Standalone WGS QC Pipeline Script – User Documentation

This document describes three Python scripts that together form the quality control (QC) reporting and analysis suite for the **WGS/WES pipeline**. The scripts generate detailed per-sample and per-chromosome QC reports, summary tables, interactive HTML plots, and advanced correlation analyses with publication-quality figures.

Table of Contents

Overview

The WGS QC pipeline produces a wealth of quality metrics from aligned BAM files. These three scripts transform raw pipeline outputs into human-readable reports and statistical analyses:

Script	Purpose
`generate_qc_report.py`	Reads per-sample chromosome-level JSON files and the sample-level summary TSV; produces summary tables (TSV) and an interactive HTML report with all plots (using Plotly).
`generate_qc_summary.py`	Aggregates all QC reports into a single summary table (CSV) with observed ranges, pass/fail counts per metric, and links to source files.
`wgs_qc_correlation.py`	Performs advanced correlation analysis between QC metrics and sample metadata (sex, age, race), creates publication-quality scatter plots, boxplots, heatmaps, and optionally splits samples by a coverage threshold for group comparisons.

All scripts are designed to be run independently, but they logically build on each other: first run `generate_qc_report.py` to create the detailed TSV reports, then `generate_qc_summary.py` to compile a summary, and finally `wgs_qc_correlation.py` to explore relationships.

Prerequisites

- **Python 3.6+**
- Required Python packages (install via `pip`):
  ```text
  text

  numpy pandas matplotlib seaborn plotly scipy adjusttext
  ```
- The scripts expect the directory structure produced by the WGS pipeline, specifically:
  - `QC_metrics/` – contains per-sample subfolders with merged JSON/TSV files.
  - `multiqc_data/` – contains MultiQC output files (`multiqc_general_stats.txt`, `mosdepth-*-plot.txt`, etc.).
  - `qc_output/` – created by `generate_qc_report.py` to store TSV reports and the HTML report.

- Optional: a sample information file (TSV) with columns like `Library_ID`, `Sex`, `Age`, `Race`, `Raw_Data_Size` for correlation analyses.

Configuration Files

Three configuration files control the behaviour of the scripts. Two are directly used (`thresholds.ini`, `summary_config.ini`), while `config.ini` is the main pipeline configuration and may be referenced for paths.

thresholds.ini

Defines pass/fail thresholds used by both `generate_qc_report.py` and `generate_qc_summary.py`.

```
[THRESHOLDS]
autosomal_coverage_cutoff = 30
percent_coverage_cutoff = 90
mean_median_ratio_cutoff = 1.5
freemix_cutoff = 0.01
mapped_percent_cutoff = 95
base_quality_cutoff = 30
cv_cutoffs = 5,10,15,20,25,30
```

Parameter	Description
`autosomal_coverage_cutoff`	Minimum acceptable mean/median autosomal coverage (X).
`percent_coverage_cutoff`	Minimum acceptable percentage of bases covered at a given depth (e.g., 15X, 30X).
`mean_median_ratio_cutoff`	Maximum acceptable ratio of mean to median coverage (indicates coverage skew).
`freemix_cutoff`	Maximum acceptable freemix contamination (VerifyBamID2).
`mapped_percent_cutoff`	Minimum percentage of reads mapped.
`base_quality_cutoff`	Minimum acceptable base quality (Phred score).
`cv_cutoffs`	List of coefficient of variation (CV) cutoffs for chromosome coverage imbalance (e.g., 5%, 10%, ...).

config.ini

This is the main pipeline configuration file (used by the Nextflow/WDL pipeline). It contains paths to reference files, BAM directories, and general settings. It is **not** directly read by the QC scripts, but its `[PATHS]` section may be useful for locating input data.

Example relevant entries:

```
[PATHS]
workdir = /path/to/analysis_dir
bamdir = /path/to/BAMs
refpath = /path/to/references
sampleinfo = /path/to/sample_info.tsv
analysis_type = WGS
...
```

summary_config.ini

Used by `generate_qc_summary.py` and `wgs_qc_correlation.py` to locate the necessary directories and files. It also specifies the optional sample information file and the 30X coverage cutoff for group analysis.

```
[Paths]
qc_output_dir = /path/to/qc_output
multiqc_data_dir = /path/to/multiqc_data
qc_metrics_dir = /path/to/QC_metrics
output_file = /path/to/output/qc_summary.csv
sex_info = /path/to/sample_info_39.tsv
group_cuttoff_30X = 75
```

Key	Description
`qc_output_dir`	Directory where `generate_qc_report.py` writes its TSV reports (e.g., `Autosomal_Coverage_Samples_report.tsv`).
`multiqc_data_dir`	Directory containing MultiQC output files (e.g., `multiqc_general_stats.txt`, `mosdepth-*-plot.txt`).
`qc_metrics_dir`	Directory containing per-sample subfolders with merged chromosome-level files (the pipeline's `QC_metrics` folder).
`output_file`	Path where the final summary CSV will be written (used by `generate_qc_summary.py`).

`sex_info`	(Optional) Path to a TSV file with sample metadata (see below).
`group_cuttoff_30X`	(Optional) Threshold for `Percent_autosome_coverage_at_30X` to split samples into High/Low groups for additional analysis in `wgs_qc_correlation.py`.

Sample information file format (tab-separated, first column must be `Library_ID`):

```
Library_ID	Sex	Age	Race	Raw_Data_Size
SAMPLE1	M	45	Asian	120
SAMPLE2	F	32	Caucasian	110
...
```

- `Sex` should be `M`/`F` or `Male`/`Female`.
- `Age` numeric.
- `Race` categorical.
- `Raw_Data_Size` numeric (e.g., GB).

Script 1: generate_qc_report.py

Purpose

Reads the per-sample per-chromosome JSON files and the sample-level summary TSV produced by the pipeline, and generates:

- Summary tables (TSV) for samples, chromosomes, contamination, coverage, etc.
- An interactive HTML report containing all plots (using Plotly).

Inputs

- `--qc-dir` : Path to the pipeline's `QC_metrics` directory (contains sample subfolders with merged JSON files).
- `--multiqc-dir` : Path to the MultiQC output directory (contains `mosdepth`, `samtools`, etc. subdirectories).
- `--outdir` : Directory where all reports and plots will be saved.
- `--sample-summary` (optional) : Path to the sample-level TSV (default: `QC_metricses_data_all_samples.tsv` inside `--qc-dir`).
- `--config` (optional) : Path to `thresholds.ini`.

Outputs

Inside the specified `--outdir`:

- **TSV reports**:
    - `Autosomal_Coverage_Samples_report.tsv` – per-sample autosomal coverage statistics.
    - `Autosomal_Coverage_Chromosomes_report.tsv` – per-chromosome averages.
    - `Samples_Contamination_report.tsv` – contamination metrics (freemix, avg_dp, etc.).
    - `Cumulative_Coverage_Samples_report.tsv` – cumulative coverage at 5X,10X,...,30X.
    - `Coverage_per_contig_Samples_Chromosome_report.tsv` – per-contig coverage and CV.
    - `Chromosome_coverage_imbalance_report.tsv` (duplicate of above).
    - `Percent_mapped_reads_report.tsv` – mapping statistics.
    - `insert_quality_report.tsv` – insert size statistics.
    - `base_quality_report.tsv` – base quality statistics.
- **HTML report**: `qc_report.html` – contains all plots (cumulative coverage, coverage distribution, per-contig boxplots, mapping stats, insert size, base quality, and numerous chromosome-wise plots).

Usage

```
python generate_qc_report.py \
    --qc-dir /path/to/QC_metrics \
    --multiqc-dir /path/to/multiqc_data \
    --outdir /path/to/qc_output \
    --config /path/to/thresholds.ini
```

If `--sample-summary` is omitted, the script looks for `QC_metricses_data_all_samples.tsv` inside `--qc-dir`.

Script 2: generate_qc_summary.py

Purpose

Aggregates all the TSV reports generated by `generate_qc_report.py` (plus the original MultiQC files) into a single summary table (CSV). For each QC metric, it reports:

- The observed range across samples.
- Pass/fail counts (based on thresholds defined in `thresholds.ini`).
- Percentages and links to source files.
- A new column `Type` categorises the metric (e.g., "Average coverage per contig", "Percentage autosome coverage").

Inputs

- A configuration file (default: `summary_config.ini`) containing the paths to the input directories and output file (see Configuration Files).
- The script automatically reads:
  - `Autosomal_Coverage_Samples_report.tsv`
  - `Samples_Contamination_report.tsv`
  - `base_quality_report.tsv`
  - `Cumulative_Coverage_Samples_report.tsv`
  - `mosdepth-coverage-per-contig-multi.txt` (from `multiqc_data`)
  - `multiqc_general_stats.txt`
  - `samtools_alignment_plot.txt`
  - `fastp-insert-size-plot.txt`
  - `QC_metricses_data_all_samples.tsv` (from `qc_metrics_dir`)

Outputs

- A CSV file (as specified by `output_file` in the config) with the following columns:
  - `QC Group` (e.g., Depth, Coverage, Alignment, Cross contamination)
  - `Type` (sub-category)
  - `QC Type` (specific metric name)
  - `Metric (from report)`
  - `Source File`
  - `Expected Threshold`
  - `Observed Value (range)`
  - `Pass Count`, `Fail Count`, `Total Samples`
  - `Pass %`, `Fail %`

Usage

```
python generate_qc_summary.py /path/to/summary_config.ini
```

If no argument is given, it defaults to `summary_config.ini` in the current directory.

Script 3: wgs_qc_correlation.py

Purpose

Performs in-depth correlation analysis between QC metrics and sample metadata (sex, age, race). It produces publication-quality plots (scatter plots, boxplots, heatmaps) and optionally splits samples into High/Low groups based on a user-defined 30X coverage threshold for additional group comparisons.

Key features:

- Merges all data sources (autosomal coverage, MultiQC stats, contamination, sample info, chromosome-level XY coverage).
- Sex check scatter plot with automatic labelling of extreme samples.
- Boxplots of coverage by sex (with Mann-Whitney U test) and by race (with Kruskal-Wallis test).
- Scatter plots of coverage vs. age/raw data size with Pearson correlation.
- Individual sample boxplots per chromosome.
- Combined heatmap of per-chromosome coverage across all samples.
- Group comparison (High vs Low coverage) including Mann-Whitney U / chi-square tests, boxplots, stacked bar charts, correlation heatmaps, mixed-type association matrices, and Cramér's V matrices.

Inputs

- **Mandatory**: a configuration file (same as for `generate_qc_summary.py`) that contains `[Paths]` with all necessary directories and the optional `group_cuttoff_30X`.
- The script reads the same files as `generate_qc_summary.py` plus:
    - `mosdepth-xy-coverage-plot.txt` (for sex chromosome coverage)

    - Per-sample chromosome TSV files (if present) from `qc_metrics_dir` (e.g., `SAMPLE_merged_all_chrom_qc_metrics.tsv`).

Outputs

All plots are saved in a subdirectory `plots/` (created alongside the `output_file` parent directory) and its subfolder `plots/boxplots/` for individual sample chromosome boxplots. Both **PNG (300 dpi)** and **PDF** versions are generated.

Main outputs:

- `sex_check_scatter.png/pdf` – scatter plot of chrX vs chrY coverage with inferred and known sex.
- `15X_coverage_by_sex.png/pdf`, `30X_coverage_by_sex.png/pdf` – boxplots with statistical annotations.
- `15X_coverage_by_race.png/pdf`, `30X_coverage_by_race.png/pdf` – boxplots with Kruskal-Wallis test.
- `15X_coverage_vs_age.png/pdf`, `30X_coverage_vs_age.png/pdf` – scatter plots.
- `15X_coverage_vs_raw_size.png/pdf`, `30X_coverage_vs_raw_size.png/pdf` – scatter plots.
- `corr_15X_<metric>.png/pdf`, `corr_30X_<metric>.png/pdf` – scatter plots for each available metric (e.g., `total_reads`, `dup_pct`, `freemix`), with low-coverage samples highlighted.
- `correlation_matrix.png/pdf` – heatmap of pairwise Pearson correlations among numeric QC metrics.
- `chrom_15X_heatmap.png/pdf`, `chrom_30X_heatmap.png/pdf` – heatmaps of per-chromosome coverage across all samples.
- `chrom_15X_group_boxplot.png/pdf`, `chrom_30X_group_boxplot.png/pdf` – boxplots per chromosome comparing low vs high coverage samples.
- Individual sample chromosome boxplots: `plots/boxplots/<sample>_chrom_boxplots.png/pdf`.

If `group_cuttoff_30X` is defined in the config, an additional folder `coverage_analysis/` is created containing:

- `analysis_md/` and `analysis_recal/` subfolders with:
    - `statistical_tests.csv` – results of Mann-Whitney U / chi-square tests.
    - Boxplots for each continuous variable.

    - Stacked bar charts for each categorical variable.
    - `correlation_heatmap_continuous.png` – correlation matrix of continuous variables.
    - `association_matrix_all.png` – mixed-type association matrix (Pearson / Kruskal-Wallis / Cramér's V).
    - `categorical_association_matrix.png` – Cramér's V for categorical-categorical pairs.

Usage

```
python wgs_qc_correlation.py --config /path/to/summary_config.ini
```

Workflow Example

1. **Run the pipeline** (not covered here) to produce BAMs and the QC outputs.
2. **Generate the detailed QC reports**:

   ```
   python generate_qc_report.py \
       --qc-dir /data/WGS_QC_Full_39/QC_metrics \
       --multiqc-dir /data/WGS_QC_Full_39/multiqc_data \
       --outdir /data/WGS_QC_Full_39/qc_output \
       --config /data/WGS_QC_Full_39/thresholds.ini
   ```
3. **Create a summary configuration file** (e.g., `summary_config.ini`) pointing to the directories and the sample info.
4. **Generate the summary CSV**:

   ```
   python generate_qc_summary.py /data/WGS_QC_Full_39/summary_config.in
   i
   ```
5. **Run the correlation analysis**:

   ```
   python wgs_qc_correlation.py --config /data/WGS_QC_Full_39/summary_c
   onfig.ini
   ```

All outputs will be organised in the `qc_output` and `report` directories as specified.

Troubleshooting

Issue	Possible Solution
`generate_qc_report.py` cannot find JSON files.	Check that `--qc-dir` points to the correct `QC_metrics` folder and that the JSON files follow the pattern `*_merged_all_chrom_qc_metrics.json`.
`find_column` raises `KeyError`.	The script expects specific column names (e.g., `Biosample_id`, `Chromosome`, `Percent_autosome_coverage_at_30X`). Verify your input files contain these columns. You may need to adjust the search patterns in the code.
MultiQC files not found.	The script tries several common suffixes (e.g., `mosdepth-cumcoverage-dist-id.txt`, `mosdepth_cumcov_dist.txt`). If your MultiQC uses different naming, modify the suffix lists in the corresponding functions.
`generate_qc_summary.py` warnings about missing files.	Ensure all paths in `summary_config.ini` are correct and that the required files exist. Some metrics may be omitted if files are missing – this is not fatal.
`wgs_qc_correlation.py` fails with `KeyError` on column names.	The script expects certain column names after merging. Check that your sample info file has the expected column headers (first column must be `Library_ID`). Run with a debugger or print the column list after each merge.
No sex check plot.	`mosdepth-xy-coverage-plot.txt` must be present in `multiqc_data`. If missing, sex-related analyses are skipped.
`adjustText` not available.	Install it (`pip install adjustText`) to improve label placement on the sex scatter plot.
Plots are too crowded.	For large sample sets, some plots (like individual chromosome boxplots) may become unreadable. Consider increasing figure size or filtering samples.

Appendix: Sample Commands

```
# 1. Generate QC reports
python /home/ubuntu/WGSQC_pipeline/scripts/generate_qc_report.py \
    --qc-dir /home/ubuntu/DATA_DRIVE/WGS_QC/WGS_QC_39/QC_metrics \
    --multiqc-dir /home/ubuntu/DATA_DRIVE/WGS_QC/WGS_QC_39/multiqc_data \
    --outdir /home/ubuntu/DATA_DRIVE/WGS_QC/WGS_QC_39/qc_output \
    --config /home/ubuntu/WGSQC_pipeline/scripts/thresholds.ini

# 2. Prepare summary_config.ini (see example above)
cat > /home/ubuntu/DATA_DRIVE/WGS_QC/WGS_QC_39/summary_config.ini << EOF
[Paths]
qc_output_dir = /home/ubuntu/DATA_DRIVE/WGS_QC/WGS_QC_39/qc_output
multiqc_data_dir = /home/ubuntu/DATA_DRIVE/WGS_QC/WGS_QC_39/multiqc_data
qc_metrics_dir = /home/ubuntu/DATA_DRIVE/WGS_QC/WGS_QC_39/QC_metrics
output_file = /home/ubuntu/DATA_DRIVE/WGS_QC/WGS_QC_39/report/qc_summary.c
sv
sex_info = /home/ubuntu/WGSQC_pipeline/data/sample_info_39.tsv
group_cuttoff_30X = 75
EOF

# 3. Generate summary CSV
python /home/ubuntu/WGSQC_pipeline/scripts/generate_qc_summary.py \
    /home/ubuntu/DATA_DRIVE/WGS_QC/WGS_QC_39/summary_config.ini

# 4. Correlation analysis
python /home/ubuntu/WGSQC_pipeline/scripts/wgs_qc_correlation.py \
    --config /home/ubuntu/DATA_DRIVE/WGS_QC/WGS_QC_39/summary_config.ini
```