# QuaC: Implementing Quality Control Best Practices for Genome Sequencing and Exome Sequencing Data

Manavalan Gajapathy, Brandon M. Wilk, Donna M. Brown, Elizabeth A. Worthey

Center for Computational Genomics and Data Science, University of Alabama at Birmingham, Birmingham, AL

## Introduction

- Quality Control (QC) of genome sequencing (GS) and exome sequencing (ES) data is necessary to ensure that data are of sufficient quality for downstream analyses. Examples of QC failures that need to be identified include poor sequencing data quality, data not matching to sequenced individual's expected metadata (sex, ancestry, relatedness), sample contamination, batch effects, etc.
- While several QC tools are available to perform QC at various levels post sequencing, their output needs to be reviewed and interpreted in a very manual process. This is a major challenge in large projects where standardization and consistency are highly desired in terms of QC metrics used and the threshold values utilized for such metrics.
- Further, logging the results of QC review and disseminating them with those involved in downstream analysis in an understandable format can be challenging. This could result in downstream users not utilizing the prior QC review results or re-reviewing them on their own and thereby wasting time and effort.

## Results

- We have developed a pipeline called QuaC to run QC tools at the various stages of secondary analysis, to summarize QC results, and to allow visualization and sharing of results in an easy-to-consume format. We further standardized the logging of QC review results.
- QuaC runs several QC tools and further consumes QC results produced by its upstream companion alignment and small variant caller pipeline (Table 1).
- It includes a tool called QuaC-Watch, which consumes results from all the QC tools, performs QC checkup, and then summarizes whether samples have passed user-defined thresholds for QC metrics of interest or not. QuaC-Watch comes with pre-configured thresholds for both GS and ES data, and they were curated based on literature, in-house analyses and prior experience (Fig. 1).
- QuaC aggregates results produced by all the QC tools as well as QuaC-Watch using MultiQC, both at the single-sample- and project-level, and generates a single stand-alone, easy-to-distribute HTML report file (Fig. 1).
- We further devised a "Sample QC database" where manual QC review results would be stored using controlled flags (Table 2).

**QuaC** makes **QC easy** for Genome and Exome Sequencing data

**Pipeline** runs several QC tools for **BAM** and **VCF** files and accepts QC results of **FASTQs**.

**QuaC-Watch** tool performs **QC checkup** based on the expected, pre-configured thresholds and **summarizes** the results for easy consumption.

**Aggregates** and **visualizes** QC results using MultiQC, both at the single sample and project levels.

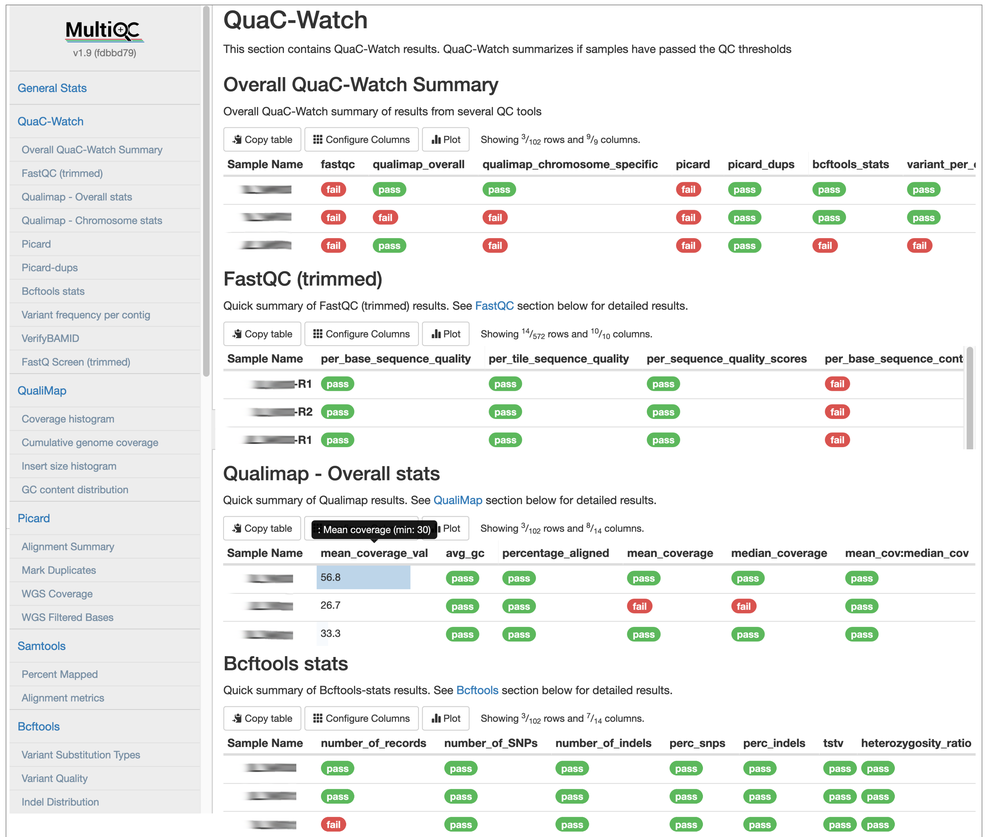


**Figure 1.** Aggregation and visualization of QC tools output and QuaC-Watch output using MultiQC at the project level. QuaC-Watch section shown here enables quick review of samples' QC results and helps identify samples that need further review. Additionally, similar MultiQC report is created at the single-sample level.

**Table 1.** QC tools included in QuaC. Tools run directly by QuaC using BAM and VCF files as input are listed here. For practical reasons, QC for FASTQ files (fastqc, FastQ Screen) as well certain QC of BAM files (Picard-MarkDuplicates) are performed by the upstream alignment and small variant calling pipeline. However, QuaC can consume their output as well for the QC aggregation step.

| Tool | Use | QC type |
|---|---|---|
| Qualimap | QCs alignment data in SAM/BAM files | BAM QC |
| Picard-CollectMultipleMetrics | Summarizes alignment metrics from a SAM/BAM file using several modules | BAM QC |
| Picard-CollectWgsMetrics | Collects metrics about coverage and performance of whole genome sequencing (WGS) experiments. | BAM QC |
| mosdepth | Fast BAM/CRAM depth calculation | BAM QC |
| indexcov | Estimate coverage from whole-genome bam or cram index (Skipped in exome mode) | BAM QC |
| covviz | Identifies large, coverage-based anomalies (Skipped in exome mode) | BAM QC |
| bcftools stats | Stats variants in VCF | VCF QC |
| verifybamid | Estimates within-species (i.e., cross-sample) contamination | Within-species contamination |
| somalier | Estimation of sex, ancestry and relatedness | Sex, ancestry and relatedness estimation |

**Table 2.** Sample QC database. As part of the manual sample QC review, results are logged in to our internal sample QC database for various QC measures at multiple levels using controlled terms. Type 1 flags includes terms pass, acceptable, poor and fail. Type 2 flags includes terms pass, fail and not applicable. This database allows the downstream GS/ES data consumers to quickly identify problematic samples along with their potential cause for failure. This empowers them to quickly determine if such sample can be used for their intended data analyses.

| Field | Explanation | Allowed values |
|---|---|---|
| Sample - Overall Status | Overall QC status considering results of all QC performed | Type 1 flags |
| FASTQ | Overall QC status considering results of all QC performed at FASTQ level | Type 1 flags |
| FASTQ Comment | Comments on QC at FASTQ level (e.g., small insert size, high adapter content, etc.) | Free text |
| BAM | Overall QC status considering results of all QC performed at BAM level | Type 1 flags |
| BAM Comment | Comments on QC at BAM level (e.g., low mean coverage, high duplication rate, etc.) | Free text |
| Other Species Contamination | Checks for sample contamination due to other species' genomic material | Type 1 flags |
| Human Cross-contamination | Checks for sample contamination due to other human's genomic material | Type 1 flags |
| Sex Check | Checks if predicted sex matches self-reported sex | Type 2 flags |
| Relatedness Check | Checks if predicted relatedness matches self-reported relatedness | Type 2 flags |
| Ancestry Check | Checks if predicted ancestry matches self-reported ancestry | Type 2 flags |
| Other Comments/Notes | Any other comments/notes concerning QC | Free text |

## Methods

- Pipeline component was developed using Snakemake, and QuaC-Watch and wrapper components were implemented using Python. It is a companion pipeline, which gets run after GS/ES samples are run through alignment and small variant calling, i.e., raw reads from FASTQs are aligned and then produced BAM and VCF files.
- QuaC isolates QC tool execution in a Conda environment within a Singularity container, as this setup provides the major advantage of reproducibility and portability.
- QuaC is run at a project level, and samples are provided as input in a pedigree file format (.ped), which allows to provide samples' relatedness and sex information to the pipeline.