Vendredi 13 février 2004

Démo 5 – HMM et profiles

Jean-Eudes Duchesne

0. Similarité vs Spécificité

Pour juger la qualité d'un programme de recherche spécifique ou d'un profile (recherche générale), il faut se définir des paramètres représentant la qualité des résultats. Dans les définitions qui suivent une occurrence est un élément trouvé (*in silico* par notre programme ou profile) de la classe recherchée.

Vrais positifs : Occurrences qui ont été trouvées. Représentent les éléments spécifiques de la classe recherchée. Par exemple, si l'on possède un profile pour l'ARNt (ARN de transfert) et qu'il est utilisé pour rechercher des ARNt dans le génome humain, les éléments trouvés qui correspondent à de vrais ARNt sont des vrais positifs.
Faux positifs : Éléments trouvés qui ne sont pas des occurrences de ce qui est recherché. Ce sont des erreurs engendrées par le programme ou le profile, possiblement parce que les contraintes sont trop larges.
Faux négatifs : Occurrences non trouvées, possiblement parce que les contraintes sont trop sévères.
Vrais négatifs : Le reste, ce qui n'a pas été considéré et n'aurait pas dû l'être de toute façon.

Un bon modèle devra donc maximiser le nombre de vrais positifs et minimiser les faux positifs ainsi que les faux négatifs.

Sensibilité : Trouvons-nous tous les éléments que nous devons trouver ? (VP/ VP + FN)
Spécificité : Trouvons-nous tous les éléments que nous sommes supposer trouver ? (VP/VP + FP)

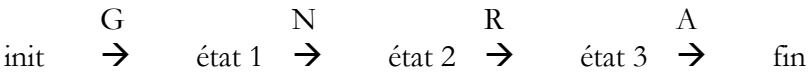
Le dilemme de la sensibilité vs la spécificité est que si l'on augmente les contraintes d'un modèle, on augmente aussi la spécificité mais on peut réduire la sensibilité et inversement si l'on assouplit les contraintes. Il faut souvent beaucoup d'expérimentation avant d'obtenir le bon équilibre entre les paramètres d'un modèle.

1. Signatures et automates finis

Une signature ressemble à une expression régulière. On peut donc la rechercher avec un automate fini.

Ex : codes IUPAC

GNRA -> G(A|C|G|T)(A|G)A



IUPAC-IUB SYMBOLS FOR NUCLEOTIDE NOMENCLATURE: Cornish-Bowden (1985) Nucl. Acids Res. 13: 3021-3030.

Symbol	Meaning	Nucleic Acid
A	A	Adenine
C	C	Cytosine
G	G	Guanine
T	T	Thymine
U	U	Uracil
M	A or C	
R	A or G	
W	A or T	
S	C or G	
Y	C or T	
K	G or T	
V	A or C or G	
H	A or C or T	
D	A or G or T	
B	C or G or T	
X	G or A or T or C	
N	G or A or T or C	

Ex : Signature prosite

La syntaxe prosite est définie comme suit :

- - est un séparateur de position. Chaque – représente une nouvelle position.
- [ ] sert à indiquer que n'importe lequel des différents acides aminés inclus entre les [ ] peut se retrouver à cette position.
- {} est le contraire du cas précédent. Les acides aminés inclus entre {} ne peuvent pas se retrouver à cette position.
- x représente n'importe lequel des acides aminés.
- x(N) où N est un entier signifies que l'on retrouve N fois de suite n'importe lequel des acides aminés.
- (n,m) représente que la position précédente peut se répéter de n (minimum) à m (maximum) fois.

Donc, avec la signature suivante :

N-[PT]-{GM}-x(2)-[ILVM]

→ N-P-K-G-H-V et N-T-L-K-G-M seront trouvées.

→ N-L-K-G-H-V et N-T-G-K-H-V ne le sont pas.

Problèmes ? Pas de possibilité d'erreurs non incluses dans le modèle (substitutions et indels), pas d'informations sur la fréquence des nucléotides.

2. Modèle de Markov : la chaîne de Markov

Problème : Beau temps, mauvais temps !

États :
1. Pluvieux (P)
2. Nuageux (N)
3. Ensoleillé(E)

Matrice des probabilités de transition d`états **:**

États	P	N	E
P	0.4	0.3	0.3
N	0.2	0.6	0.2
E	0.1	0.1	0.8

Question **:**

Calculer la probabilité d'observer une séquence EEPPENE sachant qu'aujourd'hui c'est ensoleillé donc état E.

Règle de probabilité de base : $P(A,B) = P(A|B) * P(B)$

La règle de Markov :

$$
\begin{aligned}
P(q1,q2,\ldots,qT) &= P(qT|q1, q2, \ldots qT\text{-}1)P(q1,q2,\ldots qT\text{-}1) \\
&= P(qT|qT\text{-}1)P(q1,q2,\ldots qT\text{-}1) \\
&= P(qT|qT\text{-}1)P(qT\text{-}1|qT\text{-}2)\ P(q1,q2,\ldots qT\text{-}2) \\
&= P(qT|qT\text{-}1)P(qT|qT\text{-}2)\ldots P(q2|q1)P(q1)
\end{aligned}
$$

Soit l'observation O, O = (E,E,E,P,P,E,N,E), en utilisant la règle on obtient :

$$
\begin{aligned}
P(O|\text{modèle}) &= P(E,E,E,P,P,E,N,E|\text{modèle}) \\
&= P(E)\ P(E|E)\ P(E|E)\ P(P|E)\ P(P|P)\ P(E|P)\ P(N|E)\ P(E|N) \\
&= (1)\ (0.8)\ (0.8)\ (0.1)\ (0.4)\ (0.3)\ (0.1)\ (0.2) \\
&= 1.536 * 10\ \text{-}4
\end{aligned}
$$

3. Modèle de Markov Caché (HMM)

Comme le modèle de Markov, mais on ajoute la possibilité d'émettre un caractère associé à une probabilité à l'intérieur même d'un état. Dans le cas de l'exemple précédent, on pourrait ajouter les états E+ et E-, par exemple, pour représenter une certaine qualité de l'ensoleillement (E+ signifie très ensoleillé et E- peu ensoleillé).

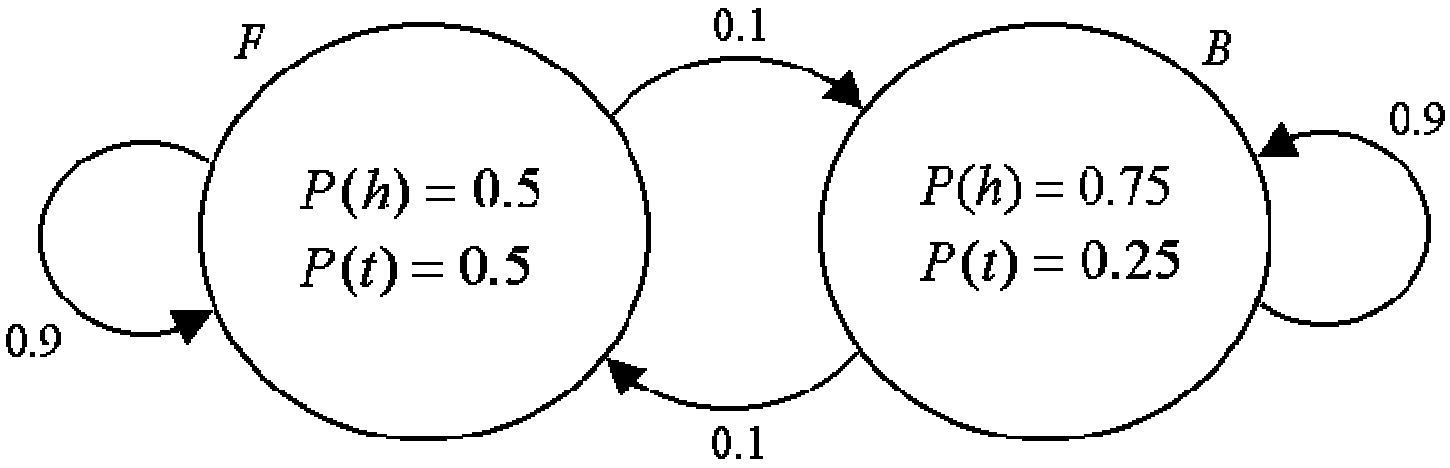
Problème : Le croupier véreux !

Admettons un croupier au Casino de Montréal dont le rôle consiste à lancer un sou. Nous savons que le croupier utilise deux sous, l'un est normal (probabilité de 50/50) mais l'autre est pipé (merci M. Duncan !!!) et tombe sur « face » 3 fois sur 4. Le croupier change de sou avec une probabilité de 0.1. Étant donné une séquence de coups, nous voulons déterminer s'il est plus probable que le sou normal ou que le sou pipé ait été utilisé.

Modèle HMM :

- Nous avons une liste d'états Q = {F,B} où F est pour « fair » (normal) et B est « biased » (pipé).
- L'alphabet est Σ = {h,t} où h est « heads » (face) et t est « tails » (pile)
- L'esemble des probabilitées est :
    - $a_{FF} = a_{BB} = 0.9$
    - $a_{FB} = a_{BF} = 0.1$
    - $e_F(h) = e_F(t) = 0.5$
    - $e_B(h) = 0.75$ et $e_B(t) = 0.25$

Figure 1: HMM pour le problème du croupier véreux



Comme pour le cas du modèle de Markov, nous avons définitivement une probabilité P(O|modèle) pour une séquence observée O et un modèle (tel qu'énoncé). Par contre, nous ne connaissons pas la séquence des états parcourus qui ont émis les caractères observés dans O. C'est pour ceci que nous disons que le chemin générateur de O est caché ! Alors comment déterminer le chemin parcouru ? Une solution : les chemins de Viterbi !

Chemins de Viterbi **:**

Le but est de calculer le chemin optimal qui obtient une observation donnée.

Initialisation :

$v_{begin} = 1$ (les états initiaux ont tous une probabilité de 1)
$\forall\ k \neq begin, v_k = 0$ (les états non initiaux ont tous une prabilité de 0)

Règle de récurrence :

$v_l(i) = e_l(x_i) * \max_{k \in Q}\{v_k(i-1) * a_{kl}\}$ où :

- k et l sont elements de Q
- l est l'état courant et k un état qui mène à l (peut être lui-même)
- e_l est la probabilité d'émission associées à l'état l
- a_{kl} est la probabilité associée à la transition de k à l

Réponse optimale :

La valeur optimale est trouvée dans la dernière colonne de la table de programmation dynamique. Il suffit de trouver la valeur maximale.

Ex :

O = FFPFF

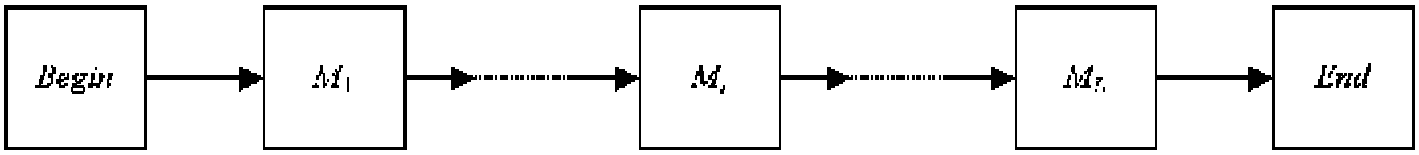
P	1	.675	0.456	0.103	0.07	**0.047**
N	1	.45	0.203	0.091	0.041	0.018

En suivant le chemin de Viterbi il est possible d'affirmer qu'il est plus probable que le croupier est utilise le sou pipé pour obtenir l'observation donnée.

Profiles HMM

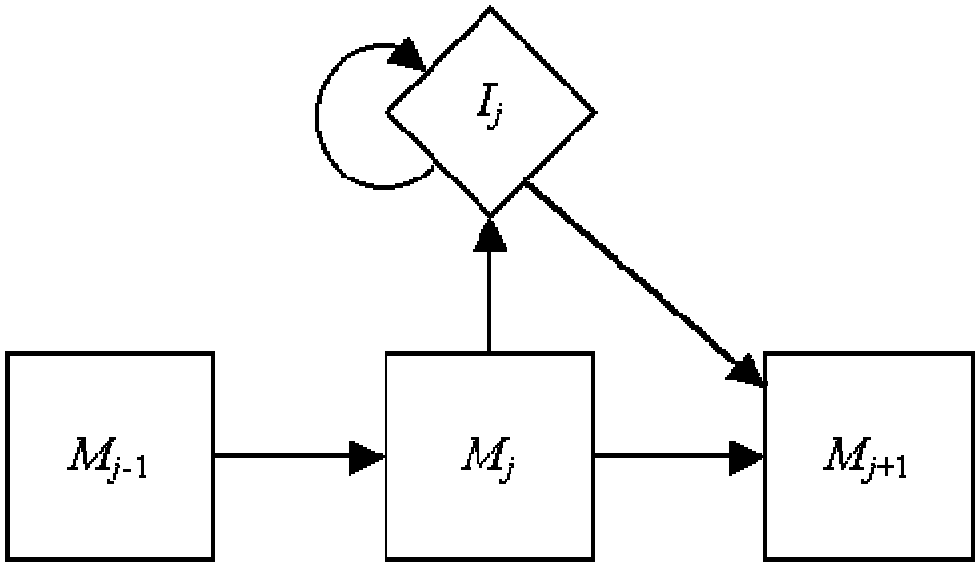
Les HMM sont intéressants, mais pour être utiles dans un contexte biologique, il est plus utile de les modifier pour l'alignement à un profile. Un profile HMM est un HMM où chaque état représente un « match » avec le profile. Donc à la base, un HMM aura la forme suivante :

Figure 2: États représentants les matchs/substitutions dans un HMM de profile.



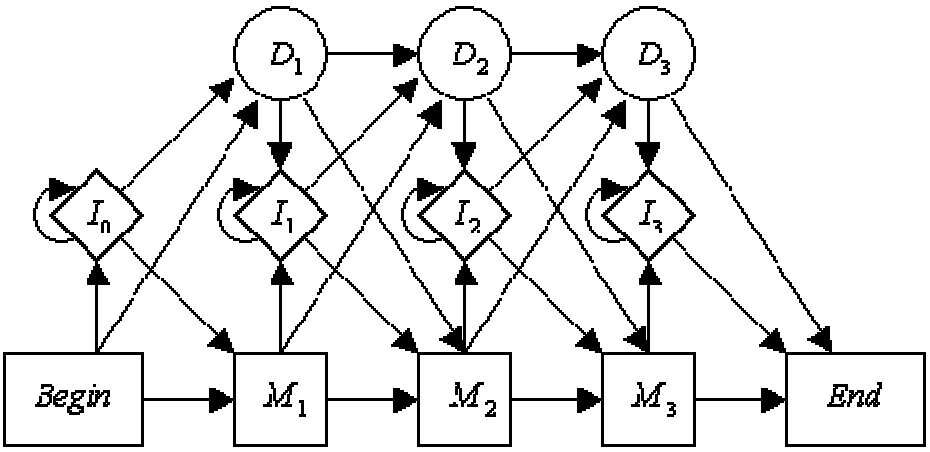
Maintenant que notre HMM est capable de représenter les matchs/substitutions d'un profile donné, augmentons le modèle en permettant les insertions à l'intérieur de la séquence à aligner au profile HMM.

Figure 3: Un HMM de profile avec insertions.



Finalement, un HMM de profile ne pourrait être complet sans permettre la délétion d'un état de l'automate. L'addition de tous ces éléments donne un HMM capable de simuler l'alignement global.

Figure 4: HMM de profile pour l'alignement global.



Et l'alignement local ?

Figure 5: HMM de profile pour l'alignement local.