

# Package ‘spillR’

May 20, 2024

**Type** Package

**Title** Spillover Compensation in Mass Cytometry Data

**Version** 1.0.0

**Description** Channel interference in mass cytometry can cause spillover and may result in miscounting of protein markers. We develop a nonparametric finite mixture model and use the mixture components to estimate the probability of spillover. We implement our method using expectation-maximization to fit the mixture model.

**biocViews** FlowCytometry, ImmunoOncology, MassSpectrometry, Preprocessing, SingleCell, Software, StatisticalMethod, Visualization, Regression

**License** LGPL-3

**Encoding** UTF-8

**LazyData** false

**Config/testthat/edition** 3

**RoxygenNote** 7.2.3

**Imports** dplyr, tibble, tidyselect, stats, ggplot2, tidyr, spatstat.geom, S4Vectors, parallel

**Depends** R (>= 4.3.0), SummarizedExperiment, CATALYST

**Suggests** knitr, rmarkdown, cowplot, testthat (>= 3.0.0), BiocStyle, hexbin

**VignetteBuilder** knitr

**git\_url** <https://git.bioconductor.org/packages/spillR>

**git\_branch** RELEASE\_3\_19

**git\_last\_commit** 20e0144

**git\_last\_commit\_date** 2024-04-30

**Repository** Bioconductor 3.19

**Date/Publication** 2024-05-19

**Author** Marco Guazzini [aut, cre] (<<https://orcid.org/0009-0007-8111-5772>>),  
 Alexander G. Reisach [aut],  
 Sebastian Weichwald [aut] (<<https://orcid.org/0000-0003-0169-7244>>),  
 Christof Seiler [aut] (<<https://orcid.org/0000-0001-8802-3642>>)

**Maintainer** Marco Guazzini <m.guazzini@student.maastrichtuniversity.nl>

## Contents

compCytof . . . . .	2
compensate . . . . .	3
compensate_naive . . . . .	4
generate_bead . . . . .	5
generate_real . . . . .	5
plotDiagnostics . . . . .	6
tfm . . . . .	7

<b>Index</b>	<b>8</b>
--------------	----------

---

compCytof	<i>Compute spillover probability and correct for spillover</i>
-----------	--

---

## Description

Compute spillover probability and correct for spillover

## Usage

```
compCytof(
  sce,
  sce_bead,
  marker_to_barcode,
  impute_value,
  overwrite = FALSE,
  n_cores = 1,
  naive = FALSE
)
```

## Arguments

sce	<a href="#">SingleCellExperiment</a> for the real cells
sce_bead	<a href="#">SingleCellExperiment</a> for the bead experiment
marker_to_barcode	Table that maps the marker to the barcode in the beads experiment
impute_value	Imputed value for counts that are declared as spillover
overwrite	logical; if TRUE data are overwritten if FALSE data are saved in new columns
n_cores	Number of computing cores
naive	logical; if TRUE use the naive version

**Value**

A `SingleCellExperiment` object

**Examples**

```
library(CATALYST)
library(dplyr)
bc_key <- c(139, 141:156, 158:176)
sce_bead <- prepData(ss_exp)
sce_bead <- assignPrelim(sce_bead, bc_key, verbose = FALSE)
sce_bead <- applyCutoffs(estCutoffs(sce_bead))
sce_bead <- computeSpillmat(sce_bead)
data(mp_cells, package = "CATALYST")
sce <- prepData(mp_cells)
marker_to_barcode <- rowData(sce_bead)[, c("channel_name", "is_bc")] |>
  as_tibble() |>
  filter(is_bc == TRUE) |>
  mutate(barcode = bc_key) |>
  select(marker = channel_name, barcode)
spillR::compCytotf(sce, sce_bead, marker_to_barcode, impute_value = NA)
```

---

 compensate

*Compute spillover probability and correct for spillover*


---

**Description**

Compute spillover probability and correct for spillover

**Usage**

```
compensate(
  tb_real,
  tb_bead,
  target_marker,
  spillover_markers,
  impute_value = NA,
  n_iter = 1000
)
```

**Arguments**

<code>tb_real</code>	Data frame or tibble with proteins counts of real experiment
<code>tb_bead</code>	Data frame or tibble with proteins counts of bead experiment
<code>target_marker</code>	Marker name in real experiment
<code>spillover_markers</code>	Marker names in bead experiment
<code>impute_value</code>	Value for counts that are declared as spillover
<code>n_iter</code>	Maximum number of EM steps

**Value**

A list of class `spillr` containing

<code>tb_compensate</code>	corrected real cells
<code>tb_spill_prob</code>	probability curve
<code>convergence</code>	covergence table of EM algorithm
<code>tb_real</code>	input real cells
<code>tb_bead</code>	input bead cells
<code>target_marker</code>	input marker in real experiment
<code>spillover_markers</code>	input markers in bead experiment

---

<code>compensate_naive</code>	<i>Compute spillover probability and correct for spillover from beads only</i>
-------------------------------	--

---

**Description**

Compute spillover probability and correct for spillover from beads only

**Usage**

```
compensate_naive(
  tb_real,
  tb_bead,
  target_marker,
  spillover_markers,
  impute_value = NA
)
```

**Arguments**

<code>tb_real</code>	Data frame or tibble with proteins counts of real experiment
<code>tb_bead</code>	Data frame or tibble with proteins counts of bead experiment
<code>target_marker</code>	Marker name in real experiment
<code>spillover_markers</code>	Marker names in bead experiment
<code>impute_value</code>	Value for counts that are declared as spillover

**Value**

A list of class `spillr` containing

<code>tb_compensate</code>	corrected real cells
<code>tb_spill_prob</code>	probability curve
<code>convergence</code>	covergence table of EM algorithm
<code>tb_real</code>	input real cells
<code>tb_bead</code>	input bead cells
<code>target_marker</code>	input marker in real experiment
<code>spillover_markers</code>	input markers in bead experiment

---

<code>generate_bead</code>	<i>Generate dataset for vignettes and simulation studies</i>
----------------------------	--

---

**Description**

Generate dataset for vignettes and simulation studies

**Usage**

```
generate_bead()
```

**Value**

`tibble` data frame

**Examples**

```
set.seed(23)
generate_bead()
```

---

<code>generate_real</code>	<i>Generate dataset for vignettes and simulation studies</i>
----------------------------	--

---

**Description**

Generate dataset for vignettes and simulation studies

**Usage**

```
generate_real()
```

**Value**

`tibble` data frame

**Examples**

```
set.seed(23)
generate_real()
```

---

plotDiagnostics	<i>Compute spillover probability and correct for spillover</i>
-----------------	--

---

**Description**

Compute spillover probability and correct for spillover

**Usage**

```
plotDiagnostics(sce, ch)
```

**Arguments**

sce	A <code>SingleCellExperiment</code> object
ch	Character string specifying the channel to plot

**Value**

A list of `ggplot2` plots

**Examples**

```
library(CATALYST)
library(dplyr)
bc_key <- c(139, 141:156, 158:176)
sce_bead <- prepData(ss_exp)
sce_bead <- assignPrelim(sce_bead, bc_key, verbose = FALSE)
sce_bead <- applyCutoffs(estCutoffs(sce_bead))
sce_bead <- computeSpillmat(sce_bead)
data(mp_cells, package = "CATALYST")
sce <- prepData(mp_cells)
marker_to_barcode <- rowData(sce_bead)[, c("channel_name", "is_bc")] |>
  as_tibble() |>
  filter(is_bc == TRUE) |>
  mutate(barcode = bc_key) |>
  select(marker = channel_name, barcode)
sce <- spillR::compCytOf(sce, sce_bead, marker_to_barcode, impute_value = NA)
plotDiagnostics(sce, "Yb173Di")
```

---

tfm	<i>Variance stabilizing transform of counts</i>
-----	---

---

**Description**

Variance stabilizing transform of counts

**Usage**

tfm(x)

**Arguments**

x                      Raw count

**Value**

A transformed count

# Index

compCytof, [2](#)  
compensate, [3](#)  
compensate\_naive, [4](#)

generate\_bead, [5](#)  
generate\_real, [5](#)  
ggplot2, [6](#)

plotDiagnostics, [6](#)

SingleCellExperiment, [2](#), [3](#), [6](#)

tfm, [7](#)  
tibble, [5](#), [6](#)