Assemblage *de novo* de longues lectures par programmation linéaire

Victor Epain[1], Rumen Andonov[1], Hristo Djidjev[2], Dominique Lavenier[1]

[1] Univ Rennes, Inria, CNRS, IRISA, Rennes, France
{victor.epain,rumen.andonov,dominique.lavenier}@irisa.fr
[2] Los Alamos National Laboratory, Los Alamos, NM 87545, USA djidjev@lanl.gov

Mots-clés : *graphe de chevauchements, PLNE, partitionnement de graphe, problème du chemin de poids maximal.*

1 Introduction

Afin d'analyser *in silico* un génome, deux étapes sont nécessaires : le séquencer chimiquement, par clonage puis en le découpant en plusieurs parties appellées lectures, et assembler ces lectures informatiquement pour reconstruire le génome. Les lectures peuvent varier en taille et, bien que le nombres d'erreurs soit positivement corrélé à la taille des lectures, les longues lectures couvrent davantage les parties du génome et permettent de séparer des régions similaires en séquence nucléotidiques, mais placées distinctement sur le génome - nommées répétitions.

L'assemblage *de novo* est une méthode qui n'a pas besoin de référence. Alors que des assembleurs longues lectures *de novo* comme wtdbg2 [6], Flye [3] ou encore Unicycler [7] usent d'heuristiques, nous proposons ici de résoudre ce problème par une approche globale avec la programmation linéaire mixte en nombres entiers.

2 Le programme LOREAS : un assembleur de longues lectures *de novo*

Afin de répondre au problème d'assemblage, nous développons le programme nommé LOREAS, structuré en deux tâches : la première consiste à donner un ordonnancement des lectures ; la deuxième, à réaliser la séquence consensus à partir du précédent ordonnancement. À ce jour, seule la première étape fut réalisée.

3 Notre approche

3.1 Abstraction

Nous proposons d'abstraire le problème d'ordonnancement des lectures par la recherche d'un chemin dans un graphe de chevauchements entre les lectures orientées. Ainsi, soit $G = (V, l, E, \lambda)$ un tel graphe, où V est l'ensemble des sommets qui représentent les lectures, l leur taille associée ; E l'ensemble des arcs orientés *i.e.* l'ensemble des chevauchements entre les lectures, pondérés par *lambda*. On souhaite trouver le chemin qui maximise le nombre de lectures participantes : c'est un sous problème du plus long chemin qui est un problème NP complet, en vertu de la NP complétude du problème de la recherche d'un chemin hamiltonien.

3.2 Les chevauchements entre les lectures

Afin de comparer les lectures entre elles, celles ci sont alignées par rapport à leur séquence nucléotidique grâce au logiciel MINIMAP2 [4]. Nous souhaitons des alignements de type "préfixe-suffixe" que l'on nomme chevauchements. Si les lectures u et v dans V se chevauchent, alors on leur associe la longueur de chevauchement λ_{uv}.

3.3 Recherche du plus long chemin dans le graphe des chevauchements

Les lectures (sommets) sont vues en tant qu'objets de taille l_v possédant des chevauchements de taille λ_{uv} avec d'autres. Donner un ordonnancement de ces lectures revient à attribuer, aux lectures participantes au plus long chemin, une coordonnée y_v (coordonnée placée à l'extrémité droite des lectures). La détermination des coordonnées des lectures est soumise à contraintes, inspirées de celles permettant l'assemblage hybride de lectures par la programmation mathématique proposé par Miller-Tucker-Zemlin (MTZ) [5] et étendue dans un des articles rédigé par des présents auteurs (DCEP) [1].

3.4 Partitionnement du graphe des chevauchements

Si le graphe est considéré comme grand (selon le nombre d'arcs), alors il est partitionné grâce à l'outil METIS [2]. Les lectures sont groupées en parties, représentées avec leurs interelations dans un graphe de partie. Il s'agit ensuite de trouver le chemin de poids maximal dans ce graphe pour déterminer l'ordre de résolution des parties, à savoir trouver le plus long chemin dans le graphe de chevauchement de chaque partie. De même, les contraintes associées à ce problème sont inspirées des méthodes MTZ et DCEP.

4 Les résultats

La tâche d'ordonnancement fut appliquée sur dix génomes bactériens. En comparaison avec les coordonées véritables des lectures sur les génomes test, il en résulta pour 8 génomes un bon ordonnancement - à l'exception d'une lecture pour un des génomes - dont 7 furent *a priori* couverts à plus de 99%. Des améliorations et re-structurations sont à amener pour résoudre plus d'instances et de plus grande taille.

Références

[1] Sébastien Francois, Rumen Andonov, Hristo Djidjev, Metodi Traikov, and Nicola Yanev. Mixed integer linear programming approach for a distance-constrained elementary path problem.

[2] George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. 20(1) :359–392.

[3] Mikhail Kolmogorov, Jeffrey Yuan, Yu Lin, and Pavel A. Pevzner. Assembly of long, error-prone reads using repeat graphs. 37(5) :540–546.

[4] Heng Li. Minimap2 : pairwise alignment for nucleotide sequences. 34(18) :3094–3100.

[5] C. E. Miller, A. W. Tucker, and R. A. Zemlin. Integer programming formulation of traveling salesman problems. 7(4) :326–329.

[6] Jue Ruan and Heng Li. Fast and accurate long-read assembly with wtdbg2.

[7] Ryan R. Wick, Louise M. Judd, Claire L. Gorrie, and Kathryn E. Holt. Unicycler : Resolving bacterial genome assemblies from short and long sequencing reads. 13(6) :e1005595.