IFT3295 -TP1

Louis-Andre Brassard
Maggie Robert

October 2022

Chevauchement de séquences

1)

La différence entre ce genre d'alignement et l'alignement global se divise en deux points

- Dans le cas de l'alignement local, nous initialisons la première rangée et la première colone a 0 ce qui donne V(i,0) = 0 et V(0,j) = 0 contrairement a l'alignement globale ou on initialise V(i,0) = i et V(0,j) = j

- Pour l'alignement local, on trouve la valeur maximal dans tout le tableau et on part de cette valeur pour et on suit les pointeurs pour retourner au 0 le plus proche. Le trajet parcourus avec les pointeurs nous donne l'alignement local le donc le score est le plus élevé. Pour l'alignement globale, nous remplissons la table de programmation dynamique au complet et on parts de la case V(i,j) pour retourner en suivant les pointeurs vers la case V(0,0).

2)

Comme mentionné plus haut, les valeurs de V(i,0) $\forall i$ et V(0,j) $\forall j$ sont de 0. Ceci s'explique par le fait que l'alignement local optimal peux commencer et terminer n'importe où dans notre mot et pas seulement à l'origine V(0,0).

3)

La fonction de récurences pour ce genre d'alignement local est:

$$V(i,0) = V(0,j) = 0 \qquad \forall i,j$$

$$V(i,j) = max \begin{cases} V(i-1,j-1) + \delta(v_i, w_j) \\ V(i-1,j) + \delta(v_i, -) \\ V(i,j-1) = \delta(-, w_j) \end{cases}$$

où

$$\delta = \begin{cases} 4 & \text{if } v_i = v_j, \\ -4 & \text{if } v_i \neq v_j, \\ -8 & \text{si } v_i \text{ ou } v_j \text{ est un indel} \end{cases}$$

4)

On commence par initialiser les pointeurs des conditions initiales. Pour chaque $V(i, 0)$ les pointeurs point vers le haut. Pour chaque $V(0, j)$ il n'y a pas de pointeurs. Ensuite on remplit la table de programmation dynamique en suivant les formules de récurence montrer plus haut. Après avoir remplit la table au complet, on prend la case de la dernière colonne avec la plus grande valeur (excluant celle de la première ligne), qui se trouve à la ligne k. k représente le nombre de lettres du deuxième mot qui appartiennent à l'alignement. On ajoute à l'envers les $len(mot2) - k$ lettres du deuxième mot et le même nombre d'espaces aux premier mot. Ensuite on suit les pointeurs, en ajoutant à l'envers les lettres de l'alignement, jusqu'à ce que l'on croise une case sans un pointeur dans la première ligne qui se trouve à la colonne l. Finalement on ajoute à l'envers les $l + 1$ premiers lettres du premier mot et le même nombre d'espaces au deuxième mot. Ceci donne le chevauchement suffixe-préfixe optimal entre les deux séquences.

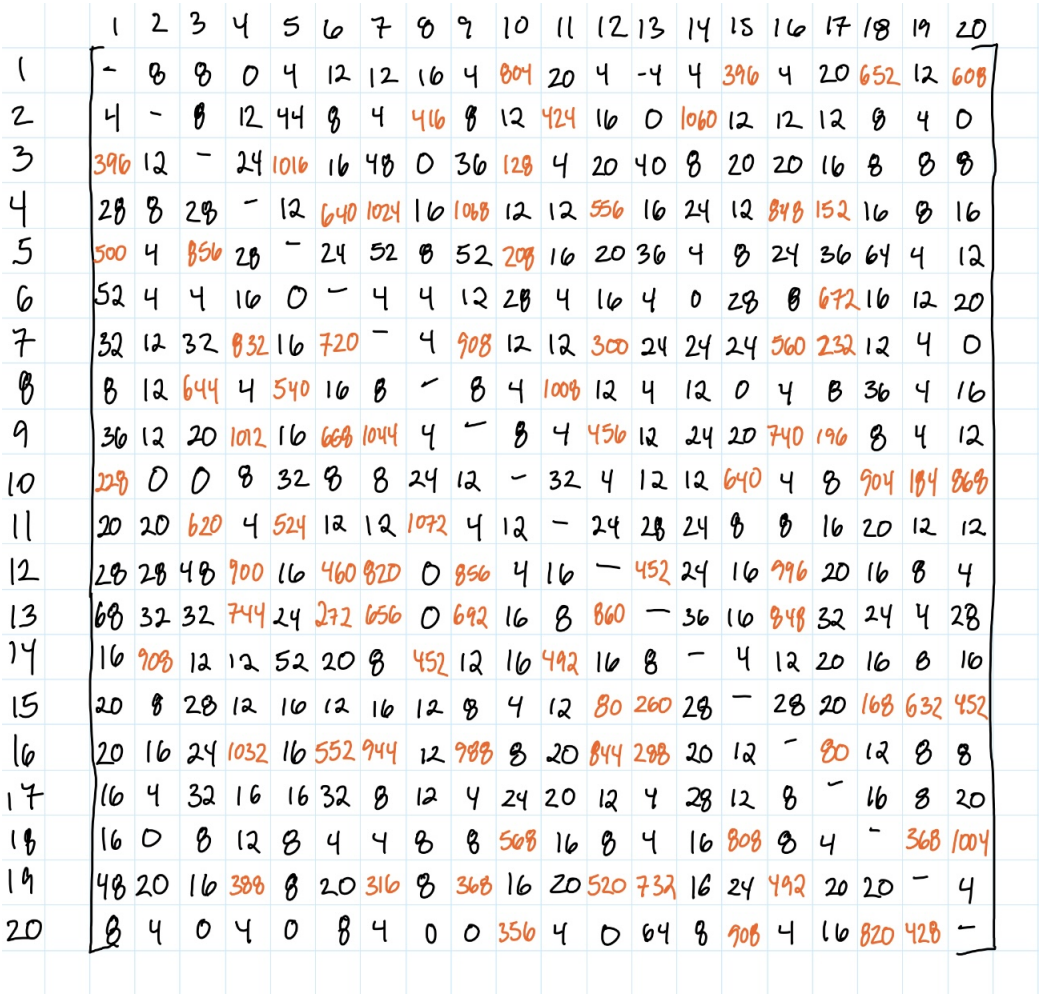
5)

Voir pièce jointe pour code

Assemblage de fragments

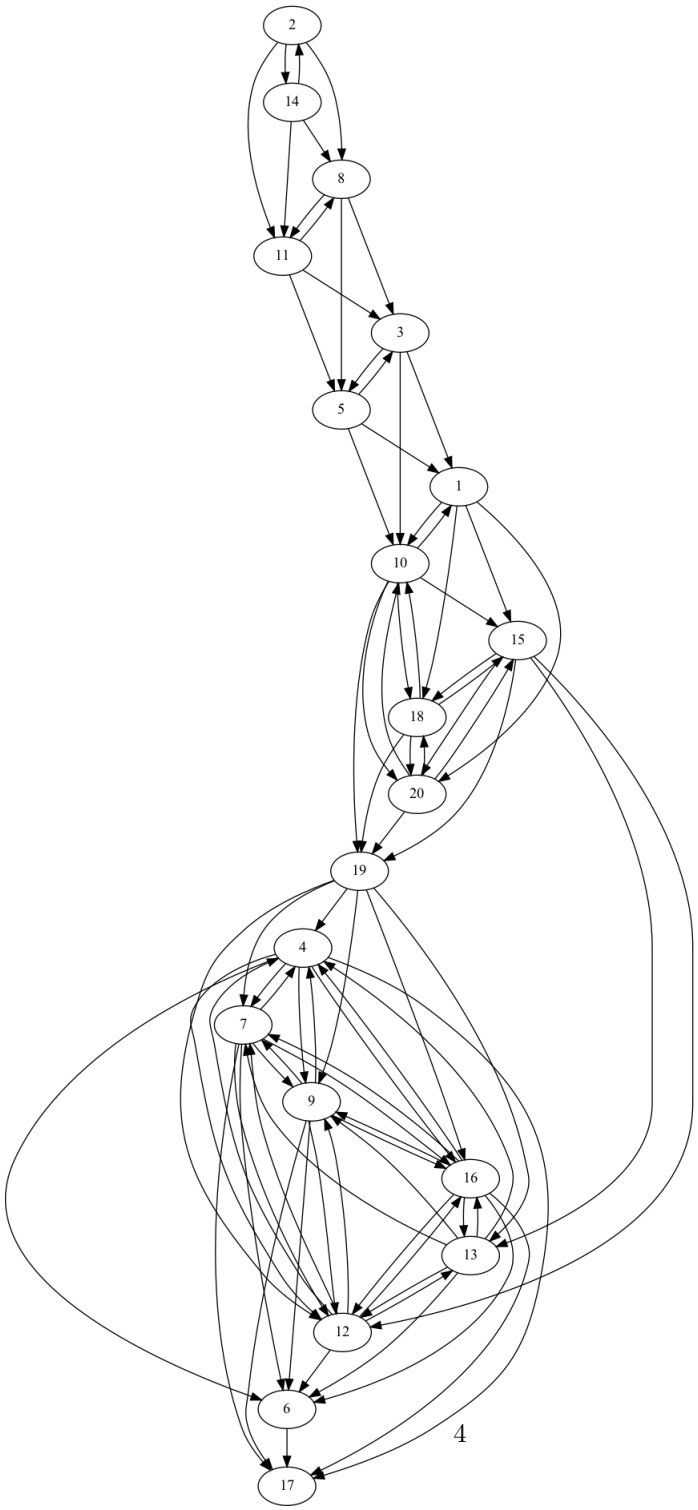
1)

La matrice des scores, avec les scores ≥ 80 en orange.



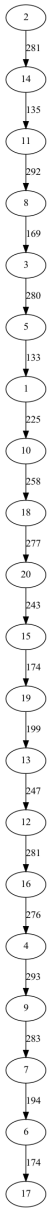
2)

a)



b)

En choisisant les arêtes avec les scores les plus elévés, on obtient les chevauchements suivants:



c)

La séquence finale est ci-dessous. Sa longueur est 1581 caractères.

CTAGGGACTTGGAGAACACAAGTATTATGAAAAGTACTGATGAAA
GTTATTAACAGGTTTCGAAAAATAACTTTACTATCTGACGTGTTGC
TTCTGCCGAGGATGACCGTTATTCCTGGTTTTGCATTTATATTTCA
CGTATGGTTAAATGTGCCAGCGTTGTGGTTTAAAACTAATAGTAAT
AATATGCTTCTTTGTTCAGTTGGCTAGAGATTTACTACATCCGTCC
TTGGAAGAGGAAAAGAAAAAACATAACAAGAAACGCCTAGTACAA
AGTCCAAATTCTTACTTTATGGATGTAAAATGTCCCGGTCAAATTT
GAAATCTTAATTCCTTTACTAAAGAAAATTTCTGAAGGGATTGCTA
GTGTGGTGTGTATAGTTAAGATACATTCGAATCCTCTGTTGAGTAG
AAGTGGGATTACAGAATTGGAAATGTGAGGGACATTTTCATAATAA
ATGTACTGAAACCATTGTAGAAACATTTCGGGGTAGTGAAAACAGG
CCTTCAAAGGAGATATGCCATTTGACTGGACTTGTGGATAATCAAA
ATGTGGATACACAAAATAGAGATGTTCTTTAATTGTAAAATACTCA
CTGGATTTTGACGATGTAGCACAGAAAAAAAATACATTGATTACAC
GTTTTTAAAAATTTTTGGGTCGCTGCTAGAAAAGTTTATGTTACAC
ATGGGGCTTGCTGTTTCATAGCACTGAAGTTAATGATTTTTTTACA
TATTACCTGAAATCTCGAACAGGTCCTGTTTTCCTCTGCTTTTCAT
TTTTAACATTGCCTTTTTTTTTTTTTAGGTTGCTACAAGATAACCA
CGGTTTTCAGCCATGCTCAGACAGTGGTTCTTTGTGTAGGTTGTTC
AACAGTGTTGTGCCAGCCTACAGGAGGAAAGGCCAGACTCACAGAA
GGTATACATTTGTCATTCTCCAACCCAGTGATGAGATTGATGATTA
TAAATGTCTCTATCTTCACTGAAAAGTTTACAGAAATCTTAATGAT
TCCCAAAATAACTTATCTCACACTGGAAGAGTTCAAGTGGATTGGC
AGCAAATCTGAGATCTATTTGGTGTGACCTGGTGAGATCTAAATAT
GGAGTCAGCACATGATTTTTTAAGAGTAATATTGCTAAGTAATATT
GCTAAGTATAGTCTGAAAATACCTCTAATCAAAATTTTTTACTTGA
GAAAAGTATTCAGTATAGTTCCTAAAAATTAAGAGTATATTTCTGG
TATAAAAGGATAAATATTCTGTATATGAGTATTAATCCAATATGCT
TAAAACTTCAGTATTTTACTTAAAAGTACTGTTTGTCATTAAAATT
ATACCAAAGTAGAATGCACTTGTTTAATATACTCTCATGATTCTTT
TGCAGGGTGTTCATTTAGAAGAAAGCAACACTAATGATTCAAACAG
CTTCCTGAATTTTAATTTTGTGTTGTCTCACAGAAAGCCTTATCAT
AAATTCCATAATTCTAATTAATTTACCAAGATAATGTCATTACATT
TGGTTATGTAAGTTATACAGCAGTAATCTCCTATTTTGGTGTCAGT
TTTTCACTAAAGTTTTTAT

Recherche d'introns et Blast

a)

En utilisant le code standard et en décortiquant le brin codant selon les trois cadre nous obtenons les suite d'acides aminé que voici

cadre 1 :
RTKFALPYVS*SCESPFLGTWRTQVL*KVLMKVINRFRKITLLSDVLLLP
RMTVIPVFACIFHVWLNVPALWFKTNSNNMLLCSVG*RFTTSVLGRGK
EKT*KETPSTKSKFLLYGCKMSR*NLKS*FLY*RKFL*GLLVWCV*LRYI
RTLC*VEVGLQNWKCQGHFHNKCTETIVETFRGK*KQAFKGDMPFDC
TCG*SKCGYSK*RCSLIVKYSLDFDDVAQKKNTLITLFLKNFVSLLEKFM
LHMWLAVS*H*SY*FFYILPEMSNRCCFPLLFIFNIAFFFF*FATRSPRFSA
MLRQWFFV*VVQQCCASLQEERPDSQKDYHLAFSNPVMRLMIINVSIFT
EKFKEILMITKITYLSLEEFKWIGSKSEIYLV*PGEI*IWSQHMIFLRVILLS
NIAKYSLKIPLIKIIYLRKVFRIVPKN*EYISGIKG*IICI*VLIQYS*NFSILLK
STFCH*NYSKGRMHLFNILS*FFCRLFI*KKATLMIQTAS*ILILCCLTESLII
NSIILINLPR*CNYIWFCKVYSSNLLFWCQFFNKVLIMGK

Plus longue séquence entre un codon start et un codon stop pour le cadre 1 :
MKVINRFRKITLLSDVLLLPRMTVIPVFACIFHVWLNVPALWFKTNSNNMLLCSVG

cadre 2 :
GQSLPYHMFPDRAKALF*GLGEHKYYEKY**KLLTGFEK*LYYLTCCFC
RG*PLFLFLHVYFTYG*MCQRCGLKLIVIICFFVQLARDLLHPSLEEEKK
KHKKKRLVQSPNSYFMDVKCPGKI*NLNSFTKENFCRDC*CGVYS*DTL
EPSVE*KWDYRIGNVRDIFIINVLKPL*KHFGVSENRHSKEICHLTALVD
NQNVDTQNRDVL*L*NTHWILTM*HRKKIH*LHCF*KILCRC*KSLCYTC
GLLFHSTEVTDFFTYYQKCRTGAVFLCFSFLTLPFFFFSLLQDHHGFQP
CSDSGSLCRLFNSVVPAYRRKGQTHRRIIIWHSPTQ**D**L*MSLSSLKSL
KKS**LPK*LISHWKSSSGLAANLRSIWCDLVRSKYGVST*FF*E*YC*VIL
LSIV*KYL*SKLFT*EKYSE*FLKIKSIFLV*KDK*SVYEY*SNILKTSVFYL
KVLFVIKIIAKVECTCLIYSHDSFADCSFRRKQH**FKQLPEF*FCVVSQK
ALS*IP*F*LIYQDNVITFGFVRYTAVISYFGVSFSIKF*LWA

Plus longue séquence entre un codon start et un codon stop pour le cadre 2 :
MCQRCGLKLIVIICFFVQLARDLLHPSLEEEKKKHKKKRLVQSPNSYFMDVKCPGKI

cadre 3 :
DKVCLTICFLIVRKPFSRDLENTSIMKSTDESY*QVSKNNFTI*RVASAED
DRYSCFCMYISRMVKCASVVV*N****YASLFSWLEIYYIRPWKRKRKNI
KRNA*YKVQILTLWM*NVQVKFEILIPLLKKISVGIASVVCIVKIH*NPLLS
RSGITELEMSGTFS**MY*NHCRNISG*VKTGIQRRYAI*LHLWIIKMWIL
KIEMFFNCKILTGF*RCSTEKKYIDYTVFKKFCVAARKVYVTHVACCFI
ALKLLIFLHITRNVEQVLFSSAFHF*HCLFFFLVCYKITTVFSHAQTVVLC

VGCSTVLCQPTGGKARLTEGLSFGILQPSDEIDDYKCLYLH*KV*RNLN
DYQNNLSLTGRVQVDWQQI*DLFGVTW*DLNMESAHDFFKSNIAK*YC*
V*SENTSNQNYLLEKSIQNSS*KLRVYFWYKRINNLYMSINPIFLKLQYFT
*KYFLSLKL*QR*NALV*YTLMILLQIVHLEESNTNDSNSFLNFNFVLSHR
KPYHKFHNSN*FTKIM*LHLVL*GIQQ*SPILVSVFQ*SFDYGQ

Plus longue séquence entre un codon start et un codon stop pour le cadre 3 :
MILLQIVHLEESNTNDSNSFLNFNFVLSHRKPYHKFHNSN

À l'aide d'un algorithme qui scan les cadres pour trouver codons de départs "M", on cherche le "*" suivant le codon de départ qui sont séparer par un nombre suffisant d'acide aminé soit 30, ce qui correspont à 90 nucléotides. Le résultat de cet algorithme nous donne une suite d'acides aminées dans chaque cadre qui nous indique la protéine que l'on cherche.

Lorsqu'on isole les séquences commencant par un "M" et finissant par un "*" on peux voir que le début de notre gèneX se trouve dans le cadre 2

b)

Pour résoudre ce genre de problème, nous croyons que le meilleur algorithme de programmation dynamique disponible est celui de pondération affine.

Cet algorithme à les conditions initiales et relations de récurences suivantes :

Conditions initiales

$$V(i,0) = F(i,0) = -\rho - i \cdot \sigma$$
$$V(0,j) = E(0,j) = -\rho - j \cdot \sigma$$

Récurences

$$V(i,j) = max \begin{cases} V(i-1,j-1) + \delta(v_i, w_j) \\ E(i,j) \\ F(i,j) \end{cases}$$

$$E(i,j) = max \begin{cases} E(i,j-1) - \sigma \\ V(i,j-1) - \rho - \sigma \end{cases}$$

$$F(i,j) = max \begin{cases} F(i-1,j) - \sigma \\ V(i-1,j) - \rho - \sigma \end{cases}$$

Où ρ est le coût d'ouvertue d'un gap et σ le coût de continuation d'un gap

Pour initier le tableau de (n+1 x m+1) cases, on doit garder en tête 3 valeurs par case soit V(i,j), E(i,j) et F(i,j). Cela nous servira a calculer les case suivante en sachant si nous avons un match, mismatch, que nous commencons un gap ou que nous le poursuivons. Les valeurs de V,E et F sont initialisé selon les règle de conditions initiales. Si elle ne figurent pas dans les conditions initiales elles sont initialisés a -∞. Une fois que le tableau est bien initialisé, on descend dans le tableau en suivant les règles de récurences montrées plus haut en prenant bien soins de mémoriser les pointeurs qui on permis de calculé la valeur observé. Une fois que le tableau est rempli, on part de la case (n+1,m+1) et on regarde la valeur la plus élevé parmis V,E,F et on suit sont pointeur jusqu'à ce qu'on arrive à V(0,0), E(0,0) ou F(0,0).

Nous avons choisi cette façon de faire puisque nous voulons minimiser les introns. Les introns peuvent être représenté par des gaps dans la séquence et la pondérations affine vise à réduire le nombre de gap dans notre alignement en lui donnant un poids élevé pour l'ouverture d'un gap, mais en ne pénalisant pas autant sévèrement la continuation d'un gap. Donc nous avons pensez que cet algorithme serait parfait pour cette tâche.

c)

La protéine X du Dr Osbourne porte le nome de ribosomal protein S27 like [Homo sapiens]. Selon le National library of Science américain, cette protéine sert à "selectively regulates the expression and alternative splicing of inflammatory and immune response genes in thyroid cancer cells"