

Exercices sur la recherche de motifs et les modèles de Markov – correction

1. Motifs exacts : algorithme de Knuth-Morris-Pratt

On désire utiliser l'algorithme de Knuth-Morris-Pratt pour trouver les occurrences du motif GCAGAGAG dans la séquence GCATCGCAGAGAGTATACAGTACG.

Question 1. Calcul du tableau `Next` pour le motif GCAGAGAG:

i	0	1	2	3	4	5	6	7	8
$\text{Next}(i)$	-1	0	0	-1	1	-1	1	-1	1

Question 2. Recherche de GCAGAGAG dans la séquence GCATCGCAGAGAGTATACAGTACG:

- étape 1:

```
GCATCGCAGAGAGTATACAGTACG
|||X
GCAGAGAG
```

Erreur en position $i = 3$: décalage de $i - \text{Next}(i) = 3 - (-1) = 4$

- étape 2:

```
GCATCGCAGAGAGTATACAGTACG
  X
  GCAGAGAG
```

Erreur en position $i = 0$: décalage de $0 - \text{Next}(0) = 1$

- étape 3:

```
GCATCGCAGAGAGTATACAGTACG
  |||||
  GCAGAGAG
```

Le motif apparaît sans erreur: décalage de $8 - \text{Next}(8) = 7$

- étape 4:

```
GCATCGCAGAGAGTATACAGTACG
  |X
  GCAGAGAG
```

Erreur en position $i = 1$: décalage de $1 - \text{Next}(1) = 1$

- étape 5:

```
GCATCGCAGAGAGTATACAGTACG
    X
    GCAGAGAG
```

Erreur en position $i = 0$: décalage de $0 - \text{Next}(0) = 1$

- étape 6:

GCATCGCAGAGAGTATACAGTACG
 X
 GCAGAGAG

Erreur en position $i = 0$: décalage de 0 – $\text{Next}(0) = 1$

• étape 7:

GCATCGCAGAGAGTATACAGTACG
 X
 GCAGAGAG

Erreur en position $i = 0$: décalage de 0 – $\text{Next}(0) = 1$

• étape 8:

GCATCGCAGAGAGTATACAGTACG
 X
 GCAGAGAG

Erreur en position $i = 0$, et on a parcouru toute la séquence.

En tout, la recherche a demandé 19 comparaisons de caractères. Avec l'algorithme par force brute, on aurait effectué 30 comparaisons de caractères.

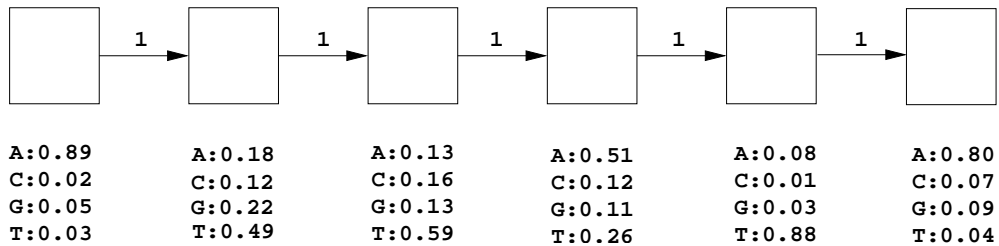
2. TATTAT

Dans le génome, les sites d'initiation de la transcription de l'ADN en ARN sont précédés d'un site *promoteur* : c'est une région de composition particulière sur laquelle peut se fixer l'ARN polymérase. Dans les organismes procaryotes, ce promoteur est représenté par le motif approché TATTAT.

Le tableau ci-dessous donne les fréquences à chaque position pour TATTAT relevées dans le génome de la bactérie *E. coli*:

	T	A	T	T	A	T
A	0.04	0.88	0.26	0.59	0.49	0.03
C	0.09	0.03	0.11	0.13	0.22	0.05
G	0.07	0.01	0.12	0.16	0.12	0.02
T	0.80	0.08	0.51	0.13	0.18	0.89

Donner un modèle de Markov caché (sans pseudo-comptes) pour représenter un promoteur sur le *brin complémentaire* de l'ADN. À titre de rappel, la correspondance entre les bases est $A \leftrightarrow T$ et $C \leftrightarrow G$.



3. BLAST

On considère les séquences ATTCAATTCATTCAATTCATTCAATTCATTCAATTC et ATTGATTGATTGATTGATTGATTGATTGATTG. Quel est, à première vue, leur pourcentage d'identité ? Quand on fait un alignement avec l'algorithme de BLAST, aucune similarité n'est trouvée. Pourquoi ?

Le pourcentage d'identité entre les deux séquences est 75%: chaque C de la première séquence est substitué en G dans la seconde. C'est un pourcentage élevé. Mais BLAST ne le détecte pas à cause de sa stratégie de construction d'alignement : il commence par rechercher des mots exacts communs aux deux séquences. Ici, les mots exacts sont trop courts, puisqu'ils sont de longueur 3.

4. Le génome de *Bizarrus Examinus*

On considère des séquences nucléotidiques d'un organisme étrange *Bizarrus Examinus*:

1. Le nucléotide actuel est un A, un C, un G ou un T avec une probabilité de 25% si les deux nucléotides précédents sont identiques.
2. Le nucléotide actuel a deux fois plus de chance d'être un C ou un G, plutôt qu'un A ou un T, si les deux nucléotides précédents sont différents. De plus, dans le choix entre C contre G, ou A contre T, les purines (A ou G) vont être préférées dans 60% des cas aux pyrimidines (C ou T).

Il y a deux possibilités pour la probabilité d'un caractère, suivant les deux dernières lettres lues.

- Les deux derniers caractères lus sont identiques (AA, CC, GG ou TT):

$$\begin{array}{ll} \text{Proba(A)}=0.25 & \text{Proba(C)}=0.25 \\ \text{Proba(G)}=0.25 & \text{Proba(T)}=0.25 \end{array}$$

- Les deux derniers caractères lus sont différents :

$$\begin{array}{ll} \text{Proba(A)}=0.6 \times 1/3 = 0.2 & \text{Proba(C)}=0.4 \times 2/3=4/15 \\ \text{Proba(G)}=0.6 \times 2/3 = 0.4 & \text{Proba(T)}=0.4 \times 1/3=2/15 \end{array}$$

En première analyse, on pourrait donc penser qu'un modèle à deux états pourrait convenir: un état pour chaque possibilité. Mais il faut garder la trace du dernier caractère lu, pour savoir dans quel état aller à la lecture d'une nouvelle position.

On doit donc créer 8 états :

AA : les deux derniers caractères lus sont des A

CC : les deux derniers caractères lus sont des C

GG : les deux derniers caractères lus sont des G

TT : les deux derniers caractères lus sont des T

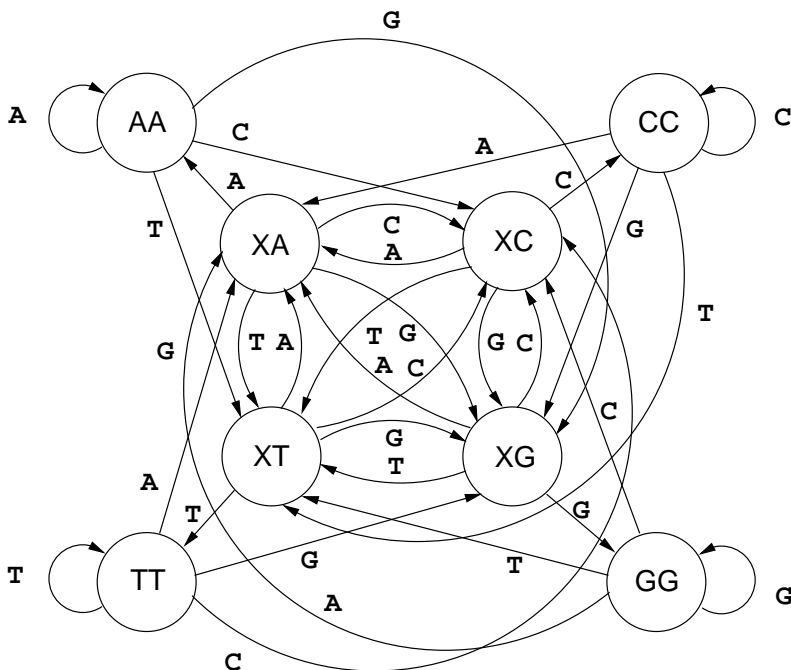
XA : les deux derniers caractères lus sont différents, et le dernier est un A

XC : les deux derniers caractères lus sont différents, et le dernier est un C

XG : les deux derniers caractères lus sont différents, et le dernier est un G

XT : les deux derniers caractères lus sont différents, et le dernier est un T

Cela donne le modèle de Markov suivant:



Les probabilités de transition sont données par le tableau ci-dessous.

(AA, A)	AA	0.25	(XA, A)	AA	0.2
(AA, C)	XC	0.25	(XA, C)	XC	4/15
(AA, G)	XG	0.25	(XA, G)	XG	0.4
(AA, T)	XT	0.25	(XA, T)	XT	2/15
(CC, A)	XA	0.25	(XC, A)	XA	0.2
(CC, C)	CC	0.25	(XC, C)	CC	4/15
(CC, G)	XG	0.25	(XC, G)	XG	0.4
(CC, T)	XT	0.25	(XC, T)	XT	2/15
(GG, A)	XA	0.25	(XG, A)	XA	0.2
(GG, C)	XC	0.25	(XG, C)	XC	4/15
(GG, G)	GG	0.25	(XG, G)	GG	0.4
(GG, T)	XT	0.25	(XG, T)	XT	2/15
(TT, A)	XA	0.25	(XT, A)	XT	0.2
(TT, C)	XC	0.25	(XT, C)	XT	4/15
(TT, G)	XG	0.25	(XT, G)	XG	0.4
(TT, T)	TT	0.25	(XT, T)	TT	2/15

La ligne $(XA, A)|AA|0.2$, par exemple, signifie : La transition étiquetée par A partant de l'état XA mène à l'état AA , et sa probabilité est 0.2. Les transitions absentes correspondent à des transitions impossibles, de probabilité nulle. Il n'y a pas de probabilités d'émission: ce n'est pas un modèle de Markov caché.