WGA-LP: A PIPELINE FOR WHOLE GENOME ASSEMBLY OF CONTAMINATED READS

A bioRxiv PREPRINT

Rossi N.

Department of Mathematics, Computer Science, and Physics.
University of Udine, Udine, 33100, Italy.
olocin.issor@gmail.com

Colautti A.

Dipartimento di Scienze agroalimentari, ambientali e animali.
University of Udine, Udine, 33100, Italy.
colautti.andrea@spes.uniud.it

Iacumin L.

Dipartimento di Scienze agroalimentari, ambientali e animali.
University of Udine, Udine, 33100, Italy.
lucilla.iacumin@uniud.it

Piazza C.

Department of Mathematics, Computer Science, and Physics.
University of Udine, Udine, 33100, Italy.
carla.piazza@uniud.it

This article has been accepted for publication in Bioinformatics Published by Oxford University Press.

October 16, 2021

ABSTRACT

Summary: Whole Genome Assembly (WGA) of bacterial genomes with short reads is a quite common task as DNA sequencing has become cheaper with the advances of its technology. The process of assembling a genome has no absolute golden standard (Del Angel et al. (2018)) and it requires to perform a sequence of steps each of which can involve combinations of many different tools. However, the quality of the final assembly is always strongly related to the quality of the input data. With this in mind we built WGA-LP, a package that connects state-of-art programs and novel scripts to check and improve the quality of both samples and resulting assemblies. WGA-LP, with its conservative decontamination approach, has shown to be capable of creating high quality assemblies even in the case of contaminated reads.

Availability and Implementation: WGA-LP is available on GitHub (<https://github.com/redsnc/WGA-LP>) and Docker Hub (<https://hub.docker.com/r/redsnc/wgalp>). The web app for node visualization is hosted by shinyapps.io (<https://redsnc.shinyapps.io/ContigCoverageVisualizer/>).

Contact: Nicolò Rossi, olocin.issor@gmail.com

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

WGA-LP: A PIPELINE FOR WHOLE GENOME ASSEMBLY OF CONTAMINATED READS

1 Introduction

A currently active challenge in the context of whole genome assembly for bacterial genomes is to produce reliable whole genome assemblies that are contaminant free (Steinberger and Salzberg (2020) and Chun et al. (2018)). In this context, we built WGA-LP, a pipeline that includes different strategies to guide the users in producing higher quality whole genome assemblies, by also including specific features to control possible contamination. Moreover, its workflow is structured to assist in the quality evaluation of the results of each step of the pipeline by providing useful plots and summaries. The current state-of-the-art for decontamination consists in the use of Kraken2, (Wood et al. (2019)) a software for read origin imputation, and of pipelines like ProDeGe (Tennessen et al. (2016)) and SIDR (Fierst and Murdock (2017)). This last is however meant for eukaryotic genomes.

2 Software description

WGA-LP software is built to be used from the command line. The procedures of the pipeline are organized by functionality and have a consistent syntax for argument passing. More details are available in the Supplementary Information material, on the GitHub, and Docker Hub web pages of the tool.

WGA-LP performs many steps that can be run independently. In order to execute the whole workflow the user is required to provide the raw reads (.fastq) and, optionally, the references that should be used for decontamination (.fasta). All the other input files can be produced using WGA-LP commands. Check the Supplementary information material for a complete explanation of all the input parameters for WGA-LP.

The first step of WGA-LP has the role of assessing the quality of the input reads and detecting possible contamination sources. To this end WGA-LP relies on Trimmomatic (Bolger et al. (2014)), FastQC (Andrews (2010)), Kraken2, and Bracken (Lu et al. (2017)). The trimming step is fully configurable so that the user can choose the right approach for his data.

The decontamination procedure is based on a custom script that includes calls to three programs: BWA mem (Li (2013)), Samtools, (Li et al. (2009)) and Bazam (Sadedin and Oshlack (2019)). We take in input the raw reads and two sets of references, one for the target organism and one for the contaminants. This procedure is presented with more details in the Figure 1 and in the Supplementary materials. This decontamination approach is conservative and reduces the probability of discarding reads of the target organism. This part of the pipeline can be used as a standalone program and can be combined with any other program for Whole Genome Assembly.

WGA-LP natively supports SPAdes (Bankevich et al. (2012)) and Minia (Chikhi and Rizk (2012)) assemblers. SPAdes is currently a common choice for bacterial WGA, while Minia is a very simple and fast assembler. The other steps of WGA-LP can support any assembler that includes in its outputs a fasta formatted assembly and a fastg assembly graph (required only for putative plasmid search with Recycler (Rozov et al. (2017))).

WGA-LP includes custom scripts to help in the visualization of node coverage by post processing the output of Samtools depth. This allows to produce coverage plots (computed by remapping the reads to the assembled genome) that can be helpful in finding anomalies, such as prophage insertions in the genome. Moreover, WGA-LP provides a web app and tools for nodes (and reads) selection that can improve the decontamination results. These act by exploiting

Figure 1: The decontamination procedure. Input reads are mapped against each reference of the contaminant independently (first three wires from left to are then merged together (Union, $\cup$) and gradually filtered (last wire from right to left), with the effect of removing all the reads that map to any The final decontaminated reads are extracted by set difference ($\setminus$) using the original input set.

WGA-LP: A PIPELINE FOR WHOLE GENOME ASSEMBLY OF CONTAMINATED READS

the assembly process as it tends to assemble nodes with reads of the same organism. Such procedures are well fitted to be combined with Kraken2, since this tool can point out problematic nodes, that can be then further evaluated with BLAST alignment (Altschul et al. (1990)) in order to validate user selections.

For node reordering, WGA-LP uses the `ContigOrderer` option from Mauve aligner (Rissman et al. (2009)). This step requires to provide a reference for the target organism.

WGA-LP offers interfaces to two programs that extract putative plasmids: `plasmidSPAdes` (Antipov et al. (2016)) and `Recycler`. It is highly recommended to check the results of these tools using BLAST.

WGA-LP includes three programs to evaluate the quality of the final result of the pipeline: `Quast` (Gurevich et al. (2013)), `CheckM` (Parks et al. (2015)) and `Mercury` (Rhie et al. (2020)). Especially, `checkM` is useful to verify the completeness and contamination of the produced assembly.

For the annotation, WGA-LP interfaces with `Prokka` (Seemann (2014)) in order to create NCBI compliant assemblies. This can be considered the final output of the pipeline and can be used for downstream analysis.

3 Results

We tested WGA-LP pipeline on real data (see Data Availability section) and we have shown how its workflow was effective in producing an high quality Whole Genome Assembly even in this challenging scenario of a contaminated genome, with improvements in comparison with less curated approaches (see Supplementary Information). Finally, we extended the comparison to include `ProDeGe`, another state-of-the-art decontamination procedure. `ProDeGe` alone was not able to filter large nodes of the contaminant, however it was possible to use WGA-LP procedures based on `kraken2` classification to refine the resulting assembly, achieving comparable results with our pipeline. However, also in this case our tool performed better on the elimination of the shorter nodes, keeping those that, in a further check, were classified from the target genome by BLAST alignment.

Both the decontamination procedure and the node selection that are the core of our pipeline, can be integrated in any other pipeline for Whole Genome Assembly, in the pre-processing and post-processing phases.

4 Data availability

The testing reads, from the organism *Lactocaseibacillus rhamnosus*, heavily contaminated with *Pediococcus acidilactici*, were deposited in the NCBI's Sequence Read Archive (SRA) with the ID SRR15265000, included in the associated BioProject PRJNA749304.

Acknowledgements

All website and links in this paper and in the supplementary materials were accessed on the 25/07/2021.

Funding

This work was funded by Ministero dell'Università e della Ricerca, Rome, Italy, FIRB n. RBFR107VML.

References

- Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410.
- Andrews, S. (2010). Babraham bioinformatics-fastqc a quality control tool for high throughput sequence data. URL: <https://www.bioinformatics.babraham.ac.uk/projects/fastqc>.
- Antipov, D., Hartwick, N., Shen, M., Raiko, M., Lapidus, A., and Pevzner, P. A. (2016). `plasmidspades`: assembling plasmids from whole genome sequencing data. *Bioinformatics*, 32(22):3380–3387.
- Bankevich, A., Nurk, S., Antipov, D., Gurevich, A. A., Dvorkin, M., Kulikov, A. S., Lesin, V. M., Nikolenko, S. I., Pham, S., Prjibelski, A. D., et al. (2012). `Spades`: a new genome assembly algorithm and its applications to single-cell sequencing. *Journal of computational biology*, 19(5):455–477.

WGA-LP: A PIPELINE FOR WHOLE GENOME ASSEMBLY OF CONTAMINATED READS

- 92 Bolger, A. M., Lohse, M., and Usadel, B. (2014). Trimmomatic: a flexible trimmer for illumina sequence data.
93 *Bioinformatics*, 30(15):2114–2120.
- 94 Chikhi, R. and Rizk, G. (2012). Space-efficient and exact de bruijn graph representation based on a bloom filter. In
95 *WABI*, volume 7534 of *Lecture Notes in Computer Science*, pages 236–248. Springer.
- 96 Chun, J., Oren, A., Ventosa, A., Christensen, H., Arahal, D. R., da Costa, M. S., Rooney, A. P., Yi, H., Xu, X.-W.,
97 De Meyer, S., et al. (2018). Proposed minimal standards for the use of genome data for the taxonomy of prokaryotes.
98 *International journal of systematic and evolutionary microbiology*, 68(1):461–466.
- 99 Del Angel, V. D., Hjerde, E., Sterck, L., Capella-Gutierrez, S., Notredame, C., Pettersson, O. V., Amselem, J., Bouri, L.,
100 Bocs, S., Klopp, C., et al. (2018). Ten steps to get started in genome assembly and annotation. *F1000Research*, 7.
- 101 Fierst, J. L. and Murdock, D. A. (2017). Decontaminating eukaryotic genome assemblies with machine learning. *BMC*
102 *bioinformatics*, 18(1):1–16.
- 103 Gurevich, A., Saveliev, V., Vyahhi, N., and Tesler, G. (2013). Quast: quality assessment tool for genome assemblies.
104 *Bioinformatics*, 29(8):1072–1075.
- 105 Li, H. (2013). Aligning sequence reads, clone sequences and assembly contigs with bwa-mem. *arXiv preprint*
106 *arXiv:1303.3997*.
- 107 Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., and Durbin, R. (2009).
108 The sequence alignment/map format and samtools. *Bioinformatics*, 25(16):2078–2079.
- 109 Lu, J., Breitwieser, F. P., Thielen, P., and Salzberg, S. L. (2017). Bracken: estimating species abundance in metagenomics
110 data. *PeerJ Computer Science*, 3:e104.
- 111 Parks, D. H., Imelfort, M., Skennerton, C. T., Hugenholtz, P., and Tyson, G. W. (2015). Checkm: assessing the quality
112 of microbial genomes recovered from isolates, single cells, and metagenomes. *Genome research*, 25(7):1043–1055.
- 113 Rhie, A., Walenz, B. P., Koren, S., and Phillippy, A. M. (2020). Merqury: reference-free quality, completeness, and
114 phasing assessment for genome assemblies. *Genome biology*, 21(1):1–27.
- 115 Rissman, A. I., Mau, B., Biehl, B. S., Darling, A. E., Glasner, J. D., and Perna, N. T. (2009). Reordering contigs of
116 draft genomes using the mauve aligner. *Bioinformatics*, 25(16):2071–2073.
- 117 Rozov, R., Brown Kav, A., Bogumil, D., Shterzer, N., Halperin, E., Mizrahi, I., and Shamir, R. (2017). Recycler: an
118 algorithm for detecting plasmids from de novo assembly graphs. *Bioinformatics*, 33(4):475–482.
- 119 Sadedin, S. P. and Oshlack, A. (2019). Bazam: a rapid method for read extraction and realignment of high-throughput
120 sequencing data. *Genome biology*, 20(1):1–6.
- 121 Seemann, T. (2014). Prokka: rapid prokaryotic genome annotation. *Bioinformatics*, 30(14):2068–2069.
- 122 Steinegger, M. and Salzberg, S. L. (2020). Terminating contamination: large-scale search identifies more than 2,000,000
123 contaminated entries in genbank. *Genome biology*, 21(1):1–12.
- 124 Tennesen, K., Andersen, E., Clingenpeel, S., Rinke, C., Lundberg, D. S., Han, J., Dangl, J. L., Ivanova, N., Woyke, T.,
125 Kyrpides, N., et al. (2016). Prodege: a computational protocol for fully automated decontamination of genomes. *The*
126 *ISME journal*, 10(1):269–272.