

Package ‘scBatchQC’

July 21, 2026

Type Package

Title Batch-Aware Cell Quality Control for Single-Cell RNA-seq

Version 0.99.3

Description scBatchQC provides a hierarchical empirical Bayes framework for quality control in multi-sample, multi-batch single-cell RNA-seq experiments. Unlike per-sample QC tools, scBatchQC jointly models QC metric distributions (library size, gene count, mitochondrial fraction) and doublet rates across batches, enabling calibrated cell-level QC calls that account for batch structure. The package operates natively on SingleCellExperiment objects and returns augmented colData with per-cell QC flags and batch-adjusted doublet scores.

License MIT + file LICENSE

URL <https://github.com/SubhadipJana1409/scBatchQC>

BugReports <https://github.com/SubhadipJana1409/scBatchQC/issues>

biocViews SingleCell, QualityControl, BatchEffect, StatisticalMethod, Transcriptomics, GeneExpression, CellBasedAssays, Sequencing, WorkflowStep

Encoding UTF-8

LazyData false

Depends R (>= 4.6.0)

Imports SingleCellExperiment, SummarizedExperiment, BiocParallel, scrapper, methods, stats, S4Vectors, ggplot2, rlang

Suggests scDblFinder, BiocStyle, knitr, rmarkdown, testthat (>= 3.0.0), TENxPBMCDData, withr

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

git_url <https://git.bioconductor.org/packages/scBatchQC>

git_branch devel

git_last_commit 9f8df9a

git_last_commit_date 2026-06-08

Repository Bioconductor 3.24

Date/Publication 2026-07-20

Author Subhadip Jana [aut, cre] (ORCID:
<https://orcid.org/0009-0003-7860-2853>)

Maintainer Subhadip Jana <subhadipjana1409@gmail.com>

Contents

scBatchQC-package	2
batchAwareQCMetrics	3
batchSummary	5
BQCResult	5
BQCResult-class	6
doubletScores	6
estimateBatchDoubletRate	7
harmonizeQCThresholds	9
plotBatchQC	10
qcFlags	11
show,BQCResult-method	12
Index	13

scBatchQC-package	<i>scBatchQC: Batch-Aware Cell Quality Control for scRNA-seq</i>
-------------------	--

Description

scBatchQC provides a hierarchical empirical Bayes framework for quality control (QC) in multi-sample, multi-batch single-cell RNA-sequencing (scRNA-seq) experiments.

Unlike per-sample QC tools such as `scuttle::isOutlier`, which apply a single global MAD threshold, scBatchQC jointly models QC metric distributions (library size, gene count, mitochondrial fraction) and doublet rates across batches. This prevents over-filtering of high-quality batches and under-filtering of low-quality ones — a common but underappreciated problem in multi-batch scRNA-seq workflows.

Main functions

batchAwareQCMetrics Compute per-cell QC metrics and flag outliers using batch-harmonized MAD thresholds.

estimateBatchDoubletRate Model expected doublet rates per batch as a function of cells loaded and protocol.

harmonizeQCThresholds Inspect and update harmonized thresholds at arbitrary MAD stringency.

plotBatchQC Visualize QC metric distributions per batch with threshold overlays.

Bioconductor data structures

All functions accept and return `SingleCellExperiment` objects. Results are stored as additional columns in `colData()`.

Author(s)

Maintainer: Subhadip Jana <subhadipjana1409@gmail.com> ([ORCID](#))

Authors:

- Subhadip Jana <subhadipjana1409@gmail.com> ([ORCID](#))

References

Amezquita RA et al. (2020). Orchestrating single-cell analysis with Bioconductor. *Nature Methods*, 17, 137-145.

See Also

Useful links:

- <https://github.com/SubhadipJana1409/scBatchQC>
- Report bugs at <https://github.com/SubhadipJana1409/scBatchQC/issues>

batchAwareQCMetrics *Batch-Aware QC Metric Computation*

Description

Computes per-cell quality control metrics and identifies outlier cells using a hierarchical empirical Bayes approach. Unlike `scuttle::isOutlier`, which applies a single global MAD threshold, `batchAwareQCMetrics` estimates batch-specific MAD scales and shrinks them toward a global prior, preventing over-filtering in high-quality batches and under-filtering in low-quality ones.

Usage

```
batchAwareQCMetrics(
  sce,
  batch = NULL,
  metrics = c("sum", "detected", "subsets_MT_percent"),
  nmads = 3,
  mt_pattern = "^MT-",
  shrink_strength = 0.5,
  BPPARAM = SerialParam()
)
```

Arguments

sce	A SingleCellExperiment object. Must have raw counts in assay(sce, "counts").
batch	A character(1) naming a column in <code>colData(sce)</code> that identifies batch membership. If NULL, falls back to standard per-dataset QC (equivalent to <code>scuttle::isOutlier</code>).
metrics	A character vector of QC metrics to evaluate. Supported: "sum" (library size), "detected" (genes detected), "subsets_MT_percent" (mitochondrial percentage). Default: all three.
nmads	A numeric(1) number of MADs to use as the outlier threshold. Default: 3.
mt_pattern	A character(1) regex used to identify mitochondrial genes. Default: "^MT-".

shrink_strength	A numeric(1) in $[0, 1]$ controlling how much per-batch estimates are shrunk toward the global prior. 0 = no shrinkage (pure per-batch); 1 = full pooling. Default: 0.5 (empirical Bayes midpoint).
BPPARAM	A BiocParallelParam object controlling parallelisation. Default: <code>SerialParam()</code> .

Details

For each QC metric m and batch b , the function estimates:

1. Per-batch median μ_b and MAD σ_b .
2. A shrinkage weight w_b based on batch cell count.
3. A global prior μ_0, σ_0 pooled across batches.
4. A harmonized threshold $\tau_b = \mu_b^* + nmads \times \sigma_b^*$ where μ_b^* and σ_b^* are the shrinkage estimates.

A cell is flagged as an outlier if any QC metric exceeds its batch-specific harmonized threshold.

Value

The input sce with the following additions to colData:

- `scBatchQC_sum`: library size (total UMI count).
- `scBatchQC_detected`: number of detected genes.
- `scBatchQC_subsets_MT_percent`: mitochondrial fraction.
- `scBatchQC_outlier`: logical flag; TRUE if the cell fails any QC threshold.
- `scBatchQC_outlier_reason`: character string naming which metric(s) caused the flag.

References

Lun ATL et al. (2016). A step-by-step workflow for low-level analysis of single-cell RNA sequencing data with Bioconductor. *F1000Research*, 5, 2122.

See Also

[estimateBatchDoubletRate](#), [harmonizeQCThresholds](#), [plotBatchQC](#)

Examples

```
library(SingleCellExperiment)

# Simulate a minimal SCE with two batches
set.seed(42)
counts <- matrix(rpois(2000, lambda = 5), nrow = 200, ncol = 100)
rownames(counts) <- paste0("Gene", seq_len(200))
rownames(counts)[1:10] <- paste0("MT-", seq_len(10))
colnames(counts) <- paste0("Cell", seq_len(100))

sce <- SingleCellExperiment(assays = list(counts = counts))
sce$batch <- rep(c("B1", "B2"), each = 50)

sce <- batchAwareQCMetrics(sce, batch = "batch")
table(sce$scBatchQC_outlier, sce$batch)
```

batchSummary	<i>Accessor for batch summary in a BQCResult</i>
--------------	--

Description

Returns the per-batch summary DataFrame.

Usage

```
batchSummary(x, ...)

## S4 method for signature 'BQCResult'
batchSummary(x, ...)
```

Arguments

x	A BQCResult object.
...	Additional arguments (not used).

Value

A DataFrame of per-batch statistics.

Examples

```
library(S4Vectors)
obj <- BQCResult(
  qcFlags = DataFrame(low_lib = c(FALSE, TRUE)),
  doubletScores = c(0.04, 0.06),
  batchSummary = DataFrame(batch = "B1", doublet_rate_est = 0.04)
)
batchSummary(obj)
```

BQCResult	<i>Constructor for BQCResult</i>
-----------	----------------------------------

Description

Create a new BQCResult object.

Usage

```
BQCResult(qcFlags, doubletScores, batchSummary, params = list())
```

Arguments

qcFlags	A DataFrame of per-cell QC flags.
doubletScores	A numeric vector of doublet scores.
batchSummary	A DataFrame of per-batch statistics.
params	A list of analysis parameters.

Value

A BQCResult object.

Examples

```
library(S4Vectors)
qf <- DataFrame(low_lib = c(FALSE, TRUE, FALSE))
obj <- BQCResult(
  qcFlags      = qf,
  doubletScores = c(0.04, 0.06, 0.05),
  batchSummary = DataFrame(batch = "B1", doublet_rate_est = 0.04)
)
obj
```

BQCResult-class

BQCResult: Batch-Aware QC Result Container

Description

An S4 class to store the output of batch-aware quality control applied to a SingleCellExperiment object. Slots hold per-cell QC flags, batch-level doublet rate estimates, and harmonized QC thresholds for each batch.

Slots

qcFlags A DataFrame with per-cell logical QC flags, one row per cell and one column per QC metric.

doubletScores A numeric vector of batch-adjusted doublet probability scores, one per cell.

batchSummary A DataFrame with one row per batch summarising estimated doublet rate and harmonized thresholds.

params A list storing the parameters used in the analysis (batch variable name, NMAD multiplier, etc.).

doubletScores

Accessor for doublet scores in a BQCResult

Description

Returns the per-cell doublet score vector.

Usage

```
doubletScores(x, ...)
```

```
## S4 method for signature 'BQCResult'
doubletScores(x, ...)
```

Arguments

`x` A `BQCResult` object.

`...` Additional arguments (not used).

Value

A numeric vector of doublet scores.

Examples

```
library(S4Vectors)
obj <- BQCResult(
  qcFlags      = DataFrame(low_lib = c(FALSE, TRUE)),
  doubletScores = c(0.04, 0.06),
  batchSummary = DataFrame(batch = "B1", doublet_rate_est = 0.04)
)
doubletScores(obj)
```

estimateBatchDoubletRate

Estimate Per-Batch Doublet Rates

Description

Estimates the expected doublet rate for each batch in a multi-sample `SingleCellExperiment` experiment. The doublet rate is modelled as a linear function of per-batch technical covariates (number of cells loaded, median library size, protocol type), enabling principled flagging of likely doublets across batches with heterogeneous capture efficiencies.

Usage

```
estimateBatchDoubletRate(
  sce,
  batch = NULL,
  cells_loaded = NULL,
  protocol = NULL,
  observed_doublets = NULL,
  return_sce = TRUE
)
```

Arguments

`sce` A [SingleCellExperiment](#) object.

`batch` A character(1) naming the batch column in `colData(sce)`.

`cells_loaded` A named numeric vector with the number of cells loaded per batch (key = batch label, value = cell count loaded). If `NULL`, the observed cell count per batch is used as a proxy (underestimates doublet rate).

`protocol` A named character vector mapping batch labels to protocol type (e.g. "10x_v3", "10x_v2", "inDrop"). Used to set the baseline k constant. Default: `NULL` (all batches assumed 10x v3).

observed_doublets	A character(1) naming a column in colData(sce) that contains externally computed doublet calls (TRUE/FALSE). When supplied, the model is calibrated against observed rates. Default: NULL.
return_sce	Logical. If TRUE (default), returns the input sce with scBatchQC_doublet_rate added to colData. If FALSE, returns a DataFrame of batch-level estimates.

Details

Doublet rates in droplet-based scRNA-seq follow approximately:

$$r_b \approx k \times N_b$$

where N_b is the number of cells loaded per batch and $k \approx 8 \times 10^{-6}$ for 10x Genomics Chromium.

estimateBatchDoubletRate allows k to vary by batch covariates (e.g. protocol, operator) by fitting a linear model on the log-transformed per-batch cell count. When external doublet simulations are not desired, this gives a lightweight alternative to full simulation-based tools like scDblFinder.

Optionally, if the user supplies observed doublet calls from an external tool in colData, the function will calibrate the rate model against those observations.

Value

If return_sce = TRUE: the input sce with a scBatchQC_doublet_rate column in colData giving the estimated doublet probability for each cell's batch. If return_sce = FALSE: a DataFrame with one row per batch and columns batch, n_cells_obs, cells_loaded, doublet_rate_est, protocol.

See Also

[batchAwareQCMetrics](#), [plotBatchQC](#)

Examples

```
library(SingleCellExperiment)

set.seed(42)
counts <- matrix(rpois(2000, lambda = 5), nrow = 200, ncol = 100)
rownames(counts) <- paste0("Gene", seq_len(200))
colnames(counts) <- paste0("Cell", seq_len(100))

sce <- SingleCellExperiment(assays = list(counts = counts))
sce$batch <- rep(c("B1", "B2"), each = 50)

cells_loaded <- c(B1 = 5000, B2 = 8000)
sce <- estimateBatchDoubletRate(sce,
  batch = "batch",
  cells_loaded = cells_loaded
)
sce$scBatchQC_doublet_rate
```

 harmonizeQCThresholds *Harmonize QC Thresholds Across Batches*

Description

Given a [SingleCellExperiment](#) that has already been processed by [batchAwareQCMetrics](#), `harmonizeQCThresholds` returns the per-batch QC threshold table and optionally updates `colData` with revised flags at a user-specified stringency.

This is useful for interactive threshold exploration or for downstream reporting: instead of re-running the full QC pipeline, the user can sweep `nmads` and inspect how the number of flagged cells changes per batch.

Usage

```
harmonizeQCThresholds(
  sce,
  batch = NULL,
  nmads = 3,
  shrink_strength = 0.5,
  update_sce = FALSE
)
```

Arguments

<code>sce</code>	A SingleCellExperiment processed by batchAwareQCMetrics .
<code>batch</code>	A <code>character(1)</code> naming the batch column.
<code>nmads</code>	A <code>numeric(1)</code> MAD multiplier. Default: 3.
<code>shrink_strength</code>	A <code>numeric(1)</code> in $[0, 1]$. Default: 0.5.
<code>update_sce</code>	Logical. If TRUE, rewrites the <code>scBatchQC_outlier</code> and <code>scBatchQC_outlier_reason</code> columns in <code>colData(sce)</code> using the new thresholds. Default: FALSE.

Value

A list with components:

`thresholds` A named list of per-metric `data.frames` (rows = batches, columns = lower and upper).

`n_flagged` A `DataFrame` with one row per batch and one column per metric showing the number of cells that would be flagged at these thresholds.

`sce` The (possibly updated) `sce`, returned only when `update_sce = TRUE`.

See Also

[batchAwareQCMetrics](#), [estimateBatchDoubletRate](#), [plotBatchQC](#)

Examples

```
library(SingleCellExperiment)

set.seed(42)
counts <- matrix(rpois(2000, lambda = 5), nrow = 200, ncol = 100)
rownames(counts) <- paste0("Gene", seq_len(200))
rownames(counts)[1:10] <- paste0("MT-", seq_len(10))
colnames(counts) <- paste0("Cell", seq_len(100))

sce <- SingleCellExperiment(assays = list(counts = counts))
sce$batch <- rep(c("B1", "B2"), each = 50)
sce <- batchAwareQCMetrics(sce, batch = "batch")

# Explore with 2.5 MADs instead of default 3
result <- harmonizeQCThresholds(sce, batch = "batch", nmads = 2.5)
result$n_flagged
```

plotBatchQC

Visualize QC Metric Distributions Across Batches

Description

Produces a panel of violin plots showing per-batch distributions of QC metrics, with harmonized threshold lines overlaid. Useful for inspecting whether `batchAwareQCMetrics` thresholds are sensible and for comparing batch quality visually.

Usage

```
plotBatchQC(
  sce,
  batch = NULL,
  metrics = NULL,
  show_thresholds = TRUE,
  nmads = 3,
  colour_by = "scBatchQC_outlier",
  point_size = 0.4,
  point_alpha = 0.4
)
```

Arguments

<code>sce</code>	A <code>SingleCellExperiment</code> processed by <code>batchAwareQCMetrics</code> .
<code>batch</code>	A character(1) naming the batch column in <code>colData(sce)</code> .
<code>metrics</code>	A character vector of QC metric names to plot (without the "scBatchQC_" prefix). Default: all available.
<code>show_thresholds</code>	Logical. If TRUE, overlays the batch-specific harmonized upper thresholds as dashed horizontal lines. Default: TRUE.
<code>nmads</code>	Passed to <code>harmonizeQCThresholds</code> to recompute thresholds for display. Default: 3.

colour_by	A character(1) naming a colData column to colour cells by (e.g. "scBatchQC_outlier"). Default: "scBatchQC_outlier".
point_size	Numeric. Jitter point size. Default: 0.4.
point_alpha	Numeric. Jitter point alpha. Default: 0.4.

Value

A ggplot2 object. Can be modified with standard ggplot2 functions or saved with ggsave().

See Also

[batchAwareQCMetrics](#), [harmonizeQCThresholds](#)

Examples

```
library(SingleCellExperiment)

set.seed(42)
counts <- matrix(rpois(2000, lambda = 5), nrow = 200, ncol = 100)
rownames(counts) <- paste0("Gene", seq_len(200))
rownames(counts)[1:10] <- paste0("MT-", seq_len(10))
colnames(counts) <- paste0("Cell", seq_len(100))

sce <- SingleCellExperiment(assays = list(counts = counts))
sce$batch <- rep(c("B1", "B2"), each = 50)
sce <- batchAwareQCMetrics(sce, batch = "batch")

plotBatchQC(sce, batch = "batch")
```

qcFlags

Accessor for QC flags in a BQCResult

Description

Returns the per-cell QC flag DataFrame.

Usage

```
qcFlags(x, ...)

## S4 method for signature 'BQCResult'
qcFlags(x, ...)
```

Arguments

x	A BQCResult object.
...	Additional arguments (not used).

Value

A DataFrame of per-cell logical QC flags.

Examples

```
library(S4Vectors)
obj <- BQCRresult(
  qcFlags      = DataFrame(low_lib = c(FALSE, TRUE)),
  doubletScores = c(0.04, 0.06),
  batchSummary = DataFrame(batch = "B1", doublet_rate_est = 0.04)
)
qcFlags(obj)
```

show,BQCRresult-method *Show method for BQCRresult*

Description

Prints a compact summary of a BQCRresult object.

Usage

```
## S4 method for signature 'BQCRresult'
show(object)
```

Arguments

object A BQCRresult object.

Value

Invisibly returns object.

Index

batchAwareQCMetrics, [2](#), [3](#), [8–11](#)
batchSummary, [5](#)
batchSummary, BQCResult-method
 (batchSummary), [5](#)
BiocParallelParam, [4](#)
BQCResult, [5](#)
BQCResult-class, [6](#)

doubletScores, [6](#)
doubletScores, BQCResult-method
 (doubletScores), [6](#)

estimateBatchDoubletRate, [2](#), [4](#), [7](#), [9](#)

harmonizeQCThresholds, [2](#), [4](#), [9](#), [10](#), [11](#)

plotBatchQC, [2](#), [4](#), [8](#), [9](#), [10](#)

qcFlags, [11](#)
qcFlags, BQCResult-method (qcFlags), [11](#)

scBatchQC (scBatchQC-package), [2](#)
scBatchQC-package, [2](#)
show, BQCResult-method, [12](#)
SingleCellExperiment, [2](#), [3](#), [7](#), [9](#)