Devoir 2

Émanuel Rollin - 20106951

Question 1

Discussion

Séquence d'ARN = S

On considère seulement les appariements canoniques (Watson Crick)

Reformuler comme un problème d'autoappariement tq on apparie S avec $S^{Reverse}$

$S = "ACGGCAACGU"$

$S^R = "ACGUUGCCGU"$ *bases opposées et ordre inverse

Critère 1 : Algorithme avec complexité calculatoire $O(n^2)$

Algorithmes connus:

- Nussinov $\rightarrow O(n^3)$ car on vérifie 2n cases n^2 fois $\rightarrow$ Maximise le nombre de base pairs
  - On peut probablement optimiser car on s'intéresse au problème ne contenant qu'une seule tige boucle
- Zuker $\rightarrow O(n^4)$ $\rightarrow$ On vérifie le cadrant inférieur gauche de max n^2 cases n^2 fois $\rightarrow$ Minimise l'énergie libre
- Sankoff? $\rightarrow Time \in O(n^2) et\ Space \in O(n^3)$ $\rightarrow$ Scénario évolutif ayant le moindre coût = hors contexte

Tentative 1 : Nussinov modifié

Rappel de l'énoncé : "On veut le repliement en *une* tige-boucle de S qui maximise le nombre d'appariements de bases."

Idée : éliminer la recherche / compairason avec la séquence divisée en deux tiges boucles, ce qui retroune l'algorithme à $Time \in O(n^2)$

Nussinov original:

$$d_{Nussinov}(i,j) = max \begin{cases} D[i+1][j] & base\ s_1\ does\ not\ pair\ with\ s_2\ (bulge) \\ D[i][j-1] & base\ s_2\ does\ not\ pair\ with\ s_1\ (bulge) \\ D[i+1][j-1] & pairing\ between\ bases \\ max_{k=i+1}^{j-2}(M(i,k)+M(k+1,j)) & Combinaison\ de\ sousséquences\ coupées\ au\ points\ k \\ & est\ plus\ stable\ que\ la\ séquence\ complète\ (sous\ boucles) \end{cases}$$

1.1. À Quoi correspondent les cellules de M sur l'anti-diagonale $M[i, |S| - i]$?

$V[i][j] = 0\ si\ j \geqslant len(s) - i - 1$ $\rightarrow$ Diagonale inverse = 0

Représentent un appariement d'un nucléotide avec soi-même. Devrait être considéré

impossible, tel que la diagonale inverse devrait être initialisée à zéro.

On peut remplir seulement la moitié de la table, car on doit balancer une moitié de la séquence avec l'autre, tel qu'on peut apparier au maximum n/2 bases et au minimum 0 (fin de la séquence avec elle même ou début avec elle-même).

1.2. Donnez les équations d'initialisation et de récurrence pour remplir la table M

$$V_{Nussinov}(i,j) = max\begin{cases} D[i+1][j] & (\longleftarrow)\ bulge\ in\ s_2 \\ D[i][j+1] & (\uparrow)\ \ bulge\ in\ s_1 \\ D[i+1][j+1] + \delta_{match} & (\nwarrow)\ \ if\ pairing\ between\ bases\ i\ and\ j \end{cases}$$

$V[i][j] = 0\ si\ j \geqslant len(s) - i - 1$ → Tout ce qui est sous la diagonale inverse = 0

Pondération: $\delta_{match} = 1$

1.3 Décrivez comment on peut retrouver une repliement en tige-boucle maximisant les appariements de nucléotides

La case $V[0][0]$ devrait contenir le nombre d'appariements maximal. Le traceback suit le chemin composé des flèches $\uparrow$, $\nwarrow$, $\longleftarrow$ et reconstruit la tige boucle en appariant la tête de la séquence avec sa queue, laissant des gaps au besoin.

Case M[0][0] retourne le nb d'appariements maximal

$S = "ACGGCAACGU"$
$S^R = "ACGUUGCCGU"$ If i == j, alors match

Real	Reverse	A	C	G	G	C	A	A	C	G	U
U	A	4	3	2	2	2	2	2	1	0	0
G	C	3	3	2	1	1	1	1	1	0	0
C	G	2	2	2	1	0	0	0	0	0	
A	U	1	1	1	1	0	0	0	0		
A	U	1	1	1	1	0	0	0			
C	G	1	1	1	1	0	0				
G	C	1	1	0	0	0					
G	C	1	1	0	0						
C	G	0	0	0							
A	U	0	0								

Maximum : 4 appariements

Traceback :

s_1	s_2	
A	A	Match ↖
C	C	Match ↖
G	G	Match ↖
U	G	Gap s_1 ↑
U	G	Gap s_1 ↑
G	G	Match ↖

Condition d'arrêt : Traceback stoppe à l'atteinte d'un zéro. On assume des gaps ensuite.

On a tracé l'appariement de la figure 1 en vert

A C G – – G
U G C A A C

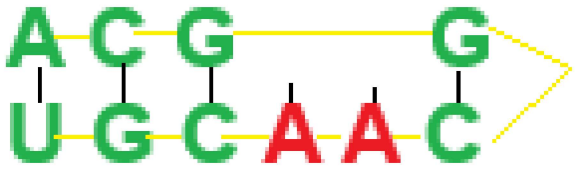


Figure 1: L'appariement optimal n'est pas nécessairement représentatif du comportement dans la nature. On remarque que le pont hydrogène en noir du dernier G-C induit un angle difficile à maintenir pour le squelette phosphaté.

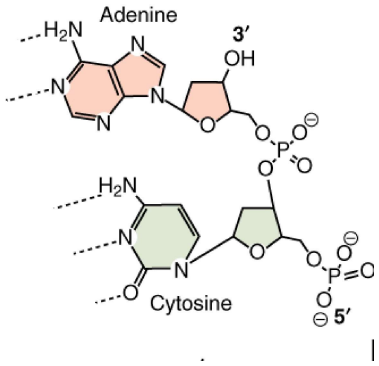
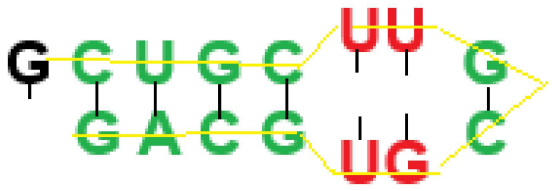


Figure 2: Difficulté à former des ponts hydrogènes avec ses voisins immédiats

1.4 Appliquez votre algorithme à "GCGUGCUUGCGUGCACG"

Real	Reverse	G	C	G	U	G	C	U	U	G	C	G	U	G	C	A	C	G
G	C	6	6	5	4	4	4	4	4	4	4	3	2	1	1	1	1	0
C	G	5	5	5	4	4	3	3	3	3	3	3	2	1	0	0	0	0
A	U	4	4	4	4	3	3	3	3	2	2	2	2	1	0	0	0	
C	G	3	3	3	3	3	2	2	2	2	1	1	1	1	0	0		
G	C	3	3	2	2	2	2	1	1	1	1	0	0	0	0			
U	A	3	3	2	2	2	2	1	1	1	1	0	0	0				
G	C	3	3	2	2	2	2	1	1	1	1	0	0					
C	G	3	2	2	2	2	1	1	1	1	0	0						
G	C	2	2	1	1	1	1	0	0	0	0							
U	A	2	1	1	1	1	0	0	0	0								
U	A	2	1	1	1	1	0	0	0									
C	G	2	1	1	1	1	0	0										
G	C	1	1	1	0	0	0											
U	A	1	1	0	0	0												
G	C	1	1	0	0													
C	G	1	0	0														
G	C	0	0															



Maximum : 4 appariements

Traceback choisi :

s_1	s_2	
–	G	Gap s_2 ←
C	C	Match ↖
U	U	Match ↖
G	G	Match ↖
C	C	Match ↖
A	–	Gap s_1 ↑
C	–	Gap s_1 ↑
–	U	Gap s_2 ←
–	U	Gap s_2 ←
G	G	Match ↖

Traceback stoppe à l'atteinte d'un zéro. On assume des gaps ensuite.

On a tracé l'appariement en vert

S_2	G	C	U	G	C	U	U	G
S_1	–	G	A	C	G	U	G	C

Certains gaps peuvent être remplacés par des missmatchs

1.5 Pour être stable, une tige-boucle doit contenir au moins 3 nucléotides. Décrivez une façon de modifier votre algorithme afin d'avoir des tige-boucles avec au moins 3 nucléotides dans la boucle.

L'idée est d'avancer la diagonale de zéros, tel que les appariements finaux lors du traceback seront forcément des gaps.

Real	*Reverse*	*G*	*C*	*G*	*U*	*G*	*C*	*U*	*U*	*G*	*C*	*G*	*U*	*G*	*C*	*A*	*C*	*G*
G	*C*															0	0	0
C	*G*														0	0	0	0
A	*U*													0	0	0	0	
C	*G*												0	0	0	0		
G	*C*											0	0	0	0			
U	*A*										0	0	0	0				
G	*C*									0	0	0	0					
C	*G*								0	0	0	0						
G	*C*							0	0	0	0							
U	*A*						0	0	0	0								
U	*A*					0	0	0	0									
C	*G*				0	0	0	0										
G	*C*			0	0	0	0											
U	*A*		0	0	0	0												
G	*C*	0	0	0	0													
C	*G*	0	0	0														
G	*C*	0	0															

Changement 1:
On change les conditions d'arrêt pour arrêter sur $M[i][j]$ *avec* $i = n - j$

Chagement 2:
On ajoute 2 diagonales de zéros:

Alors forcément, les 3 derniers appariements seront des gaps dans s_2 ou s_1 tel que

s_1	s_2	
–	...	*Gap* s_1 ↑
–	...	*Gap* s_1 ↑
–	...	*Gap* s_1 ↑

Force un minimum de 3 gaps dans la tête d'épingle.

1.6 Décrivez une modification de votre algorithme qui permet les appariements G-U

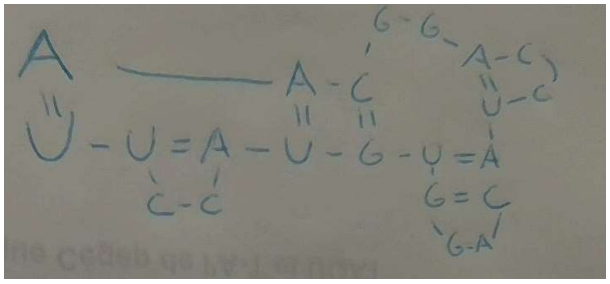
La logique de matching actuel réside dans l'idée que le complément inverse de S_2 est S_1 et permet de vérifier une identité / égalité entre le caractère à la position $S_1[j]$ et $S_2[i]$.

Si on veut ajouter cet appariement sans se débarasser de l'inverse complémentaire, on peut ajouter un quatrième cas à vérifier. Celui-ci vérifie simplement que le caractère G complémenté en C est pairé avec un U (ou inversement) et si oui, faire un match. On pourrait changer la pondération de ce cas si on le souhaite aussi en lui donnant un δ_{GU}.

$$V_{Nussinov}(i,j) = max \begin{cases} D[i+1][j] & (\longleftarrow)\ bulge\ in\ s_2 \\ D[i][j+1] & (\uparrow)\quad bulge\ in\ s_1 \\ D[i+1][j+1] + \delta_{match} & (\nwarrow)\quad if\ pairing\ between\ bases\ i\ and\ j \\ D[i+1][j+1] + \delta_{match} & (\nwarrow)\ if \\ & (S_1[j] = C\ \ \&\&\ \ S[i] = U)\ \ || \\ & (S_1[j] = A\ \ \&\&\ \ S[i] = G) \end{cases}$$

1.7 Considérez la séquence d'ARN ci-dessous avec ses paires de bases inférées. Dessinez un repliement en 2D qui illustre les sous-structures principales de la molécule.

Traits simples = Squelette phosphaté
Traits doubles = Liens hydrogènes



1.8 Supposez que l'on veuille maximiser le nombre d'empilements, c'est-à-dire le nombre d'appariements (i,j) tels que (i+1,j-1) sont aussi appariés. Décrivez un algorithme de programmation dynamique fonctionnant en temps $O(n^3)$ qui permet de retrouver un repliement en une ou plusieurs tige-boucles maximisant le nombre d'empilements.

Représente l'algorithme original de Nussinov:

Conditions initiales:
$V(i \geqslant j, j) = 0$
Pondération:
$\delta_{match} = 1$
Relation de récurrence:

$$d_{Nussinov}(i,j) = max \begin{cases} D[i+1][j] & base\ s_1\ does\ not\ pair\ with\ s_2\ (bulge) \\ D[i][j-1] & base\ s_2\ does\ not\ pair\ with\ s_1\ (bulge) \\ D[i+1][j-1] & pairing\ between\ bases \\ max_{k=i+1}^{j-2}(M(i,k)+M(k+1,j)) & Combinaison\ de\ sousséquences\ coupée\ au\ points\ k \\ & est\ plus\ stable\ que\ la\ séquence\ complète \\ & \\ & On\ compare\ toutes\ les\ coupures \end{cases}$$

Traceback commence : Coin supérieur droit
Traceback termine : Lorsqu'on rencontre la diagonale avec i=j

Chemin à parcourir : On doit être attentif aux maximums composés de la sommes de sous

épingles maximisantes, sinon traceback habituel.