Package: SeqGate (via r-universe)

October 31, 2026

Type Package

Title Filtering of Lowly Expressed Features

Version 1.23.0

Date 2021-01-22

Description Filtering of lowly expressed features (e.g. genes) is a common step before performing statistical analysis, but an arbitrary threshold is generally chosen. SeqGate implements a method that rationalize this step by the analysis of the distibution of counts in replicate samples. The gate is the threshold above which sequenced features can be considered as confidently quantified.

Encoding UTF-8

Imports stats, methods, BiocManager

Depends S4Vectors, SummarizedExperiment, GenomicRanges

Suggests testthat (>= 3.0.0), edgeR, BiocStyle, knitr, rmarkdown

License GPL (>= 2.0)

LazyData no

biocViews DifferentialExpression, GeneExpression, Transcriptomics, Sequencing, RNASeq

BugReports https://github.com/srialle/SeqGate/issues

VignetteBuilder knitr

Config/testthat/edition 3

Config/pak/sysreqs zlib1g-dev

Repository https://bioc.r-universe.dev

Date/Publication 2026-04-28 12:53:47 UTC

RemoteUrl https://github.com/bioc/SeqGate

RemoteRef HEAD

RemoteSha 4449ab3582234dba01dd8bcb69faee73364fdaa1

Contents

SeqGate-package 2
applySeqGate 3
data_MiTF_1000genes 4

Index **6**

`SeqGate-package` *Filtering of Lowly Expressed Features*

Description

SeqGate is a method to filter lowly expressed features (e.g. genes).

Details

From a matrix of counts where lines correspond to features and columns to biological samples, provided as a SummarizedExperiment object, a threshold is computed and applied in order to filter lowly expressed features. The threshold is computed based on the distribution of counts measured along with zeros within replicates of the same condition. The objective of SeqGate is to rationalize the filtering step by using the information of replicate samples. The computed threshold corresponds to the count value below which we can not be sure that the count can be considered different from zero.
The filtering is made by calling the `applySeqGate()` function.

Author(s)

Christelle Reynès <christelle.reynes@igf.cnrs.fr>,
Stéphanie Rialle <stephanie.rialle@mgx.cnrs.fr>,
Maintainer: Stéphanie Rialle <stephanie.rialle@mgx.cnrs.fr>

References

Rialle, R. et al. (2020): SeqGate: a bioconductor package to perform data-driven filtering of RNAseq datasets *manuscript in preparation*

Examples

```
# Loading of input data frame
data(data_MiTF_1000genes)
# Annotating conditions
cond<-c("A","A","B","B","A","B")
# Setting the SummarizedExperiment input
rowData <- DataFrame(row.names=rownames(data_MiTF_1000genes))
colData <- DataFrame(Conditions=cond)
counts_strub <- SummarizedExperiment(
                    assays=list(counts=data_MiTF_1000genes),
                    rowData=rowData,
                    colData=colData)
```

```
# Applying SeqGate
counts_strub <- applySeqGate(counts_strub,"counts","Conditions")
# Getting the matrix of kept genes after filtering
keptGenes <- assay(counts_strub[rowData(counts_strub)$onFilter == TRUE,])
```

`applySeqGate` *Performs a Filtering of Lowly Expressed Features*

Description

Implements the SeqGate methods to filter lowly expressed features (e.g. genes)

Usage

```
applySeqGate(readCountsSE, assayName, colCond, prop0=NULL, percentile=NULL,
             propUpThresh=0.9)
```

Arguments

`readCountsSE`	`SummarizedExperiment` object including the `assayName` assay, that is a matrix of read counts. Rows correspond to features (genes or other genomic features) and columns correspond to sample libraries. The read counts must contain zeros, as the SeqGate method is based on the distribution of counts in replicates along with zero counts.
`assayName`	character string giving the name of the assay in `readCountsSE` for which we want to perform the filtering.
`colCond`	character string giving the name of the column describing the biological conditions to which the samples belong to, in the colData DataFrame of `readCountsSE`.
`prop0`	minimal proportion of zeros among a condition to consider that the feature is not or lowly expressed. `prop0` must be > 0 and < 1. See details for default setting.
`percentile`	percentile used on the 'max' distribution to determine the filtering threshold value. `percentile` must be >= 0 and <=1. See details for default setting.
`propUpThresh`	proportion of counts to be above the threshold in at least one condition to keep the feature. `propUpThresh` must be > 0 and <= 1. See details for default setting.

Details

In order to find a theshold value to filter lowly expressed features, SeqGate analyzes the distribution of counts found in replicates along with zero counts. More specifically, features with a proportion of at least `prop0` zeros in one condition are selected. The distribution of counts found in replicates of that same condition along with those zeros is computed. The chosen threshold is the count value corresponding to the `percentile` of this distribution. Finally, features having a proportion `propUpThresh` of replicates with counts below that value in all conditions are filtered.

If `prop0` is not provided, it is set to the number of replicates minus one divided by the max total number of replicates. For example, if one of the condition of the experiment counts 2 replicates and the other condition 4 replicates, the proportion `prop0` will be set to (4-1)/4=0.75.

If percentile is not provided, it is set to 0.90 if at least one of the conditions have 5 replicates or less. Otherwise, it is set to a value comprised between 0.5 and 0.9, depending on the number or replicates.

Value

The input SummarizedExperiment with the following added elements:

`rowData(readCountsSE)$onFilter`
: vector which length = nrow(readCountsSE) indicating if features are kept after filtering (value = TRUE) or not (value = FALSE)

`metadata(readCountsSE)$threshold`
: applied filter threshold.

Author(s)

Christelle Reynès <christelle.reynes@igf.cnrs.fr>,
Stéphanie Rialle <stephanie.rialle@mgx.cnrs.fr>

References

Rialle, R. et al. (2020): SeqGate: a bioconductor package to perform data-driven filtering of RNAseq datasets (manuscript in preparation)

Examples

```
# Loading of input data frame
data(data_MiTF_1000genes)
# Annotating conditions
cond<-c("A","A","B","B","A","B")
# Setting the SummarizedExperiment input
rowData <- DataFrame(row.names=rownames(data_MiTF_1000genes))
colData <- DataFrame(Conditions=cond)
counts_strub <- SummarizedExperiment(
                    assays=list(counts=data_MiTF_1000genes),
                    rowData=rowData,
                    colData=colData)
# Applying SeqGate
counts_strub <- applySeqGate(counts_strub,"counts","Conditions")
# Getting the matrix of kept genes after filtering
keptGenes <- assay(counts_strub[rowData(counts_strub)$onFilter == TRUE,])
```

---

data_MiTF_1000genes *Extract of a Transciptome Dataset from Human Melanoma Cell Line*

Description

This data set is an extract from a dataset from human transciptome studied by Strub et al. (2011).

Usage

```
data(data_MiTF_1000genes)
```

Format

A data frame with 1,000 raw gene counts on the following 6 samples.

`A1` a numeric vector: counts from sample A1 (condition A)

`A2` a numeric vector: counts from sample A2 (condition A)

`B1` a numeric vector: counts from sample B1 (condition B)

`B2` a numeric vector: counts from sample B2 (condition B)

`A3` a numeric vector: counts from sample A3 (condition A)

`B3` a numeric vector: counts from sample B3 (condition B)

Details

The extract counts 1,000 genes measured in 6 samples, from 2 biological conditions (cells expressing the Microphtalmia Transcription Factor (MiTF) are compared to cells in which the MiTF is repressed). The raw read counts have been retrieved from the Supplementary Materials of Dillies et al. (2012). The initial dataset counts 37,720 genes.

Source

Dillies, M.A. et al. (2012) A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis. Brief. Bioinformatics 2013 Nov;14(6):671-83

References

Strub, T., Giuliano, S., Ye T,, et al. Essential role of microphthalmia transcription factor for DNA replication, mitosis and genomic stability in melanoma. Oncogene 2011;30:2319–32

Examples

```
data(data_MiTF_1000genes)
```

Index

applySeqGate, 3

data_MITF_1000genes
        *(*data_MiTF_1000genes*)*, 4
data_MiTF_1000genes, 4
data_mitf_1000genes
        *(*data_MiTF_1000genes*)*, 4

SeqGate *(*SeqGate-package*)*, 2
seqGate *(*applySeqGate*)*, 3
seqgate *(*applySeqGate*)*, 3
SeqGate-package, 2