =$ taille de la base de données examinée

σ à choisir : si σ augmente, les faux positifs augmentent

Alignement et HMM

```
A C A - - - A T G
T C A A C T A T C
A C A C - - A G C
A G A - - - A T C
A C C G - - A T C
  ↓expression
```

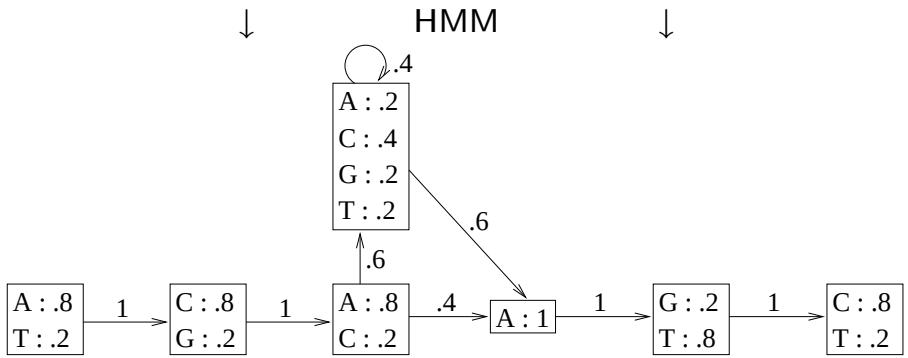
$\simeq PSSM \longrightarrow$

	1	2	3	4	5	6	7	8	9
A	.8	0	.8	.2	0	0	1	0	0
C	0	.8	.2	.2	.2	0	0	0	.8
G	0	.2	0	.2	0	0	0	.2	.2
T	.2	0	0	0	0	.2	0	.8	0
-	0	0	0	.4	.8	.8	0	0	0

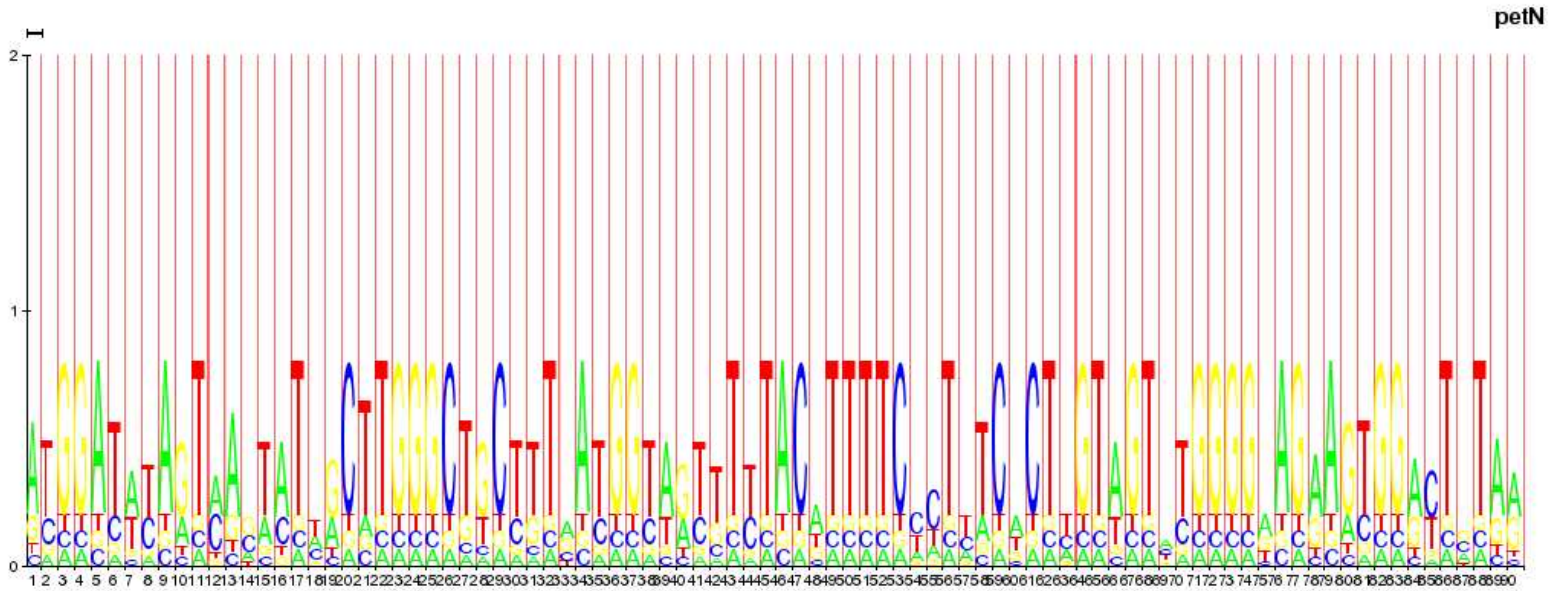
région mal alignée

régulière

$[AT][CG][AC][ACGT]^*A[TG][GC]$



HMMlogo



Validation d'un HMM

Terminologie

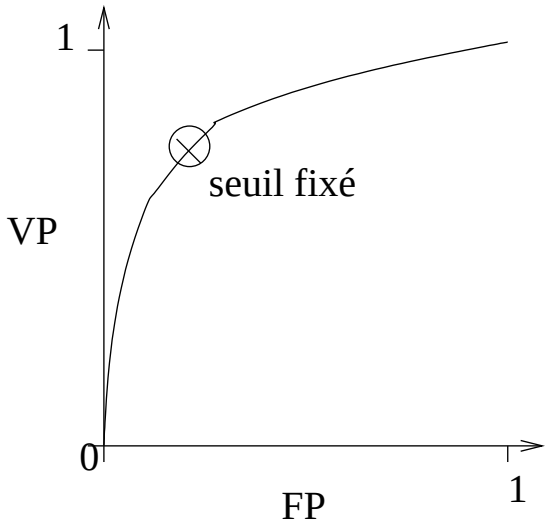
- Faux positif : $S \notin$ famille mais prédiction : $S \in$ famille
- Faux négatif : $S \in$ famille mais prédiction : $S \notin$ famille
- Vrais positifs / Vrais négatifs
- Sensibilité de la prédiction : VP/(VP+FN) idéal 100%
  Prédit-on tous les membres de la famille ?
- Spécificité de la prédiction : VN/(VN+FP) idéal 100%
  Fait-on de mauvaises prédictions positives ?

Remarque : on peut trouver des variations de ces formules, mais le concept reste le même. Exemple pour la spécificité : FP/(VN+FP)

Choix du seuil

Choix empirique : regarder la distribution des scores, prendre le minimal qui donne 0 FN, prendre le score qui laisse 5% FN...

ROC (Recieving Operating Characteristics)



Aire sous la courbe, Distance au point (0,1) : permet d'aider au choix du seuil, de comparer des modèles

Une fois choisi le prédicteur : est-il bon ?