Bioinformatics, 38(10), 2022, 2920–2921
https://doi.org/10.1093/bioinformatics/btac208
Advance Access Publication Date: 6 April 2022
Applications Note

Genome analysis

SASpector: analysis of missing genomic regions in draft genomes of prokaryotes

Cédric Lood [1,2,*], Alejandro Correa Rojo[1], Deniz Sinar[1], Emma Verkinderen[1], Rob Lavigne[2] and Vera van Noort[1,3,*]

[1]Department of Microbial and Molecular Systems, KU Leuven, 3001 Leuven, Belgium, [2]Department of Biosystems, KU Leuven, 3001 Leuven, Belgium and [3]Institute of Biology, Leiden University, 2333 Leiden, The Netherlands

*To whom correspondence should be addressed.
Associate Editor: Tobias Marschall

Received on June 21, 2021; revised on March 22, 2022; editorial decision on April 4, 2022; accepted on April 5, 2022

Abstract

Summary: Missing regions in short-read assemblies of prokaryote genomes are often attributed to biases in sequencing technologies and to repetitive elements, the former resulting in low sequencing coverage of certain loci and the latter to unresolved loops in the *de novo* assembly graph. We developed SASpector, a command-line tool that compares short-read assemblies (draft genomes) to their corresponding closed assemblies and extracts missing regions to analyze them at the sequence and functional level. SASpector allows to benchmark the need for resolved genomes, can be integrated into pipelines to control the quality of assemblies, and could be used for comparative investigations of missingness in assemblies for which both short-read and long-read data are available in the public databases.

Availability and implementation: SASpector is available at https://github.com/LoGT-KULeuven/SASpector. The tool is implemented in Python3 and available through pip and Docker (0mician/saspector).

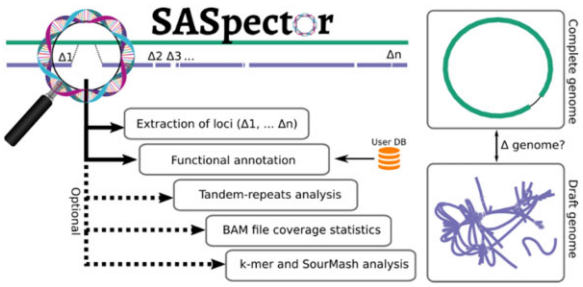
Contact: cedric.lood@kuleuven.be or vera.vannoort@kuleuven.be

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Prokaryote genome sequencing efforts are often conducted on Illumina sequencers, a technology that delivers short yet accurate reads (Goodwin *et al.*, 2016). These datasets of reads are the bedrock of many subsequent analyses which often start with *de novo* assemblies. However, these so-called draft genomes are often fragmented in hundreds of contigs (Arredondo-Alonso *et al.*, 2017; Wick *et al.*, 2017). Indeed, biases can appear during the library preparation and sequencing by synthesis (Abnizova *et al.*, 2017; Shin *et al.*, 2016), but also post-sequencing because of repetitive elements, either interspersed or in tandem repeats. Consequently, *de novo* assemblers fail to fully resolve the consensus genome based on the short-read dataset because of collapsing regions in the assembly graph or mis-assemblies (Alkan *et al.*, 2011). Long-read sequencing technologies have been welcome adjuncts to resolve assemblies, but these reads typically have a lower fidelity compared to Illumina reads (Amarasinghe *et al.*, 2020). Currently, the combination of both technologies is considered a gold-standard, resulting in hybrid assemblies of closed and accurate genomes, but consequently remain more costly (Lood et al., 2021; Wick *et al.*, 2017). The availability of both types of data for a given isolate enables systematic comparisons between the closed (hybrid) assembly and the short-read draft assembly to analyze reasons for the breaks in the draft genome, and importantly to probe what is functionally missing from these draft genomes. To address this issue, we developed SASpector, a tool that assesses missingness in short-read assemblies by comparison to reference genomes.

2 Implementation



2.1 Regions delineation and sequence analysis

SASpector uses the whole-genome alignment program progressiveMauve (Darling *et al.*, 2010) to initially map the contigs from the draft genome to the related closed genome (concatenated in the case of multiple contigs). Python3 is used to parse the alignment output and extract from the closed genome the regions not covered. The user can specify the size of the extracted flanking regions (default is 100 bp on each side). SASpector also generates a fasta file with regions from the draft assembly that did not perfectly match the reference due to indels or single nucleotide changes (so-called conflict regions).

SASpector creates two main summary files, a table for the reference genome that includes the total length of the assembly, average GC content, as well as the count and genome fraction for mapped and missing regions. A second table is generated for the missing regions with the lengths, GC contents and average amino acid residue frequencies from all six open reading frames for each region. For each of these metrics, visualizations are also produced using the matplotlib and seaborn python libraries.

2.2 What is missing in my assembly?

The functional content of the missing regions is annotated with Prokka (Seemann, 2014), with the option (--proteindb) to provide a custom trusted protein database to transfer functional annotation. Optional SASpector analyses include:

- **--coverage:** calculation of the average coverage within missing regions (as per-base read depth) based on SAMtools (Li *et al.*, 2009) and comparison with the coverage of the mapped regions. This generates a summary table for each of the regions, including locations in the reference genome, total read base count and average per-base depth, each summarized with a boxplot graph.
- **--kmers:** SASpector creates MinHash signatures of k-mers in missing and mapped regions using the Sourmash library (Pierce *et al.*, 2019) to generate a pairwise comparison by Jaccard similarity of k-mers between missing and covered regions.
- **--tandem_repeats:** tandem repeats are detected in each of the missing regions by the program Tandem Repeats Finder (Benson, 1999).
- **--quast:** SASpector wraps QUAST (Gurevich *et al.*, 2013) to assess the missing regions in relation to the complete genome. This includes the Icarus contig alignment viewer as genome viewer, which allows quick visualization of the missing regions in the genome.
- **--msh_selection:** automatic selection of a closed reference from RefSeq v202 (experimental feature).

3 Discussion

SASpector is a python-based command-line tool that compares short-read assemblies with their corresponding closed reference. It enables the systematic evaluation of missing regions in draft assemblies in terms of functional content and sequence features. We provide in Supplementary Material an example analysis of a *Pseudomonas aeruginosa* genome. The draft assembly appears to lack contiguous regions up to 7,200 bp in size, with lower GC% on average - a feature that usually indicates recently acquired mobile element in that species (San Millan *et al.*, 2015). The functional annotation of these missing regions reveals a high number of transposons, rRNA genes and modular repeat gene groups. Importantly, some genes linked to virulence appear to be missing from the draft assembly, highlighting potential impact on downstream analyses, such as annotation of pathogenicity and virulence in that isolate.

In conclusion, SASpector can help researchers to benchmark assemblies or give rationales to decide whether it is necessary to pursue long-read sequencing in a large sequencing project, for example based on the sequencing of a subset of isolates or analysis of existing data. As a python package, the tool can be integrated in pipelines and can be used for a large-scale survey that utilizes the growing amount of genomics data available in public databases like NCBI, where completed genomes are rising in number but where draft genomes vastly outnumber them and for which we currently have no systematic understanding as to what may be missing.

Funding

This work was supported by the Research Foundation—Flanders (FWO) under the scope of a PhD fellowship (1S64720N) and a postdoctoral mandate from KU Leuven (PDMt2/21/038).

Conflict of Interest: none declared.

References

Abnizova,I. *et al.* (2017) Computational errors and biases in short read next generation sequencing. *J. Proteomics Bioinform.*, **10**, 1.

Alkan,C. *et al.* (2011) Limitations of next-generation genome sequence assembly. *Nat. Methods*, **8**, 61–65.

Amarasinghe,S.L. *et al.* (2020) Opportunities and challenges in long-read sequencing data analysis. *Genome Biol.*, **21**, 30.

Arredondo-Alonso,S. *et al.* (2017) On the (im)possibility of reconstructing plasmids from whole-genome short-read sequencing data. *Microb. Genom.*, **3**, e000128.

Benson,G. (1999) Tandem repeats finder: a program to analyze DNA sequences. *Nucleic Acids Res.*, **27**, 573–580.

Darling,A.E. *et al.* (2010) progressiveMauve: multiple genome alignment with gene gain, loss and rearrangement. *PLoS One*, **5**, e11147.

Goodwin,S. *et al.* (2016) Coming of age: ten years of next-generation sequencing technologies. *Nat. Rev. Genet.*, **17**, 333–351.

Gurevich,A. *et al.* (2013) QUAST: quality assessment tool for genome assemblies. *Bioinformatics*, **29**, 1072–1075.

Li,H. *et al.* (2009) The sequence alignment/map format and SAMtools. *Bioinformatics*, **25**, 2078–2079.

Lood,C. *et al.* (2021) Genomics of an endemic cystic fibrosis *Burkholderia multivorans* strain reveals low within-patient evolution but high between-patient diversity. *PLoS Pathog.*, **17**, e1009418.

Pierce,N.T. *et al.* (2019) Large-scale sequence comparisons with sourmash. *F1000Res.*, **8**, 1006.

San Millan,A. *et al.* (2015) Interactions between horizontally acquired genes create a fitness cost in *Pseudomonas aeruginosa*. *Nat. Commun.*, **6**, 1–8.

Seemann,T. (2014) Prokka: rapid prokaryotic genome annotation. *Bioinformatics*, **30**, 2068–2069.

Shin,S. *et al.* (2016) Characterization of sequence-specific errors in various next-generation sequencing systems. *Mol. Biosyst.*, **12**, 914–922.

Wick,R.R. *et al.* (2017) Completing bacterial genome assemblies with multiplex MinION sequencing. *Microb. Genom.*, **3**, e000132.