Structures de données pour l'indexation à large echelle de donnée séquençage longue portée

Antoine Limasset

21 septembre 2021

Contexte

Vingt ans après les débuts du séquençage du génome humain, les technologies de séquençage ont successivement transformé la pratique de la biologie. Moins de dix ans plus tard, le séquençage de deuxième génération (Illumina) rend abordable le séquençage de génomes complets en réduisant les coûts de plusieurs ordres de grandeur (10 000 dollars pour un séquençage humain). Aujourd'hui les données de troisième génération (ONT, PacBio) proposent de nouvelles avancées. Les reads (séquences produites par le séquenceur) de troisième génération peuvent être 100 à 1000 fois plus longs que les reads de deuxième génération ($\approx$ 100 paires de base). Ces long reads permettent aux analyses d'éviter la plupart des problèmes liés aux répétitions génomiques et sont aux centres d'applications à fort impact : assemblage de génomes presque complets, études des variants structuraux, études des régions complexes...De plus, ces technologies n'ont pas d'étape d'amplification et n'ont donc pas les biais propres aux séquençage de seconde génération.En revanche, ces technologies présentent des taux d'erreurs importants pouvant dépasser 10%. Le traitement de ces reads radicalement différents nécessite le développement de nouvelles méthodes pour en tirer parti efficacement.

Représentation graphique d'un génome bactérien assemblé à partir de données de seconde génération (gauche) et de troisième génération (droite). L'assemblage de seconde génération est fragmenté a causes des répétitions génomiques contrairement a l'assemblage de droite qui représente le génome en une seule séquence. Extrait de `albertsenlab.org/what-is-a-good-genome-assembly/`.

Projet

Quelle que soit l'application : assemblage, alignement, détection de variant ... ; un besoin fondamental à la comparaison de séquences est de localiser les occurrences d'un motif au sein d'un jeu de donnée. Une approche efficace pour traiter les donnée de deuxième génération est d'indexer les k-mers (mot de taille k, où k est un paramètre de la méthode) apparaissant dans le jeu de données, et de leur associer les informations utile a l'analyse (nombre d'occurrences, positions,...). De nombreux travaux ont proposé des structures de données capables d'indexer des séquençage de deuxième génération et répondre à ce genre de requêtes avec un débit important et une utilisation mémoire faible [3, 2]. En raison des différences fondamentales entre

ces deux types de données (longueur, taux d'erreur, nombre de reads), ces approches ne sont pas applicables directement au séquençage de troisième génération.

Le but de ce stage est de développer un index capable d'associer à un k-mer les séquences dans lesquelles il apparaît au sein d'un séquençage de troisième génération.

Quelques chiffres pour appréhender l'envergure du problème dans le cas d'un séquençage humain avec un couverture de 40X avec des séquences de l'ordre de 10 000 bases.

— Plus de 10 millions de séquences (contenant chacun $\approx$ 10 000 mots)
— Plus de 10 milliards de k-mers distincts vus dans le jeu de données
— Les k-mers peuvent être présents dans des dizaines (voire des centaines) de séquences.

Utiliser une table de hachage standard associer une liste de séquence à chaque k-mer nécessiterait une quantité de mémoire de l'ordre du terabyte de RAM. L'enjeu de stage est donc principalement une problématique de passage a l'échelle.

Pour cela le·a candidat·e devra s'appuyer sur des structures de donnée efficaces (fonction de hachage parfaite, adressage ouvert, bitmap…) probabilistes ou déterministes (Filtre de bloom, count min sketch, Quotient filter…) pour proposer un index rapide avec une consommation mémoire modérée.

Pour diminuer davantage l'empreinte mémoire d'un tel index, le·a candidat·e pourra utiliser des techniques de compression permettant un encodage efficace des données associés aux kmers (delta encoding [1], Run-length encoding, Arithmetic coding …)

En raison de la quantité de données à traiter, une attention particulière devra être donnée a l'implémentation efficace des structures proposées via des langages compilés comme C/C++ ou Rust, l'utilisation de plusieurs coeurs ou de d'autres formes de parallélisme comme le SIMD.

Le stage sera donc l'occasion d'étudier en profondeur des aspects fondamentaux tels que les algorithmes de compression ainsi que les structures de données tout en gardant un aspect pratique via le développement des structures. Ce stage s'intègre dans un projet de recherche financé sur quatre ans, le·a candidat·e bénéficiera du soutien des cinq membres du consortium spécialistes dans les structures de données et l'utilisation des données de troisième génération. De plus, au sein de ce projet de recherche, l'équipe d'accueil dispose d'un financement de thèse permettant de poursuivre sur le sujet si le·a candidat·e le désire.

Contact

Antoine Limasset (`antoine.limasset@univ-lille.fr`)

Références

[1] D. Lemire, L. Boytsov, and N. Kurz. Simd compression and the intersection of sorted integers. Software : Practice and Experience, 46(6) :723–749, 2016.

[2] C. Marchet, M. Kerbiriou, and A. Limasset. Blight : Efficient exact associative structure for k-mers. Bioinformatics, 2021.

[3] C. Marchet, L. Lecompte, A. Limasset, L. Bittner, and P. Peterlongo. A resource-frugal probabilistic dictionary and applications in bioinformatics. Discrete Applied Mathematics, 274 :92–102, 2020.