MethodsX 16 (2026) 103955

Contents lists available at ScienceDirect

MethodsX

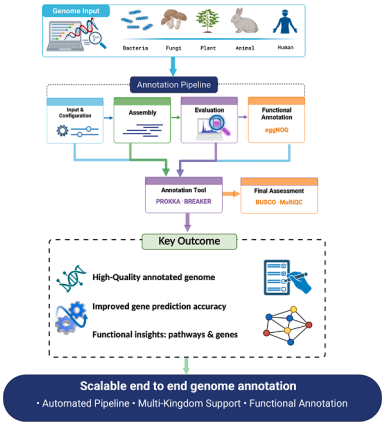
journal homepage: www.elsevier.com/locate/methodsx

AquaaG: A comprehensive pipeline for quality assessment and annotation of genomes

Jagriti Shukla, Kanka Mukherjee, Namrata Sahu, Shailesh Kumar *

Bioinformatics Lab, BRIC–National Institute of Plant Genome Research (NIPGR), Aruna Asaf Ali Marg, New Delhi 110067, India

GRAPHICAL ABSTRACT



ARTICLE INFO

Keywords:
Genome assembly
Genome annotation
BUSCO
QUAST
Prokka
BRAKER3
Bioinformatics pipeline

ABSTRACT

The rapid expansion of publicly available genome assemblies has made genome annotation an increasingly challenging task, particularly for large-scale analyses across prokaryotic and eukaryotic organisms. While several tools exist for assembly evaluation and annotation, their use often involves fragmented workflows that require extensive manual coordination. To overcome this limitation, we introduce AquaaG, an automated and reproducible genome annotation pipeline. AquaaG integrates genome assembly retrieval from NCBI, assembly quality assessment using QUAST, organism-specific annotation using Prokka for prokaryotes and BRAKER3 for eukaryotes, gene-space completeness evaluation using BUSCO, and functional annotation using EggNOG-mapper. The pipeline is configured through simple YAML files and supports species-level,

* Corresponding author.
E-mail address: shailesh@nipgr.ac.in (S. Kumar).

https://doi.org/10.1016/j.mex.2026.103955
Received 20 April 2026; Accepted 11 May 2026
Available online 12 May 2026

kingdom-level, and custom assembly-based analyses with optional submitter-based filtering. AquaaG therefore provides a practical and reproducible framework for high-throughput genome annotation and assessment.

The various features of the 'AquaaG' are as follows:

1. Genome annotation workflow: AquaaG automates genome retrieval, quality assessment with QUAST, and organism-specific annotation using Prokka and BRAKER3.

2. Configurable analysis modes: The pipeline supports species-wise, kingdom-wise, and assembly-list-based analyses, with optional submitter-based filtering.

3. Comprehensive outputs: AquaaG produces annotated genomes, BUSCO completeness reports, assembly quality reports, and metadata for downstream analyses.

Specifications table

This table provides general information on your method.

Subject area	Bioinformatics
More specific subject area	Genome annotation
Name of your method	AquaaG
Name and reference of original method	-
Resource availability	GitHub: https://github.com/skbinfo/AquaaG/tree/main; Web: www.nipgr.ac.in/AquaaG.

Background

High-throughput sequencing technologies have drastically increased the number of genomic assemblies deposited in public databases [1], providing new opportunities for comparative and evolutionary research. Despite these advances, the rapid expansion has introduced a considerable new set of practical challenges. A fundamental proportion of eukaryotic genomes remains unannotated even after assembly, leaving crucial gaps in their biological interpretation [2–5]. As a result, researchers often undertake a labor-intensive workflow that requires manually retrieving assemblies, assessing their quality, and running multiple annotation tools independently, with each step functioning separately and demanding considerable technical effort (Kitts et al., 2016; Salzberg, 2019). This fragmented workflow is slow, tedious to standardize, and susceptible to inconsistencies, ultimately hindering the efficient translation of raw genomic data into valuable biological insights [4,6].

Annotation approaches differ substantially between prokaryotic and eukaryotic systems. Prokaryotic genomes can often be annotated using streamlined tools that deliver rapid and reliable results [7]. In contrast, eukaryotic annotation must contend with

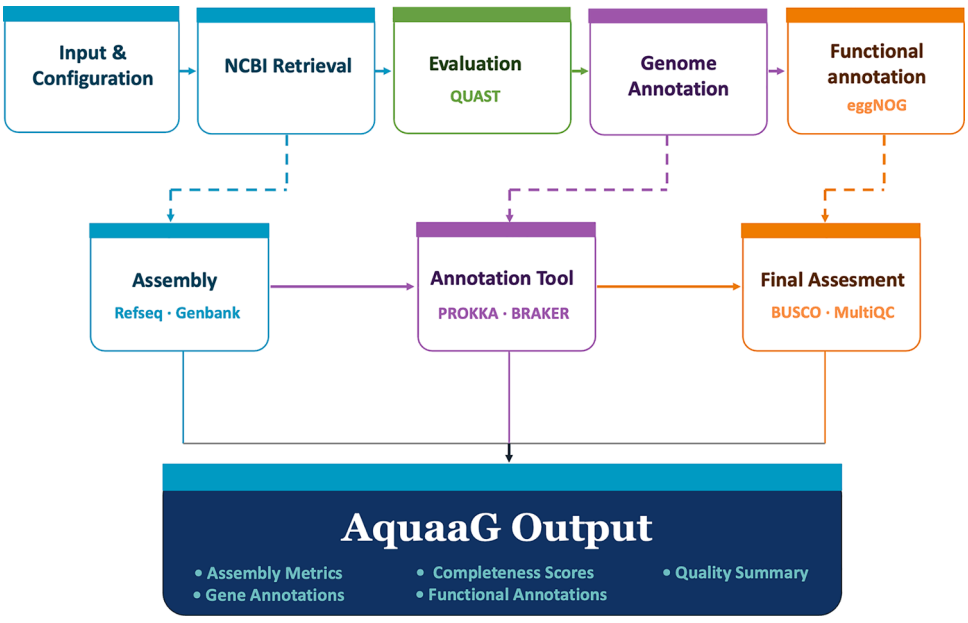


Fig. 1. Schematic overview of the AquaaG pipeline illustrating automated genome retrieval from NCBI, assembly quality assessment using QUAST, organism-specific annotation (Prokka/BRAKER), functional annotation (EggNOG-mapper), and final completeness evaluation using BUSCO.

complex gene structures, extensive repetitive content, and the need to integrate transcriptomic and protein evidence [2]. Even with recent advances in automated pipelines, the process remains technically demanding and dependent on the careful management of numerous software components, making reproducibility a persistent concern [2,5,7].

Public genomic assemblies often include heterogeneous metadata, which further complicates large-scale evaluation [6,8]. While sequence data are generated and published at unprecedented scale, information such as geographic origin, submitter affiliation, and sample context is frequently incomplete or inconsistently reported. This limits the ability to perform region-specific analyses and complicates efforts to evaluate the contributions of specific institutions or countries to global genomic resources. Researchers often resort to manual curation or custom filtering scripts that are neither scalable nor easily reproducible [3,4,6].

AquaaG was developed to address these challenges through an integrated, automated workflow for genome acquisition, quality assessment, and annotation across prokaryotic and eukaryotic organisms. A distinctive feature of the pipeline is its metadata-based regional filtering system, which uses NCBI submitter metadata to identify assemblies according to user-defined criteria. As a use case, curated keywords corresponding to Indian research institutes are included. This capability supports country-focused comparative genomics, pathogen surveillance, and biodiversity assessment in a transparent and reproducible manner. AquaaG relies on clear YAML-based configuration, supports multi-threaded execution, and ensures reproducible operation on Linux systems. For eukaryotic gene prediction, it automatically manages Docker-based environments, simplifying the deployment of complex tools. Together, these computational tools make AquaaG a practical, robust solution for large-scale genome annotation and curation (Fig. 1).

Method details

Genome assembly retrieval

AquaaG retrieves genomic assemblies from the National Center for Biotechnology Information using Python's Entrez utilities (https://biopython.org/docs/dev/Tutorial/chapter_entrez.html) and ncbi-genome-download, providing flexible support for species-level retrieval, kingdom-wide species discovery, assembly list-based downloads using user-provided accessions, and metadata-driven filtering.

AquaaG extracts key metadata fields from NCBI Assembly records, including AssemblyAccession, SpeciesName, AssemblyName, AssemblyStatus, SubmitterOrganization, TotalLength, and ContigN50, which are used for dataset organization and filtering. It automatically acquires reference genomes along with their corresponding GFF and protein FASTA files.

The metadata-based regional filtering module enables user-defined keyword-based selection (e.g., country, institution, or region). As a use case, a curated keyword set corresponding to Indian cities, institutes, and agencies is included to facilitate the identification of genomic submissions originating from India.

Assembly quality assessment

Assembly quality is assessed using QUAST, which computes a comprehensive set of metrics including N50, L50, NG50, and other contig-level statistics [9]. When corresponding GFF or FASTA reference files are available, QUAST also performs reference-guided comparisons to evaluate structural accuracy. The tool produces both HTML and text-based reports, enabling clear visualization and documentation of assembly quality. To enhance efficiency, all QUAST analyses are executed in parallel using user-defined CPU threads, allowing scalable performance across diverse datasets.

Genome annotation workflows

The annotation module in AquaaG provides dedicated workflows for both prokaryotic and eukaryotic genomes, ensuring accurate and reproducible gene prediction across diverse taxa. For bacterial and archaeal assemblies, annotation is performed using Prokka [7,10], which enables rapid structural and functional characterization of genomes. This workflow generates standardized outputs, including GFF, FAA, and FNA files, as well as detailed summary reports. Users may configure key taxonomic parameters such as genus, species, strain, and kingdom, allowing tailored annotation that aligns with the biological context of the dataset.

For eukaryotic genomes, the pipeline implements a more comprehensive annotation strategy based on BRAKER3 [5,11]. This workflow incorporates repeat identification using RepeatModeler followed by genome masking with RepeatMasker [12,13], ensuring improved accuracy during gene prediction. BRAKER3 is executed within a Docker container to maintain portability and reproducibility, and it supports the optional inclusion of external protein evidence to enhance annotation quality [11]. Configuration of underlying tools such as AUGUSTUS and GeneMark-ETP is handled automatically, reducing user intervention while maintaining robustness across different genome types [11,14]. Functional annotation of predicted genes is performed using eggNOG-mapper, enabling orthology assignment and pathway inference based on evolutionary relationships. This step complements structural annotation by assigning biological meaning to predicted gene models and generating functional categories and pathway-level insights.

BUSCO-based gene-space assessment

AquaaG employs BUSCO to assess the quality of each genome by examining the presence and status of conserved single-copy orthologs [15]. The analysis reports the proportions of complete, single-copy, duplicated, fragmented, and missing genes, offering a clear indication of how well core biological components are represented in the assembly. The workflow automatically manages the

appropriate BUSCO lineage datasets for both eukaryotic and prokaryotic genomes, such as bacteria_odb10 and fungi_odb10, enabling consistent and taxon-specific evaluation across diverse datasets.

Pipeline configuration

AquaaG offers a highly configurable framework, which allows users to tailor analyses to specific organismal groups and research objectives. All workflow parameters are managed through dedicated YAML configuration files **'Eu_config.yaml'** for eukaryotic genomes and **'Pr_config.yaml'** for prokaryotic genomes. These files allow precise control over essential components of the pipeline, including the number of computational threads, BUSCO lineage selection, annotation settings, QUAST parameters, and the optional inclusion of custom protein evidence [9,16]. Users can also specify their NCBI Entrez email to ensure compliant and stable data retrieval. This modular configuration system supports reproducible, transparent, and adaptable benchmarking across diverse species, assemblies, and experimental designs.

Modes of operation

AquaaG provides three modes of operation to accommodate different analytical scenarios and scales of genomic data. In Species Mode, the pipeline retrieves and processes genomic assemblies corresponding to a specific organism based on its scientific name. Kingdom Mode allows broader exploration by automatically discovering species within a selected kingdom, such as Bacteria or Fungi, and processing a user-defined number of representative assemblies. For more targeted or custom analyses, Assembly-File Mode enables users to supply their own list of accession numbers, bypassing metadata filters and focusing solely on the assemblies of interest. Together, these modes ensure flexibility and efficiency across both small-scale and large-scale genomic studies.

Pipeline execution

AquaaG is executed through a unified Python interface (AquaaG.py) that integrates all modules responsible for genome retrieval, quality assessment, annotation, and gene-space evaluation. The pipeline operates in three modes: Species Mode, Kingdom Mode, and Assembly-File Mode. These modes enable flexible processing at different taxonomic scales and allow users to analyse either individual genomes or large comparative datasets. All executions rely on user-defined YAML configuration files that standardize computational parameters and ensure reproducible workflows across organisms and experiments.

Computational environment

All analyses were conducted in a Linux environment configured with Conda and Docker [17]. The pipeline and dependencies were encapsulated within a custom Docker image. The setup involves cloning the repository and building the image directly from the provided Dockerfile:

```
git clone https://github.com/skbinfo/AquaaG.git.
cd AquaaG.
docker build -t aquaag-pipeline:latest .
```

Successful installation was verified by confirming that the Docker daemon was active and that the 'aquaag-pipeline:latest' image was built successfully, encompassing tools such as QUAST, Prokka, and BUSCO. To enhance reproducibility and usability, the GitHub repository provides a fully Dockerized environment for seamless execution of the pipeline. In addition, a minimal toy dataset is included to allow users to quickly validate installation and pipeline functionality.

Configuration parameter

The pipeline relies on dynamic environment variables passed directly to the Docker runtime that define parameters for eukaryotic and prokaryotic analyses. These variables govern the eukaryotic workflow, including BRAKER3, repeat annotation steps, BUSCO lineage specification, and CPU allocation, while providing equivalent settings for the prokaryotic branch and Prokka annotation. A typical eukaryotic configuration is supplied via environment variables such as:

```
-e ORGANISM="Arabidopsis thaliana".
-e TYPE="EK".
-e THREADS=50.
-e BUSCO="embryophyta_odb10".
```

Command-line executionexamples

In Species Mode, AquaaG retrieves assemblies from NCBI based on the organism's scientific name and automatically selects the most appropriate genomic record. When enabled, the metadata-based filter restricts downloads to assemblies submitted by Indian research institutes. Once the dataset is selected, the pipeline performs assembly assessment, annotation, and BUSCO evaluation in a single continuous run. An example execution for a eukaryotic organism using the Dockerized pipeline is shown below:

```
docker run –rm -it \
-u $(id -u):$(id -g) \
–group-add $(stat -c '%g' /var/run/docker.sock) \
-v /var/run/docker.sock:/var/run/docker.sock \
```

```
-v "$PWD":"$PWD" -w "$PWD"
-e HOME=/tmp \
-e EMAIL="your@email.com" \
-e TYPE="EK" \
-e THREADS=50 \
-e BUSCO="embryophyta_odb10" \
aquaag-pipeline \
-o output_eu
-s "Arabidopsis thaliana"
–num-assemblies 1 -I –run-func.
```

When the -I flag is omitted, AquaaG includes all assemblies. The same mode can be applied to prokaryotes by adjusting the environment variables.

Kingdom Mode enables large-scale comparative genomics by automatically discovering species within a specified kingdom. The following example processes three bacterial species:

```
docker run –rm -it \
-u $(id -u):$(id -g) \
–group-add $(stat -c '%g' /var/run/docker.sock) \
-v /var/run/docker.sock:/var/run/docker.sock \
-v "$PWD":"$PWD" -w "$PWD"
-e HOME=/tmp \
-e EMAIL="your@email.com" \
-e TYPE="PK" \
-e GROUP="bacteria" \
-e THREADS=50 \
-e BUSCO="bacteria_odb10" \
aquaag-pipeline \
-o output_pk_kingdom
-k "Bacteria" \
–num-species 3 \
–num-assemblies 1 -a –run-func.
```

Assembly-File Mode allows users to analyze custom datasets:

```
docker run –rm -it \
-u $(id -u):$(id -g)
–group-add $(stat -c '%g' /var/run/docker.sock)
-v /var/run/docker.sock:/var/run/docker.sock
-v "$PWD":"$PWD" -w "$PWD"
-e HOME=/tmp \
-e EMAIL="your@email.com" \
-e TYPE="PK" \
-e THREADS=50 \
-e BUSCO="bacteria_odb10" \
aquaag-pipeline \
-o output_pk_assemblies
–assembly-file assembly.txt -a –run-func.
```

Internal toolexecution

After initialization, AquaaG automatically invokes all internal modules using parameters derived from the configuration file. Assembly evaluation is carried out through QUAST, using the command:

```
quast.py ⟨assembly.fna⟩ -t ⟨threads⟩ -o quast_output.
```

Prokaryotic genome annotation is conducted using Prokka:

```
prokka –cpus ⟨threads⟩ –kingdom Bacteria ⟨assembly.fna⟩.
```

Eukaryotic repeat annotation integrates RepeatModeler and RepeatMasker through sequential execution:

```
BuildDatabase -name RM_db assembly.fna.
RepeatModeler -database RM_db -pa ⟨threads⟩.
RepeatMasker -pa ⟨threads⟩ -lib RM_db-families.fa assembly.fna.
```

BRAKER3 is run inside a Docker container to ensure consistent dependency management across systems:

```
docker run –rm -v $PWD:/data braker3_image braker.pl –genome=masked.fna –prot_seq=proteins.faa.
```

Gene-space completeness is assessed using BUSCO, which evaluates protein sets against lineage-specific ortholog databases:

```
busco -I proteins.faa -l ⟨lineage⟩ -m proteins -c ⟨threads⟩.
```

The automated orchestration of these tools ensures a uniform and reproducible workflow for both eukaryotic and prokaryotic

genomic datasets.

Method validation

AquaaG output files

Outputs produced by this pipeline in the case of both prokaryotes and eukaryotes are explained in Figs. 2 and 3, respectively. In the provided directory with the '-o' option as described above, the AquaaG pipeline provides structured outputs organized into separate directories containing individual reports for QUAST, BUSCO, and genome annotation results. Outputs are generated as structured directories containing separate reports (QUAST, BUSCO, annotation files):

Quast output

The QUAST report provides a comparative summary of assembly quality metrics for the analysed datasets, including the number of contigs and total assembly size across multiple length thresholds, which allows assessment of assembly fragmentation and continuity [9]. It reports the total count of contigs in each assembly, indicating the overall level of fragmentation, and shows how many contigs exceed increasing size cutoffs, helping to distinguish between highly fragmented assemblies and those with longer, more continuous contigs. In addition, QUAST calculates the total assembly length, representing the summed length of all contigs and enabling comparison of overall genome or sequence size across assemblies. Together, these metrics provide a clear and straightforward overview of assembly completeness and contiguity, highlighting differences in fragmentation, contig size distribution, and total sequence length between datasets.

Prokka output

The Prokka annotation provides a detailed representation of the genomic features present in the assembly. The predicted gene set includes several thousand coding sequences, reflecting a wide functional repertoire encoded within the genome [10]. In addition to protein-coding genes, Prokka identifies essential classes of non-coding RNA elements, including rRNAs, tRNAs, and tmRNA, which together support core transcriptional and translational processes. The presence of these features indicates that the assembly captures the fundamental components required for genome function and enables reliable downstream analyses such as functional characterization, comparative genomics, and evolutionary assessment.

Braker output

BRAKER produced three primary annotation outputs: the predicted protein sequences (braker.aa), the gene structure annotation file (braker.gtf), and the corresponding coding DNA sequences (braker.codingseq) [11]. The braker.gtf file lists the coordinates and exon–intron organization of each predicted gene, including transcript identifiers, CDS positions, strand information, and reading frame assignments. The 'braker.aa' file contains the translated protein sequences for all predicted transcripts, while the 'braker.codingseq' file provides their nucleotide coding regions extracted directly from the genome. Together, these outputs represent the final set of gene models generated by BRAKER and serve as the foundation for downstream structural and functional annotation.

BUSCO output

The BUSCO evaluation of the predicted proteome provides an overview of its completeness and overall quality [16]. The results indicate the presence of a substantial portion of conserved single-copy orthologs expected for the lineage, along with a noticeable fraction of duplicated and fragmented gene models. At the same time, a considerable number of expected genes were not detected. Together, this pattern suggests that while the annotation captures a core set of essential genes, there are signs of redundancy as well as gaps that likely reflect areas where the genome assembly or gene prediction pipeline may require further refinement.

MultiQC output

To facilitate the review of large-scale datasets, AquaaG automates the generation of unified summary reports. The pipeline natively executes MultiQC within the containerized environment to sweep the output directories and aggregate logs from QUAST, Prokka, and BUSCO into a single, centralized HTML report (multiqc_report.html). This provides a comprehensive technical overview of alignment statistics, annotation counts, and quality metrics in one file. Concurrently, a custom interactive dashboard (AquaG_Dashboard.html) is generated using 'Chart.js'. This dashboard provides dynamic, side-by-side visualizations of key metrics such as assembly contiguity (N50) and gene-space completeness (BUSCO), allowing users to perform rapid comparative assessments across all analyzed assemblies without manually traversing individual subdirectories.

Validation of AquaaG on prokaryotic and eukaryotic genomes

To validate the performance and versatility of AquaaG, the pipeline was applied to both prokaryotic

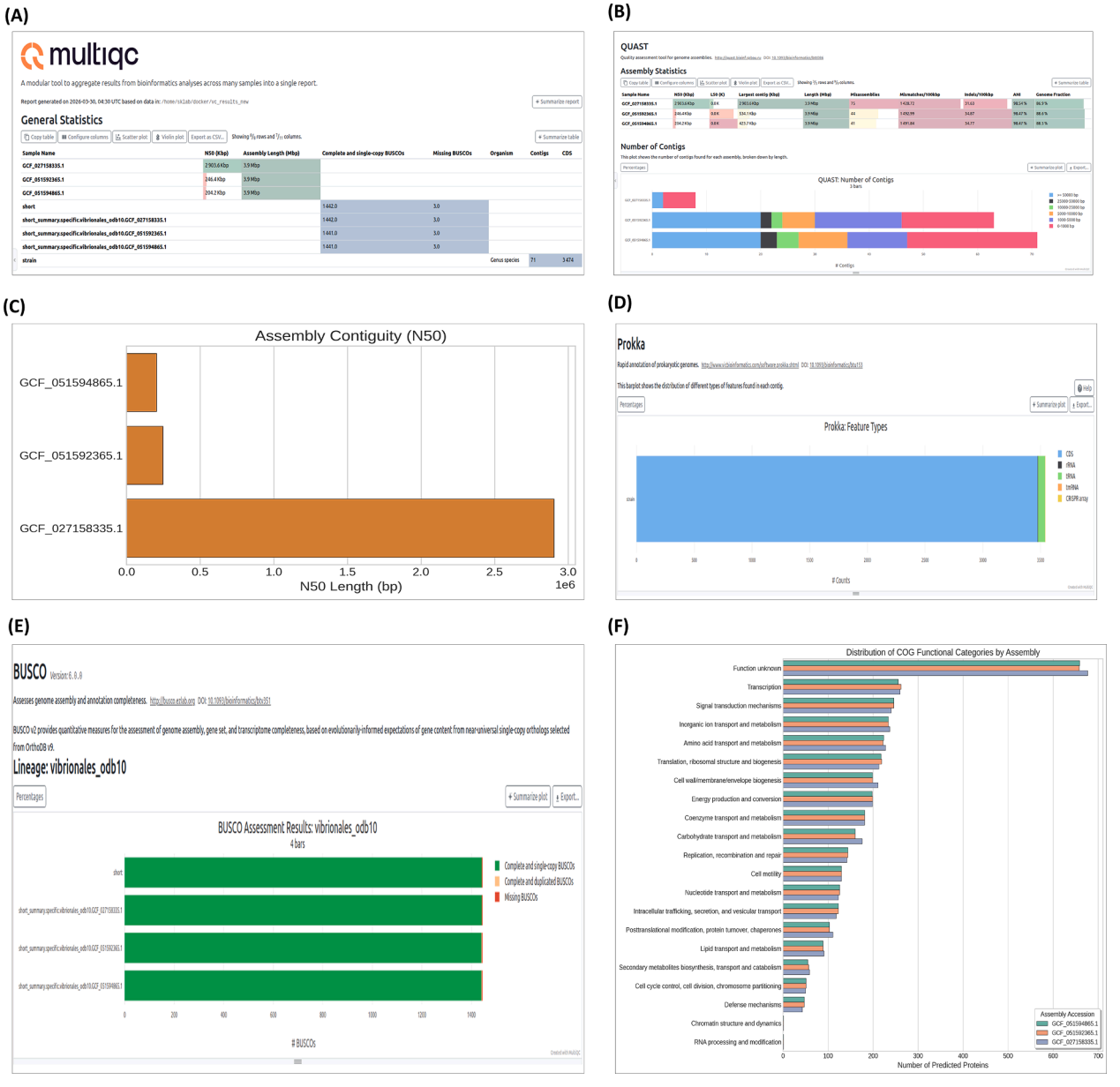


Fig. 2. Pipeline output summary for three *Vibrio* assemblies. (A) MultiQC general statistics; (B) QUAST assembly metrics and contig size distribution; (C) N50 contiguity comparison; (D) Prokka feature type annotations; (E) BUSCO completeness assessment using the *vibrionales_odb10* lineage; (F) EggNOG-mapper COG functional category distribution showing predicted protein counts per assembly.

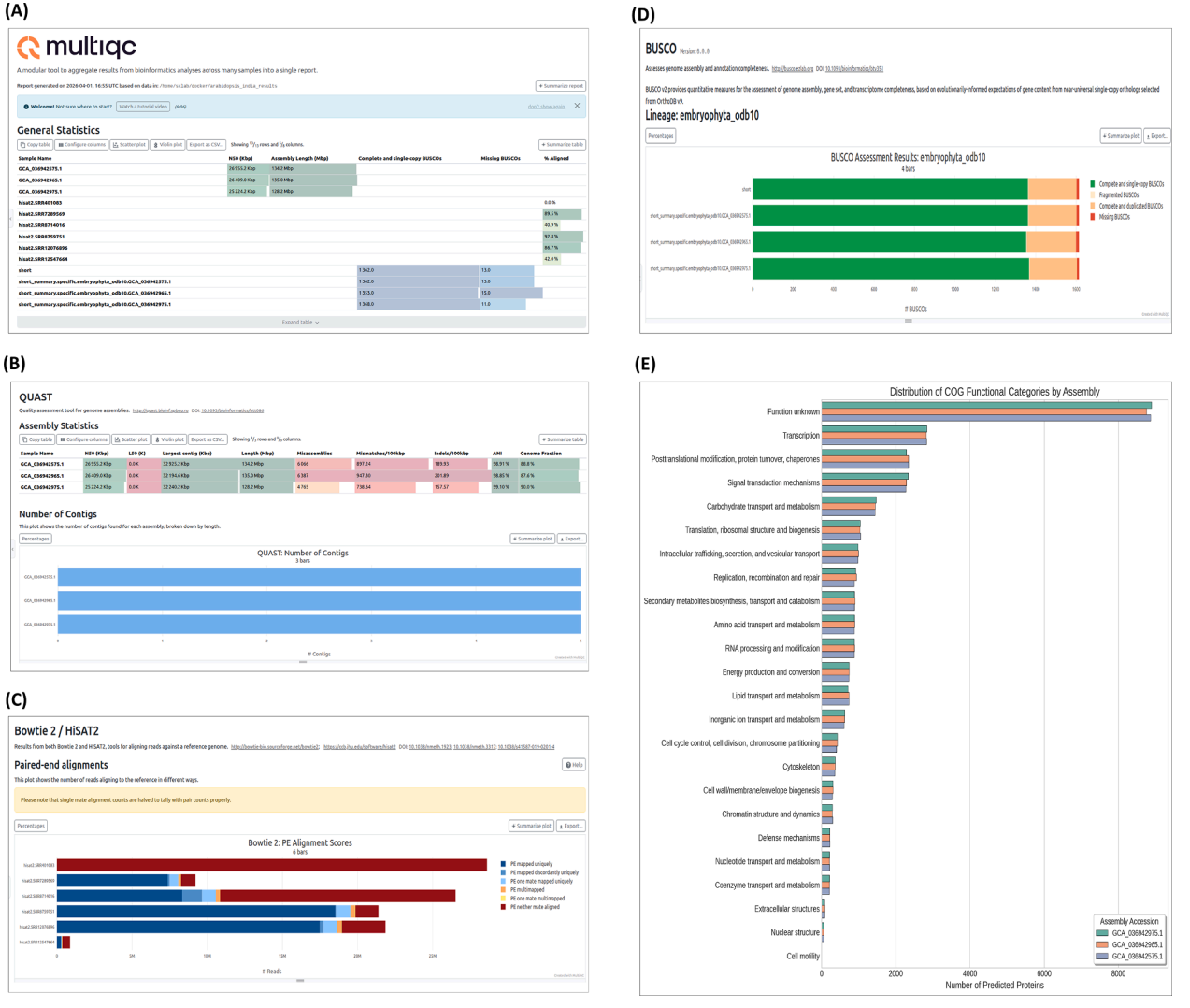


Fig. 3. Representative pipeline outputs for three *Arabidopsis* assemblies. (A) MultiQC general statistics; (B) QUAST assembly metrics and contig distribution; (C) Bowtie2/HISAT2 RNA-seq read alignment scores; (D) BUSCO completeness assessment using the *embryophyta_odb10* lineage; (E) EggNOG-mapper COG functional category distribution of predicted proteins per assembly.

(vibrionales_odb10.2019– 04– 24) and eukaryotic genome (embryophyta_odb10.2019– 11– 20) datasets from List of BUSCO.v4 lineages (https://busco.ezlab.org/list_of_lineages.html). The results demonstrate its ability to execute genome retrieval, quality assessment, annotation, and gene-space evaluation in a fully automated and reproducible manner across diverse organismal systems.

Prokaryotic genome analysis

For prokaryotic datasets, AquaaG completed all processing steps within a unified workflow, starting from genome retrieval to final annotation outputs (Fig. 2A). The pipeline executed efficiently using the Docker-based environment, ensuring reproducibility and minimal user intervention.

Assembly quality assessment using QUAST generated comprehensive reports summarizing key metrics such as total assembly length, number of contigs, and N50 values (Fig. 2B and C). These metrics provided a clear indication of genome contiguity and completeness, enabling rapid evaluation of assembly quality across datasets. Genome annotation using Prokka resulted in the identification of many protein-coding genes along with essential RNA elements, including rRNAs, tRNAs, and tmRNA (Fig. 2D). The standardized output files, including GFF, FAA, and FNA formats, facilitate downstream analyses such as functional annotation and comparative genomics. BUSCO-based assessment further validated the completeness of the annotated genomes by identifying conserved single-copy orthologs (Fig. 2E). The distribution of complete, duplicated, fragmented, and missing BUSCO genes indicated that the pipeline effectively captures core genomic features while maintaining annotation reliability. The result also includes the COG functional category distribution, showing predicted protein counts per assembly (Fig. 2F).

Overall, the prokaryotic workflow demonstrates that AquaaG provides a rapid and efficient solution for genome annotation and quality assessment in bacterial and archaeal systems.

Table 1
Comparative analysis of AquaaG and existing genome annotation pipelines. The table summarizes differences in pipeline type, organism support, automation, annotation strategies, input flexibility, and functional capabilities.

Feature	AquaaG	nf-core/ genomeAnnotator	PGAP	Funannotate
Pipeline Type	Integrated genome analysis pipeline	General annotation tool	Prokaryotic annotation pipeline	Eukaryotic genome annotation pipeline
Target Organisms	Prokaryotes & eukaryotes	Eukaryotes	Prokaryotes	Primarily eukaryotes
Input Flexibility	Species, kingdom, accession IDs, or local assemblies	Supports standard genome inputs	Predefined submission-based input	Supports genome assemblies with optional evidence
Automated Data Retrieval	Yes (NCBI-based)	Not included	Not included	Not included
Assembly Download & Filtering	Yes	Not included	Not included	Not included
Assembly Quality Assessment	Yes (QUAST)	Not included	Not included	Not included
Repeat Identification & Masking	Yes (RepeatModeler + RepeatMasker)	Basic repeat handling	Not required	Yes
Gene Prediction Strategy	Hybrid (ab initio + RNA-seq + protein evidence)	Primarily ab initio	Ab initio (GeneMarkS+)	Hybrid (AUGUSTUS, SNAP, GeneMark)
RNA-seq Integration	Yes (multiple input modes supported)	Not supported	Not supported	Yes
Protein Evidence Integration	Yes	Basic support	Not supported	Yes
Functional Annotation	Yes (EggNOG)	Basic annotation features	Standard annotation output	Yes (InterPro, EggNOG, Pfam)
Quality Assessment	Yes (BUSCO)	Not included	Internal validation	Yes (BUSCO)
Automation Level	High	Moderate	High	High
Customization	High (configurable pipeline)	Standard configuration options	Fixed workflow	Moderate
Output Formats	Standard formats (GFF3, FASTA, annotation tables)	Standard outputs	RefSeq/GenBank	GenBank + reports
Best Use Case	Research-driven, multi-omics genome analysis	General genome annotation	Genome submission to NCBI	Eukaryotic genome annotation
Limitations	Requires computational resources and setup	Focused on core annotation tasks	Restricted to prokaryotic genomes	Computationally intensive setup
Execution Time (Representative)	Fast for prokaryotes (~25–40 min); moderate–high for eukaryotes (~6–10 h) depending on genome size and threads	Moderate–high (depends on workflow modules)	Moderate (optimized for prokaryotes)	Moderate
Resource Usage	Scalable; supports multi-threading and parallel execution	High (workflow-dependent)	Moderate	Moderate
Installation & Reproducibility	Dockerized pipeline	Requires workflow setup (Nextflow/Conda)	Pre-configured NCBI pipeline	Conda-based setup required
Ease of Use	single-command execution, automated workflow	Moderate (requires workflow familiarity)	Moderate (submission-oriented)	Moderate–low (complex setup)

Eukaryotic genome analysis

For eukaryotic genomes, AquaaG implemented a more comprehensive workflow to accommodate the complexity of gene structures and repetitive elements. The pipeline successfully executed all stages, including repeat identification, masking, gene prediction, and completeness evaluation, within a Docker-based environment (Fig. 3A). Assembly quality was assessed using QUAST, which generated detailed reports describing genome contiguity and structural integrity (Fig. 3B). Metrics such as N50, total assembly size, and contig distribution provided insights into genome fragmentation and assembly quality. Gene prediction using BRAKER3 produced high-confidence gene models, including annotated gene structures, protein sequences, and coding regions. The integration of repeat masking and protein evidence improved annotation accuracy and enabled reliable gene model construction for complex eukaryotic genomes. It also includes the alignment of transcriptome reads to the assembled genome with Bowtie2/HISAT2 (Fig. 3C). The completeness of the predicted gene set was evaluated using BUSCO (Fig. 3D). The results indicated a substantial proportion of conserved orthologs, along with smaller fractions of fragmented and missing genes, reflecting both the quality of the assembly and the effectiveness of the annotation workflow. The COG functional category distribution of predicted proteins per assembly is the final step of the annotation (Fig. 3E).

Overall, the eukaryotic validation highlights the robustness of AquaaG in handling complex genomes, providing accurate annotation and standardized outputs suitable for downstream functional and comparative analyses.

Comparative evaluation of AquaaG with existing annotation pipelines

To further assess the capabilities of AquaaG in the context of existing genome annotation frameworks, a comparative evaluation was performed against widely used pipelines, including GenomeAnnotator, PGAP, and Funannotate. The comparison highlights key differences in terms of organism support, automation level, input flexibility, integration of multi-omics evidence, and overall functionality (Table 1). AquaaG demonstrates a more comprehensive and flexible design by supporting both prokaryotic and eukaryotic genomes within a unified framework. Unlike existing tools that are often restricted to specific organism types or require manual data handling, AquaaG enables automated genome retrieval, integrated quality assessment, and configurable annotation workflows. Additionally, the incorporation of RNA-seq and protein evidence, along with built-in quality evaluation using BUSCO and QUAST, positions AquaaG as a robust solution for large-scale and reproducible genome analysis.

Limitations and future directions

Despite its utility, AquaaG has several limitations. First, the performance of the pipeline depends on the quality of the input genome assemblies, and highly fragmented assemblies may adversely affect downstream annotation accuracy, particularly for complex eukaryotic genomes. Second, while AquaaG employs BUSCO to assess gene-space completeness, this evaluation is limited to conserved single-copy orthologs and may not fully capture lineage-specific or novel genes.

Third, functional annotation is currently limited to eggNOG-mapper-based orthology assignment and does not yet include extended functional characterization such as pathway reconstruction, protein domain analysis, or ontology-based classification. Lastly, the metadata-based regional filtering module relies on the completeness and consistency of NCBI submitter metadata, which may lead to the exclusion of some regionally relevant assemblies when metadata annotations are incomplete or ambiguous. Future developments of AquaaG will focus on addressing these limitations and expanding its analytical capabilities. Planned improvements include the integration of additional functional annotation layers, such as domain-based, pathway-level, and ontology-driven analyses, to provide deeper biological insight from annotated genomes. Enhancements to assembly evaluation may also include refinement steps aimed at improving annotation readiness.

Furthermore, future versions may incorporate phylogenetic inference tools such as IQ-TREE, RAxML, or FastTree to support automated evolutionary analyses. The inclusion of whole-genome alignment tools such as MUMmer4 or LASTZ is also anticipated to enable detailed genome-to-genome comparisons, including the detection of structural variations and syntenic regions. In addition, improvements in metadata parsing and validation strategies will be pursued to increase the accuracy and flexibility of geographic and institutional filtering. Together, these developments will strengthen AquaaG as a flexible, scalable, and extensible framework for large-scale genome assembly analysis and annotation across diverse organisms.

Funding

This research is supported by the BT/PR40243/BTIS/137/75/2023 project grant from the Department of Biotechnology (DBT), Government of India, and by the core grant of the BRIC—National Institute of Plant Genome Research (NIPGR) in the laboratory of SK.

CRediT author statement

Jagriti Shukla: Methodology, Software, Data curation, Formal analysis, Writing – original draft. Kanka Mukherjee: Methodology, Software, Data curation, Formal analysis, Writing – original draft. Namrata Sahu: Software. Shailesh Kumar: Conceptualization, Writing – review & editing, Validation, Supervision, Coordinated and conceived the project. All authors read and approved the final version of the manuscript.

Data availability

The AquaaG pipeline (version 1.0), including all source code and a user-friendly manual, is publicly available at the GitHub repository (https://github.com/skbinfo/AquaaG/tree/main) and the NIPGR website (https://www.nipgr.ac.in/AquaaG). Future updates, versioned releases, and change logs will be maintained through the GitHub repository to ensure transparency and reproducibility.

Declaration of competing interest

The authors have declared no competing interests.

Acknowledgments

The authors extend their gratitude to the DBT e-Library Consortium (DeLCON) for providing access to e-material and the Computational Biology & Bioinformatics Facility (CBBF) of NIPGR for their support. All the authors acknowledge the suggestions provided by Dr. Suchta Tripathy, Senior Principal Scientist, CSIR-IICB, Kolkata, India.

References

[1] P.A. Kitts, et al., Assembly: a resource for assembled genomes at NCBI, Nucleic. Acids. Res. 44 (1) (2016) D73–D80, https://doi.org/10.1093/nar/gkv1226.
[2] S. Lewis, M. Ashburner, M.G. Reese, Annotating eukaryote genomes, Curr. Opin. Struct. Biol. 10 (3) (2000) 349–354, https://doi.org/10.1016/s0959-440x(00)00095-6.
[3] S.L. Salzberg, Next-generation genome annotation: we still struggle to get it right, Biomed. Cent. Ltd. 20 (2019), https://doi.org/10.1186/s13059-019-1715-2.
[4] O.K. Tørresen, et al., An improved genome assembly uncovers prolific tandem repeats in Atlantic cod, BMC Genom. 18 (1) (2017), https://doi.org/10.1186/s12864-016-3448-x.
[5] M. Yandell and D. Ence, "A beginner's guide to eukaryotic genome annotation," 2012. doi: 10.1038/nrg3174.
[6] B. Grüning, et al., Practical computational reproducibility in the life sciences, Cell Press. 6 (2018) 631–635, https://doi.org/10.1016/j.cels.2018.03.014.
[7] T. Tatusova, et al., NCBI prokaryotic genome annotation pipeline, Nucleic. Acids. Res. 44 (14) (2016) 6614–6624, https://doi.org/10.1093/nar/gkw569.
[8] C. Tenopir, et al., Data sharing by scientists: practices and perceptions, PLoS. One 6 (6) (2011) e21101, https://doi.org/10.1371/journal.pone.0021101.
[9] A. Gurevich, V. Saveliev, N. Vyahhi, G. Tesler, QUAST: quality assessment tool for genome assemblies, Bioinformatics 29 (8) (2013) 1072–1075, https://doi.org/10.1093/bioinformatics/btt086.
[10] T. Seemann, Prokka: rapid prokaryotic genome annotation, Bioinformatics 30 (14) (2014) 2068–2069, https://doi.org/10.1093/bioinformatics/btu153.
[11] L. Gabriel, et al., BRAKER3: fully automated genome annotation using RNA-seq and protein evidence with GeneMark-ETP, AUGUSTUS, and TSEBRA, Genome Res. 34 (5) (2024) 769–777, https://doi.org/10.1101/gr.278090.123.
[12] J.M. Flynn, et al., RepeatModeler2 for automated genomic discovery of transposable element families, Proc. Natl. Acad. Sci. 117 (17) (2020) 9451–9457, https://doi.org/10.1073/pnas.1921046117.
[13] M. Tarailo-Graovac and N. Chen, "Using RepeatMasker to identify repetitive elements in genomic sequences," 2009. doi: 10.1002/0471250953.bi0410s25.
[14] M. Stanke, O. Keller, I. Gunduz, A. Hayes, S. Waack, B. Morgenstern, AUGUSTUS: a b initio prediction of alternative transcripts, Nucleic. Acids. Res. 34 (WEB. SERV. ISS) (2006) W435–W439, https://doi.org/10.1093/nar/gkl200.
[15] M. Manni, M.R. Berkeley, M. Seppey, E.M. Zdobnov, BUSCO: assessing genomic data quality and beyond, Curr. Protoc. 1 (12) (2021), https://doi.org/10.1002/cpz1.323.
[16] M, Z.E.M. Seppey Mathieu, Manni, BUSCO: assessing genome assembly and annotation completeness, in: M. Kollmar (Ed.), Gene Prediction: Methods and Protocols, Springer New York, New York, NY, 2019, pp. 227–245, https://doi.org/10.1007/978-1-4939-9173-0_14.
[17] C. Boettiger, An introduction to Docker for reproducible research, in: ACM SIGOPS Operating Systems Review 49, 2015, pp. 71–79, https://doi.org/10.1145/2723872.2723882.