TER M1 : Post-analyse de fichiers d'alignement pour la construction de graphes d'échafaudage

Annie Chateau

2013-2014

Mots-clés Algorithmique, graphe, analyse de séquence, implémentation C++, parallélisme.

1 Contexte bioinformatique

Lorsqu'un nouveau génome est séquencé, les données produites consistent en une quantité gigantesque (plusieurs milliards) de mots, ou séquences, de courte taille, appelées **lectures appariées** (ou *paired reads*). Les techniques de séquençage sont diverses et variées, et peuvent produire différentes sortes de reads, plus ou moins longs (voir [1]), mais le principe général reste le même. A noter que, comme la molécule d'ADN a une structure physique naturellement orientée, les lectures sont également orientées. Les paires ont donc également une orientation, qui peut être interne ou externe (voir Figure 1), mais l'orientation est homogène sur un même jeu de données. Dans ce qui suit, et pour simplifier le propos, on considèrera que l'orientation est interne.

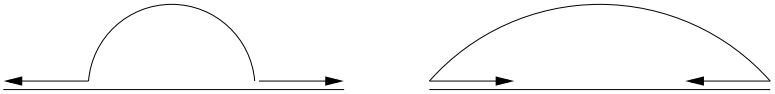

Figure 1: Un fragment d'ADN et une paire de lectures qui lui correspond. L'orientation peut être externe (gauche) ou interne (droite).

1.1 Assemblage

Etant donné un jeu de données de lectures, la phase d'assemblage consiste à reconstruire le gigantesque puzzle constitué par ces mots, en utilisant leurs chevauchements. Pour un état de l'art sur l'assemblage, voir [3]. La couverture moyenne du génome par ces lectures est généralement élevée, mais pas nécessairement homogène : certaines zones du génome ne sont pas suffisamment couvertes par des lectures chevauchantes, et ne sont donc pas retrouvées par les assembleurs. Les morceaux reconstitués sont appelés des **contigs** (voir Figure 2a). Remarque : l'information d'appariement n'est pas utilisée lors de l'étape d'assemblage. Les difficultés de l'assemblage sont également liées aux répétitions de certaines portions de génome, aux erreurs d'assemblage et à la variabilité à l'intérieur de l'échantillon.

(a) Les contigs (en-dessous) sont reconstruits en utilisant les chevauchements entre les lectures (au-dessus).

(b) Les contigs (lignes) sont orientés et ordonnés en scaffolds (boîtes) le long du génome (en-dessus).

Figure 2: Échafaudage de génome

1.2 Scaffolding

Une fois les contigs produits, l'étape d'après consiste à les orienter et à les ordonner, afin de produire de plus grandes parties du génome appelées **scaffold** (voir Figure 2b).

Pour réaliser l'échafaudage (scaffolding), on utilise cette fois-ci l'information d'appariement. Le scaffolding se définit de façon semi-formelle de la façon suivante : étant donné un ensemble de contigs $\mathcal{C} = \{C_1, \ldots, C_n\}$, et un ensemble de paires de lectures, on veut déterminer un ordre et une orientation des contigs qui soient cohérents avec un maximum de paires de lectures. On suppose pour cela que l'on a recherché la position des lectures sur les contigs (étape d'alignement), et l'on se concentre sur les paires qui sont à cheval sur deux contigs.

Etant donnée une orientation fixée des contigs, il existe quatre façons de les relier par une paire de lectures, appelées **histoires** (voir Figure 3).

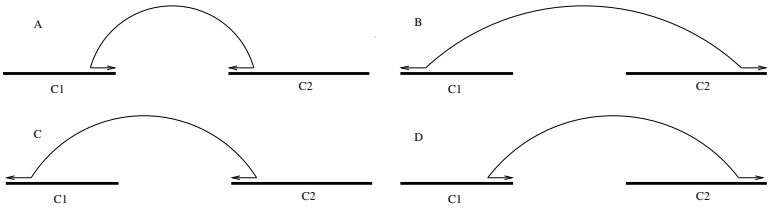

Figure 3: Les quatre histoires possibles pour relier deux contigs donnés. L'histoire A (resp. B) raconte que le contig C_1 (resp. C_2) se trouve avant le contig C_2 (resp. C_1) dans leur orientation originale (à symétrie globale près). L'histoire C (resp. D) raconte que le contig C_1 (resp. C_2), renversé, se trouve avant le contig C_2 (resp. C_1).

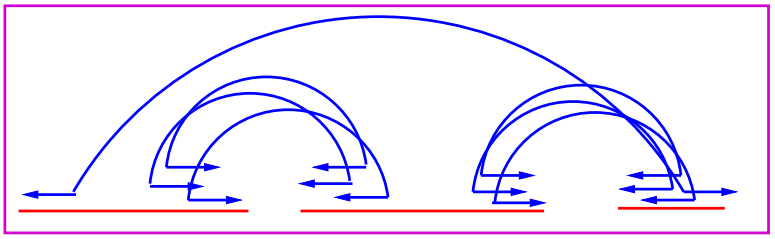

Figure 4: Trois contigs et les paires de lectures qui les relient. Remarque : deux paires peuvent raconter une même histoire.

Plusieurs paires peuvent raconter une même histoire (voir Figure 4). Dans ce cas, on considère qu'on peut les fusionner, et on attribue à l'histoire un poids égal au nombre de paires fusionnées. Le processus de fusion des paires de lectures est basé sur la distribution de la distance entre les lectures d'une paire (la taille de l'insert). Ce processus est décrit dans [2].

1.3 Alignement des lectures sur les contigs

Pour produire le graphe d'échafaudage, on a besoin de connaître la localisation et l'orientation des paires de lectures sur les contigs. Pour cela, on utilise un outil d'alignement (ou mapping) qui produit un fichier texte qui indique pour chaque lecture la position où il a été trouvé, et où se situe sa lecture appariée. On distingue alors les lectures dont les deux éléments appariés sont sur un même contig, et ceux qui sont sur des contigs différents. Il existe une troisième catégorie, des lectures qui ne s'alignent pas sur les contigs, mais dans le contexte de ce projet on ne les considère pas. Il existe un format standard pour stocker ces alignements, le format `.sam`, et sa version compressée `.bam`. Afin de déterminer de façon la plus précise possible les paramètres statistiques de la distribution des paires de lectures sur l'alignement, on considère l'ensemble des paires de reads qui sont sur un même contig. On va en déduire la distribution des tailles d'insert, et on s'intéresse plus particulièrement à la moyenne

et à l'écart-type de cette taille. Ces paramètres nous servirons par la suite à réaliser la fusion des histoires selon des critères adaptés à la distribution (typiquement, on va considérer que deux paires racontent une même histoire lorsque leurs tailles d'insert sont différents de moins d'un écart-type par exemple).

2 Objectifs du TER

L'objectif du TER est de produire une implémentation efficace (en C++, permettant de s'intégrer à un ensemble plus général d'implémentations) du traitement des fichiers d'alignement des lectures sur les contigs, de manière à produire un graphe d'échafaudage. Les tâches précises dévolues au groupe seront en particulier :

- Comprendre la structure d'un fichier d'alignement (.sam et sa version compressée, .bam)
- Implémenter une analyse syntaxique de ces fichiers permettant d'inférer les paramètres statistiques de la distribution des lectures sur les contigs, sur les paires internes aux contigs
- Extraire de ce même fichier les lectures à cheval sur deux reads, et construire le graphe d'échafaudage correspondant.
- Etant donné le volume des fichiers, on s'attachera à implémenter dès que possible une version parallélisable des algorithmes ci-dessus (en utilisant MPI, OpenMP par exemple).
- Chaque étape doit être paramétrisable (notamment la production du graphe d'échafaudage), et pipelinable (ligne de commande avec options)
- Il faudra réfléchir à une architecture des données qui permet à la fois de traiter le graphe produit comme n'importe quel graphe sous forme de listes d'adjacences, mais aussi de "retracer" les informations biologiques initiales (notamment, les identifiants des contigs représentant les sommets du graphe)

Contacts : annie.chateau@lirmm.fr

References

[1] J. Adams. DNA sequencing technologies. *Nature Education*, 1, 2008.

[2] D. H. Huson, K. Reinert, and E. W. Myers. The greedy path-merging algorithm for contig scaffolding. *J. ACM*, 49(5):603–615, September 2002.

[3] Steven L. Salzberg, Adam M. Phillippy, Aleksey Zimin, Daniela Puiu, Tanja Magoc, Sergey Koren, Todd J. Treangen, Michael C. Schatz, Arthur L. Delcher, Michael Roberts, Guillaume Marçais, Mihai Pop, and James A. Yorke. Gage: A critical evaluation of genome assemblies and assembly algorithms. *Genome Research*, 22(3):557–567, 2012.