

IFT3295 - TP1

Chevauchement de séquences

- 1) L'alignement dans ce cas-ci est le meilleur alignement entre un suffixe d'une séquence X_i et le préfixe d'une séquence X_j , contrairement à un alignement global qui se fait sur les séquences au complet.



- 2) $V(i, 0) = 0$

Étant donné qu'on considère un suffixe de X_i , cela permet de commencer notre alignement à n'importe quelle position de X_i sans pénalité.

$$V(0, j) = -8 * j$$

Étant donné qu'on considère un préfixe de X_j , cela permet de pénaliser les potentiels indels aux positions antérieures à notre préfixe de X_j .

- 3) $V(i, 0) = 0$

$$V(0, j) = -8 * j$$

$$V(i, j) = \max \{ V(i-1, j) - 8, \\ V(i, j-1) - 8, \\ V(i-1, j-1) + m \text{ où } m = 4 \text{ si match et } -4 \text{ si mismatch} \}$$

- 4) Pour retrouver le meilleur alignement, il faut partir de la valeur maximale de la dernière ligne et suivre les pointeurs jusqu'à arriver à la première colonne. Si on suit un pointeur qui va vers la gauche, on aligne la position j de X_j avec un indel. Si on prend un pointeur diagonal, on aligne la position j de X_j avec la position i de X_i . Si on prend un pointeur qui monte, on aligne la position i avec un indel.

5) Implémentation de l'algorithme de programmation dynamique.

Étape d'exécution du code :

- 1) Ouvrir le fichier `chevauchement.html` dans n'importe quel navigateur internet.
- 2) Importer un fichier FASTQ (`reads.fq` par exemple) en utilisant le bouton prévu à cet effet.

Choisir ou coller le contenu d'un fichier FASTQ :

Choisir un fichier

Aucun fichier choisi

Choisir ou coller le contenu d'un fichier FASTQ :

Choisir un fichier

Aucun fichier choisi

```
@SEQUENCE_TEST_1
CATCCTTCT
+
222222222
@SEQUENCE_TEST_2
CCTTTCACC
+
222222222
```

3) Cliquer sur le bouton "Aligner les séquences"

Aligner les séquences

Xi : CATCCTTCT

Xj : CCTTTCACC

0,-8,-16,-24,-32,-40,-48,-56,-64,-72
0,4,-4,-12,-20,-28,-36,-44,-52,-60
0,-4,0,-8,-16,-24,-32,-32,-40,-48
0,-4,-8,4,-4,-12,-20,-28,-36,-44
0,4,0,-4,0,-8,-8,-16,-24,-32
0,4,8,0,-8,-4,-4,-12,-12,-20
0,-4,0,12,4,-4,-8,-8,-16,-16
0,-4,-8,4,16,8,0,-8,-12,-20
0,4,0,-4,8,12,12,4,-4,-8
0,-4,0,4,0,12,8,8,0,-8

↑,←,←,←,←,←,←,←
↑,↖,↖,←,←,←,←,↖,↖
↑,↖,↖,↖,↖,↖,←,←
↑,↖,↖,↖,↖,←,←,←
↑,↖,↖,↑,↖,↖,←,↖
↑,↖,↖,←,↖,↖,↖,↖
↑,↖,↖,↖,↖,↖,↖,↖
↑,↖,↖,↖,↖,←,←,↖
↑,↖,↖,↑,↑,↖,←,↖
↑,↖,↖,↖,↖,↖,↖,↖

Le score d'alignement optimal est 12 à la position i=9,
j=5

L'alignement optimal est
CATCCTTCT----
---CCTT-TCACC

La longueur de chevauchement optimal est 6

Le fichier `alignerSequences.js` contient le code de l'algo de prog dynamique en lui-même, qui est chargé dans la page html.

Assemblage de fragments

- 1) Le fichier `assemblage.html` contient le code qui crée la matrice ci-dessous en utilisant l’algorithme de la section précédente sur chaque pair de reads et qui génère une liste des nœuds pour le graphe. (Le code n’est pas demandé dans l’énoncé, mais nous le mettons quand même à disposition)

	mot1	mot2	mot3
mot1	×		
mot2		×	
mot3			×

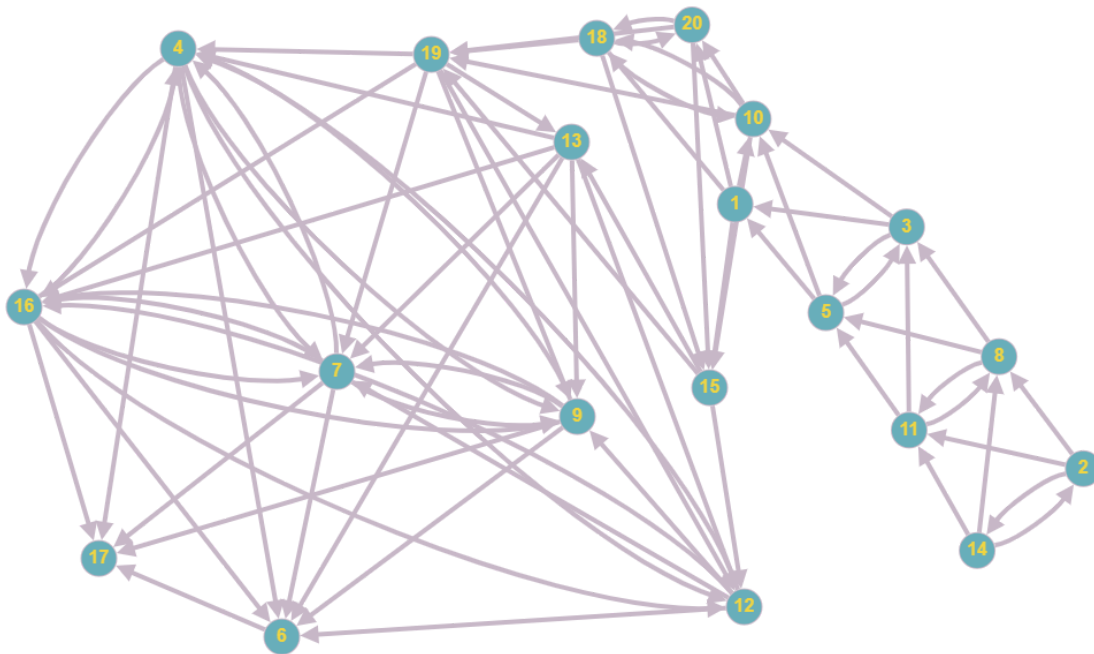
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
01	0	8	8	0	4	12	12	16	4	804	20	4	0	4	396	4	20	652	12	608
02	4	0	8	12	44	8	4	416	8	12	424	16	0	1060	12	12	12	8	4	0
03	396	12	0	24	1016	16	48	0	36	128	4	20	40	8	20	20	16	8	8	8
04	28	8	28	0	12	640	1024	16	1068	12	12	220	16	24	12	656	152	16	8	16
05	500	4	696	28	0	24	52	8	52	208	16	20	36	4	8	24	36	64	4	12
06	52	4	4	16	0	0	4	4	12	28	4	16	4	0	28	8	672	16	12	20
07	32	12	32	648	16	720	0	4	772	12	12	24	24	24	24	176	232	12	4	0
08	8	12	644	4	540	16	8	0	8	4	944	12	4	12	0	4	8	36	4	16
09	36	12	20	956	16	668	1044	4	0	8	4	64	12	24	20	492	196	8	4	12
10	4	0	0	8	32	8	8	24	12	0	32	4	12	12	640	4	8	904	184	868
11	20	20	620	4	524	12	12	1072	4	12	0	24	28	24	8	8	16	20	12	12
12	28	28	48	900	16	460	820	0	856	4	16	0	28	24	16	996	20	16	8	4
13	68	32	32	744	24	272	656	0	692	16	8	860	0	36	16	848	32	24	4	28
14	16	756	12	12	52	20	8	452	12	16	492	16	8	0	4	12	20	16	8	16
15	20	8	28	12	16	12	16	12	8	4	12	80	260	28	0	28	20	24	632	24
16	20	16	24	1032	16	552	944	12	988	8	20	692	8	20	12	0	80	12	8	8
17	16	4	32	16	16	32	8	12	4	24	20	12	4	28	12	8	0	16	8	20
18	16	0	8	12	8	4	4	8	8	240	16	8	4	16	808	8	4	0	368	1004
19	48	20	16	388	8	20	316	8	368	16	20	520	732	16	24	492	20	20	0	4
20	8	4	0	4	0	8	4	0	0	0	4	0	64	8	908	4	16	644	428	0

Avec seuil à 80

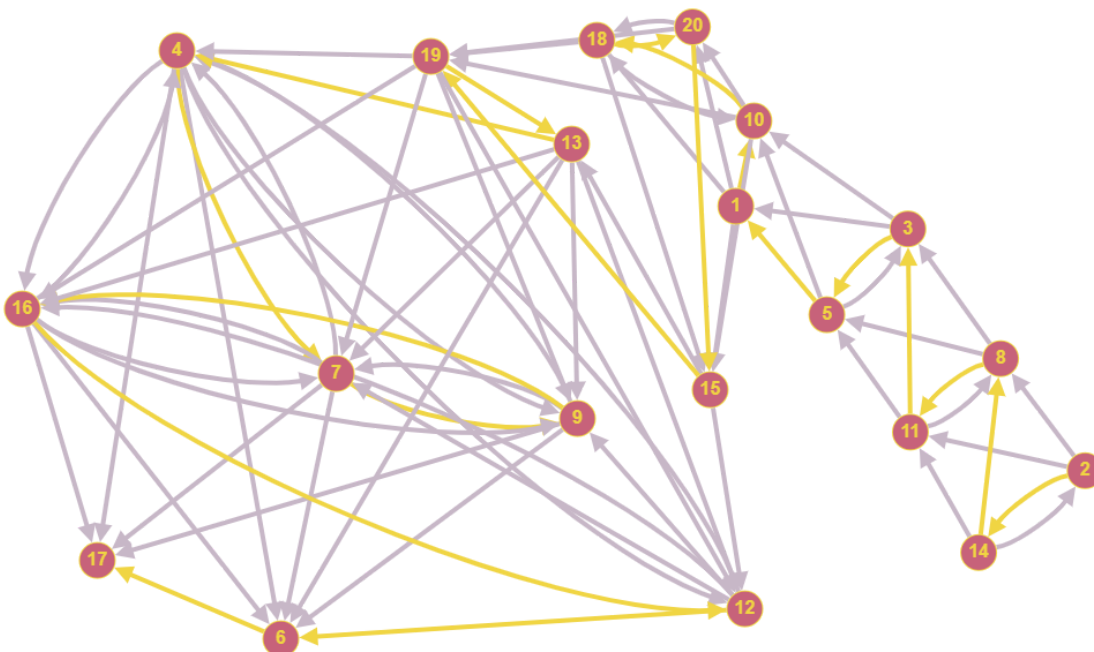
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
01	0	0	0	0	0	0	0	0	0	804	0	0	0	0	396	0	0	652	0	608
02	0	0	0	0	0	0	0	416	0	0	424	0	0	1060	0	0	0	0	0	0
03	396	0	0	0	1016	0	0	0	0	128	0	0	0	0	0	0	0	0	0	0
04	0	0	0	0	0	640	1024	0	1068	0	0	220	0	0	0	656	152	0	0	0
05	500	0	696	0	0	0	0	0	0	208	0	0	0	0	0	0	0	0	0	0
06	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	672	0	0	0
07	0	0	0	648	0	720	0	0	772	0	0	0	0	0	0	176	232	0	0	0
08	0	0	644	0	540	0	0	0	0	0	944	0	0	0	0	0	0	0	0	0
09	0	0	0	956	0	668	1044	0	0	0	0	0	0	0	0	492	196	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	640	0	0	904	184	868
11	0	0	620	0	524	0	0	1072	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	900	0	460	820	0	856	0	0	0	0	0	0	996	0	0	0	0
13	0	0	0	744	0	272	656	0	692	0	0	860	0	0	0	848	0	0	0	0
14	0	756	0	0	0	0	0	452	0	0	492	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	80	260	0	0	0	0	0	632	0
16	0	0	0	1032	0	552	944	0	988	0	0	692	0	0	0	0	80	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	240	0	0	0	0	808	0	0	0	368	1004
19	0	0	0	388	0	0	316	0	368	0	0	520	732	0	0	492	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	908	0	0	644	428	0

2)

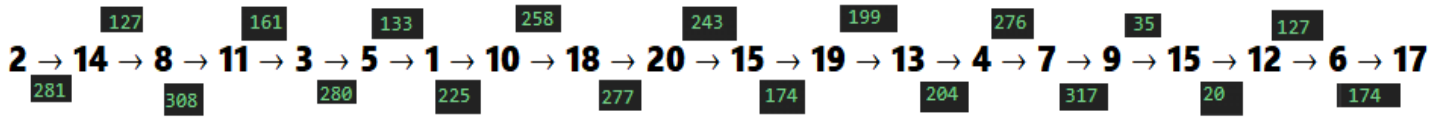
- a) Cela permet de ne garder que les séquences avec un haut score d'alignement et mieux mettre en évidence les séquences qui ont le plus long chevauchement entre eux.
- b) Nous avons utilisé <https://graphonline.ru/en/> pour produire les graphes.



Pour trouver le chemin qui passe par tous les noeuds une fois, il suffit de chercher le chemin hamiltonien (algorithme présent sur l'outil cité), ce qui nous donne le graphe réduit en jaune :



Voici donc l'ordre des reads ainsi que la longueur du chevauchement entre deux paires de reads.



c) Voici la séquence du fragment génomique reconstitué manuellement.

```
CTAGGGACTTGGAGAACACAAGTATTATGAAAAGTACTGATGAAAGTTATTAACAGGTTTCGAAAAATAACTTTACTATCTGACGT
GTTGCTTCTGCCGAGGATGACCGTTATTCTGGTTTTGCATTTATATTTACGTATGGTTAAATGTGCCAGCGTTGTGGTTTAAAA
CTAATAGTAATAATATGCTTCTTTGTTTCAGTTGGCTAGAGATTTACTACATCCGTCCTTGGAAGAGGAAAAAGAAAAACATAACAA
GAAACGCCTAGTACAAAGTCCAAATCTTACTTTATGGATGTAAAATGTCCAGGTAAAATTTGAAATCTTAATTCCTTTACTAAAG
AAAATTTCTGTAGGGATTGCTAGTGTGGTGTGTATAGTTAAGATACATTAGAATCCTCTGTTGAGTAGAAGTGGGATTACAGAATT
GGACATGTCAGGGACAATTTATAATAAATGTACTGAAACCATTGTAGAAACATTTGCGGTAAAGTAAAAACAGGCCTTCAAAGGA
GATATGCCATTTGACTGGACTTGTGGATAATCAAAATGTGGATACACAAAATAGAGATGTTCTTTAATTGTAAAATACTCACTGGA
TTTTGACGATGTAGCACAGAAAAAAATACATTGATTACACTGTTTTTAAAAATTTTGGGTCGCTGCTAGAAAAGTTTATGTTAC
ACATGGGGCTTGCTGTTTCATAGCACTGAAGTTAATGATTTTTTTTACATATTACCTGAAATCTCGAACAGGTCCTGTTTTCTCTG
CTTTTCATTTTTTAACATTGCCTTTTTTTTTTTTTAGGTTGCTACAAGATAACCACGTTTTTCAGCCATGCTCAGACAGTGGTTCTT
TGTGTAGGTTGTTCAACAGTGTGTGCCAGCCTACAGGAGGAAAGGCCAGACTCACAGAAGGTATATCATTGTGCTATTCTCCAACC
CAGTGATGAGATTGATGATTATAAATGTCTCTATCTTCACTGAAAAGTTTACAGAAATCTTAATGATTCCCAAAATAACTTATCTC
ACACTGGAAGAGTTCAAGTGGATTGGCAGCAAATCTGAGATCTATTTGGTGTGACCTGGTGAGATCTAAATATGGAGTCAGCACAT
GATTTTTTTAAGAGTAATATTGCTAATAATATTGCTAAGTATAGTCTGAAAATACCTCTAATCAAAATTATGTACTTGAGAAAAG
TTTTCAGAATAGTTCCTAAAAAGTAAGAGTATATTTCTGGTATAAAAGGATAAATAATATGTATATGAGTATTAATCCAATATTCT
TAAAACTTCAGTATTTATAGCTCTGAAGTTACTGATTTTTTTTACATATTACCAGAAATGTCGAACAGGTGCTGTTTTCTCTGCTT
TTCATTTTTTAACATTGCCTTTTTTTTTTTTTAGGTTGCTACAACATCACCACGGTTTTTCAGCCATGCTCAGACAGTGGTTCTTTGT
GTAGGTTGTTCAACAGTGGTGTCCAGCCTACAGGAGGAAAGGCCAGACTCACAGAAGGTATATCATTGGCATTCTCCAACCCAG
TGATGAGATTGATGATTTTAAATGTCTCTATATTAAGTAAAAGTTTAAAGAAATCTTAATGATTACCAAAATAACTTATCTCTCA
ATGGAAGAGTTCAAGTGGATTTCGAGCAAATCGGAGATCTATTTGGTGTGACCTGGTGAGATCTAAATATGGAGTCTGCACATGAT
TTTTTTTAGAGTTATATTGCTAAGTAATATTGCTTAGTATAGTCGAAAAATTCCTCTAATCAAAATTATTTACTTGAGAAAAGTAT
TCAGAAGAGTTCCTAAAAATTAAGAGTATATTTCTGGTATATAAGCATAAATAATCTGTATATGAGTATTAATCCAATATTCTTAA
AACTTCAGTATTTTACTTAAAAGTCCTTTTTGTCAATAAAATTATAGCAACGGTAGAATGCACCTGTTTAAATATACTCTCATGATT
CTTTTGCAGGGTGTTCATTTAGAAGAAAGCAACACTAATGATTCAAACAGCTTCCTGAATTTTAAATTTGTGTTGTCTCACAGAAA
GCCTTATCATAAATTCATAATTCTAATTAATTTACCAAGATAATGTCATTACATTTGGTTATGTAAGTTATACAGCAGTAATCTC
CTATTTTGGTGTGAGTTTTTCACTAAAGTTTTTAT
```

Taille : 2185

[illegible]

Recherche d'introns et Blast

- 1) Bien qu'écrire un script pour traduire la séquence nucléotidique de `sequence.fasta` en séquence protéique (format du gène X) en utilisant le code génétique standard dans les trois cadres de lecture ne soit pas très compliqué, nous avons juste utilisé le site

<https://web.expasy.org/translate/>

DNA or RNA sequence

AGGACAAAGTTTGCCTTACCATATGTTTCTGATCGTGCAGAACCCCTTTCTAGGACTTGGAGAACACAAAGTATTATGAAAGTACTGATGAAAGTTATTAACAGGTTTGCAGGTTTGTGTTTAACTAATAGTAATAATATGCTTCTTTGTTGCTTGGCTAGAGATTACTACATCCGCTCTTGGAGAGGAAAGAGAAAAACATAAAAGAAACGCCTAGTACAAAGTCCAAATCTTACTTTATGGATGTAAGTGTCCAGGTAAATTTGAAATCTTAATTCCTTTACTAAAGAAATTTCTGTAGGATTGCTAGTGTGTTGTATAGTTAAGATACATTAGAACCCTTGTGTGAGTAGAAGTGGGATTACAGAATTGGAAATGTCAGGACATTTTCATAATAAATGTACTGAAACCATTTGTAGAAACATTTTGGGGTAAAGTGAAGACAGGACATTCAGAGGAGATATGCCATTTGACGACATTTGGATAATCAAAATGTGGATCTCAAAATAGAGATGTTCTTTAATTGTAAAAATCACTGAGATTTTGTGATGATGACACAGAAAAAATACATTTGATTACACTGTTTTTAAAAAATTTGTGCTGCTGCTAGAAAAAGTTTATGTTTACACATGTGCTTGTCTGTTTCATAGCACTGAAGTTACTGATTTTTTACATATTACACAGAAATGTCGAACAGGTGCTGTTTTCTCTGCTTTTCACTTTTAAACATTTGCTTTTTTTTTTTAGTTTGTCTACAAGA

Output format

☐ Verbose: Met, Stop, spaces between residues

☒ Compact: M, -, no spaces

☐ Includes nucleotide sequence

☐ Includes nucleotide sequence, no spaces

DNA strands

☒ forward ☒ reverse

Genetic codes - See NCBI's genetic codes

Standard

reset

TRANSLATE!

Celui-ci nous donne les 3 cadres de lectures :

5'3' Frame 1

RTKFPALPYVS-SCESPFLGTWRTQVL-KVLMKVINRFRKITLLSDVLLPRMTVIPVFACIFHVWLNVPALWFKTNSNNMLLCSVG-RFTTSVLGRGKEKT-KETPSTKSKFLLYGCKMSR-NLKS-FLY-RKFL-GLLVWCV-LRYIRTL-VEVGLQNWKCQGHFNKCTETIVETFRGK-KQAFKGLMPFDCTCG-SKCGYSK-RCSLIVKYSLDFDDVAQKKNLTITLFLKNFVSLLEKFMMLHMLAVS-H-SY-FFYILPEMSNRCCFPLLFIFNIAFFFF-PATRSRFSAMLQWFFV-VVQCCASLQEEPRDPSQKDYHLAFSNPVMRLMIINVSIFTEKFEILMITKITYLSLEEFKWIWSKSEIYLV-PGEI-IWSQHMIFLRVILLSNIAKYSKLIPLIKIYLRKVFRIVPKN-EYISGIRG-IICI-VLIQYS-NFSILLKSTFCH-NYSKGRMHLFNILS-FFCRLFI-KKATIMIQAS-ILILCCLTESLIINSIILINLPR-CNYIWFCVKYSSNLLFWCQFFNKVILMGR

5'3' Frame 2

GQSLPYHMFPDRAKALF-GLGEHKYKEY--KLLTGFEK-LYYLTCCFCRG-PLFLFLHVFYTYG-MCQRCLGLLIVICFFVQLARDLLHPSLEEEKKKKKKRLVQSPNSYFMDVKCPGKI-NLNSFTKENFCRDC-CGVYS-DTLEPSVE-KWDYRIGNVRDIFIINVLKPL-KHFGVSENRRHSKEICHLTALVDNQNVDTQNRDVL-L-NTHWILTM-HRKKIH-LHCF-KILCRC-KSLCYTCGLLFHSTEVTDFFTYYQKCRGTGAVFLCFSLTLPFFFFSLLQDHHGFQPCSDSGSLCRLFNSSVVPAYRRGQTHRRIIIWHSPTQ--D--L-MSLSSLKSLKS--LPK-LISHWSSSGLAANLRSIWCDLVRSKYGVST-FFE-YC-VILLSIV-KYL-SKLFT-EKYSE-FLKIKSIFLV-KDK-SVYEV-SNILKTSVFLKVLVFIKIIAKVECTCLYSHDSFADCSFRRKQH--FKQLPEF-FCVVSQKALS-IP-F-LIYQDNVITFGFVRYTAVISYFGVSFSIKF-LWA

5'3' Frame 3

DKVCLTICFLIVRKPFSDLENTSINKSTDESY-QVSKNNFTI-RVASAEDDRYSCFCMYISRMVKCASVVV-N----YASLFSWLEIYYIRPWKRKRKNIKRNA-YKVOILTLM-NVQVKFEILIPLLKISVGIASVVCIVKIH-NPLLSRSGITELEMSGTF--MY-NHCRNISG-VKTGIQRRYAI-LHLWIKMWILKIEFFNCKILTGF-RCSTEKKYIDYTVFKKFCVAARKVYVTHVACCFIALKLLIFLHITRNVQVLFSSAFHF-HCLFFFLVCYKITTVFSHAQTVVLCVGCSTVLCQPTGGKARLTEGLSFGILQPSDEIDDYKCLYLH-KV-RNLNDYQNNLSLTGRVQVDWQI-DLFGVTW-DLNMESAHDFPKSNIAR-YC-V-SENTSNQNYLLEKSIQNSS-KLRVYFYKRIINLYMSINPIFLKLQYFT-KYFLSLKL-QR-NALV-YTLMILLQIVHLEESNTNDSNSFLNFNFVLSHRKPYKHFHNSN-PTKIM-LHLVL-GIQQ-SPILVSVFQ-SFDYQ

- a) En faisant une recherche du début de la séquence du gène X dans la page, on obtient qu'il se trouve dans le second cadre de lecture.

web.expasy.org/translate/

reset TRANSLATE! MCQRCGLKLVIIICFFVQLARD 1/1

Results of translation

- Open reading frames are highlighted in red
- Select your initiator on one of the following frames to retrieve your amino acid sequence

Download all the translated frames

5'3' Frame 1—

RTKFPALPYVS-SCSPFLGTWRTQVL-KVLMKVINRFRKITLLSDVLLPRMTVIPVFACIFHVWLNVPALWFKTNSNNMLLCSVG-RFTTSVLGRGKEKT-KETPSTKSKFLLYGCKMSR-NLKS-FLY-RKFL-GLLVWCV-LRYIRTLC-VEVLQNWKCQGHFNKCTETIVETFRGK-KQAFKGDMPFDCTCG-SKCGYSK-RCSLIVKYSYSLDFDDVAQKNTLITLFLKNFVSLLKEFMLHMLAVS-H-SY-PFY-ILPEMSNRCCFPFLFIENIAFFFE-FATRSRFSAMLRCWFEV-VVQCCASLQEEPRDSQKDYHLAFSNPVMRLMIINVSIETEFKEILMITKITIYLSLEEFKWIWSKSEIYLV-PGEI-IWSQHMIEL-RVILLSNIAKYSKIPLIKIIYLRKVFRIVFKN-EYISGIGK-IICI-VLIQYS-NFSILLKSTFCH-NYSKGRMHLFNILS-PFCRLFI-KKATIMIQTAS-ILILCCLTESLIINSIILINLPR-CNYI-WPCKVYSSNLLFWCQFFNKVLIIMGK

5'3' Frame 2—

GQSLPYHMFPPDRAKALF-GLGEHYYEYK--KLLTGFEK-LYYLTCCFCRG-PLFLFLHVYFTYG-MCQRCGLKLVIIICFFVQLARDLLHPSLEEKKKKKKKRLVQSPNSYFMDVKCPGKI-NLNSFTK-ENFCRDC-CGVYS-DTLEPSVE-KWDYRIGNVRDIFIINVLPKPL-KHFGVSENRSKEICHTALVDNQNVDTQNRDVL-L-NTHWILT-M-HRKKIH-LHCF-KILCRC-KSLCYTCGLLFHSTEVTDFFT-YYQKCRGTGAFLCFSLTLTFFFSLLQDHGFGPCSDSGSLCRLFNSVVPAYRRKGQTHRRIIWHSPQ--D--L-MSLSLKLKKS--LPK-LISHWKSSSGLAANLRSIWCDLVRSKYGVST-PF-E-YC-VILLSIV-KYL-SKLFT-EKYSE-FLKIKSIFLV-KDK-SVYEV-SNLIKTSVPYLVKLVFIKIAKVECTCLYSHDSFADCSFRRKQH--FKQLPEF-FCVVSQKALS-IP-F-LIYQDNVITF-GFVRYTAVISYFGVSFSIKF-LWA

5'3' Frame 3—

DKVCLTICFLIVRKPFPSRDLENTSIMKSTDESY-QVSKNNFTI-RVASAEDDRYSCFCMYISRMVKCASVVV-N---YASLFSWLEIYYIRPWKKRKNIKRNA-YKVQILTLM-NVQVKFEILIPLLK-KISVGIASVVCIVKIH-NPLLSRSGITELMSGTFS--MY-NHCRNIG-VKTGIQRRYAI-LHLWIIKMWILKIEFMFNCKILTGF-RCSTEKKYIDYTVFKKFCVAARKVYVTHVACCFIALKLLIFLH-ITRNVEQVLFSSAFHF-HCLFFFLVCYKITTVFSHAQTVVLCVGCSTVLCQPTGGKARLTEGLSFGILQPSDEIDYKCLYLH-KV-RNLNDYQNNLSLTGRVQVDWQOI-DLFGVTW-DLNMESAHDFEK-SNIAK-YC-V-SENTSNONYLLEKSIONSS-KLRVYFWYKRNNLYMSINPIFLKLOYFT-KYFLSLKL-QR-NALV-YTLMILLOIVHLEESNTNDSNLSFLNFVLSHRKPKYHKPHNSN-FTKIM-LHL-VL-GIQQ-SPILVSVFQ-SFDYQG

- b) La recherche d'un gène dans une séquence génomique consiste tout simplement en une recherche de motif dans une séquence. Les équations de récurrences sont celles de la distance d'édition. Dans la table de programmation dynamique, la séquence est Xj et le gène est Xi.

$$V(i, 0) = i$$

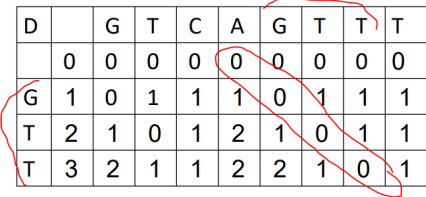
$$V(0, j) = 0$$

$$V(i, j) = \min \{ V(i-1, j), V(i, j-1), V(i-1, j-1) + m \text{ où } m = 1 \text{ si match et } 0 \text{ si mismatch} \}$$

Pour retrouver l'alignement optimal dans la table de programmation dynamique, il faut partir de la case avec le score minimal sur la dernière ligne et suivre les pointeurs pour remonter jusqu'à la première ligne. Lorsqu'on prend un pointeur diagonal, on aligne la position i du gène avec la position j de la séquence. Dans le cas idéal où le score minimal à la dernière ligne est 0 et qu'il n'y aurait que des pointeurs diagonaux de cette même case jusqu'à la première ligne, cela voudrait dire qu'il y a une occurrence exacte du gène à la position j de la séquence.

Exemple d'algorithme pour trouver l'alignement optimal :

Comme expliqué précédemment, on veut parcourir le tableau dynamique créé avec les règles précédentes pour trouver le meilleur alignement possible. On garde toujours en mémoire l'alignement optimal trouvé. Pour éviter de considérer tous les alignements possibles, nous allons adopter la stratégie suivante:



D		G	T	C	A	G	T	T	T
	0	0	0	0	0	0	0	0	0
G	1	0	1	1	1	0	1	1	1
T	2	1	0	1	2	1	0	1	1
T	3	2	1	1	2	2	1	0	1

Nous voulons parcourir chaque élément de la dernière ligne et les ajouter à une liste en ordre croissant. C'est-à-dire de la valeur minimale jusqu'à la valeur maximale. On voudra garder en mémoire chacune des valeurs et positions de nos éléments de la dernière ligne.

Une fois notre liste prête, on va la parcourir. Pour chaque élément de la liste, on suit les pointeurs et valeurs minimales jusqu'à la première ligne. On rappelle que c'est possible, car chaque élément de la liste contient une valeur et sa position dans le tableau dynamique. De plus, il faut suivre seulement les pointeurs sur la diagonale pour éviter les erreurs. Comme dit précédemment, on garde toujours en mémoire notre meilleur alignement trouvé. Si on trouve un alignement parfait, c'est-à-dire une diagonale contenant que des zéros, alors on peut retourner cet alignement et l'algorithme se termine. Sinon, on continue en gardant en mémoire le meilleur alignement trouvé. On le retourne quand l'algorithme se termine. L'Algorithme peut se terminer lorsqu'on a parcouru toute notre liste et il peut aussi se terminer dans un autre cas que nous allons expliquer.

Les valeurs dans la liste vont être entre 0 et m , où m est le nombre de lignes. Il est inutile de parcourir les valeurs égale ou supérieure à $m-1$ si nous avons des valeurs égale ou inférieure à m dans notre liste. La raison est que les chemins que nous allons trouver à partir de ces éléments ne seront pas meilleurs que les chemins trouvés avec les éléments précédents. Donc, si on rencontre une valeur inférieure ou égale à $m-1$, l'algorithme s'arrête et retourne le meilleur chemin lorsqu'on rencontre dans notre liste un élément de valeur m . Cela revient à dire que $k = m-1$ lorsque nous avons un élément plus petit ou égale à $m-1$ dans la liste.

2) En utilisant uniprot, on trouve que la protéine X est la protéine **40S ribosomal protein S27** dont la fonction est de se lier à un ion de zinc. De plus, selon la bibliothèque nationale de médecine, "des mutations dans ce gène ont été identifiées chez de nombreux patients atteints de mélanome et chez au moins un patient atteint d'anémie de Diamond-Blackfan (DBA). Une expression élevée de ce gène a été observée dans divers cancers humains. Comme cela est typique pour les gènes codant pour des protéines ribosomales, il existe de multiples pseudogènes transformés de ce gène dispersés dans le génome."

Source : <https://www.ncbi.nlm.nih.gov/gene?Db=gene&Cmd=DetailsSearch&Term=6232>