

# MultiQC - Comprehensive Reference Guide

## Overview

MultiQC is a modular tool that aggregates results from multiple bioinformatics analyses across many samples into a single interactive HTML report. It searches for analysis output files, parses their content, and creates visualizations that enable cross-sample comparison and quality assessment.

**Website:** <https://multiqc.info/>

**Publication:** Ewels et al. Bioinformatics 2016

**Current Version:** 1.15+

**Applicable to:** Any NGS workflow producing standard tool outputs (RNA-seq, WGS, WES, ChIP-seq, ATAC-seq, BS-seq, metagenomics, etc.)

---

## What MultiQC Does

### Core Function

MultiQC is a **meta-analysis tool** that:

1. **Searches directories** for recognized output files from bioinformatics tools
2. **Parses metrics** from each tool's output format
3. **Aggregates data** across all samples
4. **Generates visualizations** comparing samples side-by-side
5. **Creates interactive HTML report** with plots, tables, and statistics
6. **Exports raw data** for further analysis

### Key Concept: Aggregation vs Individual Analysis

Tool Type	Function	Example
<b>Individual Analysis</b>	Analyzes one sample	FastQC, STAR, Salmon
<b>Meta-Analysis (MultiQC)</b>	Compares across samples	MultiQC

### Analogy:

- Individual tools = Taking temperature of each patient
  - MultiQC = Creating a chart comparing all patients' temperatures
-

## Why MultiQC is Essential

### Problems It Solves

#### Without MultiQC:

Project with 50 samples:

- 50 FastQC HTML reports (must open each individually)
- 50 STAR log files (must parse each manually)
- 50 RSeQC outputs (scattered across directories)
- No easy way to compare samples
- Outliers difficult to identify
- Hours of manual review

#### With MultiQC:

Project with 50 samples:

- 1 HTML report with all QC metrics
- Side-by-side comparison plots
- Interactive filtering and sorting
- Outliers immediately visible
- 5 minutes to review entire project

### Value Proposition

1. **Time Savings:** Review 100 samples in minutes vs hours
  2. **Outlier Detection:** Spot problematic samples instantly
  3. **Cross-Sample Trends:** Identify batch effects or systematic issues
  4. **Comprehensive Overview:** All QC metrics in one place
  5. **Publication Ready:** Export high-quality plots for papers
- 

## Supported Tools

MultiQC supports 100+ bioinformatics tools across various domains:

### Sequencing Quality Control

Tool	Input Files	Metrics Extracted
FastQC	*_fastqc.zip	Quality scores, GC content, adapters, duplication
FastQ Screen	*_screen.txt	Contamination screening results
Cutadapt	*.cutadapt.log	Trimming statistics, adapter removal
Trim Galore	*_trimming_report.txt	Trimming and quality filtering stats
fastp	fastp.json	Quality filtering, adapter trimming

## Alignment

Tool	Input Files	Metrics Extracted
<b>STAR</b>	<code>Log.final.out</code>	Mapping rates, splice junctions, mismatch rates
<b>HISAT2</b>	<code>*.hisat2.log</code>	Alignment rates, concordant pairs
<b>Bowtie2</b>	<code>*.bowtie2.log</code>	Mapping statistics
<b>BWA</b>	Parsed from SAM/BAM flags	Alignment metrics
<b>Bismark</b>	<code>*_report.txt</code>	Bisulfite conversion, methylation stats

## RNA-seq Quantification

Tool	Input Files	Metrics Extracted
<b>Salmon</b>	<code>meta_info.json, quant.sf</code>	Mapping rates, library type detection
<b>Kallisto</b>	<code>run_info.json</code>	Pseudoalignment stats
<b>featureCounts</b>	<code>*.summary</code>	Gene assignment statistics
<b>HTSeq</b>	<code>*.htseq.log</code>	Read counting statistics
<b>RSEM</b>	<code>*.cnt</code>	Expression quantification stats

## RNA-seq QC

Tool	Input Files	Metrics Extracted
<b>RSeQC</b>	Various <code>*.txt</code> outputs	Read distribution, gene body coverage, strand specificity
<b>Qualimap</b>	<code>qualimapReport.html</code>	Coverage statistics, insert size
<b>Picard</b>	Various <code>*_metrics.txt</code>	RNA metrics, duplication, insert size

## Variant Calling

Tool	Input Files	Metrics Extracted
<b>bcftools stats</b>	<code>*.bcftools_stats.txt</code>	Variant statistics, Ts/Tv ratios
<b>GATK</b>	Various GATK outputs	Variant quality, filtering stats
<b>VEP</b>	<code>*.vep.txt</code>	Variant annotation statistics
<b>SnpEff</b>	<code>*.snpEff.summary.html</code>	Variant effect statistics

## ChIP-seq / ATAC-seq

Tool	Input Files	Metrics Extracted
<b>MACS2</b>	<code>*_peaks.xls</code>	Peak calling statistics
<b>deepTools</b>	Various outputs	Coverage, fingerprint plots

Tool	Input Files	Metrics Extracted
<b>Phantompeakqualtools</b>	<code>*.spp.out</code>	ChIP-seq quality metrics

## General BAM/SAM Statistics

Tool	Input Files	Metrics Extracted
<b>Samtools</b>	<code>*.samtools.stats</code>	Alignment statistics, error rates
<b>Picard</b>	Various metrics files	Duplication, insert size, GC bias
<b>Sambamba</b>	<code>*.sambamba.log</code>	Duplicate marking stats
<b>Mosdepth</b>	<code>*.mosdepth.summary.txt</code>	Depth of coverage statistics

## Other Domains

- **Metagenomics:** Kraken, Centrifuge, MetaPhlAn
- **Amplicon:** QIIME, mothur
- **Long reads:** NanoPlot, NanoStat
- **Custom:** Custom data via JSON/YAML

Full list: <https://multiqc.info/modules/>

---

## How MultiQC Works

### File Discovery Process

1. Start in specified directory  
â†"
2. Recursively search all subdirectories  
â†"
3. Check each file against known patterns  
â†"
4. If match found â†' Parse file with appropriate module  
â†"
5. Extract metrics into internal data structure  
â†"
6. Aggregate across all samples  
â†"
7. Generate HTML report + data directory

### File Pattern Recognition

MultiQC uses **filename patterns** to identify tool outputs:

**Examples:**

```

*_fastqc.zip           â†' FastQC module
Log.final.out          â†' STAR module
*_screen.txt           â†' FastQ Screen module
meta_info.json          â†' Salmon module
*.RSeQC.read_distribution.txt  â†' RSeQC module

```

**Key point:** No need to specify which files are which - MultiQC auto-detects!

---

## Basic Usage

### Simplest Command

```
multiqc .
```

### What it does:

- Searches current directory (.) recursively
- Finds all recognized output files
- Creates `multiqc_report.html`
- Creates `multiqc_data/` directory

### Recommended Production Usage

```

multiqc \
    --force \
    --clean-up \
    --title "My Project RNA-seq QC" \
    --filename project_report \
    .

```

### Parameters explained:

Parameter	Purpose	Default
<code>--force</code>	Overwrite existing report	Off (fails if exists)
<code>--clean-up</code>	Remove intermediate files	Off (keeps tmp files)
<code>--title</code>	Report header title	"MultiQC Report"
<code>--filename</code>	Output filename (no .html)	"multiqc_report"
.	Directory to search	Required argument

---

## Command Line Options

### Directory and File Handling

#### Search Path

```

# Current directory (default)
multiqc .

# Specific directory
multiqc /path/to/results/

# Multiple directories
multiqc dir1/ dir2/ dir3/

# Specific files (not recommended - use directory search)
multiqc sample1_fastqc.zip sample2_fastqc.zip

```

**Best practice:** Use directory search, not individual files

**Why?:**

- Avoids "argument list too long" errors
  - Automatically finds all relevant files
  - Handles nested directory structures
  - More maintainable
- 

**--ignore / --ignore-samples** **Purpose:** Exclude specific files or samples

```

# Ignore files matching pattern
multiqc --ignore "*tmp" .

# Ignore specific directories
multiqc --ignore "**/backup/**" .

# Ignore samples by name
multiqc --ignore-samples "Failed_*" .

```

**Use cases:**

- Exclude failed samples
  - Ignore test runs
  - Skip backup directories
- 

## Output Customization

**--outdir** **Purpose:** Specify output directory

```
multiqc --outdir reports/ .
```

**Creates:**

```
reports/
```

```
multiqc_report.html  
multiqc_data/
```

---

**--filename Purpose:** Custom report filename

```
multiqc --filename project_QC .
```

**Creates:** project\_QC.html (note: .html added automatically)

---

**--title Purpose:** Report header title

```
multiqc --title "RNA-seq Batch 2 - January 2026" .
```

**Appears:** Top of HTML report, browser tab

---

**--comment Purpose:** Add text to report header

```
multiqc --comment "Preliminary analysis - review outliers" .
```

**Use cases:**

- Analysis notes
  - Warnings
  - Version information
- 

### Sample Name Handling

**--sample-names Purpose:** Rename samples using TSV file

```
multiqc --sample-names rename.tsv .
```

**rename.tsv format:**

```
old_name      new_name  
Sample1_R1_fastqc    Patient_001  
Sample2_R1_fastqc    Patient_002
```

**Use cases:**

- Anonymize patient IDs
  - Use meaningful names
  - Match metadata
-

**--sample-filters** **Purpose:** Filter samples by name pattern

```
multiqc --sample-filters "Patient_*" .
```

**Use case:** Include only specific samples

---

### Report Appearance

**--config** **Purpose:** Use custom configuration file

```
multiqc --config my_config.yaml .
```

**Configuration options (YAML):**

```
title: "My Custom Report"
subtitle: "Quality Control Analysis"
intro_text: "This report shows QC metrics for 50 RNA-seq samples."

# Custom colors
custom_plot_config:
    general_stats:
        - FastQC:
            min: 0
            max: 100

# Module order
module_order:
    - fastqc
    - star
    - salmon
    - rseqc

# Remove modules
exclude_modules:
    - cutadapt
    - bowtie2
```

**Use cases:**

- Consistent branding across projects
  - Custom plot configurations
  - Module reordering
- 

**--template** **Purpose:** Choose report template

```
multiqc --template default .
```

```
multiqc --template simple .
```

## Available templates:

- `default`: Standard interactive report (recommended)
  - `simple`: Minimal static report
  - `geo`: Simplified for GEO submission
  - Custom templates (advanced)
- 

## Data Export

`--data-format / --data-dir` Purpose: Control data directory output

```
# Include data directory (default)
multiqc --data-format tsv .

# Also export as JSON
multiqc --data-format tsv --data-format json .

# Custom data directory name
multiqc --data-dir custom_data_dir .

# Don't create data directory
multiqc --no-data-dir .
```

Data directory contents (`multiqc_data/`):

```
multiqc_data/
    multiqc_general_stats.txt      # Summary table
    multiqc_sources.txt           # List of parsed files
    multiqc_data.json             # All data (JSON format)
    multiqc_fastqc.txt            # FastQC metrics
    multiqc_star.txt              # STAR metrics
    multiqc_salmon.txt            # Salmon metrics
```

---

## Performance Options

`--quiet / --verbose` Purpose: Control logging output

```
# Suppress progress messages
multiqc --quiet .

# Detailed logging (debugging)
multiqc --verbose .
multiqc -v . # Short form
```

---

```
--flat / --fn_as_s_name  Purpose: Handle flat directory structures
# Ignore directory structure, use filenames as sample names
multiqc --flat .

# Use full filename as sample name
multiqc --fn_as_s_name .
```

**Use case:** All files in single directory with descriptive names

---

## Report Structure

### Interactive HTML Report

The MultiQC report consists of several sections:

#### 1. General Statistics Table Location: Top of report

**Contains:**

- Most important metrics from all tools
- One row per sample
- Sortable, filterable, searchable

**Example columns:**

Sample	FastQC Total Seq	STAR Uniquely Mapped%	Salmon Mapping%	RSeQC Assigned%
S001	45.2M	85.3%	91.2%	78.5%
S002	52.1M	87.1%	92.5%	80.3%

**Features:**

- Click column headers to sort
  - Search box to filter samples
  - Configure which columns to show
  - Export as CSV
- 

#### 2. FastQC Section Contains:

- Sequence quality histograms
- Per-sequence quality scores
- Per-sequence GC content
- Adapter content
- Sequence duplication levels

**Key plots:**

### **Sequence Quality Histograms (Heatmap):**

- Rows: Samples
- Columns: Base positions
- Color: Mean quality score at that position
- **Use:** Quickly identify samples with quality issues

### **Adapter Content:**

- Line plot showing % adapter per position
  - **Use:** Verify trimming worked
- 

### **3. Alignment Section (STAR/HISAT2/etc.) Contains:**

- Uniquely mapped reads percentage
- Multi-mapping reads
- Unmapped reads breakdown
- Mismatch rates

#### **Key plots:**

##### **Alignment Rates Bar Plot:**

- Stacked bars per sample
- Categories: Uniquely mapped, multi-mapped, unmapped
- **Use:** Identify low-mapping samples

##### **Mapping Rate Distribution:**

- Violin plot or histogram
  - **Use:** Assess overall mapping quality across project
- 

### **4. Quantification Section (Salmon/Kallisto/etc.) Contains:**

- Mapping rates
- Library type detection
- Fragment length distribution

#### **Key plots:**

##### **Mapping Rate Comparison:**

- Bar chart of mapping percentages
- **Use:** Verify consistent quantification

##### **Library Type:**

- Table showing detected strand-specificity
- **Use:** Confirm library prep type

---

## 5. RSeQC Section Contains:

- Read distribution (CDS, UTR, intergenic, intronic)
- Gene body coverage
- Inner distance (insert size)
- Read duplication

### Key plots:

#### Read Distribution (Stacked bar):

- Shows where reads map (exons, introns, intergenic)
- **Use:** Detect gDNA contamination, poor library prep

#### Gene Body Coverage (Line plot):

- Coverage across gene length (5' to 3')
  - **Use:** Detect 3' bias or degradation
- 

## 6. Module-Specific Sections

Each tool gets its own section with relevant plots and metrics.

---

### Data Directory

**Purpose:** Machine-readable data export

### Key files:

**multiqc\_general\_stats.txt** **Format:** Tab-separated values (TSV)

**Contains:** Summary table from report

### Example:

Sample	FastQC_total_sequences	STAR_uniquely_mapped_percent	Salmon_percent_mapped
S001	45200000	85.3	91.2
S002	52100000	87.1	92.5

### Use in R/Python:

```
# R
stats <- read.table("multiqc_data/multiqc_general_stats.txt",
                      header=TRUE, sep="\t")

# Python
import pandas as pd
stats = pd.read_csv("multiqc_data/multiqc_general_stats.txt", sep="\t")
```

---

**multiqc\_sources.txt** Format: Tab-separated

Contains: List of all files parsed by MultiQC

Example:

```
Sample Module File
S001 fastqc /path/to/S001_fastqc.zip
S001 star /path/to/S001.Log.final.out
S002 fastqc /path/to/S002_fastqc.zip
```

Use: Verify which files were included

---

**multiqc\_data.json** Format: JSON

Contains: All data in structured format

Use:

- Custom visualizations
  - Programmatic access
  - Integration with other tools
- 

**Tool-Specific Files Examples:**

```
multiqc_fastqc.txt      # FastQC metrics
multiqc_star.txt        # STAR alignment stats
multiqc_salmon.txt      # Salmon quantification
multiqc_rseqc_*.txt     # RSeQC metrics
```

Format: TSV, one row per sample

---

## Interpreting MultiQC Reports

### General Statistics Table

What to check:

1. **Total sequences:** Similar across samples?
  - Large variations may indicate technical issues
  - Expected variation: <2-fold difference
2. **Quality metrics:** Passing thresholds?
  - FastQC % Duplicates: <70% for RNA-seq

- GC%: Within expected range for organism
3. **Mapping rates:** Consistent and high?
    - RNA-seq: >70% uniquely mapped
    - WGS/WES: >90% mapped
  4. **Sort by each column:** Identify outliers
- 

### Cross-Sample Comparison Strategy

- Step 1: Review General Statistics  
    † Identify outlier samples
- Step 2: Check FastQC plots  
    † Verify quality, adapters, contamination
- Step 3: Check Alignment plots  
    † Confirm good mapping rates
- Step 4: Check Quantification  
    † Verify expected library type
- Step 5: Check RSeQC (RNA-seq)  
    † Confirm proper read distribution
- Step 6: Document findings  
    † Flag problematic samples  
    † Decide on exclusions
- 

### Red Flags to Watch For

Observation	Possible Issue	Action
Low total reads (<10M)	Under-sequencing	Re-sequence or exclude
High duplication (>80%)	Low library complexity	Check library prep
Low mapping rate (<50%)	Wrong reference, contamination	Investigate with FastQ Screen
Outlier GC content	Contamination	Check overrepresented sequences
3' bias (RNA-seq)	Degraded RNA	Check RIN scores
High intergenic reads	gDNA contamination	Check DNase treatment
Inconsistent library type	Mixed protocols	Verify sample metadata

---

## Workflow-Specific Interpretation

### RNA-seq Expectations

#### Good RNA-seq sample:

##### FastQC:

- Total sequences: 20-50M per sample
- % Duplicates: 40-70% (normal for RNA-seq!)
- GC content: Match transcriptome (~50% for human)
- Adapters: <1% after trimming

##### STAR:

- Uniquely mapped: >75%
- Multi-mapped: <20%
- Unmapped: <10%

##### Salmon:

- Mapping rate: >80%
- Library type: Consistent across samples

##### RSeQC:

- CDS exons: >60%
- Intronic: <15%
- Intergenic: <10%
- Gene body coverage: Relatively flat (5' to 3')

#### Common issues:

- High intronic/intergenic: gDNA contamination
  - 3' bias: RNA degradation
  - Low CDS%: Poor library quality
- 

### WGS/WES Expectations

#### Good WGS sample:

##### FastQC:

- Total sequences: 100M+ (30x coverage)
- % Duplicates: <20%
- GC content: Match genome (~40% human)

##### Alignment (BWA/Bowtie2):

- Mapped: >95%
- Properly paired: >90%
- Duplicates: <20%

**Coverage (Mosdepth/Qualimap):**

- Mean coverage: 30x
- % bases >10x: >90%

**Good WES sample:**

**FastQC:**

- Total sequences: 50-100M
- % Duplicates: 20-40% (higher than WGS, normal)

**Alignment:**

- Mapped: >95%
- On-target: >70%

**Coverage:**

- Mean on-target: 100x
  - % targets >20x: >90%
- 

## **ChIP-seq / ATAC-seq Expectations**

**Good ChIP-seq:**

**FastQC:**

- Total sequences: 20-40M
- % Duplicates: 30-60% (enrichment creates duplicates)

**Alignment:**

- Mapped: >70%
- Mitochondrial: <50%

**Peaks (MACS2):**

- Peaks called: 1,000-50,000
- FRIP score: >1%

**Good ATAC-seq:**

**FastQC:**

- Total sequences: 25-50M
- Fragment size: Nucleosome pattern visible

**Alignment:**

- Mapped: >80%
  - Mitochondrial: <50% (ideally <25%)
  - Duplicates: 40-60%
-

## Advanced Features

### Interactive Plot Features

All plots support:

1. **Hover tooltips:** See exact values
2. **Click to highlight:** Click sample to highlight across all plots
3. **Zoom and pan:** Drag to zoom, double-click to reset
4. **Export plots:** Download as PNG, SVG, or CSV
5. **Hide/show series:** Click legend to toggle samples

Example workflow:

1. Click outlier sample in General Stats  
→ Sample highlighted across ALL plots
  2. Review FastQC plots for that sample  
→ Identify quality issues
  3. Check alignment plots  
→ See if quality affects mapping
  4. Export plot showing the issue  
→ Include in QC report
- 

### Plot Toolbox

Located: Top-right of each plot

Features:

Button	Function
Download plot	Export as PNG or SVG
Download data	Export plot data as CSV
Configure plot	Adjust axes, colors, thresholds
Show/hide samples	Filter samples by name
Switch plot type	Change visualization (bar, line, scatter)

---

### Sample Filtering

Search box (top of General Stats):

```
# Show only samples matching pattern
Patient_0*
```

```
# Exclude samples  
-Failed_*  
  
# Multiple patterns  
Patient_001 Patient_002  
  
# Regex  
/^Control_/  
  
Use cases:
```

- Focus on subset of samples
  - Compare treatment vs control
  - Exclude failed samples temporarily
- 

## Table Configuration

**Toolbox** (General Statistics table):

**Features:**

- **Configure Columns:** Show/hide metrics
- **Sort:** Click any column header
- **Search:** Filter by sample name
- **Export:** Download as CSV
- **Highlight:** Color-code by thresholds

**Conditional formatting:**

- Values automatically colored by quality
  - Red: Below threshold
  - Orange: Warning
  - Green: Good
- 

## Configuration Files

**Custom Config (YAML)**

**Purpose:** Customize report appearance and behavior

**Basic structure:**

```
# Report metadata  
title: "My Project QC Report"  
subtitle: "RNA-seq Batch 5"  
intro_text: "Quality control for 48 RNA-seq samples sequenced January 2026."
```

```

# Customization
custom_logo: "/path/to/logo.png"
custom_logo_url: "https://mylab.edu"
custom_logo_title: "My Lab"

# Sample name cleaning
fn_clean_sample_names: true
sample_names_rename_buttons:
  - "Remove _R1"
  - "Remove _trimmed"

# Plot configuration
custom_plot_config:
  fastqc_sequence_quality_plot:
    ymin: 0
    ymax: 40

# Module order
module_order:
  - fastqc:
      name: "FastQC"
      info: "Quality control of raw sequencing data"
  - star:
      name: "STAR Alignment"
  - salmon:
      name: "Salmon Quantification"
  - rseqc:
      name: "RSeQC Analysis"

# Exclude specific modules
exclude_modules:
  - bowtie2
  - tophat

# Table columns configuration
table_columns_visible:
  FastQC:
    percent_duplicates: false
    percent_gc: true
  STAR:
    uniquely_mapped_percent: true
    multimapped_percent: false

Usage:
multiqc --config my_config.yaml .

```

---

## Sample Name Cleaning

**Purpose:** Clean up sample names automatically

**Common patterns:**

```
# Remove file extensions
fn_clean_exts:
  - ".fastq.gz"
  - "_fastqc"
  - ".sorted"

# Trim from sample names
fn_clean_trim:
  - ".Aligned.sortedByCoord.out"
  - "_001"

# Sample name regex replacement
sample_names_replace:
  - ["Sample_", ""]
  - ["_R[12]", ""]
  - ["_L00[1-4]", ""]
```

---

## Custom Content

**Purpose:** Add custom data to report

**Method 1: Custom JSON/YAML**

```
# custom_data.yaml
id: "my_custom_section"
section_name: "Custom Metrics"
description: "Laboratory metadata"
plot_type: "table"
data:
  Sample1:
    RIN: 8.5
    DV200: 75
    Concentration: 250
  Sample2:
    RIN: 7.8
    DV200: 70
    Concentration: 180

multiqc --custom-content custom_data.yaml .
```

## Method 2: TSV Files

```
# custom_metrics.tsv
Sample  RIN Library_Conc      Fragment_Size
S001    8.5 250 350
S002    7.8 180 320
```

MultiQC auto-detects \*.tsv files and includes them.

---

## Troubleshooting

### No Data / Empty Report

**Symptom:** Report generated but shows "No analysis results found"

**Causes:**

1. Wrong directory

```
# Check you're in right place
ls -R | grep -E "(fastqc.zip|Log.final.out|meta_info.json)"
```

2. Files not recognized

```
# Check file naming
ls *fastqc.zip      # Should find FastQC files
ls *Log.final.out   # Should find STAR logs
```

3. Files in subdirectories (but should be found)

```
# Force flat search
multiqc --flat .
```

4. Old MultiQC version

```
multiqc --version
pip install --upgrade multiqc
```

---

## Missing Samples

**Symptom:** Some samples in report, others missing

**Diagnosis:**

```
# Check multiqc_sources.txt
cat multiqc_data/multiqc_sources.txt | grep "missing_sample"
```

**Causes:**

1. File naming inconsistency

- Sample1\_fastqc.zip found
- Sample2\_fastqc.zip not found (typo? different location?)

## 2. Sample filtering

- Check if `--sample-filters` or `--ignore-samples` used

## 3. File parsing failed

```
# Run with verbose logging
multiqc -v .
# Check for parsing errors
```

---

## Module Not Appearing

**Symptom:** Tool outputs exist but module not in report

**Causes:**

### 1. Module excluded in config

```
# Check config
exclude_modules:
  - module_name # Remove this line
```

### 2. Incompatible file format

- Verify file matches expected pattern
- Check MultiQC documentation for exact format

### 3. Empty/corrupt file

```
# Check file size
ls -lh problematic_file

# Check first few lines
head problematic_file
```

---

## Very Large Report (>100MB)

**Symptom:** HTML file huge, slow to load

**Causes:**

### 1. Many samples (>500)

- Normal, but consider splitting

### 2. High-resolution plots

**Solutions:**

```
# Reduce plot data points
multiqc --flat --no-data-dir .

# Split into batches
multiqc --sample-filters "Batch1_*" -o batch1/
multiqc --sample-filters "Batch2_*" -o batch2/

# Use simple template
multiqc --template simple .
```

---

## Plot Display Issues

**Symptom:** Plots not rendering or appearing broken

**Causes:**

1. **JavaScript disabled in browser**
    - Enable JavaScript
  2. **Old browser**
    - Use modern browser (Chrome, Firefox, Safari)
  3. **Report moved without data directory**
    - Keep `multiqc_report.html` and `multiqc_data/` together
    - Or use `--no-data-dir` (embeds data in HTML, larger file)
- 

## Performance Optimization

**Large Projects (>100 samples)**

**Recommendations:**

```
# Use multiple cores (if available in MultiQC version)
multiqc --threads 4 .

# Disable data directory if not needed
multiqc --no-data-dir .

# Clean up intermediate files
multiqc --clean-up .

# Quiet mode (faster logging)
multiqc --quiet .
```

---

## Very Large Datasets (>1000 samples)

**Strategy:** Split into batches

```
# Option 1: By sample groups
multiqc --sample-filters "Group_A_*" -o groupA/
multiqc --sample-filters "Group_B_*" -o groupB/

# Option 2: By directory
multiqc batch1/ -o batch1_report/
multiqc batch2/ -o batch2_report/

# Option 3: Programmatic batching
for i in {1..10}; do
    multiqc --sample-filters "Batch${i}_*" -o batch${i}/
done
```

---

## Best Practices

### For Routine Analysis

âœ... **Always:**

- Run MultiQC at the end of pipeline
- Include all QC outputs in search path
- Review report before downstream analysis
- Keep data directory for further analysis
- Document outliers and exclusions

âœ... **Recommend:**

- Use descriptive title and filename
  - Add project comments/notes
  - Export general statistics table
  - Archive report with project
  - Share with collaborators
- 

### For Publication

âœ... **Include:**

- MultiQC report as supplementary material
- General statistics table
- Key plots (mapping rates, quality metrics)
- Sample exclusion criteria

âœ... **Customize:**

- Professional title
- Remove unnecessary modules
- Configure plot aesthetics
- Add lab logo (if allowed by journal)

**Example publication config:**

```
title: "RNA-seq Quality Control"
subtitle: "Supplementary Data"
intro_text: "Quality control metrics for 48 RNA-seq samples described in Doe et al. 2026."

# Clean appearance
custom_logo: null
show_analysis_time: false

# Essential modules only
exclude_modules:
  - cutadapt
  - fastq_screen

# High-quality plot export
export_plots: true
plots_force_flat: true
```

---

**For Collaboration**

âœ... **Share:**

- HTML report (standalone, no dependencies)
- Data directory (for custom analysis)
- Configuration file (for reproducibility)

âœ... **Document:**

- Software versions
- Quality thresholds used
- Samples excluded and why

**Example README:**

Project: RNA-seq Batch 5

Date: January 2026

Analyst: John Doe

MultiQC Report: project\_multiqc\_report.html

Quality Thresholds:

- Minimum reads: 15M

- Minimum mapping rate: 70%
- Maximum intergenic reads: 15%

Excluded Samples:

- Sample\_042: Low total reads (8.2M)
- Sample\_073: Low mapping rate (45%)

Software Versions:

- MultiQC: 1.15
  - FastQC: 0.12.1
  - STAR: 2.7.10a
  - Salmon: 1.10.0
- 

## Integration with Pipelines

### Nextflow Integration

```
process MULTIQC {
    publishDir "${params.outdir}/multiqc", mode: 'copy'

    input:
    path(all_files)

    output:
    path("multiqc_report.html"), emit: html
    path("multiqc_data"), emit: data

    script:
    """
        multiqc \
            --force \
            --title "${params.project_name}" \
            --filename multiqc_report \
    """
}

// Collect all QC outputs
workflow {
    // ... upstream processes ...

    all_qc = Channel.empty()
        .mix(FASTQC.out.zip)
        .mix(STAR.out.log)
        .mix(SALMON.out.results)
```

```

    .mix(RSEQC.out.stats)
    .collect()

    MULTIQC(all_qc)
}

```

---

### Snakemake Integration

```

rule multiqc:
    input:
        expand("fastqc/{sample}_fastqc.zip", sample=SAMPLES),
        expand("star/{sample}.Log.final.out", sample=SAMPLES),
        expand("salmon/{sample}/quant.sf", sample=SAMPLES)
    output:
        html="multiqc/multiqc_report.html",
        data=directory("multiqc/multiqc_data")
    params:
        title=config["project_name"]
    log:
        "logs/multiqc.log"
    conda:
        "envs/multiqc.yaml"
    shell:
        """
        multiqc \
            --force \
            --title "{params.title}" \
            --filename multiqc_report \
            --outdir multiqc \
            {input} \
            2> {log}
        """

```

---

### Shell Script Integration

```

#!/bin/bash
# run_qc_pipeline.sh

PROJECT="MyProject"
OUTDIR="results"

# Run individual QC tools
fastqc raw_data/*.fastq.gz -o ${OUTDIR}/fastqc/

```

```

# ... run STAR, Salmon, RSeQC ...

# Aggregate with MultiQC
multiqc \
    --force \
    --clean-up \
    --title "${PROJECT} Quality Control" \
    --filename ${PROJECT}_multiqc_report \
    --outdir ${OUTDIR}/multiqc \
    ${OUTDIR}/

echo "QC complete. Report: ${OUTDIR}/multiqc/${PROJECT}_multiqc_report.html"

```

---

## Comparison with Other QC Tools

### MultiQC vs Individual Tool Reports

Aspect	MultiQC	Individual Reports
<b>Scope</b>	Cross-sample	Single sample
<b>Outlier detection</b>	Excellent	Difficult
<b>Time to review</b>	Minutes	Hours
<b>Format</b>	Interactive HTML	Various (HTML, TXT, PDF)
<b>Data export</b>	Unified TSV/JSON	Tool-specific formats
<b>Customization</b>	High	Low

**Conclusion:** MultiQC complements, not replaces, individual tools

---

### MultiQC vs Similar Tools

Tool	Purpose	Pros	Cons
<b>MultiQC</b>	Multi-tool aggregation	100+ tools, interactive, active development	Requires multiple tools
<b>QualiMap</b>	BAM QC	Detailed coverage stats	Single tool
<b>FastQC + custom scripts</b>	DIY aggregation	Full control	Requires manual effort
<b>nf-core/rnaseq</b>	Full pipeline with QC	Integrated, automated	Pipeline complexity

---

## Common Use Cases

### Use Case 1: Batch Effect Detection

**Scenario:** Samples sequenced across multiple batches

**Strategy:**

1. Run MultiQC on all samples
2. Color samples by batch in General Stats
3. Look for batch-specific patterns:
  - GC content clustering by batch
  - Quality score differences
  - Mapping rate variations

**Solution if found:**

- Include batch as covariate in DE analysis
  - Consider batch correction (ComBat, limma)
- 

**Use Case 2: Sample Swap Detection**

**Scenario:** Verify sample identities

**Strategy:**

1. Check expected sex (XY/XX) vs gene expression
2. Verify known positive controls
3. Check biological replicates cluster together

**In MultiQC:**

- Use custom data for expected metadata
  - Compare with RSeQC chromosome counts
  - Flag mismatches for investigation
- 

**Use Case 3: Contamination Screening**

**Scenario:** Check for non-target organism contamination

**Strategy:**

1. Run FastQ Screen (aligns to multiple genomes)
2. Include in MultiQC report
3. Review % mapping to each organism

**Expected:**

Human samples:

- Human: 90%+
- Mouse: <1%
- Bacteria: <0.5%
- Adapters: <1%

**Red flags:**

- High mouse in human samples: Lab contamination
  - High bacteria: Environmental contamination
  - High adapters: Poor trimming
- 

#### Use Case 4: Library Prep QC

**Scenario:** Evaluate library preparation quality

**Metrics to check:**

1. **Duplication rate:** Amplification efficiency
2. **Insert size:** Size selection effectiveness
3. **GC bias:** PCR bias
4. **Fragment length:** Fragmentation quality

**MultiQC plots to review:**

- Picard duplication metrics
  - Picard insert size distribution
  - Picard GC bias
  - FastQC sequence length distribution
- 

## Command Reference

### Common Commands

```
# Basic usage
multiqc .

# Production usage
multiqc --force --title "Project QC" --filename report .

# Custom output directory
multiqc --outdir qc_reports/ .

# Ignore specific samples
multiqc --ignore-samples "Failed_*" .

# Use config file
multiqc --config config.yaml .

# Verbose logging
multiqc -v .

# Multiple directories
```

```

multiqc dir1/ dir2/ dir3/

# Export data only (no HTML)
multiqc --data-format tsv --no-data-dir .

```

---

## Config File Examples

### Minimal config:

```
title: "My Project"
```

### Standard config:

```

title: "RNA-seq QC Report"
subtitle: "Batch 5 - January 2026"
intro_text: "Quality control for 48 samples."

module_order:
  - fastqc
  - star
  - salmon
  - rseqc

exclude_modules:
  - cutadapt

```

### Advanced config:

```

title: "Publication Supplementary Data"
custom_logo: "lab_logo.png"
custom_logo_url: "https://lab.edu"

fn_clean_exts:
  - ".fastq.gz"
  - "_fastqc"

sample_names_rename_buttons:
  - "Remove _R1"
  - "Simplify names"

custom_plot_config:
  general_stats:
    read_count:
      min: 0
      max: 50000000

table_columns_visible:

```

```
FastQC:  
    total_sequences: true  
    percent_duplicates: true  
STAR:  
    uniquely_mapped_percent: true
```

---

## Related Documentation

- **FastQC:** docs/fastqc.md - Individual FASTQ quality metrics
  - **STAR Alignment:** docs/star\_align.md - RNA-seq alignment
  - **Salmon Quantification:** docs/salmon\_quant.md - Transcript quantification
  - **RSeQC:** docs/rseqc.md - RNA-seq quality control
  - **MultiQC Website:** <https://multiqc.info/>
  - **MultiQC Documentation:** <https://multiqc.info/docs/>
- 

**Document Version:** 2.0

**Last Updated:** January 2026

**MultiQC Version:** 1.15+

**Applicable to:** All NGS workflows producing standard tool outputs

---

## Quick Reference

### When to Use MultiQC

âœ... Use when:

- Analyzing multiple samples (3+)
- Need cross-sample comparison
- Want unified QC report
- Publishing results

âŒ Not needed when:

- Single sample analysis
- Tool doesn't have MultiQC module
- Need raw tool output details

### Most Important Checks

1. **General Statistics:** All samples have reasonable metrics
2. **FastQC:** Quality good, no contamination
3. **Alignment:** High mapping rates, consistent
4. **Quantification:** Expected library types

5. **Outliers:** Flag for exclusion or re-sequencing

### Quick Start

```
# After running FastQC, STAR, Salmon, etc.  
cd /path/to/results/  
multiqc --force --title "My Project" .  
firefox multiqc_report.html
```

That's it!