How to involve repetitive regions in scaffolding improvement

Rémy Costa[1,4], Quentin Delorme[1,2], Yasmine Mansour[1,2,3],
Anna-Sophie Fiston-Lavier[1,3], Annie Chateau[1,2]

[1]Université de Montpellier
[2]Laboratoire d'Informatique, Robotique et Micro-électronique de Montpellier
[3]Institut des Sciences de l'Évolution de Montpellier
[4]Master Science et Numérique pour la Santé, parcours Bioinformatique, Connaissances, Données

August 23, 2019

Abstract

Context and motivation. Repetitive regions (RR) in DNA sequences are present in almost all organisms and may represent over 80% of the genome size [3]. Fundamental source of genetic plasticity and diversity, yet they are a source of complication when it comes to assemble genomes [10]. Assembly produces contigs of various sizes, sometimes really smaller than the original chromosome size. To reduce the fragmentation of chromosomes, the scaffolding process involves additional information, for instance pairing between reads, to infer how contigs are relatively organized [5]. Repetitive regions are disturbing both assembly and scaffolding processes, which are based on graphs. Most of assembly and scaffolding methods use a repeat filter or abort extension of contigs in presence of RR. One way to untangle ambiguous parts of these graphs is to use long reads, produced by third-generation sequencing technologies. However, this is not always possible due to high cost and lower quality. Here we propose to use RR sequences themselves to enhance the scaffolding step.

Methodology. The scaffold graph is defined as follows: vertices represent contig extremities, while edges are of two kinds: (1) contig edges, linking both extremities of a contig, and (2) inter-contig edges relating the pairing-information. A weight function on the inter-contig edges indicates how many pairs are supporting this edge. Due to repeats, some of the inter-contigs edges are erroneous and have to be removed from the graph. In other cases, they are supported by RR. Our method is based on a pipeline progressively refining inter-contig edges through RR analysis, described as follows:

1. find the known RR sequences using a repeat database [1], map them on contigs, tag the contigs with this information, and cluster them according to these tags;
2. inside each cluster, determine inter-contig edges sharing coherent RR sequence parts;
3. modify the weight of the validated inter-contig edges;
4. delete edges incoherent with RR composition or length;
5. after scaffolding, use the RR canonical sequence to fill the gaps between contigs.

An additional knowledge about well-documented RRs (such as Transposable Elements) may help to improve Step 2, and answer the following question: do assembly errors come essentially from recent RRs ? Step 3 can be achieved in different ways, thus we propose to try several weight function perturbations. Step 4 is quite expeditious and may be smoothed by introducing a probabilistic measure to ponder the inter-contig weight instead of deleting it.

Validation. The benchmark is composed of organisms offering different repetition rates and sizes. To validate our approach, we use simulated data from model species, amongst them very high quality genomes such as *Drosophila melanogaster* and *Caenorhabditis elegans*. We used ART [4] to generate our paired-end reads with a 20X coverage, simulating Illumina's HighSeq2000. To realise the assembly, we used Minia [2] and Spades [9] (Bankevich et. al, 2012) in order to compare the most effecicient tool. The mapping was realised with BWA [7] and Minimap2 [6]. Genome quality is measured using the QUAST tool [8].

Results. Results show a slight reduction of the covered genome fraction and the NG50, but an improvement in the reduction of misassemblies up to 26% with SPAdes (and no improvement with minia). To analyse further these misassemblies, we aligned them on the reference genome to observe if RR were implicated. RRs are implicated in 60 to 70% of the misassemblies. Even if we found some tandem repeats and pseudogenes, the vast majority is composed of transposable elements.

References

[1] Weidong Bao, Kenji K. Kojima, and Oleksiy Kohany. Repbase Update, a database of repetitive elements in eukaryotic genomes. *Mob DNA*, 6:11, 2015.

[2] Rayan Chikhi and Guillaume Rizk. Space-efficient and exact de bruijn graph representation based on a bloom filter. In *WABI*, volume 7534 of *Lecture Notes in Computer Science*, pages 236–248. Springer, 2012.

[3] Ingrid Garbus, José R. Romero, Miroslav Valarik, Hana Vanžurova, Miroslava Karafiatova, Mario Caccamo, Jaroslav Doležel, Gabriela Tranquilli, Marcelo Helguera, and Viviana Echenique. Characterization of repetitive DNA landscape in wheat homeologous group 4 chromosomes. *BMC Genomics*, 16:375, May 2015.

[4] Weichun Huang, Leping Li, Jason R Myers, and Gabor T Marth. Art: a next-generation sequencing read simulator. *Bioinformatics*, 28(4):593–594, 2011.

[5] Martin Hunt, Chris Newbold, Matthew Berriman, and Thomas D. Otto. A comprehensive evaluation of assembly scaffolding tools. *Genome Biology*, 15(3):R42, Mar 2014.

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

[7] Heng Li and Richard Durbin. Fast and accurate short read alignment with burrows–wheeler transform. *bioinformatics*, 25(14):1754–1760, 2009.

[8] Alla Mikheenko, Andrey Prjibelski, Vladislav Saveliev, Dmitry Antipov, and Alexey Gurevich. Versatile genome assembly evaluation with quast-lg. *Bioinformatics*, 34(13):i142–i150, 2018.

[9] S. Nurk, A. Bankevich, D. Antipov, A. A. Gurevich, A. Korobeynikov, A. Lapidus, A. D. Prjibelski, A. Pyshkin, A. Sirotkin, Y. Sirotkin, R. Stepanauskas, S. R. Clingenpeel, T. Woyke, J. S. McLean, R. Lasken, G. Tesler, M. A. Alekseyev, and P. A. Pevzner. Assembling single-cell genomes and mini-metagenomes from chimeric MDA products. *J. Comput. Biol.*, 20(10):714–737, Oct 2013.

[10] Haixu Tang. Genome assembly, rearrangement, and repeats. *Chemical Reviews*, 107(8):3391–3406, 2007.