Package 'miQC'

November 12, 2026

Type Package

Title Flexible, probabilistic metrics for quality control of scRNA-seq data

Version 1.20.0

Description Single-cell RNA-sequencing (scRNA-seq) has made it possible to profile gene expression in tissues at high resolution. An important preprocessing step prior to performing downstream analyses is to identify and remove cells with poor or degraded sample quality using quality control (QC) metrics. Two widely used QC metrics to identify a 'low-quality' cell are (i) if the cell includes a high proportion of reads that map to mitochondrial DNA encoded genes (mtDNA) and (ii) if a small number of genes are detected. miQC is data-driven QC metric that jointly models both the proportion of reads mapping to mtDNA and the number of detected genes with mixture models in a probabilistic framework to predict the low-quality cells in a given dataset.

URL https://github.com/greenelab/miQC

BugReports https://github.com/greenelab/miQC/issues

License BSD_3_clause + file LICENSE

Imports SingleCellExperiment, flexmix, ggplot2, splines

Suggests scRNAseq, scater, BiocStyle, knitr, rmarkdown

biocViews SingleCell, QualityControl, GeneExpression, Preprocessing, Sequencing

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 7.2.1

LazyData TRUE

git_url https://git.bioconductor.org/packages/miQC

git_branch RELEASE_3_23

git_last_commit cec28ae

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Ariel Hippen [aut, cre],
Stephanie Hicks [aut]

Maintainer Ariel Hippen <ariel.hippen@gmail.com>

Contents

filterCells 2
get1DCutoff 3
metrics 4
mixtureModel 5
plotFiltering 6
plotMetrics 7
plotModel 8

Index **10**

`filterCells` *filterCells*

Description

Find those cells probabilistically determined to be compromised by the mixture model and remove them from the dataset.

Usage

```
filterCells(
sce,
model = NULL,
posterior_cutoff = 0.75,
keep_all_below_boundary = TRUE,
enforce_left_cutoff = TRUE,
verbose = TRUE
)
```

Arguments

`sce`	(SingleCellExperiment) Input data object.
`model`	(flexmix) Output of mixtureModel function, which should be explicitly called first to ensure stability of model parameters. Default = NULL.
`posterior_cutoff`	(numeric) The posterior probability of a cell being part of the compromised distribution, a number between 0 and 1. Any cells below the appointed cutoff will be marked to keep. Default = 0.75
`keep_all_below_boundary`	(boolean) Ensures that no cells below the intact cell distribution are removed. This should almost always be set to true. Default = TRUE

enforce_left_cutoff
: (boolean) Prevents a U-shape in the filtering plot. Identifies the cell with the lowest mitochondrial fraction that is set to be discarded, it ensures that no cells with lower library complexity (further left) and higher mitochondrial percentage (further up) than it are kept. Default = TRUE

verbose
: (boolean) Whether to report how many cells (columns) are being removed from the SingleCellExperiment object. Default = TRUE

Value

Returns a SingleCellExperiment object, the same as the input except with a new column in colData, prob_compromised, and all cells with greater than the set posterior probability removed from the dataset.

Examples

```
library(scRNAseq)
library(scater)
sce <- ZeiselBrainData()
mt_genes <- grepl("^mt-",  rownames(sce))
feature_ctrls <- list(mito = rownames(sce)[mt_genes])
sce <- addPerCellQC(sce, subsets = feature_ctrls)
model <- mixtureModel(sce)
sce <- filterCells(sce, model)
```

---

get1DCutoff *get1DCutoff*

Description

When the model is generated based only on mitochondrial percentage, e.g. run_model(sce, model_type = "one_dimensional"), there will be a discrete cutoff point at which to remove cells with at least that mitochondrial percentage. This function identifies this cutoff based on a given posterior probability threshold. This number can then be passed as a cutoff to other modalities.

Usage

```
get1DCutoff(
  sce,
  model = NULL,
  posterior_cutoff = 0.75,
  subsets_mito_percent = "subsets_mito_percent"
)
```

Arguments

sce	(SingleCellExperiment) Input data object.
model	(flexmix) Output of mixtureModel function, which should be explicitly called first to ensure stability of model parameters. Default = NULL.
posterior_cutoff	(numeric) The posterior probability of a cell being part of the compromised distribution, a number between 0 and 1. Any cells below the appointed cutoff will be marked to keep. Default = 0.75
subsets_mito_percent	(character) Column name in sce giving the percent of reads mapping to mitochondrial genes. This name is inherited from scater's addPerCellQC() function, provided the subset "mito" with names of all mitochondrial genes is passed in. See examples for details.

Value

Returns a single numeric value, the percent mitochondrial cutoff at which to filter cells.

Examples

```
library(scRNAseq)
library(scater)
sce <- ZeiselBrainData()
mt_genes <- grepl("^mt-",  rownames(sce))
feature_ctrls <- list(mito = rownames(sce)[mt_genes])
sce <- addPerCellQC(sce, subsets = feature_ctrls)
model <- mixtureModel(sce, model_type = "one_dimensional")
get1DCutoff(sce, model)
```

---

metrics *Basic scRNA-seq QC metrics from an ovarian tumor*

Description

Basic QC metrics from a high-grade serous ovarian cancer (HGSOC) sample. The information included is the minimum needed to generate a miQC model and make plots. Count data for the full tumor is available through GEO, accession number GSM4816047.

Usage

```
metrics
```

Source

GEO Accession GSM4816047

mixtureModel *mixtureModel*

Description

Function to fit a two-distribution mixture model on a SingleCellExperiment object.

Usage

```
mixtureModel(
sce,
model_type = "linear",
detected = "detected",
subsets_mito_percent = "subsets_mito_percent"
)
```

Arguments

sce	(SingleCellExperiment) Input data object.
model_type	(character) What type of model to generate. A linear mixture model ("linear") based on mitochondrial percentage and library complexity is recommended. B-spline ("spline") and two-degree polynomial ("polynomial") models are also supported. For a simpler model, a one-dimensional gaussian mixture model ("one_dimensional") based on mitochondrial percentage only is available. Default = "linear".
detected	(character) Column name in sce giving the number of unique genes detected per cell. This name is inherited by default from scater's addPerCellQC() function.
subsets_mito_percent	(character) Column name in sce giving the percent of reads mapping to mitochondrial genes. This name is inherited from scater's addPerCellQC() function, provided the subset "mito" with names of all mitochondrial genes is passed in. See examples for details.

Value

Returns a flexmix object with mixture model parameters, which is used to calculate posterior probability for each cell being compromised and make final filtering decisions.

Examples

```
library(scRNAseq)
library(scater)
sce <- ZeiselBrainData()
mt_genes <- grepl("^mt-",  rownames(sce))
feature_ctrls <- list(mito = rownames(sce)[mt_genes])
sce <- addPerCellQC(sce, subsets = feature_ctrls)
model <- mixtureModel(sce)
```

plotFiltering *plotFiltering*

Description

Function to plot which cells will be kept and removed given their posterior probability of belonging to the compromised distribution.

Usage

```
plotFiltering(
sce,
model = NULL,
posterior_cutoff = 0.75,
keep_all_below_boundary = TRUE,
enforce_left_cutoff = TRUE,
palette = c("#999999", "#E69F00"),
detected = "detected",
subsets_mito_percent = "subsets_mito_percent"
)
```

Arguments

sce	(SingleCellExperiment) Input data object.
model	(flexmix) Output of mixtureModel function, which should be explicitly called first to ensure stability of model parameters. Default = NULL.
posterior_cutoff	(numeric) The posterior probability of a cell being part of the compromised distribution, a number between 0 and 1. Any cells below the appointed cutoff will be marked to keep. Default = 0.75
keep_all_below_boundary	(boolean) Ensures that no cells below the intact cell distribution are removed. This should almost always be set to true. Default = TRUE
enforce_left_cutoff	(boolean) Prevents a U-shape in the filtering plot. Identifies the cell with the lowest mitochondrial fraction that is set to be discarded, it ensures that no cells with lower library complexity (further left) and higher mitochondrial percentage (further up) than it are kept. Default = TRUE
palette	(character) Color palette. A vector of length two containing custom colors. Default = c("#999999", "#E69F00").
detected	(character) Column name in sce giving the number of unique genes detected per cell. This name is inherited by default from scater's addPerCellQC() function.
subsets_mito_percent	(character) Column name in sce giving the percent of reads mapping to mitochondrial genes. This name is inherited from scater's addPerCellQC() function, provided the subset "mito" with names of all mitochondrial genes is passed in. See examples for details.

Value

Returns a ggplot object. Additional plot elements can be added as ggplot elements (e.g. title, customized formatting, etc).

Examples

```
library(scRNAseq)
library(scater)
sce <- ZeiselBrainData()
mt_genes <- grepl("^mt-",  rownames(sce))
feature_ctrls <- list(mito = rownames(sce)[mt_genes])
sce <- addPerCellQC(sce, subsets = feature_ctrls)
model <- mixtureModel(sce)
plotFiltering(sce, model)
```

---

plotMetrics *plotMetrics*

Description

A function to plot the QC parameters used for a miQC model, number of unique genes expressed and percent mitochondrial reads. This function can be run before calling mixtureModel() to assess if miQC is appropriate given the data distribution. See vignette for examples of cases where miQC is and isn't a good choice for filtering.

Usage

```
plotMetrics(
sce,
detected = "detected",
subsets_mito_percent = "subsets_mito_percent",
palette = "#33ADFF"
)
```

Arguments

sce	(SingleCellExperiment) Input data object.
detected	(character) Column name in sce giving the number of unique genes detected per cell. This name is inherited by default from scater's addPerCellQC() function.
subsets_mito_percent	(character) Column name in sce giving the percent of reads mapping to mitochondrial genes. This name is inherited from scater's addPerCellQC() function, provided the subset "mito" with names of all mitochondrial genes is passed in. See examples for details.
palette	(character) Specifies the color to plot cells as. Default is "#33ADFF".

Value

Returns a ggplot object. Additional plot elements can be added as ggplot elements (e.g. title, customized formatting, etc).

Examples

```
library(scRNAseq)
library(scater)
sce <- ZeiselBrainData()
mt_genes <- grepl("^mt-",  rownames(sce))
feature_ctrls <- list(mito = rownames(sce)[mt_genes])
sce <- addPerCellQC(sce, subsets = feature_ctrls)
plotMetrics(sce)
```

`plotModel` *plotModel*

Description

Function to plot quality characteristics of cells in dataset, parameters of compromised and intact distributions, and posterior probability of each cell belonging to the compromised distribution.

Usage

```
plotModel(
sce,
model = NULL,
detected = "detected",
subsets_mito_percent = "subsets_mito_percent"
)
```

Arguments

`sce`	(SingleCellExperiment) Input data object.
`model`	(flexmix) Output of mixtureModel function, which should be explicitly called first to ensure stability of model parameters. Default = NULL.
`detected`	(character) Column name in sce giving the number of unique genes detected per cell. This name is inherited by default from scater's addPerCellQC() function.
`subsets_mito_percent`	(character) Column name in sce giving the percent of reads mapping to mitochondrial genes. This name is inherited from scater's addPerCellQC() function, provided the subset "mito" with names of all mitochondrial genes is passed in. See examples for details.

Value

Returns a ggplot object. Additional plot elements can be added as ggplot elements (e.g. title, customized formatting, etc).

Examples

```
library(scRNAseq)
library(scater)
sce <- ZeiselBrainData()
mt_genes <- grepl("^mt-",  rownames(sce))
feature_ctrls <- list(mito = rownames(sce)[mt_genes])
sce <- addPerCellQC(sce, subsets = feature_ctrls)
model <- mixtureModel(sce)
plotModel(sce, model)
```

Index

∗ **datasets**
    metrics, 4

filterCells, 2

get1DCutoff, 3

metrics, 4
mixtureModel, 5

plotFiltering, 6
plotMetrics, 7
plotModel, 8