# IFT3295 - TP1

Assemblage de séquences

17 septembre 2019

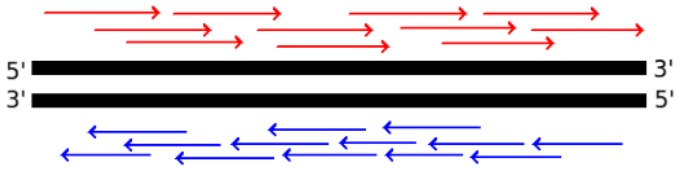
## Évaluation

- Ce TP est à faire seul ou en équipe de deux (pas d'équipe de trois).
- Vous devez le rendre au plus tard le Mercredi 2 Octobre avant la démo.
- Deux fichiers à remettre : le code pour les programmes demandés et un pdf pour les autres questions.
- Utiliser le système de remise du DIRO tel qu'indiqué sur la page des démonstrations. Une version papier des questions écrites est aussi correcte.
- Votre programme doit être un code source qui est facilement exécutable sur le terminal (console) sur linux.
- Votre code doit être bien lisible et compréhensible (bonne indentation, noms de variables et de fonctions représentatifs, commentaires, etc.).
- Langage de programmation de votre choix parmi les langages usuels : Java, JavaScript, Python, C, C++, Perl, etc. Svp écrivez en commentaire au début de votre code les instructions pour l'exécuter.

## Mise en situation

Vous commencez un stage dans le laboratoire du Dr Osborn. Ce dernier étudie activement un nouveau virus et a récemment identifié une protéine humaine X impliquée dans la synthèse de protéines virales. Il décide donc d'étudier cette nouvelle protéine pour mieux comprendre son rôle dans les infections virales. Pour ce faire, Dr Osborn séquence la région génomique

contenant le gène de la protéine X . La technologie qu'il utilise produit des petits fragments chevauchant appelés *reads* qui recouvrent la région d'intérêt, dans les deux sens de lecture (voir figure suivante).




En temps normal, les séquences des *reads* sont retournées dans deux fichiers séparés : l'un contenant le sens *forward* (5'-3') de lecture et le second, le sens *reverse* (3'-5') du brin complémentaire. Malheureusement, dû à une maladresse, les *reads reverse* et les *reads forward* se retrouvent dans le même fichier de reads. Le Dr Osborn requiert votre assistance pour identifier les *reads reverse* et assembler tous les reads afin d'obtenir la séquence nucléotidique de la région génomique.

## Chevauchement de séquences (30pts)

Tout au long de ce TP, veuillez considérer les pondérations suivantes :

— "match" : +4

— "mismatch" : -4

— "indel" (insertion ou suppression) : -8

Pour assembler les reads, vous devez considérer pour chaque paire de séquence $X_i, X_j$ , le meilleur alignement tel que le chevauchement (tel illustré sur la figure suivante) entre $X_i$ et $X_j$ soit de score maximal.

Afin de développer un algorithme pour le problème du chevauchement maximal entre deux séquences, répondez aux questions suivantes :

1. Quelle est la différence entre un tel alignement et l'alignement global ?
2. 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 ? Justifiez votre réponse.

3. Quelles sont les équations de récurrence à utiliser pour remplir la table de programmation dynamique ?
4. Comment peut-on retrouver l'alignement avec le meilleur chevauchement à partir de la table de programmation dynamique ? Vous devez décrire la procédure entière pour retrouver l'alignement.
5. Déduisez-en un algorithme pour trouver l'alignement maximisant le chevauchement entre deux séquences et son score en fonction des coûts donnés plus haut. Vous devez remettre un code qui retourne pour une paire de séquences : le score, l'alignement correspondant et la longueur du chevauchement. Votre programme doit prendre comme argument un fichier contenant les deux séquences.

# Assemblage de fragments (45pts)

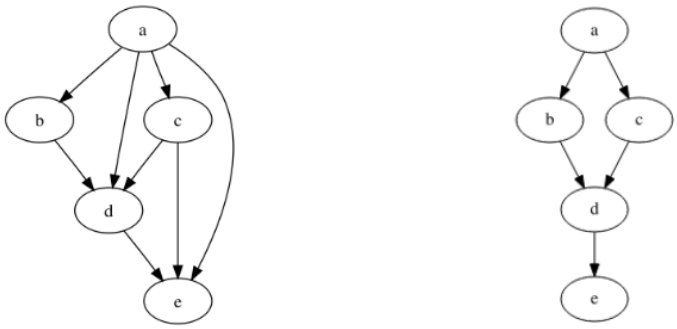
En vous servant de votre algorithme précédent, vous devez identifier les reads *reverse* puis procéder à l'assemblage des séquences. À cet effet, vous disposez d'un fichier *reads.fq* qui contient les séquences et identifiants de 20 reads en format FASTQ. Une description du format FASTQ est disponible sur wikipédia.

1. Pour chaque paire de reads $R_i, R_j$ , calculer le score de l'alignement correspondant au chevauchement maximal entre $R_i$ et $R_j$ . On vous demande une matrice 20x20. Notez que la matrice n'est pas symétrique et que sa diagonale est nulle.
2. En déduire le graphe orienté de chevauchement $G = (V, E)$ où $V$ désigne l'ensemble des reads. Il existe une arête $e_{ij} = (R_i, R_j) \in E$ entre $R_i$ et $R_j$ s'il existe un chevauchement entre un suffixe de $R_i$ et un préfixe de $R_j$. Pour vous aider à répondre aux questions suivantes, vous pouvez utiliser cet outil de visualisation : https://mrnoutahi.com/dagViz/ ou un autre outil de visualisation de graphes, comme graphviz.
   (a) Afin de ne considérer que les chevauchements pertinents, filtrez les alignements par leurs scores, en considérant un seuil minimum de 80 en dessous duquel les chevauchements sont ignorés. Quel effet a ce seuil sur le graphe résultant ?
   (b) Sachant que le read $@READS\_2$ est définitivement sur le brin *forward*, déduisez-en l'ensemble des reads *reverse*.

3. Puisque vous venez d'identifier les reads *reverse* , vous pouvez remplacer leur séquences par les séquences complémentaires inverses correspondantes. À titre d'exemple, la séquence complémentaire de $X = ACTGCAA$ est $X_c = TGACGTT$ et la séquence complémentaire inverse est $X_c^r = TTGCAGT$ .

   (a) Construire à nouveau le graphe de chevauchement avec les nouvelles séquences en applicant le seuil précédent. Que remarquez vous ?

   (b) Afin d'assembler les reads, vous devez vous servir du graphe de chevauchement. Cependant, ce dernier contient plusieurs arêtes "inutiles" et peut donc être simplifié. Vous utiliserez le principe de la réduction transitive (illustrée sur la figure suivante) qui enlève les arêtes entre deux noeuds, tant qu'il existe encore un chemin simple reliant ces deux noeuds. On vous demande l'ordre des reads dans le graphe réduit, ainsi que la longueur du chevauchement entre deux reads consécutifs.




   (c) Déduire la séquence du fragment génomique séquencé et sa longueur.

# Recherche d'introns et Blast (25pts)

Grâce à votre aide, le Dr Osborn a finalement obtenu la séquence génomique désirée (*sequence.fasta*). À partir d'une autre expérience, il a réussi à identifier la séquence protéique du gène X (*geneX.fasta*). Il désire maintenant identifier la séquence de l'ARNm du gène ainsi que ses introns.

1. Identifiez la position de la protéine X au sein de la région génomique. Pour ce faire traduisez d'abord la séquence nucléotidique de *sequence.fasta* dans les trois différents cadres de lecture, en considérant qu'il s'agit du brin codant. Utilisez le code génétique standard (disponible ici). Veuillez noter que les nucléotides et les acides aminés sont toujours représentés par un seul caractère dans les séquences et que le signal de terminaison de la traduction (stop) peut être représenté par une étoile (*). Répondez aux questions suivantes :
   (a) Dans quel cadre de lecture se trouve le codon start de la séquence protéique ?
   (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. Vous ne devez considérer que les plus longs fragments qui ne se chevauchent pas. Comme indice, vous pouvez supposer qu'il ne devrait pas avoir de mismatch entre les fragments de la séquence protéique et la traduction des régions nucléotidiques correspondantes.
   (c) En déduire les intervalles de positions contenant les exons (au sein de la séquence génomique) ainsi que la séquence de l'ARNm mature (après épissage).
2. En vous servant de l'outil Blastp et/ou de uniprot , identifiez le nom de la protéine X ainsi que sa fonction.