UNIVERSITÉ DE MONTPELLIER
PLACE EUGÈNE BATAILLON 34095 MONTPELLIER CEDEX 5
Master Bioinformatique – 2023/2024

Développement opérationnel avancé – Projet

Mapping de données issues de Nouvelles Générations de Séquenceurs (NGS) sur un génome de référence.

Alban MANCHERON

1 Objectif

L'objectif du projet est de définir une stratégie de « *mapping* » de courtes séquences génomiques – appelées « *reads* » – sur un génome de référence donné. Les *reads* sont supposés obtenus par une technique de séquençage haut-débit (*Illumina*, 454, ...) de données génomiques ou transcriptomiques. Ainsi, ces *reads* peuvent comporter des erreurs de séquençages ou des variations biologiques (petites mutations, translocations, copies variables de répétitions en tandem, présence d'introns dans le cas des données transcriptomiques, ...) par rapport au génome de référence.

La méthode que nous vous proposons d'implémenter consiste à considérer les placements exacts des sous-mots de longueur k (appelé k-mers) de chaque *read* sur le génome complet et de définir des critères algorithmiques permettant de localiser la ou les occurrences des *reads* sur le génome, en analysant les différentes situations pouvant se produire :

- le cas le plus trivial correspond à la situation où tous les k-mers du *read* analysé sont localisés de façon unique, dans le bon ordre et sur le même brin du génome, alors le *read* a une occurrence unique exacte à la position définie par l'occurrence du premier k-mers du *read* sur le génome. Cette situation est *a priori* très rare, bien évidemment.
- une première variante de ce cas correspond à la situation où les *reads* sont localisés sur le brin complémentaire inversé.
- un cas relativement trivial est similaire aux deux premiers, à ceci près que certains k-mers ont plusieurs occurrences, mais qu'il n'existe qu'une seule suite ordonnée valide de positions des k-mers. Cette situation est *a priori* plus réaliste que la précédente mais demeure un cas encore trop particulier.
- un cas plus fréquent correspond à la présence d'une petite mutation ou d'une erreur de séquençage dans le *read*. Deux cas de figure se présentent alors :
  - Supposons que cette mutation ou erreur survienne vers le milieu du *read*. Alors les premiers et derniers k-mers du *read* peuvent permettre d'une part de positionner correctement le *read* sur le génome, mais également de diagnostiquer la variation observée. En effet, si les k-mers du milieu (qui ne sont pas positionnés « au bon endroit », voire pas positionnés du tout, sur le génome de référence) ont un nombre d'occurrences similaire aux k-mers correctement positionnés, alors la variation entre le *read* et le génome est probablement d'origine biologique. Au contraire, si les k-mers du milieu ont un nombre d'occurrence très en dessous de celui des k-mers correctement positionnés, alors il s'agit vraisemblablement d'une erreur de séquençage. Dans tous les cas, le nombre de k-mers qui ne sont pas correctement positionnés ainsi que le positionnement des k-mers aux extrémités permettra de définir, par un simple calcul, l'origine de la variation (insertion, suppression, substitution). Le nombre d'occurrences d'un k-mer est appelé son « support ».
  - Supposons maintenant que cette mutation ou erreur survienne vers une des extrémités du *read*. Alors selon que le nombre de k-mers correctement localisés est représentatif ou pas il sera possible ou non de localiser correctement le *read*. Il sera toujours possible d'identifier s'il s'agit d'une variation d'origine biologique ou bien d'une erreur, cependant il ne sera pas possible de diagnostiquer plus finement la variation.
- d'autres cas de figure sont possibles et correspondent à des situations « topographiques » des localisations des k-mers sur le génome de référence. Ces situations sont nombreuses, plus complexes et ne font pas l'objet de ce projet. Cependant rien ne vous interdit d'en évoquer quelques unes avec les solutions que vous pourriez mettre en place, et il est fortement conseillé de structurer votre code de sorte que ces cas puissent être facilement intégrés dans votre analyse.

2 Mise en œuvre

Pour élaborer votre méthode de *mapping*, il est nécessaire de pouvoir lire et analyser des fichiers descriptifs de séquences génomiques comme de données de séquençage.

La première étape du projet consiste donc à définir une API permettant de manipuler des fichiers FASTA/FASTQ et nettoyer des *reads*, en C++.

Dans un second temps, il vous faudra définir une structure de donnée permettant d'indexer des mots de longueur fixe ($k > 0$) à partir d'un texte t de longueur n et capable de répondre efficacement aux questions suivantes :

- Étant donné un mot w de longueur k, est-il présent (ou son complémentaire inversé) dans le texte indexé t ?
  - le cas échéant, combien de fois apparaît-il (quel est son support), à quelles positions et sur quel brin ?
- Quel est le mot de longueur k présent à la position i ($1 \leq i \leq n$) dans le texte t ?

Dans le choix de la structure, vous veillerez bien à prendre en compte le fait que :

1. le texte t est formé sur un petit alphabet (`A`, `C`, `G` et `T`).
2. le texte t peut-être très grand ($n > 10^9$)
3. une structure de donnée peut-être constituée de plusieurs structures distinctes.

Enfin, vous implémenterez votre solution algorithmique de *mapping* d'un fichier de *reads* sur un génome de référence. La sortie de votre algorithme sera **au minimum** la localisation des *reads* sur le génome.

3 Évaluation

Toute méthode nécessite d'être testée et évaluée, de même que nous nous devons d'évaluer votre travail.

3.1 Validation de votre méthode

Ainsi, vous comparerez les résultats et performances de votre algorithme de *mapping* avec les principaux outils disponibles (les plus couramment utilisés).

Vous devrez présenter vos jeux d'essais, justifier vos choix en mettant clairement en avant les points positifs comme les points négatifs de votre méthode par rapport à d'autres méthodes renommées (nous n'espérons pas que votre outil soit meilleur. Si c'est le cas, c'est que soit vos jeux d'essais sont irréalistes, soit votre travail n'est pas vôtre, ou soit vous avez une brillante carrière de chercheur devant vous).

Vous veillerez à mesurer et à présenter les complexités théoriques (spatiale et temporelle) de votre algorithme, ainsi que les consommations réelles observées lors de vos expérimentations.

3.2 Validation de votre travail

Vous devrez fournir un rapport d'une quinzaine de pages expliquant vos différents choix stratégiques, le travail que vous avez réalisé, les pistes que vous pourriez développer si vous en aviez le temps, votre protocole expérimental de comparaison de votre méthode aux autres méthodes et les résultats obtenus.

Vous devrez également présenter votre travail lors de la dernière séance de TP pendant un petit entretien de 20 minutes, et notamment les résultats que vous aurez obtenus sur les jeux d'essais qui vous seront communiqués la semaine précédent votre évaluation.

L'objet ne sera pas d'évaluer la qualité ou l'efficacité de votre code, mais votre degré de compréhension des concepts abordés tout au long de cet enseignement.

Bon Courage. . .