The Journal of Open Source Software

BactScout: A Python pipeline for quality assessment and taxonomic profiling of bacterial sequencing data

Nabil-Fareed Alikhan [1,3], **Varun Shamanna** [2], **and GHRU Project Contributors** [3]

1 Centre for Genomic Pathogen Surveillance, University of Oxford, United Kingdom **2** KIMS **3** Global Healthcare and Research Unit

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Published: unpublished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. BactScout evaluates sequencing data across multiple quality dimensions—read quality, coverage depth, species purity, GC content, and strain typing—producing clear, interpretable quality metrics for downstream applications such as genome assembly, antimicrobial resistance prediction, genotyping, and phylogenetic inference.

The pipeline features a modular and extensible architecture with configurable quality thresholds, parallel sample processing, and detailed per-sample and batch-level reporting. A single command can process hundreds of samples, automatically generating summaries and visual outputs suitable for laboratory or high-performance computing environments.

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.

BactScout addresses this gap by providing an automated, standardized workflow for assessing sequencing quality of bacterial isolates. It integrates established, fast-performing tools with relatively low memory and CPU requirements to evaluate read quality, taxonomic purity, and strain identity, applying clear pass/fail criteria based on configurable thresholds and species-specific quality benchmarks (as defined in QualiBact).

The pipeline is designed for typical isolate sequencing tasks encountered in public health surveillance and clinical microbiology, where rapid and reproducible decisions on sample quality are essential before downstream analyses such as genome assembly or resistance prediction. By formalizing interpretation and integrating species-aware thresholds, BactScout reduces subjective decision-making and improves consistency in bacterial genomics quality control. We invision BactScout as an initial rapid screening step to identify high-quality samples suitable for further analysis, working along side more comprehensive pipelines, which would explore the genome assembly quality in greater depth.

BactScout Development

BactScout is implemented in **Python (3.11)** and designed for ease of deployment and extensibility.
The pipeline provides a **command-line interface** (CLI) built with *Typer*, supporting both individual and batch processing modes. Dependencies—including Fastp, Sylph, and StringMLST—are fully containerized and managed through **Pixi**, ensuring deterministic environments across Linux and macOS systems.

System-level tests, mock data, and example configurations are included to validate installations and future development. Continuous integration is performed via GitHub Actions on multiple platforms (Ubuntu 22.04 and macOS latest).
Parallelization is implemented through Python's thread pool executor, enabling efficient processing of large sample sets on high-performance or cloud computing environments.

Tools Utilized in BactScout

BactScout orchestrates three primary external tools to evaluate sequencing quality and taxonomic composition (Table 1).

Tool	Function	Quality Dimension
Fastp	Read-level quality control and adapter trimming	Read quality
Sylph	Taxonomic profiling and species purity estimation	Species identification
StringMLST	Multi-locus sequence typing (MLST) assignment	Strain typing

Each module outputs standardized JSON or tabular results that are parsed and evaluated against BactScout's threshold schema. Quality decisions (PASS/WARNING/FAIL) are derived from metrics such as: - Mean Q30 score and read length (Fastp) - Genome coverage and species composition (Sylph) - MLST type validity and completeness (StringMLST)

Default thresholds are defined in YAML configuration files and can be customized to project or organism-specific standards.

Quality Assessment and Reporting

BactScout performs quality evaluation in four primary domains:

1. **Read Quality:** Calculates mean read length and percentage of bases (Q30) from Fastp outputs.

2. **Coverage Depth:** Estimates genome coverage both from read counts and Sylph-derived genome size.

3. **Species Purity:** Quantifies dominant species proportion and flags cross-species contamination.

4. **Strain Typing:** Runs StringMLST when a single dominant species is detected to validate strain-level assignment.

Each sample is assigned an overall **status**—PASS, WARNING, or FAIL—with explanatory notes for each metric.
Reports include: - **Per-sample CSV summaries** with metric breakdowns
- **Batch-level summaries** aggregating performance across all samples
- **Optional Fastp HTML reports** for visual inspection

Outputs are human-readable and machine-parseable, facilitating integration with LIMS systems or downstream pipelines such as **Nextflow**.

Applications

BactScout is applicable across multiple domains:

- **Genome assembly projects** – Pre-assembly QC to identify high-quality inputs

- **Epidemiological surveillance** – Rapid strain verification and contamination detection

- **Sequencing QA/QC** – Standardized acceptance criteria for clinical or public health laboratories

- **Multi-center research cohorts** – Harmonized quality reporting across institutions

By producing interpretable, standardized outputs, BactScout helps ensure that only high-quality, biologically relevant sequencing data progress to downstream analysis.

Source Code and Documentation

Source code for **BactScout** is available at https://github.com/ghruproject/bactscout under the **GPLv3 License**.
Comprehensive documentation—covering installation, usage, configuration, and troubleshooting—is hosted on GitHub Pages.
It includes API references, example datasets, and developer contribution guidelines.

Acknowledgements

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 **Global Health Research Unit (GHRU)** for feedback during design and testing, and the open-source community for providing the foundational libraries and infrastructure enabling this work.

References

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

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