Package: omicslog (via r-universe)

November 18, 2026

Title Logging for High-Throughput Omics Data Analysis

Version 0.99.0

Date 2026-11-16

Description Provides logging capabilities for SingleCellExperiment and SummarizedExperiment objects. This package extends these classes to include tracking of operations such as filtering and column/row-level modifications. The log is displayed when the object is printed, helping users keep track of the data transformations applied during analysis. This is particularly useful for complex analysis workflows where multiple filtering and transformation steps are applied.

License GPL-3

Encoding UTF-8

URL https://github.com/tidyomics/omicslog

BugReports https://github.com/tidyomics/omicslog/issues

Roxygen list(markdown = TRUE)

biocViews Infrastructure, DataRepresentation, Software, SingleCell, Transcriptomics

Imports methods, dplyr, tidyr, tibble, rlang, S4Vectors, BiocGenerics, utils

Suggests airway, tidySummarizedExperiment, tidySingleCellExperiment, knitr, rmarkdown, BiocStyle, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Depends R (>= 4.5.0), SummarizedExperiment, SingleCellExperiment

Config/roxygen2/version 8.1.0

Config/pak/sysreqs libicu-dev zlib1g-dev

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

Date/Publication 2026-11-17 10:31:36 UTC

RemoteUrl https://github.com/BiocStaging/omicslog

RemoteRef HEAD

RemoteSha c26fdfb6665736048ff0bf9ed5f4ba514f0cbdd2

Contents

[,SummarizedExperimentLogged,ANY,ANY,ANY-method . . . 2
$<-,SummarizedExperimentLogged-method . . . 3
assays<-,SummarizedExperimentLogged,SimpleList-method . . . 4
colData<-,ExperimentLogged,DataFrame-method . . . 5
ExperimentLogged-class . . . 5
extract.SummarizedExperimentLogged . . . 6
filter.SummarizedExperimentLogged . . . 7
log_start . . . 8
mutate.SummarizedExperimentLogged . . . 9
reducedDim<-,SingleCellExperimentLogged,character-method . . . 9
reducedDims<-,SingleCellExperimentLogged-method . . . 10
rowData<-,ExperimentLogged,DataFrame-method . . . 11
select.SummarizedExperimentLogged . . . 12
SingleCellExperimentLogged-class . . . 13
slice.SummarizedExperimentLogged . . . 13
SummarizedExperimentLogged-class . . . 14

Index **15**

`[,SummarizedExperimentLogged,ANY,ANY,ANY-method`
Subset a ExperimentLogged object

Description

Subset a ExperimentLogged object

Usage

```
## S4 method for signature 'SummarizedExperimentLogged,ANY,ANY,ANY'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'SingleCellExperimentLogged,ANY,ANY,ANY'
x[i, j, ..., drop = TRUE]
```

Arguments

`x`	A ExperimentLogged object
`i, j, ...`	Indices for subsetting
`drop`	Passed to the `[` method of the parent class

Value

The subset ExperimentLogged object, with removed features and observations added to its log.

Examples

```
# Create a logged SummarizedExperiment
if (requireNamespace("tidySummarizedExperiment", quietly = TRUE)) {
  se <- tidySummarizedExperiment::pasilla
  se_logged <- log_start(se)

  se_logged <- se_logged[1:10,1:5]
  se_logged
}
```

$<-,SummarizedExperimentLogged-method

Assign column to colData using `$<-` *for ExperimentLogged*

Description

Assign column to colData using `$<-` for ExperimentLogged

Usage

```
## S4 replacement method for signature 'SummarizedExperimentLogged'
x$name <- value

## S4 replacement method for signature 'SingleCellExperimentLogged'
x$name <- value
```

Arguments

`x`	An ExperimentLogged object
`name`	Name of the `colData` column to add or modify
`value`	Values for the column

Value

The modified ExperimentLogged object, with the change added to its log.

Examples

```
# Create a logged SummarizedExperiment
if (requireNamespace("tidySummarizedExperiment", quietly = TRUE)) {
  se <- tidySummarizedExperiment::pasilla
  se_logged <- log_start(se)

  # Modify existing column
```

```
  colData(se_logged)$condition <- tolower(colData(se_logged)$condition)

  # Add new column
  colData(se_logged)$new_column <- rep("test", ncol(se_logged))
  se_logged
}
```

`assays<-,SummarizedExperimentLogged,SimpleList-method`
Assign List to assays using `assays<-` *for ExperimentLogged*

Description

Assign List to assays using `assays<-` for ExperimentLogged

Usage

```
## S4 replacement method for signature 'SummarizedExperimentLogged,SimpleList'
assays(x, withDimnames = TRUE, ...) <- value

## S4 replacement method for signature 'SingleCellExperimentLogged,SimpleList'
assays(x, withDimnames = TRUE, ...) <- value
```

Arguments

`x`	An ExperimentLogged object
`withDimnames, ...`	Passed to the `assays<-` method of the parent class
`value`	A SimpleList of assays

Value

The modified ExperimentLogged object, with added and modified assays added to its log.

Examples

```
# Create a logged SingleCellExperiment
if (requireNamespace("tidySingleCellExperiment", quietly = TRUE)) {
  data(pbmc_small, package="tidySingleCellExperiment")
  sce_logged <- log_start(pbmc_small)

  assays(sce_logged)$log2.counts <- log2(assays(sce_logged)$counts)
  sce_logged
}
```

colData<-,ExperimentLogged,DataFrame-method

Assign column to colData using `colData<-` *for ExperimentLogged*

Description

Assign column to colData using `colData<-` for ExperimentLogged

Usage

```
## S4 replacement method for signature 'ExperimentLogged,DataFrame'
colData(x) <- value

## S4 replacement method for signature 'ExperimentLogged,DFrame'
colData(x) <- value
```

Arguments

`x`	An ExperimentLogged object
`value`	A DataFrame with the new column data

Value

The modified ExperimentLogged object, with added and modified `colData` columns added to its log.

Examples

```
# Create a logged SingleCellExperiment
if (requireNamespace("tidySingleCellExperiment", quietly = TRUE)) {
  data(pbmc_small, package="tidySingleCellExperiment")
  sce_logged <- log_start(pbmc_small)

  colData(sce_logged)$num.idents <- as.numeric(colData(sce_logged)$letter.idents)
  sce_logged
}
```

ExperimentLogged-class

ExperimentLogged class

Description

A class union of SummarizedExperimentLogged and SingleCellExperimentLogged, used to define logging methods once for both classes.

Examples

```
counts <- matrix(rpois(20, 10), nrow = 5)
se_logged <- log_start(SummarizedExperiment(assays = list(counts = counts)))
sce_logged <- log_start(SingleCellExperiment(assays = list(counts = counts)))
is(se_logged, "ExperimentLogged")
is(sce_logged, "ExperimentLogged")
```

---

extract.SummarizedExperimentLogged

Extract values from a column into multiple columns in An Experiment-Logged object

---

Description

`extract()` is re-exported from **tidyr**, so it is available after `library(omicslog)`. These methods log the operation on ExperimentLogged objects; see `tidyr::extract()` for the generic.

Usage

```
## S3 method for class 'SummarizedExperimentLogged'
extract(
  data,
  col,
  into,
  regex = "([[:alnum:]]+)",
  remove = TRUE,
  convert = FALSE,
  ...
)

## S3 method for class 'SingleCellExperimentLogged'
extract(
  data,
  col,
  into,
  regex = "([[:alnum:]]+)",
  remove = TRUE,
  convert = FALSE,
  ...
)
```

Arguments

`data`	An ExperimentLogged object
`col`	Column to extract from
`into`	Names of new variables to create

regex	A regular expression to extract values
remove	If TRUE, remove input column from output
convert	If TRUE, runs type.convert() on each new column
...	Additional arguments passed to tidyr::extract

Value

An ExperimentLogged object with the extracted colData columns, with the extraction added to its log.

Examples

```
if (requireNamespace("tidySummarizedExperiment", quietly = TRUE)) {
  se <- tidySummarizedExperiment::pasilla
  se_logged <- log_start(se)
  result <- se_logged |>
    mutate(type = sub("_end","",colData(se)$type)) |>
    extract(type, into = "exp")
  result
}
```

---

filter.SummarizedExperimentLogged

Filter rows and columns of An ExperimentLogged object

Description

filter() is re-exported from **dplyr**, so it is available after library(omicslog). These methods log the operation on ExperimentLogged objects; see dplyr::filter() for the generic.

Usage

```
## S3 method for class 'SummarizedExperimentLogged'
filter(.data, ...)

## S3 method for class 'SingleCellExperimentLogged'
filter(.data, ...)
```

Arguments

.data	An ExperimentLogged object
...	Logical expressions used for filtering

Value

An ExperimentLogged object with the features and observations that match the conditions, with the number removed added to its log.

Examples

```
if (requireNamespace("tidySummarizedExperiment", quietly = TRUE)) {
se <- tidySummarizedExperiment::pasilla
  se_logged <- log_start(se)
  result <- se_logged |>
    filter(condition == "treated")
  result
}
```

`log_start` *Create a logging-enabled SummarizedExperiment object*

Description

This function wraps a SummarizedExperiment object with logging capabilities. Operations performed on the resulting object will be tracked and displayed when the object is printed.

Usage

```
log_start(se)

## S4 method for signature 'SummarizedExperimentLogged'
show(object)

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

Arguments

`se`	A SummarizedExperiment or derived object
`object`	An ExperimentLogged object

Value

`log_start()` returns an ExperimentLogged object (SummarizedExperimentLogged or SingleCellExperimentLogged) with an empty log. `show()` prints the object followed by its operation log and returns `NULL` invisibly.

Examples

```
if (requireNamespace("tidySummarizedExperiment", quietly = TRUE)) {
se <- tidySummarizedExperiment::pasilla
  se_logged <- log_start(se)
  result <- se_logged |>
    filter(condition == "treated")
  result
}
```

mutate.SummarizedExperimentLogged

Modify columns of An ExperimentLogged object

Description

`mutate()` is re-exported from **dplyr**, so it is available after `library(omicslog)`. These methods log the operation on ExperimentLogged objects; see `dplyr::mutate()` for the generic.

Usage

```
## S3 method for class 'SummarizedExperimentLogged'
mutate(.data, ...)

## S3 method for class 'SingleCellExperimentLogged'
mutate(.data, ...)
```

Arguments

`.data`	An ExperimentLogged object
`...`	Name-value pairs of expressions used to modify columns

Value

An ExperimentLogged object with the new or modified columns, with the change added to its log.

Examples

```
if (requireNamespace("tidySummarizedExperiment", quietly = TRUE)) {
  se <- tidySummarizedExperiment::pasilla
  se_logged <- log_start(se)
  result <- se_logged |>
    mutate(type = sub("_end","",colData(se)$type))
  result
}
```

reducedDim<-,SingleCellExperimentLogged,character-method

Assign a matrix to a named reduced dimension using `reducedDim<-` *for SingleCellExperimentLogged*

Description

Assign a matrix to a named reduced dimension using `reducedDim<-` for SingleCellExperiment-Logged

Usage

```
## S4 replacement method for signature 'SingleCellExperimentLogged,character'
reducedDim(x, type, withDimnames = TRUE, ...) <- value

## S4 replacement method for signature 'SingleCellExperimentLogged,numeric'
reducedDim(x, type, withDimnames = TRUE, ...) <- value
```

Arguments

x	A SingleCellExperimentLogged object
type	Name or index of the reduced dimension to set
withDimnames, ...	Passed to the reducedDim<- method of SingleCellExperiment
value	A matrix with one row per cell

Value

The modified SingleCellExperimentLogged object, with the change added to its log.

Examples

```
# Create a logged SingleCellExperiment
if (requireNamespace("tidySingleCellExperiment", quietly = TRUE)) {
  data(pbmc_small, package="tidySingleCellExperiment")
  sce_logged <- log_start(pbmc_small)

  reducedDim(sce_logged) <- reducedDim(sce_logged)[,1:2]
  sce_logged
}
```

---

reducedDims<-,SingleCellExperimentLogged-method

Assign a List of matrices to reducedDims using reducedDims<- *for SingleCellExperimentLogged*

Description

Assign a List of matrices to reducedDims using reducedDims<- for SingleCellExperimentLogged

Usage

```
## S4 replacement method for signature 'SingleCellExperimentLogged'
reducedDims(x, withDimnames = TRUE, ...) <- value
```

Arguments

x	A SingleCellExperimentLogged object
withDimnames, ...	Passed to the reducedDims<- method of SingleCellExperiment
value	A list of matrices, each with one row per cell

Value

The modified SingleCellExperimentLogged object, with added and modified reduced dimensions added to its log.

Examples

```
# Create a logged SingleCellExperiment
if (requireNamespace("tidySingleCellExperiment", quietly = TRUE)) {
  data(pbmc_small, package="tidySingleCellExperiment")
  sce_logged <- log_start(pbmc_small)

  reducedDims(sce_logged)$TSNE <- abs(reducedDims(sce_logged)$TSNE)
  sce_logged
}
```

---

rowData<-,ExperimentLogged,DataFrame-method

Assign column to rowData using rowData<- *for ExperimentLogged*

Description

Assign column to rowData using rowData<- for ExperimentLogged

Usage

```
## S4 replacement method for signature 'ExperimentLogged,DataFrame'
rowData(x) <- value

## S4 replacement method for signature 'ExperimentLogged,DFrame'
rowData(x) <- value
```

Arguments

x	An ExperimentLogged object
value	A DataFrame with the new row data

Value

The modified ExperimentLogged object, with added and modified rowData columns added to its log.

Examples

```
# Create a logged SingleCellExperiment
if (requireNamespace("tidySingleCellExperiment", quietly = TRUE)) {
  data(pbmc_small, package="tidySingleCellExperiment")
  sce_logged <- log_start(pbmc_small)

  rowData(sce_logged)$vst.variable <-  ifelse(rowData(sce_logged)$vst.variable,1,0 )
  sce_logged
}
```

`select.SummarizedExperimentLogged` *Filter columns of An ExperimentLogged object*

Description

`select()` is re-exported from **dplyr**, so it is available after `library(omicslog)`. These methods log the operation on ExperimentLogged objects; see `dplyr::select()` for the generic.

Usage

```
## S3 method for class 'SummarizedExperimentLogged'
select(.data, ...)

## S3 method for class 'SingleCellExperimentLogged'
select(.data, ...)
```

Arguments

`.data`	An ExperimentLogged object
`...`	Name of columns to select or deselect

Value

An ExperimentLogged object with the selected `colData` columns, with the number of removed columns added to its log. If the selection drops the key columns (`.feature`, `.sample`), tidySummarizedExperiment returns a tibble, which is returned unchanged and without a log.

Examples

```
if (requireNamespace("tidySummarizedExperiment", quietly = TRUE)) {
  se <- tidySummarizedExperiment::pasilla
  se_logged <- log_start(se)

  # Drop a colData column: still a logged object
  result <- se_logged |>
    select(!type)
  result
}
```

SingleCellExperimentLogged-class

SingleCellExperimentLogged class

Description

A class extending SingleCellExperiment to include logging capabilities. This class tracks operations performed on the object and displays a log when the object is printed.

Slots

`log_history` A data.frame storing the history of operations

Examples

```
counts <- matrix(rpois(40, 5), nrow = 5,
                 dimnames = list(paste0("gene", 1:5), paste0("cell", 1:8)))
sce <- SingleCellExperiment(assays = list(counts = counts))
sce_logged <- log_start(sce)
is(sce_logged, "SingleCellExperimentLogged")
sce_logged[, 1:4]
```

slice.SummarizedExperimentLogged

Slice rows from An ExperimentLogged object

Description

`slice()` is re-exported from **dplyr**, so it is available after `library(omicslog)`. These methods log the operation on ExperimentLogged objects; see `dplyr::slice()` for the generic.

Usage

```
## S3 method for class 'SummarizedExperimentLogged'
slice(.data, ..., .preserve = FALSE)

## S3 method for class 'SingleCellExperimentLogged'
slice(.data, ..., .preserve = FALSE)
```

Arguments

`.data`	An ExperimentLogged object
`...`	Row selection expressions
`.preserve`	If TRUE, preserves the grouping structure of the data

Value

An ExperimentLogged object with the selected rows, with the number of kept and removed rows added to its log.

Examples

```
if (requireNamespace("tidySummarizedExperiment", quietly = TRUE)) {
  se <- tidySummarizedExperiment::pasilla
  se_logged <- log_start(se)
  result <- se_logged |>
    slice(1)
  result
}
```

SummarizedExperimentLogged-class

SummarizedExperimentLogged class

Description

A class extending SummarizedExperiment to include logging capabilities. This class tracks operations performed on the object and displays a log when the object is printed.

Slots

`log_history` A data.frame storing the history of operations

Examples

```
counts <- matrix(rpois(20, 10), nrow = 5,
                 dimnames = list(paste0("gene", 1:5), paste0("s", 1:4)))
se <- SummarizedExperiment(assays = list(counts = counts))
se_logged <- log_start(se)
is(se_logged, "SummarizedExperimentLogged")
se_logged[1:3, ]
```

Index

[,SingleCellExperimentLogged,ANY,ANY,ANY-method
(`[,SummarizedExperimentLogged,ANY,ANY,ANY-method`), 2
[,SummarizedExperimentLogged,ANY,ANY,ANY-method, 2
$<-,SummarizedExperimentLogged-method, 3
$<-,SingleCellExperimentLogged-method
(`$<-,SummarizedExperimentLogged-method`), 3

assays<-,SummarizedExperimentLogged,SimpleList-method, 4
assays<-,SingleCellExperimentLogged,SimpleList-method
(`assays<-,SummarizedExperimentLogged,SimpleList-method`), 4

colData<-,ExperimentLogged,DataFrame-method, 5
colData<-,ExperimentLogged,DFrame-method
(`colData<-,ExperimentLogged,DataFrame-method`), 5

dplyr::filter(), *7*
dplyr::mutate(), *9*
dplyr::select(), *12*
dplyr::slice(), *13*

ExperimentLogged-class, 5
extract
(`extract.SummarizedExperimentLogged`), 6
extract.SummarizedExperimentLogged, 6

filter
(`filter.SummarizedExperimentLogged`), 7
filter.SummarizedExperimentLogged, 7

log_start, 8

mutate
(`mutate.SummarizedExperimentLogged`), 9
mutate.SummarizedExperimentLogged, 9

reducedDim<-,SingleCellExperimentLogged,character-method, 9
reducedDim<-,SingleCellExperimentLogged,numeric-method
(`reducedDim<-,SingleCellExperimentLogged,character-method`), 9
reducedDims<-,SingleCellExperimentLogged-method, 10
rowData<-,ExperimentLogged,DataFrame-method, 11
rowData<-,ExperimentLogged,DFrame-method
(`rowData<-,ExperimentLogged,DataFrame-method`), 11

select
(`select.SummarizedExperimentLogged`), 12
select.SummarizedExperimentLogged, 12
show,SingleCellExperimentLogged-method
(`log_start`), 8
show,SummarizedExperimentLogged-method
(`log_start`), 8
SingleCellExperimentLogged-class, 13
slice
(`slice.SummarizedExperimentLogged`), 13
slice.SummarizedExperimentLogged, 13
SummarizedExperimentLogged-class, 14

tidyr::extract(), *6*