Package ‘MultimodalExperiment’

July 10, 2025

Title Integrative Bulk and Single-Cell Experiment Container

Version 1.8.0

Description MultimodalExperiment is an S4 class that integrates bulk and single-cell experiment data; it is optimally storage-efficient, and its methods are exceptionally fast. It effortlessly represents multimodal data of any nature and features normalized experiment, subject, sample, and cell annotations, which are related to underlying biological experiments through maps. Its coordination methods are opt-in and employ database-like join operations internally to deliver fast and flexible management of multimodal data.

License Artistic-2.0

Encoding UTF-8

LazyData true

Depends R (>= 4.3.0), IRanges, S4Vectors

Imports BiocGenerics, MultiAssayExperiment, methods, utils

Suggests BiocStyle, knitr, rmarkdown

biocViews DataRepresentation, Infrastructure, SingleCell

VignetteBuilder knitr

RoxygenNote 7.3.1

Roxygen list(markdown = TRUE)

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

git_branch RELEASE_3_21

git_last_commit 210558b

git_last_commit_date 2025-04-15

Repository Bioconductor 3.21

Date/Publication 2025-07-09

Author Lucas Schiffer [aut, cre] (ORCID: <`https://orcid.org/0000-0003-3628-0326`>)

Maintainer Lucas Schiffer <`schiffer.lucas@gmail.com`>

Contents

`annotation-methods` *MultimodalExperiment Annotation Methods*

Description

`joinAnnotations` joins all annotations into an unnormalized DataFrame object.

Usage

```
## S4 method for signature 'MultimodalExperiment'
joinAnnotations(x)
```

Arguments

`x`	a MultimodalExperiment object

Value

`joinAnnotations` returns a DataFrame object.

See Also

`browseVignettes("MultimodalExperiment")`

Examples

```
ME <-
    MultimodalExperiment()

bulkExperiments(ME) <-
    ExperimentList(
        pbRNAseq = pbRNAseq
```

```
    )

singleCellExperiments(ME) <-
    ExperimentList(
        scADTseq = scADTseq,
        scRNAseq = scRNAseq
    )

subjectMap(ME)[["subject"]] <-
    "SUBJECT-1"

sampleMap(ME)[["subject"]] <-
    "SUBJECT-1"

cellMap(ME)[["sample"]] <-
    "SAMPLE-1"

ME <-
    propagate(ME)

experimentData(ME)[["published"]] <-
    c(NA_character_, "2018-11-19", "2018-11-19") |>
    as.Date()

subjectData(ME)[["condition"]] <-
    as.character("healthy")

sampleData(ME)[["sampleType"]] <-
    as.character("peripheral blood mononuclear cells")

cellType <- function(x) {
    if (x[["CD4"]] > 0L) {
        return("T Cell")
    }

    if (x[["CD14"]] > 0L) {
        return("Monocyte")
    }

    if (x[["CD19"]] > 0L) {
        return("B Cell")
    }

    if (x[["CD56"]] > 0L) {
        return("NK Cell")
    }

    NA_character_
}

cellData(ME)[["cellType"]] <-
    experiment(ME, "scADTseq") |>
    apply(2L, cellType)
```

```
joinAnnotations(ME)
```

`coordination-methods` *MultimodalExperiment Coordination Methods*

Description

Propagate or harmonize indices of a MultimodalExperiment object.

Usage

```
## S4 method for signature 'MultimodalExperiment'
propagate(x)

## S4 method for signature 'MultimodalExperiment'
harmonize(x)
```

Arguments

`x`	a MultimodalExperiment object

Details

`propagate` inserts experiment, subject, sample, and cell indices into all relevant tables by taking their union and adding missing indices.

`harmonize` deletes experiment, subject, sample, and cell indices from all relevant tables by taking their intersection and removing extraneous indices.

Value

`propagate` returns a MultimodalExperiment object.

`harmonize` returns a MultimodalExperiment object.

See Also

`browseVignettes("MultimodalExperiment")`

Examples

```
ME <-
    MultimodalExperiment()

bulkExperiments(ME) <-
    ExperimentList(
        pbRNAseq = pbRNAseq
    )
```

```
singleCellExperiments(ME) <-
    ExperimentList(
        scADTseq = scADTseq,
        scRNAseq = scRNAseq
    )

subjectMap(ME)[["subject"]] <-
    "SUBJECT-1"

sampleMap(ME)[["subject"]] <-
    "SUBJECT-1"

cellMap(ME)[["sample"]] <-
    "SAMPLE-1"

ME <-
    propagate(ME)

experimentData(ME)[["published"]] <-
    c(NA_character_, "2018-11-19", "2018-11-19") |>
    as.Date()

subjectData(ME)[["condition"]] <-
    as.character("healthy")

sampleData(ME)[["sampleType"]] <-
    as.character("peripheral blood mononuclear cells")

cellType <- function(x) {
    if (x[["CD4"]] > 0L) {
        return("T Cell")
    }

    if (x[["CD14"]] > 0L) {
        return("Monocyte")
    }

    if (x[["CD19"]] > 0L) {
        return("B Cell")
    }

    if (x[["CD56"]] > 0L) {
        return("NK Cell")
    }

    NA_character_
}

cellData(ME)[["cellType"]] <-
    experiment(ME, "scADTseq") |>
    apply(2L, cellType)
```

```
isMonocyte <-
    cellData(ME)[["cellType"]] %in% "Monocyte"

cellData(ME) <-
    cellData(ME)[isMonocyte, , drop = FALSE]

harmonize(ME)
```

`example-data` *MultimodalExperiment Example Data*

Description

Human peripheral blood mononuclear cells (PBMCs) from a single healthy donor were profiled by cellular indexing of transcriptomes and epitopes by sequencing (CITE-seq) to generate single-cell antibody-derived tag sequencing (`scADTseq`) and single-cell RNA sequencing (`scRNAseq`) data simultaneously; the `scRNAseq` data was summed into pseudo-bulk RNA sequencing (`pbRNAseq`) data. The dimensions of resulting matrices were reduced to conserve storage because these data are only used to demonstrate the functionality of the MultimodalExperiment class.

Usage

```
pbRNAseq

scADTseq

scRNAseq
```

Format

An object of class `matrix` (inherits from `array`) with 3000 rows and 1 columns.

An object of class `matrix` (inherits from `array`) with 8 rows and 5000 columns.

An object of class `matrix` (inherits from `array`) with 3000 rows and 5000 columns.

Source

PBMCs of a Healthy Donor - 5' Gene Expression with a Panel of TotalSeq™-C Antibodies, Single Cell Immune Profiling Dataset by Cell Ranger 3.0.0, 10x Genomics, (2018, November 19).

Examples

```
pbRNAseq[1:4, 1:1, drop = FALSE]

scADTseq[1:4, 1:4, drop = FALSE]

scRNAseq[1:4, 1:4, drop = FALSE]
```

`experiment-methods` *MultimodalExperiment Experiment Methods*

Description

Extract or replace experiments of a MultimodalExperiment object by index, name, or type.

Usage

```
## S4 method for signature 'MultimodalExperiment'
experiment(x, i)

## S4 replacement method for signature 'MultimodalExperiment'
experiment(x, i) <- value

## S4 method for signature 'MultimodalExperiment'
bulkExperiments(x)

## S4 replacement method for signature 'MultimodalExperiment'
bulkExperiments(x) <- value

## S4 method for signature 'MultimodalExperiment'
singleCellExperiments(x)

## S4 replacement method for signature 'MultimodalExperiment'
singleCellExperiments(x) <- value
```

Arguments

`x`	a MultimodalExperiment object
`i`	an integer or character index
`value`	a replacement value

Details

The term matrix-like objects refers to matrix objects or Bioconductor S4 objects that contain them (SummarizedExperiment, SingleCellExperiment, etc.) where rows represent features and columns represent observations.

Value

`experiment` returns a matrix-like object.

`bulkExperiments` returns an ExperimentList of matrix-like objects.

`singleCellExperiments` returns an ExperimentList of matrix-like objects.

See Also

`browseVignettes("MultimodalExperiment")`

Examples

```
ME <-
    MultimodalExperiment()

bulkExperiments(ME) <-
    ExperimentList(
        pbRNAseq = pbRNAseq
    )

singleCellExperiments(ME) <-
    ExperimentList(
        scADTseq = scADTseq,
        scRNAseq = scRNAseq
    )

subjectMap(ME)[["subject"]] <-
    "SUBJECT-1"

sampleMap(ME)[["subject"]] <-
    "SUBJECT-1"

cellMap(ME)[["sample"]] <-
    "SAMPLE-1"

ME <-
    propagate(ME)

experimentData(ME)[["published"]] <-
    c(NA_character_, "2018-11-19", "2018-11-19") |>
    as.Date()

subjectData(ME)[["condition"]] <-
    as.character("healthy")

sampleData(ME)[["sampleType"]] <-
    as.character("peripheral blood mononuclear cells")

cellType <- function(x) {
    if (x[["CD4"]] > 0L) {
        return("T Cell")
    }

    if (x[["CD14"]] > 0L) {
        return("Monocyte")
    }

    if (x[["CD19"]] > 0L) {
        return("B Cell")
```

```
    }

    if (x[["CD56"]] > 0L) {
        return("NK Cell")
    }

    NA_character_
}

cellData(ME)[["cellType"]] <-
    experiment(ME, "scADTseq") |>
    apply(2L, cellType)

experiment(ME, 2L) <-
    experiment(ME, 2L)[1:4, 1:4]

experiment(ME, 2L)

experiment(ME, "scRNAseq") <-
    experiment(ME, "scRNAseq")[1:4, 1:4]

experiment(ME, "scRNAseq")

bulkExperiments(ME) <-
    bulkExperiments(ME)[1L]

bulkExperiments(ME)

singleCellExperiments(ME) <-
    singleCellExperiments(ME)[2L]

singleCellExperiments(ME)
```

`map-methods` *MultimodalExperiment Map Methods*

Description

`joinMaps` joins all maps into an unnormalized DataFrame object.

Usage

```
## S4 method for signature 'MultimodalExperiment'
joinMaps(x)
```

Arguments

`x`	a MultimodalExperiment object

Value

`joinMaps` returns a DataFrame object.

See Also

`browseVignettes("MultimodalExperiment")`

Examples

```
ME <-
    MultimodalExperiment()

bulkExperiments(ME) <-
    ExperimentList(
        pbRNAseq = pbRNAseq
    )

singleCellExperiments(ME) <-
    ExperimentList(
        scADTseq = scADTseq,
        scRNAseq = scRNAseq
    )

subjectMap(ME)[["subject"]] <-
    "SUBJECT-1"

sampleMap(ME)[["subject"]] <-
    "SUBJECT-1"

cellMap(ME)[["sample"]] <-
    "SAMPLE-1"

ME <-
    propagate(ME)

experimentData(ME)[["published"]] <-
    c(NA_character_, "2018-11-19", "2018-11-19") |>
    as.Date()

subjectData(ME)[["condition"]] <-
    as.character("healthy")

sampleData(ME)[["sampleType"]] <-
    as.character("peripheral blood mononuclear cells")

cellType <- function(x) {
    if (x[["CD4"]] > 0L) {
        return("T Cell")
    }

    if (x[["CD14"]] > 0L) {
        return("Monocyte")
```

```
    }

    if (x[["CD19"]] > 0L) {
        return("B Cell")
    }

    if (x[["CD56"]] > 0L) {
        return("NK Cell")
    }

    NA_character_
}

cellData(ME)[["cellType"]] <-
    experiment(ME, "scADTseq") |>
    apply(2L, cellType)

joinMaps(ME)
```

`MultimodalExperiment` *MultimodalExperiment Constructor Function*

Description

`MultimodalExperiment` constructs a MultimodalExperiment object.

Usage

```
MultimodalExperiment(
    experimentData = DataFrame(),
    subjectData = DataFrame(),
    sampleData = DataFrame(),
    cellData = DataFrame(),
    experimentMap = DataFrame(
        type = character(),
        experiment = character()
    ),
    subjectMap = DataFrame(
        experiment = character(),
        subject = character()
    ),
    sampleMap = DataFrame(
        subject = character(),
        sample = character()
    ),
    cellMap = DataFrame(
        sample = character(),
```

```
        cell = character()
    ),
    experiments = ExperimentList(),
    metadata = list()
)
```

Arguments

`experimentData`	a DataFrame of experiment annotations with experiment indices as rownames
`subjectData`	a DataFrame of subject annotations with subject indices as rownames
`sampleData`	a DataFrame of sample annotations with sample indices as rownames
`cellData`	a DataFrame of cell annotations with cell indices as rownames
`experimentMap`	a DataFrame of type (bulk or single-cell) to experiment (index) mappings
`subjectMap`	a DataFrame of experiment (index) to subject (index) mappings
`sampleMap`	a DataFrame of subject (index) to sample (index) mappings
`cellMap`	a DataFrame of sample (index) to cell (index) mappings
`experiments`	an ExperimentList of matrix-like objects
`metadata`	a list of metadata objects

Details

The term matrix-like objects refers to matrix objects or Bioconductor S4 objects that contain them (SummarizedExperiment, SingleCellExperiment, etc.) where rows represent features and columns represent observations.

Value

`MultimodalExperiment` returns a MultimodalExperiment object.

See Also

`browseVignettes("MultimodalExperiment")`

Examples

```
MultimodalExperiment()
```

`MultimodalExperiment-class`
MultimodalExperiment Class Definition

Description

MultimodalExperiment is an S4 class that integrates bulk and single-cell experiment data; it is optimally storage-efficient, and its methods are exceptionally fast. It effortlessly represents multimodal data of any nature and features normalized experiment, subject, sample, and cell annotations, which are related to underlying biological experiments through maps. Its coordination methods are opt-in and employ database-like join operations internally to deliver fast and flexible management of multimodal data.

Details

The term matrix-like objects refers to matrix objects or Bioconductor S4 objects that contain them (SummarizedExperiment, SingleCellExperiment, etc.) where rows represent features and columns represent observations.

Slots

`experimentData` a DataFrame of experiment annotations with experiment indices as rownames

`subjectData` a DataFrame of subject annotations with subject indices as rownames

`sampleData` a DataFrame of sample annotations with sample indices as rownames

`cellData` a DataFrame of cell annotations with cell indices as rownames

`experimentMap` a DataFrame of type (bulk or single-cell) to experiment (index) mappings

`subjectMap` a DataFrame of experiment (index) to subject (index) mappings

`sampleMap` a DataFrame of subject (index) to sample (index) mappings

`cellMap` a DataFrame of sample (index) to cell (index) mappings

`experiments` an ExperimentList of matrix-like objects

`metadata` a list of metadata objects

See Also

`browseVignettes("MultimodalExperiment")`

---

name-methods *MultimodalExperiment Name Methods*

Description

Extract or replace names of a MultimodalExperiment object.

Usage

```
## S4 method for signature 'MultimodalExperiment'
names(x)

## S4 replacement method for signature 'MultimodalExperiment'
names(x) <- value

## S4 method for signature 'MultimodalExperiment'
rownames(x)

## S4 replacement method for signature 'MultimodalExperiment'
rownames(x) <- value

## S4 method for signature 'MultimodalExperiment'
colnames(x)

## S4 replacement method for signature 'MultimodalExperiment,ANY'
colnames(x) <- value

## S4 method for signature 'MultimodalExperiment'
dimnames(x)

## S4 replacement method for signature 'MultimodalExperiment,ANY'
dimnames(x) <- value

## S4 method for signature 'MultimodalExperiment'
experimentNames(x)

## S4 replacement method for signature 'MultimodalExperiment'
experimentNames(x) <- value
```

Arguments

`x`	a MultimodalExperiment object
`value`	a replacement value

Value

`names` returns a CharacterList object.

`rownames` returns a CharacterList object.

`colnames` returns a CharacterList object.

`dimnames` returns a list object.

`experimentNames` returns a character vector.

See Also

`browseVignettes("MultimodalExperiment")`

Examples

```
ME <-
    MultimodalExperiment()

bulkExperiments(ME) <-
    ExperimentList(
        pbRNAseq = pbRNAseq
    )

singleCellExperiments(ME) <-
    ExperimentList(
        scADTseq = scADTseq,
        scRNAseq = scRNAseq
    )

subjectMap(ME)[["subject"]] <-
    "SUBJECT-1"

sampleMap(ME)[["subject"]] <-
    "SUBJECT-1"

cellMap(ME)[["sample"]] <-
    "SAMPLE-1"

ME <-
    propagate(ME)

experimentData(ME)[["published"]] <-
    c(NA_character_, "2018-11-19", "2018-11-19") |>
    as.Date()

subjectData(ME)[["condition"]] <-
    as.character("healthy")

sampleData(ME)[["sampleType"]] <-
    as.character("peripheral blood mononuclear cells")

cellType <- function(x) {
    if (x[["CD4"]] > 0L) {
        return("T Cell")
    }
```

```
    if (x[["CD14"]] > 0L) {
        return("Monocyte")
    }

    if (x[["CD19"]] > 0L) {
        return("B Cell")
    }

    if (x[["CD56"]] > 0L) {
        return("NK Cell")
    }

    NA_character_
}

cellData(ME)[["cellType"]] <-
    experiment(ME, "scADTseq") |>
    apply(2L, cellType)

names(ME) <-
    names(ME) |>
    tolower()

names(ME)

rownames(ME) <-
    rownames(ME) |>
    toupper()

rownames(ME)

colnames(ME) <-
    colnames(ME) |>
    tolower()

colnames(ME)

dimnames(ME)[[2L]] <-
    dimnames(ME)[[2L]] |>
    toupper()

dimnames(ME)[[2L]]

experimentNames(ME) <-
    experimentNames(ME) |>
    gsub(pattern = "seq", replacement = "-seq")

experimentNames(ME)
```

`reexports` *Objects exported from other packages*

Description

These objects are imported from other packages. Follow the links below to see their documentation.

MultiAssayExperiment `ExperimentList`

`show-method` *MultimodalExperiment Show Method*

Description

Display details about a MultimodalExperiment object.

Usage

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

Arguments

`object`	a MultimodalExperiment object

Value

show returns `NULL` invisibly.

See Also

`browseVignettes("MultimodalExperiment")`

Examples

```
ME <-
    MultimodalExperiment()

bulkExperiments(ME) <-
    ExperimentList(
        pbRNAseq = pbRNAseq
    )

singleCellExperiments(ME) <-
    ExperimentList(
        scADTseq = scADTseq,
        scRNAseq = scRNAseq
```

```
    )

subjectMap(ME)[["subject"]] <-
    "SUBJECT-1"

sampleMap(ME)[["subject"]] <-
    "SUBJECT-1"

cellMap(ME)[["sample"]] <-
    "SAMPLE-1"

ME <-
    propagate(ME)

experimentData(ME)[["published"]] <-
    c(NA_character_, "2018-11-19", "2018-11-19") |>
    as.Date()

subjectData(ME)[["condition"]] <-
    as.character("healthy")

sampleData(ME)[["sampleType"]] <-
    as.character("peripheral blood mononuclear cells")

cellType <- function(x) {
    if (x[["CD4"]] > 0L) {
        return("T Cell")
    }

    if (x[["CD14"]] > 0L) {
        return("Monocyte")
    }

    if (x[["CD19"]] > 0L) {
        return("B Cell")
    }

    if (x[["CD56"]] > 0L) {
        return("NK Cell")
    }

    NA_character_
}

cellData(ME)[["cellType"]] <-
    experiment(ME, "scADTseq") |>
    apply(2L, cellType)

show(ME)
```

`slot-methods` *MultimodalExperiment Slot Methods*

Description

Extract or replace slots of a MultimodalExperiment object.

Usage

```
## S4 method for signature 'MultimodalExperiment'
experimentData(object)

## S4 replacement method for signature 'MultimodalExperiment'
experimentData(object) <- value

## S4 method for signature 'MultimodalExperiment'
subjectData(object)

## S4 replacement method for signature 'MultimodalExperiment'
subjectData(object) <- value

## S4 method for signature 'MultimodalExperiment'
sampleData(object)

## S4 replacement method for signature 'MultimodalExperiment'
sampleData(object) <- value

## S4 method for signature 'MultimodalExperiment'
cellData(object)

## S4 replacement method for signature 'MultimodalExperiment'
cellData(object) <- value

## S4 method for signature 'MultimodalExperiment'
experimentMap(object)

## S4 replacement method for signature 'MultimodalExperiment'
experimentMap(object) <- value

## S4 method for signature 'MultimodalExperiment'
subjectMap(object)

## S4 replacement method for signature 'MultimodalExperiment'
subjectMap(object) <- value

## S4 method for signature 'MultimodalExperiment'
sampleMap(object)
```

```
## S4 replacement method for signature 'MultimodalExperiment'
sampleMap(object) <- value

## S4 method for signature 'MultimodalExperiment'
cellMap(object)

## S4 replacement method for signature 'MultimodalExperiment'
cellMap(object) <- value

## S4 method for signature 'MultimodalExperiment'
experiments(object)

## S4 replacement method for signature 'MultimodalExperiment'
experiments(object) <- value
```

Arguments

`object`	a MultimodalExperiment object
`value`	a replacement value

Value

Extract methods return the value of the slot.

See Also

`browseVignettes("MultimodalExperiment")`

Examples

```
ME <-
    MultimodalExperiment()

bulkExperiments(ME) <-
    ExperimentList(
        pbRNAseq = pbRNAseq
    )

singleCellExperiments(ME) <-
    ExperimentList(
        scADTseq = scADTseq,
        scRNAseq = scRNAseq
    )

subjectMap(ME)[["subject"]] <-
    "SUBJECT-1"

sampleMap(ME)[["subject"]] <-
    "SUBJECT-1"
```

```
cellMap(ME)[["sample"]] <-
    "SAMPLE-1"

ME <-
    propagate(ME)

experimentData(ME)[["published"]] <-
    c(NA_character_, "2018-11-19", "2018-11-19") |>
    as.Date()

subjectData(ME)[["condition"]] <-
    as.character("healthy")

sampleData(ME)[["sampleType"]] <-
    as.character("peripheral blood mononuclear cells")

cellType <- function(x) {
    if (x[["CD4"]] > 0L) {
        return("T Cell")
    }

    if (x[["CD14"]] > 0L) {
        return("Monocyte")
    }

    if (x[["CD19"]] > 0L) {
        return("B Cell")
    }

    if (x[["CD56"]] > 0L) {
        return("NK Cell")
    }

    NA_character_
}

cellData(ME)[["cellType"]] <-
    experiment(ME, "scADTseq") |>
    apply(2L, cellType)

experimentData(ME)

subjectData(ME)

sampleData(ME)

cellData(ME)

experimentMap(ME)

subjectMap(ME)

sampleMap(ME)
```

```
cellMap(ME)

experiments(ME)
```

`subset-methods` *MultimodalExperiment Subset Methods*

Description

Extract or replace parts of a MultimodalExperiment object.

Usage

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

## S4 replacement method for signature 'MultimodalExperiment,ANY,ANY,ANY'
x[i, j] <- value
```

Arguments

`x`	a MultimodalExperiment object
`i`	a list, List, LogicalList, IntegerList, or CharacterList of elements to extract or replace
`j`	a list, List, LogicalList, IntegerList, or CharacterList of elements to extract or replace
`...`	ignored, required by generic
`drop`	ignored, required by generic
`value`	a replacement value

Value

`[` returns a MultimodalExperiment object.

See Also

`browseVignettes("MultimodalExperiment")`

Examples

```
ME <-
    MultimodalExperiment()

bulkExperiments(ME) <-
    ExperimentList(
        pbRNAseq = pbRNAseq
    )

singleCellExperiments(ME) <-
    ExperimentList(
        scADTseq = scADTseq,
        scRNAseq = scRNAseq
    )

subjectMap(ME)[["subject"]] <-
    "SUBJECT-1"

sampleMap(ME)[["subject"]] <-
    "SUBJECT-1"

cellMap(ME)[["sample"]] <-
    "SAMPLE-1"

ME <-
    propagate(ME)

experimentData(ME)[["published"]] <-
    c(NA_character_, "2018-11-19", "2018-11-19") |>
    as.Date()

subjectData(ME)[["condition"]] <-
    as.character("healthy")

sampleData(ME)[["sampleType"]] <-
    as.character("peripheral blood mononuclear cells")

cellType <- function(x) {
    if (x[["CD4"]] > 0L) {
        return("T Cell")
    }

    if (x[["CD14"]] > 0L) {
        return("Monocyte")
    }

    if (x[["CD19"]] > 0L) {
        return("B Cell")
    }

    if (x[["CD56"]] > 0L) {
        return("NK Cell")
```

```
    }

    NA_character_
}

cellData(ME)[["cellType"]] <-
    experiment(ME, "scADTseq") |>
    apply(2L, cellType)

i <-
    rownames(ME) |>
    endoapply(sample, 4L)

j <-
    colnames(ME) |>
    endoapply(sample, 1L)

ME[i, j] <-
    0L

experiment(ME[i, j], "pbRNAseq")

experiment(ME[i, j], "scADTseq")

experiment(ME[i, j], "scRNAseq")
```

Index