

UNIVERSITÉ DE MONTRÉAL

IFT 3295 – BIO-INFORMATIQUE

---

## Devoir 1

---

par :  
André Lalonde  
(20024885)

Maude Sabourin  
(p1141140)

2 octobre 2017

## Chevauchement de séquences

1. Quelle est la différence entre un tel alignement et l'alignement global ?

**Réponse :** Un tel alignement recherche le meilleur match au travers d'une séquence sans obligatoirement devoir aligner la séquence entière. Ainsi, pour l'alignement global, chaque indel doit être pénalisé peu importe sa position. L'alignement préfixe-suffixe (et inversement) vise plutôt à s'assurer qu'une séquence commence son alignement à partir du début, et qu'une autre complète l'alignement avec la fin de sa séquence.

Quelles doivent être les valeurs de la première ligne ( $V(0, j) \forall j$ ) ? et celles de la première colonne ( $V(i, 0) \forall i$ ) de la table de programmation dynamique  $V$  ?

**Réponse :** Puisque l'alignement permet d'avoir un préfixe et/ou un suffixe, la première rangée ainsi que la première colonne ne contiennent que des 0. Étant donné que l'on n'a aucune restriction par rapport au nombre de caractères qui doivent être "matchés" dans la séquence, on ne pénalise pas un "décalage initiale", puisque l'on peut démarrer de n'importe quelle des deux séquences.

3. Quelles sont les équations de récurrence à utiliser pour remplir la table de programmation dynamique ?

**Réponse :** Les équations sont

$$\max \begin{cases} V(i-1, j-1) + : \begin{cases} +4 \text{ si } v_i = w_j \\ -4 \text{ si } v_i \neq w_j \end{cases} \\ V(i-1, j) - 8 \\ V(i, j-1) - 8 \end{cases} \quad (1)$$

4. Comment peut-on retrouver l'alignement avec le meilleur chevauchement à partir de la table de programmation dynamique ?

**Réponse :** Premièrement, on cherche dans le tableau la plus grande valeur sur la dernière ligne ou la dernière colonne, qui sera la case de départ. Par la suite, on vérifie le score de la case sur laquelle on se trouve et les trois cases ( $\uparrow, \nwarrow, \leftarrow$ ) pour s'assurer que le score équivaut bien aux équations de récurrence à gauche. Lorsque l'on se déplace dans la direction  $\nwarrow$ , les deux case ont un "match" et on écrit donc les deux caractères de la case  $V(i, j)$ . Lorsque l'on se déplace dans la direction  $\uparrow$ , alors on "match" le caractère de la ligne  $i$  avec un indel. De façon similaire, lorsque l'on se déplace vers  $\leftarrow$ , alors on "match" le caractère de la ligne  $j$  avec un indel. Lorsque l'on arrive sur une case qui vaut 0, on arrête la procédure et on se dirige vers la case  $V(0, 0)$  en alignant avec des indels le reste de la séquence.

5. Voir le fichier TP1.py. Lancer l'application. Peser sur 1, puis entrez le fichier désiré. Le format fonctionnel est d'écrire une séquence par ligne jusqu'à concurrence de deux.

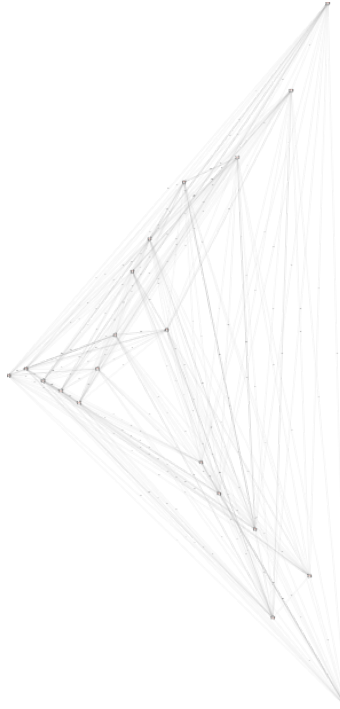
## Assemblage de fragments

1. Pour chaque paire de reads  $\{R_x, R_y\}$ , calculer le score de l'alignement correspondant au chevauchement maximal entre  $R_x$  et  $R_y$ .

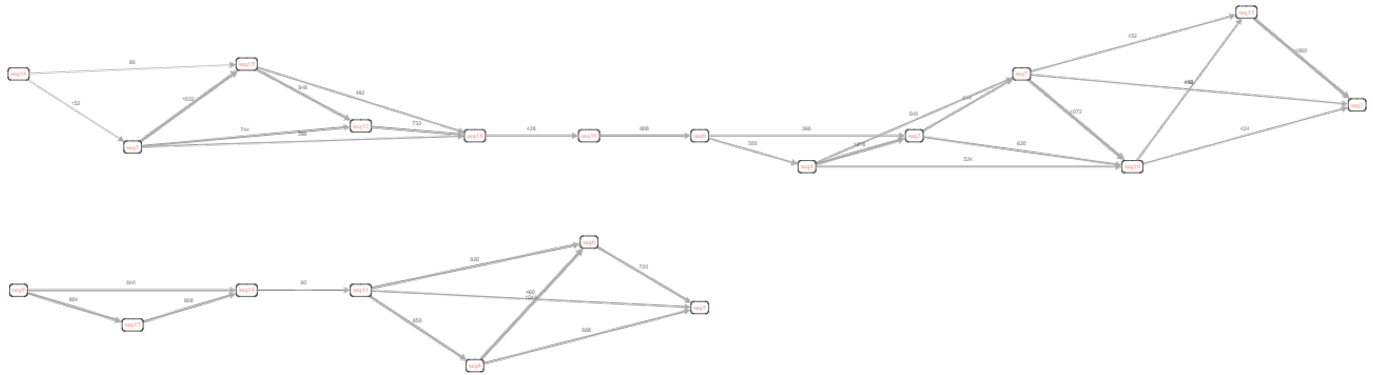
1.1. Voir le fichier TP1.py. Lancer l'application. Peser sur 2, puis patienter. Une fois toute la recherche effectuée, la matrice s'affichera.

[	0.	0.	396.	28.	500.	0.	24.	0.	0.	8.	0.	0.	68.	16.	0.	20.	0.	0.	48.	0.]
[	8.	0.	12.	0.	0.	24.	0.	0.	0.	0.	0.	0.	32.	0.	0.	16.	0.	16.	20.	4.]
[	0.	0.	0.	28.	0.	28.	0.	644.	0.	0.	620.	0.	0.	12.	0.	24.	32.	0.	16.	0.]
[	0.	12.	0.	0.	28.	24.	0.	0.	0.	4.	0.	0.	744.	0.	0.	1032.	0.	0.	388.	0.]
[	0.	44.	1016.	0.	0.	0.	0.	540.	0.	0.	524.	0.	0.	52.	0.	0.	0.	0.	8.	0.]
[	4.	0.	0.	0.	16.	0.	0.	28.	0.	28.	32.	0.	0.	0.	28.	16.	36.	16.	0.	0.]
[	0.	16.	32.	32.	12.	720.	0.	0.	0.	12.	0.	0.	12.	12.	24.	0.	28.	12.	0.	0.]
[	16.	416.	0.	16.	0.	0.	4.	0.	16.	4.	1072.	0.	0.	452.	0.	12.	12.	0.	8.	0.]
[	12.	8.	16.	64.	20.	668.	1044.	0.	0.	0.	20.	0.	28.	16.	20.	48.	52.	0.	12.	4.]
[	0.	8.	12.	0.	20.	0.	0.	0.	12.	0.	16.	0.	0.	0.	640.	8.	0.	904.	0.	16.]
[	20.	424.	0.	12.	0.	0.	8.	0.	0.	0.	0.	0.	0.	492.	0.	20.	20.	0.	20.	0.]
[	20.	12.	20.	44.	20.	460.	820.	8.	856.	4.	8.	0.	36.	0.	0.	76.	0.	16.	16.	0.]
[	0.	0.	40.	0.	36.	12.	0.	4.	0.	16.	28.	0.	0.	0.	0.	0.	0.	20.	732.	64.]
[	0.	1060.	0.	24.	0.	20.	0.	0.	0.	12.	0.	24.	36.	0.	0.	20.	28.	32.	16.	0.]
[	24.	12.	16.	32.	12.	0.	0.	16.	0.	0.	20.	80.	28.	16.	0.	32.	24.	0.	32.	0.]
[	0.	0.	0.	0.	24.	0.	40.	0.	0.	0.	0.	0.	848.	0.	0.	0.	0.	16.	492.	0.]
[	20.	12.	0.	152.	36.	0.	0.	0.	0.	12.	0.	28.	32.	0.	0.	80.	0.	28.	20.	0.]
[	4.	0.	4.	12.	8.	0.	0.	8.	8.	0.	8.	0.	0.	0.	808.	0.	0.	0.	0.	0.]
[	0.	0.	0.	0.	0.	16.	12.	0.	0.	8.	0.	0.	0.	0.	0.	0.	0.	8.	0.	428.]
[	608.	0.	8.	16.	12.	12.	12.	16.	0.	0.	12.	12.	0.	16.	4.	8.	20.	12.	0.	0.])

2. En déduire le graphe orienté de chevauchement  $G = (V, E)$ .



a) Quel effet à un seuil minimum de score de 80 sur le graphe résultant ?

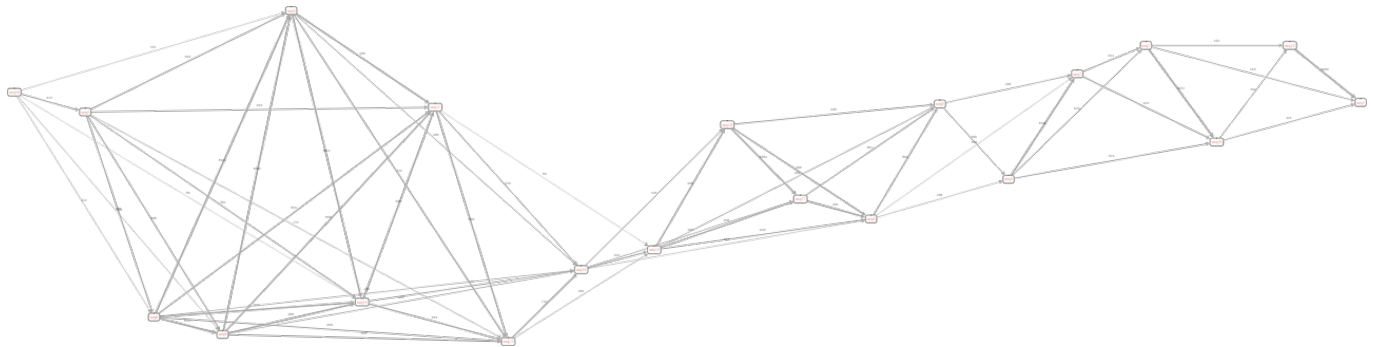


**Réponse :** Comme on peut le constater en comparant les deux figures précédentes, l'ajout d'un seuil a eu pour effet de diluer la grande majorité des arêtes de  $V$ . Cela a transformé le **tas** de séquences en une **série de séquences** presque linéaire, ce qui nous permet de distinguer les **patterns** dans les séquences.

b) Sachant que le read @READS\_2 est forward, déduisez-en l'ensemble des reads reverse.

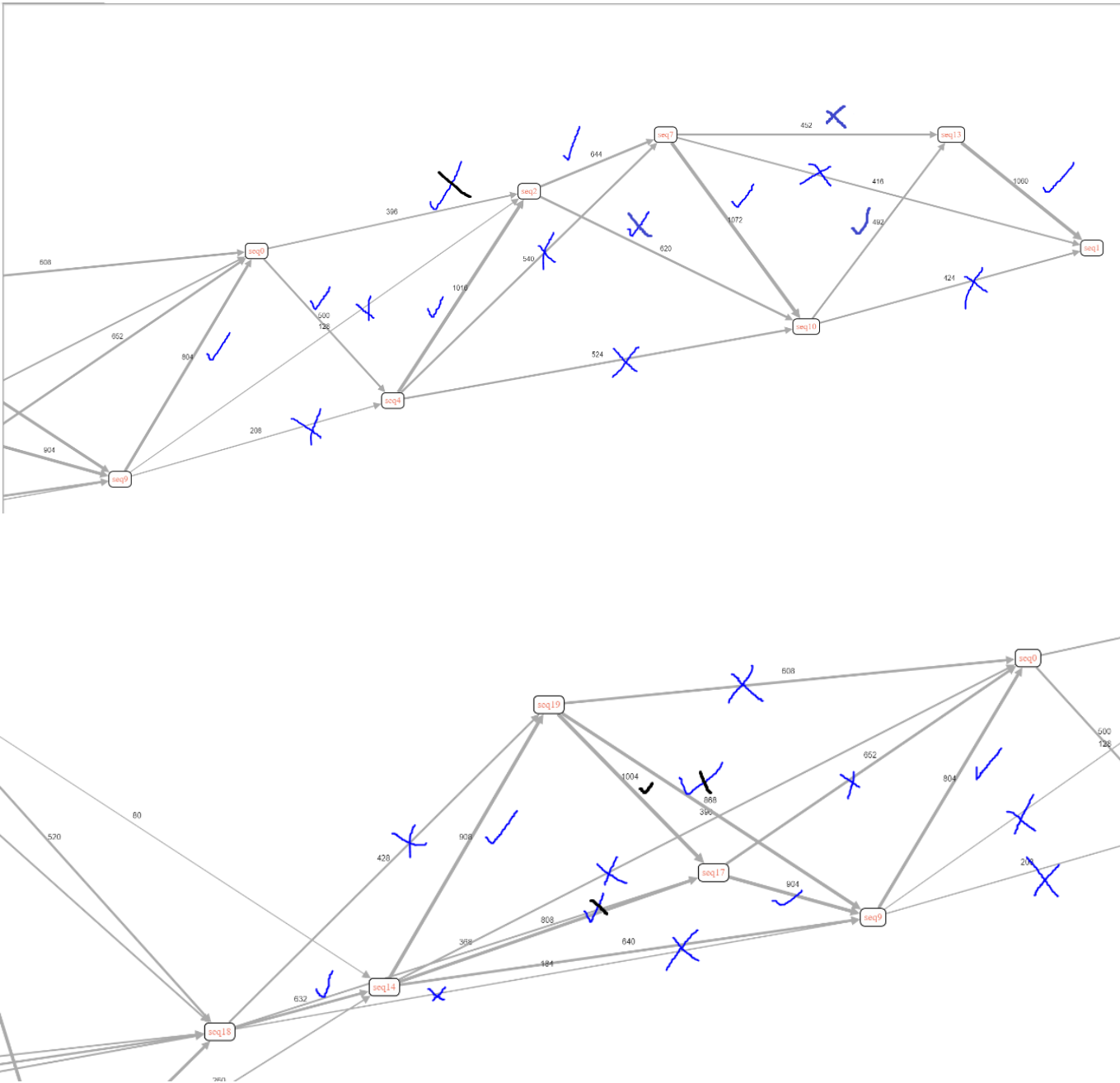
**Réponse :** L'ensemble des reads reverse est : 5,6,8,9,11,14 et 17. La séparation des reads en deux graphes ne laisse aucun doute sur l'appartenance de chacun à un groupe distinct.

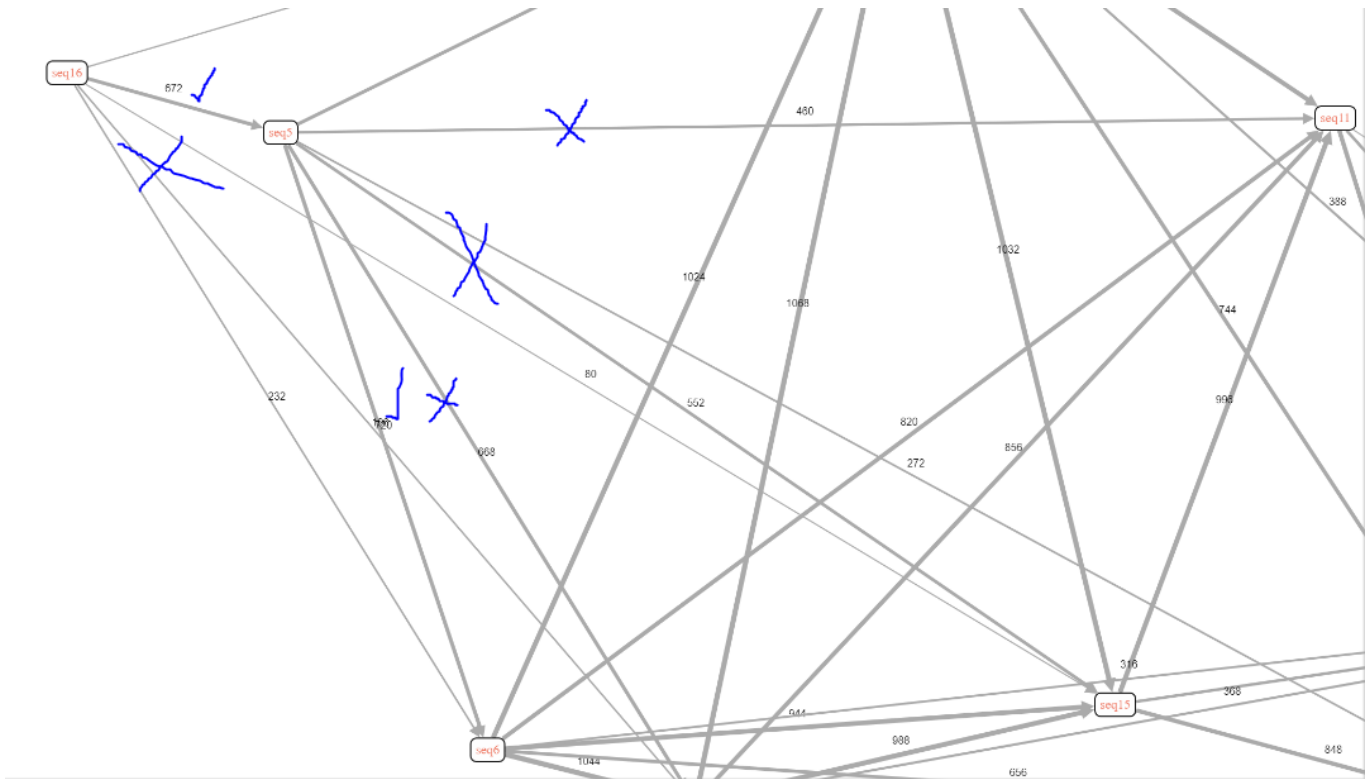
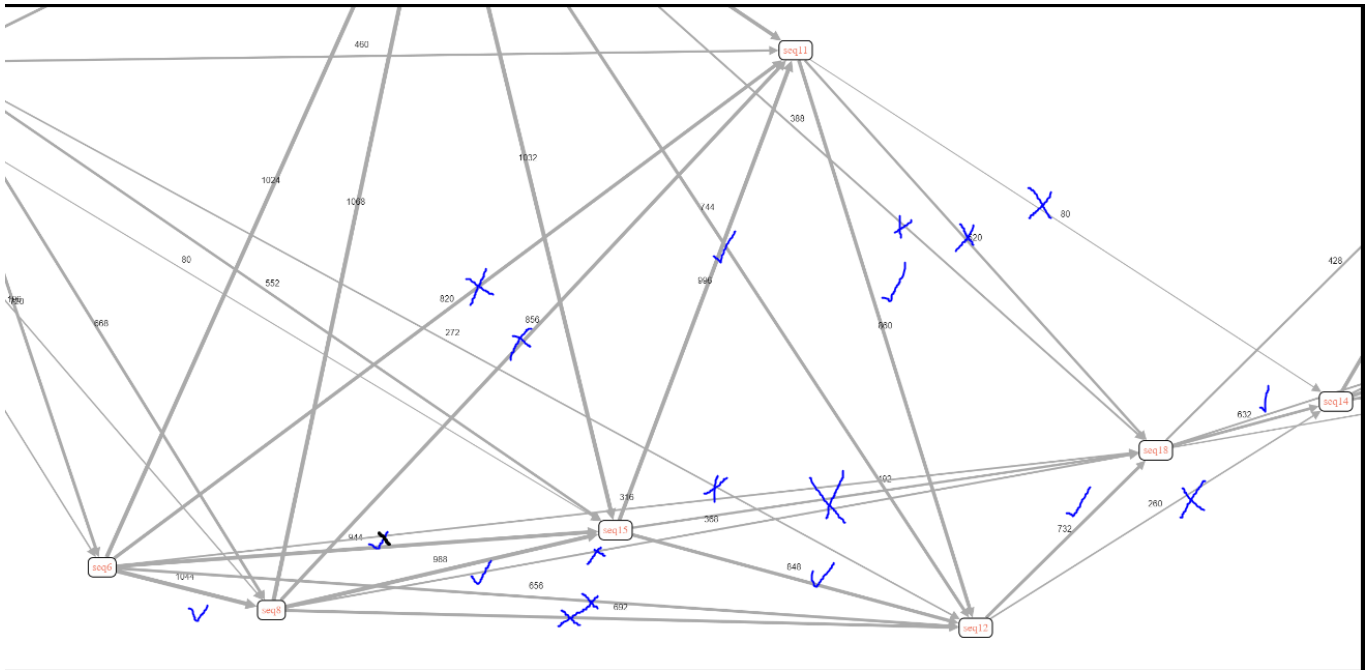
3. a) Construire le nouveau graphe de chevauchement en remplaçant les séquences identifiées comme étant des reads reverse par leur complément  $X_c^r$ .

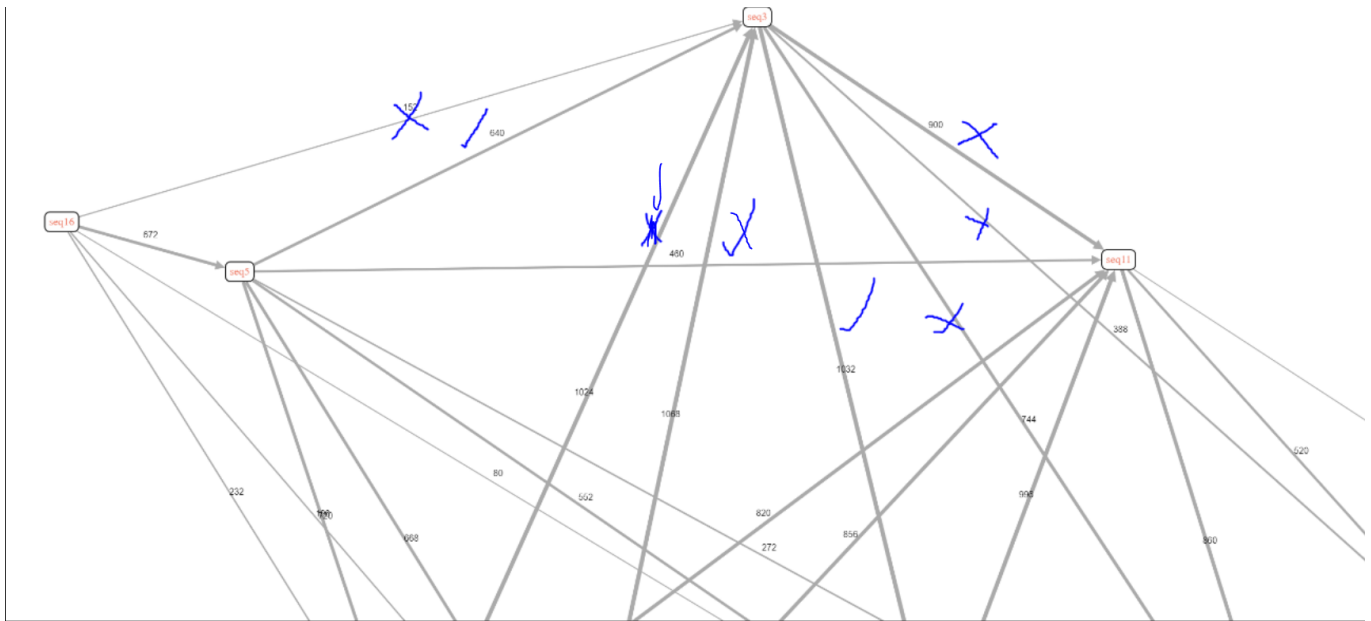


Le graphe est maintenant connexe, ce qui veut dire que l'on peut le traverser d'un bout à l'autre par un chemin. De plus, la forme spéciale du graphe aide à voir rapidement les reads qui ont beaucoup de liens. b)

Appliquez la réduction transitive au graphe ci-dessus.







Ordre des reads : 16->5->6->8->3->15->11->12->18->14->19->17->9->0->4->2->7->10->13->1

Longueur chevauchement :

- Chevauchement 16->5 : 174
- Chevauchement 5->6 : 194
- Chevauchement 6->8 : 283
- Chevauchement 8->3 : 293
- Chevauchement 3->15 : 276
- Chevauchement 15->11 : 281
- Chevauchement 11->12 : 247
- Chevauchement 12->18 : 199
- Chevauchement 18->14 : 174
- Chevauchement 14->19 : 243
- Chevauchement 19->17 : 277
- Chevauchement 17->9 : 258
- Chevauchement 9->0 : 225
- Chevauchement 0->4 : 133
- Chevauchement 4->2 : 280
- Chevauchement 2->7 : 169
- Chevauchement 7->10 : 292
- Chevauchement 10->13 : 135
- Chevauchement 13->1 : 281

c) D duire la s quence du fragment g nomique s quenc  et sa longueur.

La s quence est : CATTCTCCAACCCAGTGATGAGATTGATGATTATAAATGTCACTATCTTCACT-

GAAAAGTTTAAAGAAATCTTAATGATT

ACCAAATAACTTATCTCTCACTGGAAGAGTTCAAGTGGATTGGCAGCAAATCTGAGATCTATTTGGTGTC

AGTCAGCACATGATTTTTTTAAGAGTAATATTGCTAAGTAATATTGCTAAGTATAGTCTGAAAATACCTCTA

AGTATTCAGAATAGTTCCTAAAAATTAAGAGTATATTTCTGGTATAAAAGGATAAATATTCTGT

ATATGAGTATTAATCCAATATGCTTAAAACCTTCAGTATTTTACTTAAAAGTACTGTTTGTCATTAAAATTATA

CATGATTCTTTTGCAGGGTGTTTCATTTAGAAGAAAGCAACACTAATGATTCAAACAGCTTCCTGAATTTTA

avec une longueur de 526. Les calculs ont été effectués dans le document calculfinal et le résultat complet se trouve dans le fichier totalefinale



## Recherche d'introns et Blast

1. Identifier la protéine X au sein de la région génomique.

a) Dans quel cadre de lecture se trouve le codon start de la séquence protéique ?

**Réponse :** Dans le deuxième brin codant. En effet, on remarque la combinaison ATG(M), suivi de TGC(C), CAG(Q), etc. à partir de la séquence commençant par GGA

b) Décrivez un algorithme de programmation dynamique qui vous permet de retrouver les différents fragments de la protéine X au sein de la séquence nucléotidique.

**Réponse :** Nous pouvons utiliser un algorithme d'**alignement total**. Dès que l'on obtient un mismatch ou un indel, on assigne cela à **un intron**. Si l'on doit assigner un extra à gauche ou bien à droite, on l'assigne à **un exon**.

c) En déduire les intervalles de positions contenant les exons ainsi que la séquence de l'ARNm mature.

**Réponse :** Pour trouver la séquence d'ARN messenger après épissage, il suffit de prendre le geneX et de le transformer à l'aide de la table génétique. En ce qui concerne les intervalles de positions, nous avons d'abord la séquence [1-55], puis la séquence [277-285] qui donne la séquence MCQRCGLKLIVICFFVQLARDLLHPSLEEEKKKHKKKRLVQSPNSYFMDVKCPGCSFRRKQH.

2. En vous servant de l'outil uniprot, identifiez le nom de la protéine X ainsi que sa fonction.

**Réponse :** La protéine en question est une **Ribosomal protein S27 homo sapiens (Human)**. Elle sert à contraindre des acides désoxyribonucléique ainsi que des acides ribonucléiques dans des ribosomes, ou elle lie des ions de Zinc.