Linear Integer Programming Approaches for Chloroplast Genome Scaffolding

Victor Epain, Rumen Andonov

Univ. Rennes, Inria, IRISA, F-35000 Rennes, France
{victor.epain, rumen.andonov}@irisa.fr

Mots-clés : *genome assembly, contigs graph, inverted repeats, nested pairs*

1 Introduction

Genome assembly aims to assemble one genome DNA strand from a large amount of small DNA sequences denoted as *reads*. Due to sequencing technologies, and the fact that the DNA molecule is composed of two complemented strands, the reads must be considered in two exclusive *orientations*. Arbitrary, the as-input state is denoted *forward* orientation, while the other state is denoted the *reverse* one. The reverse read's orientation is obtained by reversing and complementing (DNA alphabet bijection) its nucleotide sequence.

The genome assembly process can be roughly separated in two main steps. First, the oriented reads are merged into larger sequences named *contigs* (also oriented). They correspond to paths in a directed graph modelling successions between the oriented reads. An upper bound of the number of time a contig is repeated in the genome, denoted *multiplicity*, is also provided. Our study exclusively focuses on the second stage of the genome assembly, *scaffolding*, in which contigs are put in correct order and orientation towards the completion of the final assembly. Usually this step uses additional data e.g. mate pairs distances, or homology references from near-species. In contrast to that, we demonstrate here that chloroplasts genomes can be assembled without any raw additional data, but only using the available biological knowledge on some chloroplast genome structures [2]. As a case example, Figure (1) illustrates one of them, in which two genomic regions are repeated, but one's sequence is the reverse-complement of the other (*inverted repeats*).

2 Method

The first step outputs a directed graph. Each contig gives two (forward and reverse orientation) times its multiplicity vertices. So each vertex corresponds to one orientation (forward or reverse) of a contig, enriched by an occurrence integer (from 0 to $multiplicity - 1$). Edges represent oriented multiplied contigs' successions. We model the genome assembly as an elementary circuit in this graph, see Figure (2). We formulate the inverted repeats with linear constraints, and we search for such a circuit using Integer Linear Programming similarly to [1].

Inverted repeats correspond to sequences of occurrences of contigs paired with other occurrences of them but in reverse orientation, see Figure (1). Therefore, paired contigs' positions on the assembled sequence must satisfy nested-pairs pattern (the blue dotted lines are parallel in Figure 1). The goal is to find the longest contiguous inverted repeats. We formulate the above constraints in terms of linear program where the objective is to maximise the nested-pairs number. Thus, we generalise a similar approach applied for RNA folding [4]. However, in contrast to the latter approach where the vertices correspond to bases with known sequence indices, in our case the positions of the contigs are variables. Our tool is implemented with Python 3 and uses the open-source PuLP package which integrates a free solver CBC to solve the above optimisation problem.

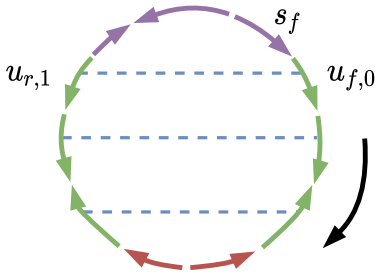

FIG. 1 – An illustration of chloroplast genome structure. It is a quadripartite circuit with a pair of inverted repeat regions (in green), separated by two unique regions (in purple and red). Each arrow represents a (multiplied) contig, where its orientation is implied by the orientation of the walk (black arrow). $u_{f,0}$ is an occurrence of contig u, paired with its reversed complement $u_{r,1}$ — another occurrence. Blue dash lines visualise other nested pairs. The circuit starts from and finishes to contig s in forward orientation.

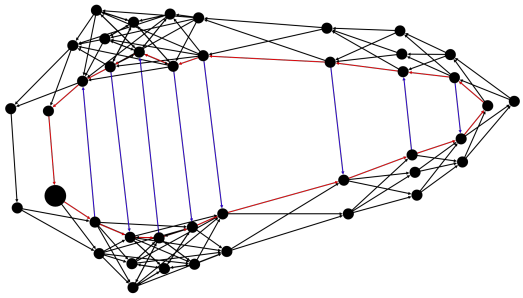

FIG. 2 – The input contigs were multiplied by their multiplicity number, then doubled according to two DNA strands. The obtained graph illustrated here possesses 42 vertices and 130 edges. Oriented contigs candidate to participate in reverse repeats have their two oriented versions linked by a blue edge. The solution path (the assembled genome) is represented in red. It begins from the biggest vertex (a given starter) and finishes to the same vertex as chloroplast genomes are circular.

3 Results

We verified our method with the well-known assembly evaluation tool QUAST [3] on a dataset that contains 42 chloroplast genome with inverted repeats, the same as in [5]. We ran the 42 instances on a laptop (16GB RAM, 8 cores), and we obtained very encouraging preliminary results, with high genome coverage (mostly $> 99\%$), and very low mismatches and indels rates (metrics computed from sequences alignment between the known genome reference and the sequences produced by our tool). Assuming that all successions between contigs are provided, our approach permits finishing pre-assembled genomes in just a few seconds, for graphs that not exceed 154 vertices and 240 edges and a dozen of candidate nested pairs.

Références

[1] Rumen Andonov, Hristo Djidjev, Sebastien François, and Dominique Lavenier. Complete assembly of circular and chloroplast genomes based on global optimization. *Journal of Bioinformatics and Computational Biology*, 17(3):1950014, June 2019.

[2] Ralph Bock and Volker Knoop, editors. *Genomics of Chloroplasts and Mitochondria*, volume 35 of *Advances in Photosynthesis and Respiration*. Springer Netherlands, Dordrecht, 2012.

[3] Alexey Gurevich, Vladislav Saveliev, Nikolay Vyahhi, and Glenn Tesler. QUAST : quality assessment tool for genome assemblies. *Bioinformatics*, 29(8):1072–1075, April 2013.

[4] Dan Gusfield. The RNA-Folding Problem. In Dan Gusfield, editor, *Integer Linear Programming in Computational and Systems Biology : An Entry-Level Text and Course*, pages 105–121. Cambridge University Press, Cambridge, 2019.

[5] Jian-Jun Jin, Wen-Bin Yu, Jun-Bo Yang, Yu Song, Claude W. dePamphilis, Ting-Shuang Yi, and De-Zhu Li. GetOrganelle : a fast and versatile toolkit for accurate de novo assembly of organelle genomes. *Genome Biology*, 21(1):241, September 2020.