Exercice 1: Pratique de l'alignement semi-global

Score: identité +2, substitution -1, insertion ou délétion -1

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

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

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

Score de l'alignement optimal: 9

```
3  A C T G T G 8        T T A C T G T G
   | | | |   |              | | | |   |
1  A C T G A G 6            A C T G A G A C
```

Exercice 2: Dissimilarité versus similarité

$$\begin{aligned} \texttt{Indel}(x) &> 0\text{, coût de l'insertion ou de la délétion du nucléotide } x, \\ \texttt{Diff}(x,y) &> 0 \text{ si } x \neq y\text{, et } 0 \text{ si } x = y. \end{aligned}$$

Question 1: formule de récurrence pour calculer la distance `Dist`

Exercice 2: Dissimilarité versus similarité

$\texttt{Indel}(x) > 0$, coût de l'insertion ou de la délétion du nucléotide x,
$\texttt{Diff}(x,y) > 0$ si $x \neq y$, et 0 si $x = y$.

Question 1: formule de récurrence pour calculer la distance Dist

$$
\begin{aligned}
\texttt{Dist}(0,0) &= 0 \\
\texttt{Dist}(0,j) &= \texttt{Dist}(0,j-1) + \texttt{Indel}(V(j)) \\
\texttt{Dist}(i,0) &= \texttt{Dist}(i-1,0) + \texttt{Indel}(U(i)) \\
\texttt{Dist}(i,j) &= \min \begin{cases} \texttt{Dist}(i-1,j-1) + \texttt{Diff}(U(i),V(j)) \\ \texttt{Dist}(i-1,j) + \texttt{Indel}(U(i)) \\ \texttt{Dist}(i,j-1) + \texttt{Indel}(V(j)) \end{cases}
\end{aligned}
$$

Méthode : programmation dynamique

Question 2: quid de l'alignement local ?

- pas adapté
- la distance est une fonction croissante: les différences s'accumulent. Il faut une fonction non monotone pour l'alignement local.

Exercice 3: Les palindromes génétiques

- palindrome génétique: mot de la forme $u\bar{u}$ où u est un mot et $\bar{u}$ est obtenu à partir de u en l'inversant et en le complémentant
- problème du palindrome maximal

  Donnée : une séquence S d'ADN
  Question : quel est le plus long sous-mot de S qui soit un palindrome génétique?

- longueur du palindrome maximal dans $ACCGGATT$?

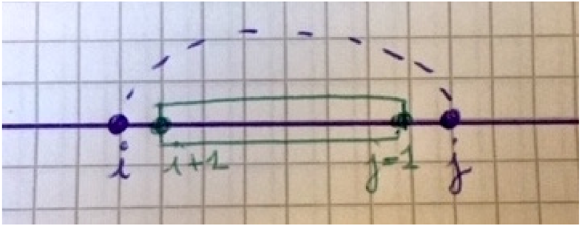
$$P(i,j) = j - i + 1 \text{ si } S(i..j) \text{ est un palindrome, 0 sinon}$$

Question 1: formule de récurrence pour calculer P

$P(i,j){=}0, \text{si } j <= i$

$P(i,i+1){=}2, \text{ si } S(i) = S(i+1), \text{ 0 sinon}$

$P(i,j){=}2 + P(i+1,j-1) \text{ si } S(i) = S(j) \text{ et } P(i+1,j-1) > 0,$
0 sinon



Question 2: implémentation

- matrice à deux dimensions
- deux boucles imbriquées
- ordre de calcul des indices

```
for i in range(n,0,-1):
    P(i,i)=1
    P(i,i+1)=...
    for j in range(i,n+1):
        P(i,j)=...
```

- le palindrome le plus long est donné par la valeur la plus élevée de la matrice. Les positions de début et fin de ce palindrome sont les indices i et j de la valeur maximale.

A propos des palindromes:
https://en.wikipedia.org/wiki/Palindromic_sequence

Exercice 4: Réplication hasardeuse

Une bactérie est atteinte par un virus qui affecte la machinerie de la réplication de la sorte

- chaque `A` peut être remplacé par 2 `A`,
- chaque `C` peut être remplacé par 2 `C`,

La multiplication des bases n'est pas systématique, et il se peut qu'à une position donnée la copie soit correcte.

Question 1: algorithme qui pour deux séquences U et V détermine si U peut être une version infectée de V.
On définit

$$\begin{aligned} I(i,j) &= \texttt{vrai} \text{ si } U(1..i) \text{ est une version infectée de } V(1..j) \\ &= \texttt{faux} \text{ sinon} \end{aligned}$$

Question 1: algorithme qui pour deux séquences U et V détermine si U peut être une version infectée de V.
On définit

$$\begin{aligned} I(i,j) &= \texttt{vrai} \text{ si } U(1..i) \text{ est une version infectée de } V(1..j) \\ &= \texttt{faux} \text{ sinon} \end{aligned}$$

$$\begin{aligned} I(0,0) &= \texttt{vrai} \\ I(0,j) &= \texttt{faux}\ (j > 0) \\ I(i,0) &= \texttt{faux}\ (i > 0) \\ I(i,j) &= \text{si } U(i) \neq V(j), \text{ alors } \texttt{faux} \\ &\quad \text{sinon si } U(i) \in \{G, T\}, \text{ alors } I(i-1, j-1) \\ &\quad \text{sinon on a } U(i) \in \{A, C\} \\ &\qquad \text{si } U(i-1) \neq U(i), \text{ alors } I(i-1, j-1) \\ &\qquad \text{sinon } (I(i-1, j-1) \text{ ou } I(i-2, j-1)) \end{aligned}$$

Question 2: En plus d'induire une copie multiple d'une position, il est également possible que la base soit oubliée, provoquant une délétion. Modifiez l'algorithme précédent pour prendre en compte ce nouveau phénomène.

$$
\begin{aligned}
I(0,0) &= \texttt{vrai} \\
I(0,j) &= \texttt{vrai}\ (j>0) \\
I(i,0) &= \texttt{faux}\ (i>0) \\
I(i,j) &=
\end{aligned}
$$

si $U(i) \neq V(j)$, alors $I(i,j-1)$

sinon si $U(i) \in \{G,T\}$, alors $I(i-1,j-1)$ $\texttt{ou}$ $I(i,j-1)$

sinon on a $U(i) \in \{A,C\}$

si $U(i-1) \neq U(i)$, alors $(I(i-1,j-1)$ $\texttt{ou}$ $I(i,j-1))$

sinon $(I(i-1,j-1)$ $\texttt{ou}$ $I(i-2,j-1)$ $\texttt{ou}$ $I(i,j-1))$

Exercice 5: Tous les alignements locaux

		t	*a*	*c*	*g*	*c*	*g*	*t*	*g*	*g*	*a*	*t*	*t*	*g*	*a*	*t*	*c*
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0
c	0	0	0	2	1	1	0	0	0	0	0	0	0	0	0	0	1
g	0	0	0	1	3	2	2	1	1	1	0	0	0	1	0	0	0
a	0	0	1	0	2	1	1	0	0	0	2	1	0	0	2	1	0
t	0	1	0	0	1	0	0	2	1	0	1	3	2	1	1	3	2
c	0	0	0	1	0	2	1	1	0	0	0	2	1	0	0	2	4
g	0	0	0	0	2	1	3	2	2	1	0	1	0	2	1	1	3
a	0	0	1	0	1	0	2	1	1	0	2	1	0	1	3	2	2
t	0	1	0	0	0	0	1	3	2	1	1	3	2	1	2	4	3
g	0	0	0	0	1	0	1	2	4	3	2	2	1	3	2	3	2
a	0	0	1	0	0	0	0	1	3	2	4	3	2	2	4	3	2
t	0	1	0	0	0	0	0	1	2	1	3	5	4	3	3	5	4
a	0	0	2	1	0	0	0	0	1	0	2	4	3	2	4	4	3

Quels sont les poids pour une identité, une substitution, un indel ?

Exercice 5: Tous les alignements locaux

		t	*a*	*c*	*g*	*c*	*g*	*t*	*g*	*g*	*a*	*t*	*t*	*g*	*a*	*t*	*c*
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0
c	0	0	0	2	1	1	0	0	0	0	0	0	0	0	0	0	1
g	0	0	0	1	3	2	2	1	1	1	0	0	0	1	0	0	0
a	0	0	1	0	2	1	1	0	0	0	2	1	0	0	2	1	0
t	0	1	0	0	1	0	0	2	1	0	1	3	2	1	1	3	2
c	0	0	0	1	0	2	1	1	0	0	0	2	1	0	0	2	4
g	0	0	0	0	2	1	3	2	2	1	0	1	0	2	1	1	3
a	0	0	1	0	1	0	2	1	1	0	2	1	0	1	3	2	2
t	0	1	0	0	0	0	1	3	2	1	1	3	2	1	2	4	3
g	0	0	0	0	1	0	1	2	4	3	2	2	1	3	2	3	2
a	0	0	1	0	0	0	0	1	3	2	4	3	2	2	4	3	2
t	0	1	0	0	0	0	0	1	2	1	3	5	4	3	3	5	4
a	0	0	2	1	0	0	0	0	1	0	2	4	3	2	4	4	3

Quels sont les poids pour une identité, une substitution, un indel ?

identité 1, substitution -2, indel -1

Question 1: les alignements de score optimal

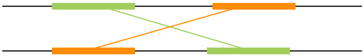
		t	a	c	g	c	g	t	g	g	a	t	t	g	a	t	c
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0
c	0	0	0	2	1	1	0	0	0	0	0	0	0	0	0	0	1
g	0	0	0	1	3	2	2	1	1	1	0	0	0	1	0	0	0
a	0	0	1	0	2	1	1	0	0	0	2	1	0	0	2	1	0
t	0	1	0	0	1	0	0	2	1	0	1	3	2	1	1	3	2
c	0	0	0	1	0	2	1	1	0	0	0	2	1	0	0	2	4
g	0	0	0	0	2	1	3	2	2	1	0	1	0	2	1	1	3
a	0	0	1	0	1	0	2	1	1	0	2	1	0	1	3	2	2
t	0	1	0	0	0	0	1	3	2	1	1	3	2	1	2	4	3
g	0	0	0	0	1	0	1	2	4	3	2	2	1	3	2	3	2
a	0	0	1	0	0	0	0	1	3	2	4	3	2	2	4	3	2
t	0	1	0	0	0	0	0	1	2	1	3	5	4	3	3	5	4
a	0	0	2	1	0	0	0	0	1	0	2	4	3	2	4	4	3

Question 2:

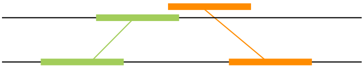
- Les alignements locaux peuvent être composés de plusieurs zones similaires colinéaires, c'est-à-dire sans croisement ni chevauchement.
- Donnez un tel alignement dont le score total est 8.

Alignements colinéaires

Alignements non-colinéaires



croisement



chevauchement

		t	a	c	g	c	g	t	g	g	a	t	t	g	a	t	c
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0
c	0	0	0	2	1	1	0	0	0	0	0	0	0	0	0	0	1
g	0	0	0	1	3	2	2	1	1	1	0	0	0	1	0	0	0
a	0	0	1	0	2	1	1	0	0	0	2	1	0	0	2	1	0
t	0	1	0	0	1	0	0	2	1	0	1	3	2	1	1	3	2
c	0	0	0	1	0	2	1	1	0	0	0	2	1	0	0	2	4
g	0	0	0	0	2	1	3	2	2	1	0	1	0	2	1	1	3
a	0	0	1	0	1	0	2	1	1	0	2	1	0	1	3	2	2
t	0	1	0	0	0	0	1	3	2	1	1	3	2	1	2	4	3
g	0	0	0	0	1	0	1	2	4	3	2	2	1	3	2	3	2
a	0	0	1	0	0	0	0	1	3	2	4	3	2	2	4	3	2
t	0	1	0	0	0	0	0	1	2	1	3	5	4	3	3	5	4
a	0	0	2	1	0	0	0	0	1	0	2	4	3	2	4	4	3

		t	a	c	g	c	g	t	g	g	a	t	t	g	a	t	c
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0
c	0	0	0	2	1	1	0	0	0	0	0	0	0	0	0	0	1
g	0	0	0	1	3	2	2	1	1	1	0	0	0	1	0	0	0
a	0	0	1	0	2	1	1	0	0	0	2	1	0	0	2	1	0
t	0	1	0	0	1	0	0	2	1	0	1	3	2	1	1	3	2
c	0	0	0	1	0	2	1	1	0	0	0	2	1	0	0	2	4
g	0	0	0	0	2	1	3	2	2	1	0	1	0	2	1	1	3
a	0	0	1	0	1	0	2	1	1	0	2	1	0	1	3	2	2
t	0	1	0	0	0	0	1	3	2	1	1	3	2	1	2	4	3
g	0	0	0	0	1	0	1	2	4	3	2	2	1	3	2	3	2
a	0	0	1	0	0	0	0	1	3	2	4	3	2	2	4	3	2
t	0	1	0	0	0	0	0	1	2	1	3	5	4	3	3	5	4
a	0	0	2	1	0	0	0	0	1	0	2	4	3	2	4	4	3

		t	a	c	g	c	g	t	g	g	a	t	t	g	a	t	c
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0
c	0	0	0	2	1	1	0	0	0	0	0	0	0	0	0	0	1
g	0	0	0	1	3	2	2	1	1	1	0	0	0	1	0	0	0
a	0	0	1	0	2	1	1	0	0	0	2	1	0	0	2	1	0
t	0	1	0	0	1	0	0	2	1	0	1	3	2	1	1	3	2
c	0	0	0	1	0	2	1	1	0	0	0	2	1	0	0	2	4
g	0	0	0	0	2	1	3	2	2	1	0	1	0	2	1	1	3
a	0	0	1	0	1	0	2	1	1	0	2	1	0	1	3	2	2
t	0	1	0	0	0	0	1	3	2	1	1	3	2	1	2	4	3
g	0	0	0	0	1	0	1	2	4	3	2	2	1	3	2	3	2
a	0	0	1	0	0	0	0	1	3	2	4	3	2	2	4	3	2
t	0	1	0	0	0	0	0	1	2	1	3	5	4	3	3	5	4
a	0	0	2	1	0	0	0	0	1	0	2	4	3	2	4	4	3

quel est le problème ?

Question 3: modélisation sous forme de graphes

- score seuil s
- chaque alignement est délimité par deux cellules dans la matrice: (i,j) en haut à gauche et (k,ℓ) en bas à droite
- noeuds: les alignements au delà d'un score seuil, étiquetés par le score de l'alignement
- arc entre deux alignements $(i_1,j_1)-(k_1,\ell_1)$ et $(i_2,j_2)-(k_2,\ell_2)$ si, et seulement si, $k_1 < i_2$ et $\ell_1 < j_2$
- alignement local final: chemin de poids maximum

Exercice 6: Alignement co-optimaux

Question 1: Exemple de deux alignments co-optimaux
Indel: -2, substitution : -1, identité: 1

```
G C - - T A G T C      G C T A - - G T C
| |     | |   | |      | | | |       | |
G C T A T A A T C      G C T A T A A T C
```

Score: 2

1) séquences ATTC et ATC

	-	A	T	T	C
-	0	-2	-4	-6	-8
A	-2	1	-1	-3	-5
T	-4	-1	2	0	-2
C	-6	-3	0	1	1

Alignements :

ATTC
AT_C   score = 1

ATTC
A_TC   score = 1

Question 2

- U et V, les deux séquences à aligner
- $\texttt{S}(i,j)$: score de similarité entre $U(1..i)$ et $V(1..j)$, donné par la matrice de programmation dynamique
- $\texttt{Coop}(i,j)$: nombre d'alignements co-optimaux entre $U(1..i)$ et $V(1..j)$

$\texttt{Coop}(0,0)$=1

$\texttt{Coop}(0,j)$ =1

$\texttt{Coop}(i,0)$ =1

$$\texttt{Coop}(i,j) = \sum \begin{cases} \texttt{Coop}(i-1,j-1), \text{si } \texttt{S}(i,j) = \texttt{S}(i-1,j-1) + Sub(U(i),V(j)) \\ \texttt{Coop}(i-1,j), \text{si } \texttt{S}(i,j) = \texttt{S}(i-1,j) + Del(U(i)) \\ \texttt{Coop}(i,j-1), \text{si } \texttt{S}(i,j) = \texttt{S}(i,j-1) + Ins(V(j)) \end{cases}$$

Exercice 7: Alignement de séquences codantes

Si on ne tient pas compte des codons STOP

$$
\begin{aligned}
\mathtt{S}(i,j) &= 0, \text{si } i < 3 \text{ et } j < 3 \\
\mathtt{S}(0,j) &= \mathtt{S}(0,j-3) + \mathtt{Indel}(V(j-2..j)) \\
\mathtt{S}(i,0) &= \mathtt{S}(i-3,0) + \mathtt{Indel}(U(i-2..i)) \\
\mathtt{S}(i,j) &= \max \begin{cases} \mathtt{S}(i-3,j-3) + \mathtt{Sub}(U(i-2..i), V(j-2..j)) \\ \mathtt{S}(i-3,j) + \mathtt{Indel}(U(i-2..i)) \\ \mathtt{S}(i,j-3) + \mathtt{Indel}(V(j-2..j)) \end{cases}
\end{aligned}
$$

Avec les codons STOP: on définit les fonctions

$$\begin{aligned}
\texttt{SIndel}(w) &= \text{si } \texttt{STOP}(w) \text{ alors } -\infty, \text{ sinon } \texttt{Indel}(w)\\
\texttt{SSub}(v,w) &= \text{si } \texttt{STOP}(v) \text{ ou } \texttt{STOP}(w), \text{ alors } -\infty, \text{ sinon } \texttt{Sub}(v,w)\\
\texttt{SMax}(a,b,c) &= \text{si } a=-\infty, b=-\infty \text{ ou } c=-\infty, \text{ alors } -\infty,\\
&\quad\ \text{sinon } \max(a,b,c)
\end{aligned}$$

$$\begin{aligned}
\texttt{S}(i,j) &= 0, \text{si } i<3 \text{ et } j<3\\
\texttt{S}(0,j) &= \texttt{S}(0,j-3)+\texttt{SIndel}(V(j-2..j))\\
\texttt{S}(i,0) &= \texttt{S}(i-3,0)+\texttt{SIndel}(U(i-2..i))\\
\texttt{S}(i,j) &= \texttt{SMax}\begin{cases}\texttt{S}(i-3,j-3)+\texttt{SSub}(U(i-2..i),V(j-2..j))\\ \texttt{S}(i-3,j)+\texttt{SIndel}(U(i-2..i))\\ \texttt{S}(i,j-3)+\texttt{SIndel}(V(j-2..j))\end{cases}
\end{aligned}$$

Exercice 8: Pas d'insertion après une délétion, s'il vous plait

Initialisation (inchangée)

$$
\begin{array}{rcl}
a(0,0) & = & 0 \\
a(i,0) & = & -\infty \\
a(0,j) & = & -\infty \\
 & & \\
b(i,0) & = & -\infty \\
b(0,j) & = & \mathtt{Ouv} + \mathtt{Ext} \times j \\
 & & \\
c(i,0) & = & \mathtt{Ouv} + \mathtt{Ext} \times i \\
c(0,j) & = & -\infty
\end{array}
$$

Cas général, aucune restriction

$$a(i,j) = Sub(i,j) + \max \begin{cases} a(i-1,j-1) \\ b(i-1,j-1) \\ c(i-1,j-1) \end{cases}$$

$$b(i,j) = \max \begin{cases} \mathtt{Ouv} + \mathtt{Ext} + a(i,j-1) \\ \mathtt{Ext} + b(i,j-1) \\ \mathtt{Ouv} + \mathtt{Ext} + c(i,j-1) \end{cases}$$

$$c(i,j) = \max \begin{cases} \mathtt{Ouv} + \mathtt{Ext} + a(i-1,j) \\ \mathtt{Ouv} + \mathtt{Ext} + b(i-1,j) \\ \mathtt{Ext} + c(i-1,j) \end{cases}$$

Cas général pour l'exercice

$$a(i,j) = Sub(i,j) + \max \begin{cases} a(i-1,j-1) \\ b(i-1,j-1) \\ c(i-1,j-1) \end{cases}$$

$$b(i,j) = \max \begin{cases} \texttt{Ouv} + \texttt{Ext} + a(i,j-1) \\ \texttt{Ext} + b(i,j-1) \\ \cancel{\texttt{Ouv} + \texttt{Ext} + c(i,j-1)} \end{cases}$$

$$c(i,j) = \max \begin{cases} \texttt{Ouv} + \texttt{Ext} + a(i-1,j) \\ \cancel{\texttt{Ouv} + \texttt{Ext} + b(i-1,j)} \\ \texttt{Ext} + c(i-1,j) \end{cases}$$

Exercice 9: Algorithme de Hirschberg

Cas 1:

$$\mathcal{A}\begin{pmatrix} U \\ V \end{pmatrix} = \mathcal{A}\begin{pmatrix} U(1..i-1) \\ V(1..j-1) \end{pmatrix} + \begin{pmatrix} U(i) \\ V(j) \end{pmatrix} + \mathcal{A}\begin{pmatrix} U(i+1..m) \\ V(j+1..n) \end{pmatrix}$$

Cas 2 :

$$\mathcal{A}\begin{pmatrix} U \\ V \end{pmatrix} = \mathcal{A}\begin{pmatrix} U(1..i-1) \\ V(1..j-1) \end{pmatrix} + \begin{pmatrix} U(i) \\ - \end{pmatrix} + \mathcal{A}\begin{pmatrix} U(i+1..m) \\ V(j..n) \end{pmatrix}$$

Exercice 9: Algorithme de Hirschberg

Cas 1: la position i de U est alignée avec une position j de V

$$\mathcal{A}\begin{pmatrix} U \\ V \end{pmatrix} = \mathcal{A}\begin{pmatrix} U(1..i-1) \\ V(1..j-1) \end{pmatrix} + \begin{pmatrix} U(i) \\ V(j) \end{pmatrix} + \mathcal{A}\begin{pmatrix} U(i+1..m) \\ V(j+1..n) \end{pmatrix}$$

Cas 2 : la position i de U est délétée

$$\mathcal{A}\begin{pmatrix} U \\ V \end{pmatrix} = \mathcal{A}\begin{pmatrix} U(1..i-1) \\ V(1..j-1) \end{pmatrix} + \begin{pmatrix} U(i) \\ - \end{pmatrix} + \mathcal{A}\begin{pmatrix} U(i+1..m) \\ V(j..n) \end{pmatrix}$$

Pour toute valeur de i, $0 \leq i < m$, l'algorithme de Needleman et Wunsh permet de calculer l'ensemble des scores d'alignements globaux entre $U(1..i-1)$ et tous les préfixes de V en espace linéaire par rapport à la longueur de V et en temps quadratique.

Vrai - à peu près vu en cours

On calcule le score avec la formule de récurrence de Needleman et Wunsh avec deux vecteurs pour la mémoire: $U(1..i-1)$ en vertical, V en horizontal.

La case j de la dernière ligne donne le score entre $U(1..i-1)$ et $V(1..j)$.

Question 3 :

3) Pour calculer les alignements globaux entre $U(i+1 \ldots m)$ et les suffixes de T pour chaque i :

- On inverse les 2 séquences U et T.

  Si $U =$ TCACGA, $U^R =$ AGCACT.

Puis on utilise l'algorithme NW pour les aligner jusqu'à i fixé et les suffixes.

Léa Vandamme

Question 3 : On peut écrire des formules de récurrence qui travaillent sur les suffixes, au lieu des préfixes

- U de longueur m et V de longueur n.
- $T(i,j)$: score d'alignement entre $U(i..m)$ et $V(j..n)$

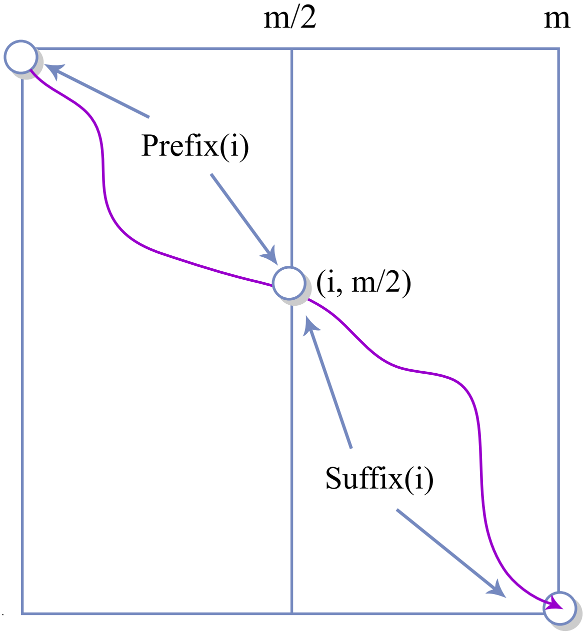
$$
\begin{array}{lcl}
\mathtt{T}(n+1, m+1) & = & 0 \\
\mathtt{T}(n+1, j) & = & \mathtt{T}(0, j+1) + Ins(V(j)) \\
\mathtt{T}(i, m+1) & = & \mathtt{T}(i-1, 0) + Del(U(i)) \\
\mathtt{T}(i, j) & = & \max \left\{ \begin{array}{l} Sub(U(i), V(j)) + \mathtt{T}(i+1, j+1) \\ Del(U(i)) + \mathtt{T}(i+1, j) \\ Ins(V(j)) + \mathtt{T}(i, j+1) \end{array} \right.
\end{array}
$$

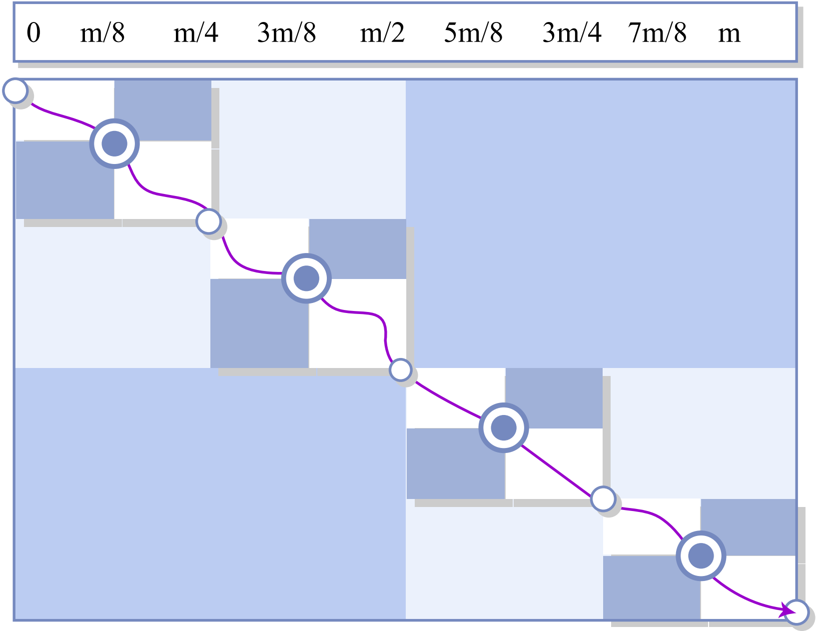
Question 4: Fonction `Score(U, V, M/2, J, Cas)`

1. On calcule le tableau `Prefix`, qui donne pour tout j le score de $U(1..i-1)$ aligné avec $V(1..j)$
2. On calcule le tableau `Suffix`, qui donne pour tout j le score de $U(i+1..m)$ aligné avec $V(j..m)$
3. Cela permet de déterminer J et `Cas` (1 ou 2), avec une boucle sur j.

Complexité:

Question 5: algo récursif pour l'alignement global optimal

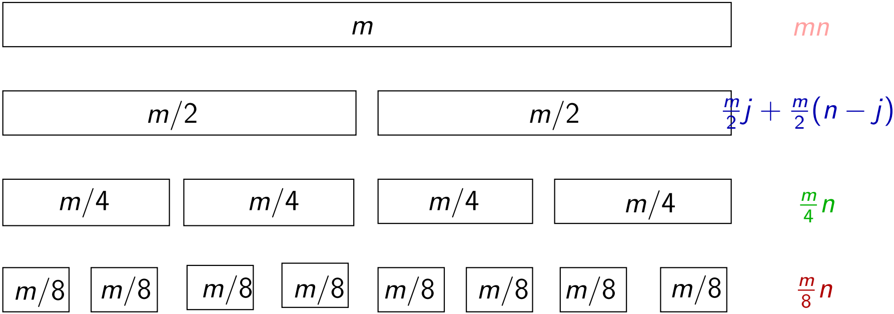



0
m/8
m/4
3m/8
m/2
5m/8
3m/4
7m/8
m

```
function Align(U,V:Sequence)
   M:longueur de U
   N:longueur de V
begin
   Si M==0 then
      return (1..N =>'-', V);
   Sinon si N==0 then
      return (U, 1..M =>'-');
   Sinon
      Score(U, V, M/2, J, Cas);
      Si Cas == 1: substitution
         return Align(U(1..M/2-1),V(1..J-1))
           &(U(M/2), V(J))
           &Align(U(M/2+1..M),V(J+1..N));
      Sinon -- cas 2: deletion
         return Align(U(1..M/2-1),V(1..J))
           &(U(M/2), '-')
           &Align(U(M/2+1..M),V(J+1..N));
end
```

Complexité en temps

U



$$n\left(1+\frac{1}{2}+\frac{1}{4}+\frac{1}{8}+\cdots\right)=2mn$$

Complexité en espace