EasyChair Preprint

№ 2573

# Long Reads Assembly Using Integer Linear Programming

Victor Epain, Rumen Andonov, Hristo Djidjev and Dominique Lavenier



February 7, 2020

# 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, plusieurs étapes sont nécessaires : le séquencer chimiquement, par clonage, le découper en plusieurs parties appellées lectures, et assembler ces lectures informatiquement pour reconstruire le génome. Les lectures peuvent varier en taille et, bien que le nombre 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 du génome.

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 Méthodologie

Afin de répondre au problème d'assemblage, nous développons une stratégie structurée en deux phases : la première consiste à produire un ordonnancement des lectures ; la seconde, à construire la séquence consensus à partir de l'ordonnancement. À ce jour, seule la première étape a été réalisée.

### 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.

### Chevauchements entre les lectures

Afin de comparer les lectures entre elles, celles ci sont alignées par rapport à leur séquence nucléotidique grâce au logiciel spécialisé dans l'alignement de longues lectures 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 $\lambda_{uv}$.

### 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 $\lambda_{uv}$ 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 proposée par Miller-Tucker-Zemlin (MTZ) [5] et étendue dans un des articles rédigé par des présents auteurs (DCEP) [1].

### 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 de partitionnement de graphe METIS [2]. Les lectures sont groupées en parties, représentées avec leurs interelations dans un graphe de parties. 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.

## 3 Résultats

La tâche d'ordonnancement a été appliquée sur dix génomes bactériens. En comparaison avec les coordonées véritables des lectures sur les génomes test, 8 génomes possèdent un bon ordonnancement - à l'exception d'une lecture pour un des génomes. 7 sont *a priori* couverts à plus de 99%. Des améliorations et re-structurations sont en cours pour résoudre des instances 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. *hal.inria.fr*, June 2018.

[2] George Karypis and Vipin Kumar. A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs. *SIAM J. Sci. Comput.*, 20(1) :359–392, December 1998.

[3] Mikhail Kolmogorov, Jeffrey Yuan, Yu Lin, and Pavel A. Pevzner. Assembly of long, error-prone reads using repeat graphs. *Nature Biotechnology*, 37(5) :540–546, May 2019.

[4] Heng Li. Minimap2 : pairwise alignment for nucleotide sequences. *Bioinformatics*, 34(18) :3094–3100, September 2018.

[5] C. E. Miller, A. W. Tucker, and R. A. Zemlin. Integer Programming Formulation of Traveling Salesman Problems. *J. ACM*, 7(4) :326–329, October 1960.

[6] Jue Ruan and Heng Li. Fast and accurate long-read assembly with wtdbg2. *Nature Methods*, 17(2) :155–158, February 2020.

[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. *PLOS Computational Biology*, 13(6) :e1005595, June 2017.