The Journal of Open Source Software

BactScout: A reproducible Python pipeline for bacterial genome sequencing quality control

Nabil-Fareed Alikhan [1,2,4], **Varun Shamanna** [3,4], **Natacha Couto** [1,2,4], **GHRU2 Project Contributors**[4], **and David M Aanensen** [1,2,4]

1 Centre for Genomic Pathogen Surveillance, University of Oxford, United Kingdom **2** WHO Collaborating Centre on Genomic Surveillance of AMR, University of Oxford, United Kingdom **3** Central Research Laboratory, KIMS, Bengaluru, India **4** NIHR Global Health Research Unit on Genomics and enabling data for the Surveillance of AMR

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

BactScout is a Python-based pipeline for rapid, standardized quality assessment and taxonomic profiling of sequencing data from cultured bacterial isolates. It integrates tools like Fastp for read quality control, Sylph for species-level taxonomic profiling, and StringMLST for multi-locus sequence typing into a single, reproducible workflow. Designed for microbiologists and bioinformaticians working with large-scale bacterial genome data, BactScout provides interpretable quality metrics across read quality, species purity, GC content, and strain typing. These outputs support downstream applications such as genome assembly, resistance prediction, and phylogenetic analysis. The pipeline features a modular and extensible architecture with configurable quality thresholds, parallel sample processing, and detailed per-sample and batch-level reporting.

BactScout emphasizes reproducibility and ease of use through deterministic, containerized environments managed by Pixi, ensuring consistent results across platforms. By combining quality control, taxonomic profiling, and strain typing in a unified, automated framework, BactScout reduces manual effort and improves standardization in bacterial genomics workflows.

Statement of need

Quality assessment of bacterial sequencing data is a critical and often under-standardized step in genomic analysis pipelines, particularly for applications requiring high-confidence genome assemblies. Common challenges include contamination, low sequencing yield, poor read quality, and variable fragment lengths. While existing tools can report these metrics, interpreting their biological relevance typically requires manual assessment that depends on the species and sequencing context. This leads to inconsistent quality decisions and reduced reproducibility across projects and laboratories.

Existing tools such as *FastQC* (Andrews, 2010) provide detailed summaries of read-level sequencing quality but do not assess biological metrics like species purity or strain identity. Similarly, taxonomic profilers such as Kraken2 (Lu et al., 2022) require additional integration and interpretation to be useful in a quality control context, with species-specific decisions often left to the discretion of the operator. BactScout bridges this gap by combining technical and biological quality metrics within a single, reproducible workflow, automatically generating pass/fail assessments informed by species-aware benchmarks defined in QualiBact.

The pipeline is designed for routine isolate sequencing tasks commonly encountered in public health surveillance and clinical microbiology, where rapid and reproducible assessment of sequencing quality is essential prior to downstream analyses such as genome assembly or antimicrobial resistance prediction. By formalizing quality interpretation and applying species-aware thresholds, BactScout minimizes subjective decision-making and enhances consistency across bacterial genomics workflows. It serves as a rapid, automated screening step to identify high-quality samples suitable for deeper analyses, complementing more comprehensive assembly evaluation pipelines. BactScout was developed to meet the practical needs of bioinformaticians and has been successfully deployed within the Global Health Research Unit (GHRU) on Genomic Surveillance of Antimicrobial Resistance, demonstrating its robustness and scalability in high-throughput laboratory environments.

Implementation

BactScout automates post-sequencing quality control by integrating *Fastp* (Chen et al., 2018), *Sylph* (Unckless et al., 2023), and *StringMLST* (Datta et al., 2016) to identify problematic samples rapidly. It applies a configurable two-tier threshold system (WARN/FAIL) to classify samples as PASS, WARNING, or FAIL, summarizing results per sample and across batches. By combining standardized thresholds with species-aware criteria, BactScout streamlines decision-making and enhances reproducibility in bacterial WGS analysis.

BactScout is implemented in modern Python (3.11+) as a compact, modular pipeline designed for reproducibility and scalability across environments. It follows a CLI-first design (via *Typer*) with distinct modules for tool wrappers, metrics aggregation, and QC decision logic. External tool adapters (for *Fastp*, *Sylph*, *StringMLST*) isolate dependencies, ensuring a stable command-line and output schema. BactScout is packaged with Pixi and container images for reproducible deployment. It supports standalone use or integration within workflow managers such as Nextflow, job arrays, or GNU Parallel, ensuring consistent execution from laptops to HPC environments. The repository includes manifests and container recipes to facilitate reproducible builds.

Runtime behavior is configured through a single YAML file defining database paths, defaults, and quality thresholds. A consistent two-tier WARN/FAIL model allows users to adjust sensitivity to their context (e.g., stricter for clinical pipelines, relaxed for research).Each run generates per-sample directories containing structured CSV and JSON summaries, tool logs, and reports. The summaries include stable, well-documented field names for easy integration into LIMS or dashboards. A `summary` command aggregates all per-sample outputs into a batch-level report (`final_summary.csv`). The typical runtime for a 2 million read sample is 30-60 seconds with less than 8 GB RAM on one thread, scaling linearly with sample number.

The codebase follows a modular structure with distinct submodules for I/O, quality assessment, and result aggregation. Automated tests (∼75% code coverage) validate parsing, schema stability, and cross-platform compatibility via GitHub Actions. The project includes contribution guidelines, changelog, and code of conduct. New modules follow a simple results-dictionary contract to maintain compatibility.

Availability

Source code and documentation are available at https://github.com/ghruproject/bactscout under the GPLv3 license. Installation instructions, usage examples, and container images are provided via the project's documentation site. Example datasets and expected outputs are included for testing and demonstration purposes in the documentation.

Acknowledgements

This work was supported by National Institute for Health Research (NIHR133307). BactScout builds upon three outstanding open-source tools: **Fastp** (Chen et al., 2018), **Sylph** (Unckless et al., 2023), and **StringMLST** (Datta et al., 2016).

We thank contributors from the **NIHR Global Health Research Unit on Genomics and enabling data for the Surveillance of AMR** for feedback during design and testing, and the open-source community for providing the foundational libraries and infrastructure enabling this work.

References

Andrews, S. (2010). *FastQC: A quality control tool for high throughput sequence data.*

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

Datta, V., Oneš, I., Lima, L., Santos, T., Diaz, G. A., Rossi, P., Costa, C., Blom, J., & Pereira, P. M. (2016). stringMLST: String distance based MLST allele calling and core genome MLST strain typing. *Bioinformatics*, *32*(11), 1640–1642. https://doi.org/10.1093/bioinformatics/btw055

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), 2815–2839. https://doi.org/10.1038/s41596-022-00738-y

Unckless, R. L., Garcia, S. L., & Schatz, M. C. (2023). Sylph: Fast whole-genome average nucleotide identity (ANI) calculation using spaced k-mers. *bioRxiv*. https://doi.org/10.1101/2023.06.16.545235