

Recherche des structures secondaires d'une chaîne d'ADN

Projet 2I003

LI Mengda, GE Zhichun

26 novembre 2017

1 Exercice 1

1.1

Comme un nucléotide ne peut former une paire avec lui même, donc

$$\forall i \in \{1, \dots, n\}, \quad S_{i,i} = \{\} \text{ et } E_{i,i} = 0$$

1.2

1.2.1

Si ni i , ni j ne sont couplés dans $S_{i,j}$, alors

$$S_{i,j} = S_{i+1,j-1} \quad E_{i,j} = E_{i+1,j-1}$$

1.2.2

Si j n'est pas couplée dans $S_{i,j}$, alors

$$S_{i,j} = S_{i,j-1} \quad E_{i,j} = E_{i,j-1}$$

1.2.3

Si $(i, j) \in S_{i,j}$, alors

$$S_{i,j} = S_{i+1,j-1} \cup (i, j) \quad E_{i,j} = E_{i+1,j-1} + 1$$

1.2.4

Si $(k, j) \in S_{i,j}$ avec $k \in \{i+1, \dots, j-1\}$, alors

$$S_{i,j} = S_{i,k-1} \cup S_{k,j} \quad E_{i,j} = E_{i,k-1} + E_{k,j}$$

1.3

1.3.1

Par la question 1.2, résumons en distinguant tous les cas possibles :

1. Soit ni i , ni j ne sont couplés dans $S_{i,j}$, alors $E_{i,j} = E_{i+1,j-1}$,
sinon on a toujours $E_{i,j} \geq E_{i+1,j-1}$
2. Soit j n'est pas couplée dans $S_{i,j}$, alors $E_{i,j} = E_{i,j-1}$,
sinon on a toujours $E_{i,j} \geq E_{i,j-1}$
3. Soit j est couplé avec i , c'est à dire $(i,j) \in S_{i,j}$, alors $E_{i,j} = E_{i+1,j-1} + 1$
4. Soit j est couplé avec un $k \in \{i+1, \dots, j-1\}$, alors $E_{i,j} = E_{i,k-1} + E_{k,j}$,
sinon $E_{i,j} \geq E_{i,k-1} + E_{k,j}$ car $S_{i,j} \supset S_{i,k-1} \cup S_{k,j} \quad \forall k \in \{i+1, \dots, j-1\}$

Soit la fonction $e : \{1, \dots, n\} \times \{1, \dots, n\} \rightarrow \{0, 1\}$ qui vaut 1 ssi i et j peuvent être couplés.

Dans le cas 1 et cas 3, $E_{i,j} = E_{i+1,j-1} + e(i,j)$ et on a toujours $\overline{E_{i,j} \geq E_{i+1,j-1} + e(i,j)}$ car
— si j est couplé avec i , $e(i,j) = 1$, $E_{i,j} = E_{i+1,j-1} + 1 = E_{i+1,j-1} + e(i,j)$,
— sinon $e(i,j) = 0$, on a toujours $E_{i,j} \geq E_{i+1,j-1} = E_{i+1,j-1} + e(i,j)$ par 1.

Dans le cas 2, $E_{i,j} = E_{i,j-1}$ et on a toujours $\overline{E_{i,j} \geq E_{i,j-1}}$ car $S_{i,j} \supset S_{i,j-1}$

Par le cas 4, $\forall k \in \{i+1, \dots, j-1\}$ $E_{i,j} \geq E_{i,k-1} + E_{k,j}$. Donc $\overline{E_{i,j} \geq \max_{i < k < j} E_{i,k-1} + E_{k,j}}$.

Et s'il existe $k_0 \in \{i+1, \dots, j-1\}$ tel que $(k_0, j) \in S_{i,j}$, $E_{i,j} = E_{i,k_0-1} + E_{k_0,j}$

Dans ce cas-là, le max est atteint :

$$E_{i,k_0-1} + E_{k_0,j} \geq \max_{i < k < j} E_{i,k-1} + E_{k,j} \text{ et } \max_{i < k < j} E_{i,k-1} + E_{k,j} \geq E_{i,k_0-1} + E_{k_0,j}$$

$$\text{D'où } E_{i,j} = E_{i,k_0-1} + E_{k_0,j} = \max_{i < k < j} E_{i,k-1} + E_{k,j}$$

En conclusion : $E_{i,j}$ est un majorant de l'ensemble $\{E_{i+1,j-1} + e(i,j), E_{i,j-1}, \max_{i < k < j} E_{i,k-1} + E_{k,j}\}$, et ce majorant est atteint : il y a toujours un cas où la condition d'égalité est vrai. Donc

$$E_{i,j} = \max\{E_{i+1,j-1} + e(i,j), E_{i,j-1}, \max_{i < k < j} E_{i,k-1} + E_{k,j}\}$$

...

1.3.2

Si $k = j$, $E_{k,j} = E_{j,j} = 0$ par 1.1, donc $E_{i,j-1} \in \{i < k \leq j \mid E_{i,k-1} + E_{k,j}\}$, et

$$E_{i,j-1} \leq \max\{i < k \leq j \mid E_{i,k-1} + E_{k,j}\}$$

On en déduit que $E_{i,j} = \max\{E_{i+1,j-1} + e(i,j), \max_{i < k < j} E_{i,k-1} + E_{k,j}\}$

1.4

```
1 def tailleMaxRec(a: str, i: int, j: int) -> int:
    n = len(a)
3     assert 1 <= i <= n and 1 <= j <= n, 'impossible de tronquer'

5     if a == '' or i >= j:
        return 0

7
8     def couple(i: int, j: int) -> bool:
9         return {a[i-1], a[j-1]} in [ {'A', 'T'}, {'C', 'G'} ]

11    def e(i: int, j: int) -> {0,1}:
12        if couple(i, j):
13            return 1
14        else:
15            return 0

17    return max(tailleMaxRec(a, i+1, j-1) + e(i, j),
18               max([tailleMaxRec(a, i, k-1) + tailleMaxRec(a, k, j)
19                    for k in range(i+1, j+1)]))
```

1.5

Cette fonction se termine car chaque fois on appelle récursivement sur une sous-partie de séquence $[i+1, j-1]$, le début de sous-séquence i est strictement croissant et la fin de sous-séquence j est strictement décroissante. Dans \mathbb{N} , l'ordre est bien fondé donc i, j se rencontre en certaine récursion. Lorsque i, j se rencontre, c'est à dire $i \leq j$, la fonction retourne une valeur et se termine en dépillant récursivement.

Cette fonction calcule exactement $\max\{E_{i+1,j-1} + e(i,j), \max_{i < k < j} E_{i,k-1} + E_{k,j}\}$, elle est valide selon la question 1.3.

1.6

1.6.1

Pour $u_0, (p=0) \Rightarrow (i=j)$. Donc l'appel de fonction retourne simplement 0, $u_0 = 1$

Pour $u_1, (p=1) \Rightarrow (i+1=j)$, donc cet appel retourne

```
1 max(tailleMaxRec(a, i+1, j-1) + e(i, j),
2     max([tailleMaxRec(a, i, k-1) + tailleMaxRec(a, k, j)
3          for k in range(i+1, j+1)]))
```

Vu que $i+1=j$, $\text{tailleMaxRec}(a, i+1, j-1)$ retourne simplement 0 car $i+1=j \geq j-1$.

Analysons la dernière itérable

```
[tailleMaxRec(a, i, k-1) + tailleMaxRec(a, k, j) for k in range(i+1, j+1)]
```

$\text{range}(j, j+1)$ ne contient que j , donc cette liste n'a qu'un élément, cela vaut dire qu'on appelle deux fois tailleMaxRec . En sommant, $u_1 = 3$

1.6.2

— Cas de base : pour $p = 2, j - i = 2$

— On appelle d'abord une fois la fonction elle-même : $\text{tailleMaxRec}(a, i, j)$, on a 1

— Et puis on appelle $\text{tailleMaxRec}(a, i+1, j-1)$,
 $(j-1) - (i+1) = (j-i) - 2 = 0$, on a donc u_0

— Enfin, pour

$\text{tailleMaxRec}(a, i, k-1) + \text{tailleMaxRec}(a, k, j)$ for k in $\text{range}(i+1, j+1)$

$\text{range}(i+1, j+1)$ équivaut à $\llbracket j-1, j \rrbracket$

Lorsque $k = j-1$,

on appelle $\text{tailleMaxRec}(a, i, j-2)$, on a u_0 et

on appelle $\text{tailleMaxRec}(a, j-1, j)$, on a u_1

Lorsque $k = j$,

on appelle $\text{tailleMaxRec}(a, i, j-1)$, on a u_1 et

on appelle $\text{tailleMaxRec}(a, j, j)$, on a u_0

On a bien $u_2 = u_0 + 1 + 2 \sum_{i=0}^1 u_i$

— Induction :