

# Baargin: a Nextflow workflow for the automatic analysis of bacterial genomics data with a focus on Antimicrobial Resistance

Juliette Hayer                       <img alt="ORCID icon" data-bbox="11335 255 1135

workflows are available, like Bactopia ([Petit III, 2020](#)), which is highly flexible and complete in term of tools available. Therefore, we needed a lighter workflow, with only a few tools and databases installed for our collaborators that have only limited computing resources and storage. Also, *baargin* is specifically designed for detecting AMR genes and plasmid features, and include a decontamination step of the assembly, allowing the downstream analyses to be performed especially on the contigs belonging to the targeted species.

## Materials and Methods

### Features

Baargin is designed to automatically parallelise workflow steps. It does not require manual intervention from the users between steps. Each workflow step, called process, uses containers, via Docker or Singularity, which also greatly improves traceability and reproducibility. Additional processes can be easily added in the future as the workflow is designed in modules, making it flexible for adding or removing steps.

### Workflow

[Figure 1](#) describes the workflow:

1. Input can be either a folder containing paired-end short reads (fastq format), a folder containing already assembled contigs (fasta format files), or an index file containing the paths to pair-end short reads (fastq files) and to long reads (ONT, fastq file) for the same sample/strain in order to perform hybrid assembly. If assembled contigs are provided, the analysis will start at step 4.
2. Quality check and adapters removal is performed on the short reads using Fastp ([Chen et al., 2018](#)).
3. De novo assembly is run using SPAdes ([Prjibelski et al., 2020](#)) if only short reads were provided, and with Unicycler ([Wick et al., 2017](#)) for a hybrid assembly when short and long reads are provided.
4. Taxonomic assignment of the contigs is performed using Kraken2 ([Wood et al., 2019](#)) and the contigs classified at the taxonomic level provided by the user (with the taxid, and including the children taxa) are retrieved and therefore named as “*deconta*” for decontaminated contigs ([Lu et al., 2022](#)). The dataset containing all the contigs are named as “*raw*”. From here all the steps except the annotation (8) will be performed on both sets of contigs “*raw*” and “*deconta*”.
5. A quality check of the assembly is conducted using Quast ([Gurevich et al., 2013](#)) and BUSCO ([Manni et al., 2021](#)). For BUSCO, the users have the possibility to specify the taxonomic lineage database to use for searching the housekeeping genes (at the class level of the strain to analyse for example: *enterobacteriales\_odb10*)
6. The contigs (*raw* and *deconta*) are then screened to identify the sequence type of the strain using the MLST tool (Multi-Locus Sequence Typing) ([Seemann, 2022](#)).
7. The contigs are subsequently submitted to plasmid identification using PlasmidFinder ([Carattoli et al., 2014](#)) and additionally with Platon if the user provides a database for it ([Schwengers et al., 2020](#)).
8. Antimicrobial Resistance Genes (ARGs) are then searched in the contigs using both CARD RGI ([Alcock et al., 2023](#)) and the NCBI AMRFinderPlus ([Feldgarden et al., 2021](#)). For certain species only, AMRFinderPlus can also detect some mutations conferring resistance, if the user provides that option.
9. A genome annotation is performed on the *deconta* contigs using Prokka by default ([Seemann, 2014](#)), or using Bakta ([Schwengers et al., 2021](#)) if the user provides a database for it.
10. Once all the strains datasets provided are annotated a pangenome analysis is done using Roary ([Page et al., 2015](#)).

## Output

The results are located in a nested folder architecture. For each dataset, a folder with the sampleID is created, and contains 5 subfolders: - qc - assembly - AMR - annotation - plasmids. These subfolders contain the main outputs from the concerned analyses. Additionally, at the root of the results folder (indicated by the user) 2 folders are created: *pangenome*, containing the results for Roary, and *compile\_results* that contains summary files for each analysis where the results for all datasets provided in input are compiled as presence/absence matrices for further comparative analyses.

## Discussion and conclusions

We presented here an easy-to-use workflow for Bacterial Assembly and Antimicrobial Resistance Genes detection In Nextflow: *baargin*. It allows the users to analyse genomic datasets from short and long sequencing reads, of several bacterial strains from the same species in one command line. The workflow will automatically assemble the genomes, check for contamination and specifically extract the sequences that belong to the expected taxon. It will then identify their sequence type and screen the assemblies for plasmids sequences and ARGs. The fact that *baargin* is implemented in Nextflow and is based on containers makes the analyses reproducible. Its modular design makes it easy to customise and extend, by adding new modules for new processes.

## Figures

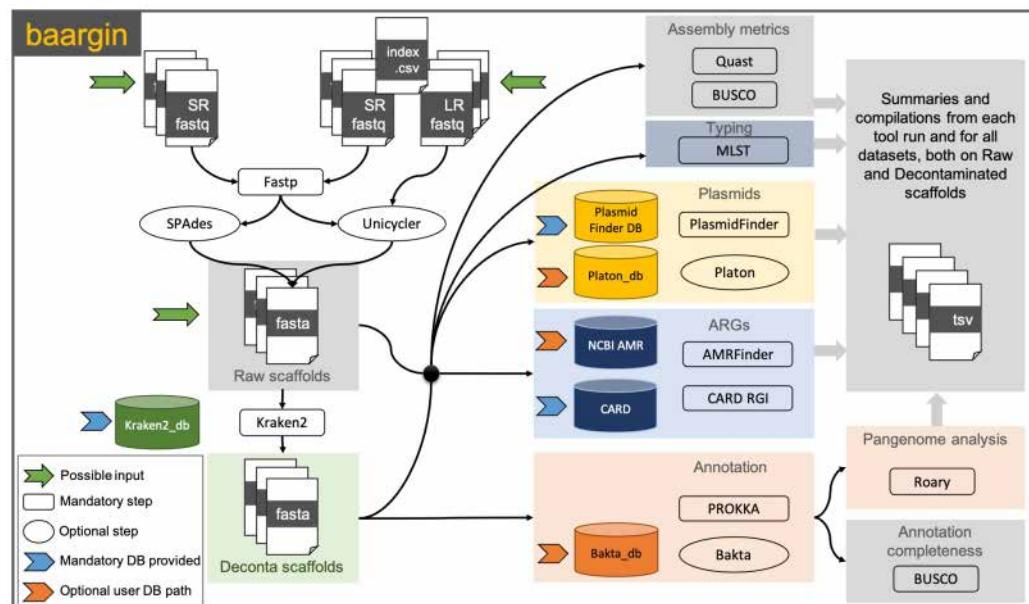


Figure 1: Flowchart of baargin workflow.

## Acknowledgements

We acknowledge contributions from Son Thai Nguyen and Julio Benavides during the development of this workflow.

The authors acknowledge the ISO 9001 certified IRD i-Trop HPC (South Green Platform) at IRD Montpellier for providing HPC resources to develop the workflow reported within this

paper. <https://bioinfo.ird.fr> <http://www.southgreen.fr>

## References

Alcock, B. P., Huynh, W., Chalil, R., Smith, K. W., Raphenya, A. R., Wlodarski, M. A., Edalatmand, A., Petkau, A., Syed, S. A., Tsang, K. K., Baker, S. J. C., Dave, M., McCarthy, M. C., Mukiri, K. M., Nasir, J. A., Golbon, B., Imtiaz, H., Jiang, X., Kaur, K., ... McArthur, A. G. (2023). CARD 2023: expanded curation, support for machine learning, and resistome prediction at the Comprehensive Antibiotic Resistance Database. *Nucleic Acids Research*, 51(D1). <https://doi.org/10.1093/NAR/GKAC920>

Carattoli, A., Zankari, E., Garcíá-Fernández, A., Larsen, M. V., Lund, O., Villa, L., Aarestrup, F. M., & Hasman, H. (2014). In silico detection and typing of plasmids using PlasmidFinder and plasmid multilocus sequence typing. *Antimicrobial Agents and Chemotherapy*, 58(7), 3895–3903. <https://doi.org/10.1128/AAC.02412-14>

Chen, S., Zhou, Y., Chen, Y., & Gu, J. (2018). fastp: an ultra-fast all-in-one FASTQ preprocessor. *Bioinformatics*, 34(17), i884–i890. <https://doi.org/10.1093/BIOINFORMATICS/BTY560>

DI Tommaso, P., Chatzou, M., Floden, E. W., Barja, P. P., Palumbo, E., & Notredame, C. (2017). Nextflow enables reproducible computational workflows. *Nature Biotechnology* 2017 35:4, 35(4), 316–319. <https://doi.org/10.1038/nbt.3820>

Feldgarden, M., Brover, V., Gonzalez-Escalona, N., Frye, J. G., Haendiges, J., Haft, D. H., Hoffmann, M., Pettengill, J. B., Prasad, A. B., Tillman, G. E., Tyson, G. H., & Klimke, W. (2021). AMRFinderPlus and the Reference Gene Catalog facilitate examination of the genomic links among antimicrobial resistance, stress response, and virulence. *Scientific Reports* 2021 11:1, 11(1), 1–9. <https://doi.org/10.1038/s41598-021-91456-0>

Gurevich, A., Saveliev, V., Vyahhi, N., & Tesler, G. (2013). QUAST: quality assessment tool for genome assemblies. *Bioinformatics*, 29(8), 1072–1075. <https://doi.org/10.1093/BIOINFORMATICS/BTT086>

Lu, J., Rincon, N., Wood, D. E., Breitwieser, F. P., Pockrandt, C., Langmead, B., Salzberg, S. L., & Steinegger, M. (2022). Metagenome analysis using the Kraken software suite. *Nature Protocols*, 17(12), 2815. <https://doi.org/10.1038/S41596-022-00738-Y>

Manni, M., Berkeley, M. R., Seppey, M., Simão, F. A., & Zdobnov, E. M. (2021). BUSCO Update: Novel and Streamlined Workflows along with Broader and Deeper Phylogenetic Coverage for Scoring of Eukaryotic, Prokaryotic, and Viral Genomes. *Molecular Biology and Evolution*, 38(10), 4647–4654. <https://doi.org/10.1093/MOLBEV/MSAB199>

Page, A. J., Cummins, C. A., Hunt, M., Wong, V. K., Reuter, S., Holden, M. T. G., Fookes, M., Falush, D., Keane, J. A., & Parkhill, J. (2015). Roary: rapid large-scale prokaryote pan genome analysis. *Bioinformatics*, 31(22), 3691–3693. <https://doi.org/10.1093/BIOINFORMATICS/BTV421>

Petit III, & R., R. A. (2020). Bactopia: a flexible pipeline for complete analysis of bacterial genomes. *Msystems*. <https://doi.org/10.1128/mSystems.00190-20>

Prjibelski, A., Antipov, D., Meleshko, D., Lapidus, A., & Korobeynikov, A. (2020). Using SPAdes De Novo Assembler. *Current Protocols in Bioinformatics*, 70(1), e102. <https://doi.org/10.1002/CPBI.102>

Schwengers, O., Barth, P., Falgenhauer, L., Hain, T., Chakraborty, T., & Goesmann, A. (2020). Platon: identification and characterization of bacterial plasmid contigs in short-read draft assemblies exploiting protein sequence-based replicon distribution scores. *Microbial Genomics*, 6(10), 1–12. <https://doi.org/10.1099/MGEN.0.000398>

Schwengers, O., Jelonek, L., Dieckmann, M. A., Beyvers, S., Blom, J., & Goesmann, A. (2021). Bakta: rapid and standardized annotation of bacterial genomes via alignment-free sequence identification. *Microbial Genomics*, 7(11), 685. <https://doi.org/10.1099/MGEN.0.000685>

Seemann, T. (2014). Prokka: rapid prokaryotic genome annotation. *Bioinformatics*, 30(14), 2068–2069. <https://doi.org/10.1093/BIOINFORMATICS/BTU153>

Seemann, T. (2022). *MLST*. <https://github.com/tseemann/mlst>

Thompson, T. (2022). The staggering death toll of drug-resistant bacteria. *Nature*. <https://doi.org/10.1038/D41586-022-00228-X>

Wick, R. R., Judd, L. M., Gorrie, C. L., & Holt, K. E. (2017). Unicycler: Resolving bacterial genome assemblies from short and long sequencing reads. *PLOS Computational Biology*, 13(6), e1005595. <https://doi.org/10.1371/JOURNAL.PCBI.1005595>

Wood, D. E., Lu, J., & Langmead, B. (2019). Improved metagenomic analysis with Kraken 2. *Genome Biology*, 20(1), 1–13. <https://doi.org/10.1186/S13059-019-1891-0>