

# Package ‘sccomp’

September 21, 2023

**Title** Robust Outlier-aware Estimation of Composition and Heterogeneity  
for Single-cell Data

**Version** 1.5.1

**Description** A robust and outlier-aware method for testing differential tissue composition from single-cell data. This model can infer changes in tissue composition and heterogeneity, and can produce realistic data simulations based on any existing dataset. This model can also transfer knowledge from a large set of integrated datasets to increase accuracy further.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.2.3

**Biarch** true

**Depends** R (>= 4.2.0)

**Imports** methods, Rcpp (>= 0.12.0), RcppParallel (>= 5.0.1), rstantools (>= 2.1.1), rstan (>= 2.26.0), SeuratObject, SingleCellExperiment, parallel, dplyr, tidyr, purrr, magrittr, rlang, tibble, boot, lifecycle, stats, tidyselect, utils, ggplot2, ggrepel, patchwork, forcats, readr, scales, stringr, glue

**Suggests** BiocStyle, testthat (>= 3.0.0), markdown, knitr, tidyseurat, tidySingleCellExperiment, loo

**Enhances** furr, extraDistr

**LinkingTo** BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), RcppParallel (>= 5.0.1), rstan (>= 2.26.0), StanHeaders (>= 2.26.0)

**SystemRequirements** GNU make

**VignetteBuilder** knitr

**RdMacros** lifecycle

**biocViews** ImmunoOncology, Normalization, Sequencing, RNASeq, Software, GeneExpression, Transcriptomics, SingleCell, Clustering

**LazyDataCompression** xz

**Config/testthat/edition** 3

**URL** <https://github.com/stemangiola/scomp>

**BugReports** <https://github.com/stemangiola/scomp/issues>

**Additional\_repositories** <https://mc-stan.org/r-packages/>

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

**git\_branch** devel

**git\_last\_commit** a6bb085

**git\_last\_commit\_date** 2023-09-18

**Date/Publication** 2023-09-20

**Author** Stefano Mangiola [aut, cre]

**Maintainer** Stefano Mangiola <mangiolastefano@gmail.com>

## R topics documented:

scomp-package . . . . .	3
counts_obj . . . . .	3
glm_dirichlet_multinomial . . . . .	3
glm_dirichlet_multinomial_generate_quantities . . . . .	4
glm_dirichlet_multinomial_imputation . . . . .	4
glm_multi_beta . . . . .	5
glm_multi_beta_generate_data . . . . .	5
multi_beta_glm . . . . .	6
plot_summary . . . . .	7
remove_unwanted_variation . . . . .	7
scomp_glm . . . . .	8
scomp_predict . . . . .	12
scomp_replicate . . . . .	13
sce_obj . . . . .	14
seurat_obj . . . . .	15
simulate_data . . . . .	15
test_contrasts . . . . .	17
<b>Index</b>	<b>19</b>

---

scomp-package      *The 'scomp' package.*

---

**Description**

A DESCRIPTION OF THE PACKAGE

**References**

Stan Development Team (2020). RStan: the R interface to Stan. R package version 2.21.2.  
<https://mc-stan.org>

---

counts\_obj      *counts\_obj*

---

**Description**

Example data set containing cell counts per cell cluster

**Usage**

```
data(counts_obj)
```

**Format**

A tidy data frame.

---

glm\_dirichlet\_multinomial  
*glm\_dirichlet\_multinomial*

---

**Description**

This object is mostly for internal use and comparative purposes, if the `dirichlet_multinomial` is chosen as noise model.

**Usage**

```
data(glm_dirichlet_multinomial)
```

**Format**

A text file containing stan code for the Dirichlet model.

---

```
glm_dirichlet_multinomial_generate_quantities  
glm_dirichlet_multinomial_generate_quantities
```

---

**Description**

This object is mostly for internal use and comparative purposes, if the `dirichlet_multinomial` is chosen as noise model.

**Usage**

```
data(glm_dirichlet_multinomial_generate_quantities)
```

**Format**

A text file containing stan code for the Dirichlet model.

---

```
glm_dirichlet_multinomial_imputation  
glm_dirichlet_multinomial_imputation
```

---

**Description**

This object is mostly for internal use and comparative purposes, if the `dirichlet_multinomial` is chosen as noise model.

**Usage**

```
data(glm_dirichlet_multinomial_imputation)
```

**Format**

A text file containing stan code for the Dirichlet model.

---

`glm_multi_beta`      *glm\_multi\_beta*

---

**Description**

This object is mostly for internal use and comparative purposes, if the `multi_beta` is chosen as noise model.

**Usage**

```
data(glm_multi_beta)
```

**Format**

A text file containing stan code for the Beta only model.

---

`glm_multi_beta_generate_data`  
*glm\_multi\_beta\_generate\_data*

---

**Description**

This object is mostly for internal use and comparative purposes, if the `multi_beta` is chosen as noise model.

**Usage**

```
data(glm_multi_beta_generate_data)
```

**Format**

A text file containing stan code for the Beta only model.

---

multi\_beta\_glm      *multi\_beta\_glm main*

---

### Description

This function runs the data modelling and statistical test for the hypothesis that a cell\_type includes outlier biological replicate.

### Usage

```
multi_beta_glm(
  .data,
  formula = ~1,
  .sample,
  check_outliers = FALSE,
  approximate_posterior_inference = TRUE,
  cores = detect_cores(),
  seed = sample(1e+05, 1)
)
```

### Arguments

.data	A tibble including a cell_type name column   sample name column   read counts column   factor columns   Pvalue column   a significance column
formula	A formula. The sample formula used to perform the differential cell_type abundance analysis
.sample	A column name as symbol. The sample identifier
check_outliers	A boolean. Whether to check for outliers before the fit.
approximate_posterior_inference	A boolean. Whether the inference of the joint posterior distribution should be approximated with variational Bayes. It confers execution time advantage.
cores	An integer. How many cored to be used with parallel calculations.
seed	An integer. Used for development and testing purposes

### Value

A nested tibble tbl with cell\_type-wise information: sample wise data | plot | ppc samples failed | exposure deleterious outliers

---

plot_summary	<i>plot_summary</i>
--------------	---------------------

---

**Description**

This function plots a summary of the results of the model.

**Usage**

```
plot_summary(.data, significance_threshold = 0.025)
```

**Arguments**

`.data` A tibble including a `cell_group` name column | sample name column | read counts column | factor columns | Pvalue column | a significance column

`significance_threshold` A real. FDR threshold for labelling significant cell-groups.

**Value**

A ggplot

**Examples**

```
data("counts_obj")

estimate =
  sccomp_glm(
    counts_obj ,
    ~ type, ~1, sample, cell_group, count,
    approximate_posterior_inference = "all",
    check_outliers = FALSE,
    cores = 1
  )

# estimate |> plot_summary()
```

---

remove_unwanted_variation	<i>remove_unwanted_variation</i>
---------------------------	----------------------------------

---

**Description**

This function uses the model to remove unwanted variation from a dataset using the estimated of the model. For example if you fit your data with this formula `~ factor_1 + factor_2` and use this formula to remove unwanted variation `~ factor_1`, the `factor_2` will be factored out.

**Usage**

```
remove_unwanted_variation(
  .data,
  formula_composition = ~1,
  formula_variability = NULL
)
```

**Arguments**

`.data` A tibble. The result of `scomp_glm`.

`formula_composition` A formula. The formula describing the model for differential abundance, for example `~treatment`. This formula can be a sub-formula of your estimated model; in this case all other factor will be factored out.

`formula_variability` A formula. The formula describing the model for differential variability, for example `~treatment`. In most cases, if differentially variability is of interest, the formula should only include the factor of interest as a large amount of data is needed to define variability depending to each factors. This formula can be a sub-formula of your estimated model; in this case all other factor will be factored out.

**Value**

A nested tibble `tbl` with `cell_group`-wise statistics

**Examples**

```
data("counts_obj")

estimates = scomp_glm(
  counts_obj ,
  ~ type, ~1, sample, cell_group, count,
  approximate_posterior_inference = "all",
  check_outliers = FALSE,
  cores = 1
)

remove_unwanted_variation(estimates)
```



## Description

The function for linear modelling takes as input a table of cell counts with three columns containing a cell-group identifier, sample identifier, integer count and the factors (continuous or discrete). The user can define a linear model with an input R formula, where the first factor is the factor of interest. Alternatively, scomp accepts single-cell data containers (Seurat, SingleCellExperiment44, cell metadata or group-size). In this case, scomp derives the count data from cell metadata.

## Usage

```
scomp_glm(
  .data,
  formula_composition = ~1,
  formula_variability = ~1,
  .sample,
  .cell_group,
  .count = NULL,
  contrasts = NULL,
  prior_mean_variable_association = list(intercept = c(5, 2), slope = c(0, 0.6),
    standard_deviation = c(20, 40)),
  check_outliers = TRUE,
  bimodal_mean_variability_association = FALSE,
  enable_loo = FALSE,
  cores = detectCores(),
  percent_false_positive = 5,
  approximate_posterior_inference = "none",
  test_composition_above_logit_fold_change = 0.2,
  .sample_cell_group_pairs_to_exclude = NULL,
  verbose = FALSE,
  noise_model = "multi_beta_binomial",
  exclude_priors = FALSE,
  use_data = TRUE,
  mcmc_seed = sample(1e+05, 1),
  max_sampling_iterations = 20000,
  pass_fit = TRUE
)
```

## Arguments

<code>.data</code>	A tibble including a <code>cell_group</code> name column   sample name column   read counts column (optional depending on the input class)   factor columns.
<code>formula_composition</code>	A formula. The formula describing the model for differential abundance, for example <code>~treatment</code> .
<code>formula_variability</code>	A formula. The formula describing the model for differential variability, for example <code>~treatment</code> . In most cases, if differentially variability is of interest, the formula should only include the factor of interest as a large amount of data is needed to define variability depending to each factors.

<code>.sample</code>	A column name as symbol. The sample identifier
<code>.cell_group</code>	A column name as symbol. The cell_group identifier
<code>.count</code>	A column name as symbol. The cell_group abundance (read count). Used only for data frame count output. The variable in this column should be of class integer.
<code>contrasts</code>	A vector of character strings. For example if your formula is $\sim 0 + \text{treatment}$ and the factor treatment has values yes and no, your contrast could be <code>contrasts = c("treatmentyes - treatmentno")</code> .
<code>prior_mean_variable_association</code>	A list of the form <code>list(intercept = c(5, 2), slope = c(0, 0.6), standard_deviation = c(20, 40))</code> . Where for intercept and slope parameters, we specify mean and standard deviation, while for standard deviation, we specify shape and rate. This is used to incorporate prior knowledge about the mean/variability association of cell-type proportions.
<code>check_outliers</code>	A boolean. Whether to check for outliers before the fit.
<code>bimodal_mean_variability_association</code>	A boolean. Whether to model the mean-variability as bimodal, as often needed in the case of single-cell RNA sequencing data, and not usually for CyTOF and microbiome data. The <code>plot_summary_plot()\$credible_intervals_2D</code> can be used to assess whether the bimodality should be modelled.
<code>enable_loo</code>	A boolean. Enable model comparison by the R package LOO. This is helpful when you want to compare the fit between two models, for example, analogously to ANOVA, between a one factor model versus a intercept-only model.
<code>cores</code>	An integer. How many cores to be used with parallel calculations.
<code>percent_false_positive</code>	A real between 0 and 100 non included. This used to identify outliers with a specific false positive rate.
<code>approximate_posterior_inference</code>	A boolean. Whether the inference of the joint posterior distribution should be approximated with variational Bayes. It confers execution time advantage.
<code>test_composition_above_logit_fold_change</code>	A positive integer. It is the effect threshold used for the hypothesis test. A value of 0.2 correspond to a change in cell proportion of 10% for a cell type with baseline proportion of 50%. That is, a cell type goes from 45% to 50%. When the baseline proportion is closer to 0 or 1 this effect threshold has consistent value in the logit unconstrained scale.
<code>.sample_cell_group_pairs_to_exclude</code>	A column name that includes a boolean variable for the sample/cell-group pairs to be ignored in the fit. This argument is for pro-users.
<code>verbose</code>	A boolean. Prints progression.
<code>noise_model</code>	A character string. The two noise models available are <code>multi_beta_binomial</code> (default) and <code>dirichlet_multinomial</code> .
<code>exclude_priors</code>	A boolean. Whether to run a prior-free model, for benchmarking purposes.
<code>use_data</code>	A boolean. Whether to sun the model data free. This can be used for prior predictive check.

mcmc_seed	An integer. Used for Markov-chain Monte Carlo reproducibility. By default a random number is sampled from 1 to 999999. This itself can be controlled by <code>set.seed()</code>
max_sampling_iterations	An integer. This limit the maximum number of iterations in case a large dataset is used, for limiting the computation time.
pass_fit	A boolean. Whether to pass the Stan fit as attribute in the output. Because the Stan fit can be very large, setting this to <code>FALSE</code> can be used to lower the memory imprint to save the output.

## Value

A nested tibble tbl, with the following columns

- `cell_group` - column including the cell groups being tested
- `parameter` - The parameter being estimated, from the design matrix dscribed with the input `formula_composition` and `formula_variability`
- `factor` - The factor in the formula corresponding to the covariate, if exists (e.g. it does not exist in case og Intercept or contrasts, which usually are combination of parameters)
- `c_lower` - lower (2.5%) quantile of the posterior distribution for a composition (c) parameter.
- `c_effect` - mean of the posterior distribution for a composition (c) parameter.
- `c_upper` - upper (97.5%) quantile of the posterior distribution fo a composition (c) parameter.
- `c_pH0` - Probability of the null hypothesis (no difference) for a composition (c). This is not a p-value.
- `c_FDR` - False-discovery rate of the null hypothesis (no difference) for a composition (c).
- `c_n_eff` - Effective sample size - the number of independent draws in the sample, the higher the better ([mc-stan.org/docs/2\\_25/cmdstan-guide/stansummary.html](http://mc-stan.org/docs/2_25/cmdstan-guide/stansummary.html)).
- `c_R_k_hat` - R statistic, a measure of chain equilibrium, should be within 0.05 of 1.0 ([mc-stan.org/docs/2\\_25/cmdstan-guide/stansummary.html](http://mc-stan.org/docs/2_25/cmdstan-guide/stansummary.html)).
- `v_lower` - Lower (2.5%) quantile of the posterior distribution for a variability (v) parameter
- `v_effect` - Mean of the posterior distribution for a variability (v) parameter
- `v_upper` - Upper (97.5%) quantile of the posterior distribution for a variability (v) parameter
- `v_pH0` - Probability of the null hypothesis (no difference) for a variability (v). This is not a p-value.
- `v_FDR` - False-discovery rate of the null hypothesis (no difference), for a variability (v).
- `v_n_eff` - Effective sample size for a variability (v) parameter - the number of independent draws in the sample, the higher the better ([mc-stan.org/docs/2\\_25/cmdstan-guide/stansummary.html](http://mc-stan.org/docs/2_25/cmdstan-guide/stansummary.html)).
- `v_R_k_hat` - R statistic for a variability (v) parameter, a measure of chain equilibrium, should be within 0.05 of 1.0 ([mc-stan.org/docs/2\\_25/cmdstan-guide/stansummary.html](http://mc-stan.org/docs/2_25/cmdstan-guide/stansummary.html)).
- `count_data` Nested input count data.

**Examples**

```

data("counts_obj")

estimate =
  scomp_glm(
    counts_obj ,
    ~ type,
    ~1,
    sample,
    cell_group,
    count,
    check_outliers = FALSE,
    cores = 1
  )

```

---

scomp\_predict

*scomp\_predict*


---

**Description**

This function replicates counts from a real-world dataset.

**Usage**

```

scomp_predict(
  fit,
  formula_composition = NULL,
  new_data = NULL,
  number_of_draws = 500,
  mcmc_seed = sample(1e+05, 1)
)

```

**Arguments**

<code>fit</code>	The result of <code>scomp_glm</code> .
<code>formula_composition</code>	A formula. The formula describing the model for differential abundance, for example <code>~treatment</code> . This formula can be a sub-formula of your estimated model; in this case all other factor will be factored out.
<code>new_data</code>	A sample-wise data frame including the column that represent the factors in your formula. If you want to predict proportions for 10 samples, there should be 10 rows. T
<code>number_of_draws</code>	An integer. How may copies of the data you want to draw from the model joint posterior distribution.
<code>mcmc_seed</code>	An integer. Used for Markov-chain Monte Carlo reproducibility. By default a random number is sampled from 1 to 999999. This itself can be controlled by <code>set.seed()</code>

**Value**

A nested tibble tbl with cell\_group-wise statistics

**Examples**

```
data("counts_obj")

if(.Platform$OS.type == "unix")
  scomp_glm(
    counts_obj ,
    ~ type, ~1, sample, cell_group, count,
    approximate_posterior_inference = "all",
    check_outliers = FALSE,
    cores = 1
  ) |>

scomp_predict()
```

---

scomp\_replicate      *scomp\_replicate*

---

**Description**

This function replicates counts from a real-world dataset.

**Usage**

```
scomp_replicate(
  fit,
  formula_composition = NULL,
  formula_variability = NULL,
  number_of_draws = 1,
  mcmc_seed = sample(1e+05, 1)
)
```

**Arguments**

**fit**                    The result of scomp\_glm.

**formula\_composition**  
A formula. The formula describing the model for differential abundance, for example ~treatment. This formula can be a sub-formula of your estimated model; in this case all other factor will be factored out.

**formula\_variability**  
A formula. The formula describing the model for differential variability, for example ~treatment. In most cases, if differentially variability is of interest, the formula should only include the factor of interest as a large amount of data is needed to define variability depending to each factors. This formula can be a

	sub-formula of your estimated model; in this case all other factor will be factored out.
number_of_draws	An integer. How may copies of the data you want to draw from the model joint posterior distribution.
mcmc_seed	An integer. Used for Markov-chain Monte Carlo reproducibility. By default a random number is sampled from 1 to 999999. This itself can be controlled by <code>set.seed()</code>

**Value**

A nested tibble tbl with cell\_group-wise statistics

**Examples**

```
data("counts_obj")

if(.Platform$OS.type == "unix")
  sccomp_glm(
    counts_obj ,
    ~ type, ~1, sample, cell_group, count,
    approximate_posterior_inference = "all",
    check_outliers = FALSE,
    cores = 1
  ) |>

  sccomp_replicate()
```

---

sce\_obj

*sce\_obj*

---

**Description**

Example SingleCellExperiment data set. SingleCellExperiment data objects can be directly used with `sccomp_glm` function.

**Usage**

```
data(sce_obj)
```

**Format**

A SingleCellExperiment object. SingleCellExperiment data objects can be directly used with `sccomp_glm` function.

---

seurat_obj	<i>seurat_obj</i>
------------	-------------------

---

**Description**

Example Seurat data set. Seurat data objects can be directly used with `scomp_glm` function.

**Usage**

```
data(seurat_obj)
```

**Format**

A Seurat object

---

simulate_data	<i>simulate_data</i>
---------------	----------------------

---

**Description**

This function simulates counts from a linear model.

**Usage**

```
simulate_data(
  .data,
  .estimate_object,
  formula_composition,
  formula_variability = NULL,
  .sample = NULL,
  .cell_group = NULL,
  .coefficients = NULL,
  variability_multiplier = 5,
  number_of_draws = 1,
  mcmc_seed = sample(1e+05, 1)
)
```

**Arguments**

<code>.data</code>	A tibble including a <code>cell_group</code> name column   sample name column   read counts column   factor columns   Pvalue column   a significance column
<code>.estimate_object</code>	The result of <code>scomp_glm</code> execution. This is used for sampling from real-data properties.

<code>formula_composition</code>	A formula. The sample formula used to perform the differential cell_group abundance analysis
<code>formula_variability</code>	A formula. The formula describing the model for differential variability, for example <code>~treatment</code> . In most cases, if differentially variability is of interest, the formula should only include the factor of interest as a large amount of data is needed to define variability depending to each factors.
<code>.sample</code>	A column name as symbol. The sample identifier
<code>.cell_group</code>	A column name as symbol. The cell_group identifier
<code>.coefficients</code>	The column names for coefficients, for example, <code>c(b_0, b_1)</code>
<code>variability_multiplier</code>	A real scalar. This can be used for artificially increasing the variability of the simulation for benchmarking purposes.
<code>number_of_draws</code>	An integer. How may copies of the data you want to draw from the model joint posterior distribution.
<code>mcmc_seed</code>	An integer. Used for Markov-chain Monte Carlo reproducibility. By default a random number is sampled from 1 to 999999. This itself can be controlled by <code>set.seed()</code>

**Value**

A nested tibble tbl with cell\_group-wise statistics

**Examples**

```
data("counts_obj")
library(dplyr)

estimate =
  sccomp_glm(
    counts_obj ,
    ~ type, ~1, sample, cell_group, count,
    approximate_posterior_inference = "all",
    check_outliers = FALSE,
    cores = 1
  )

# Set coefficients for cell_groups. In this case all coefficients are 0 for simplicity.
counts_obj = counts_obj |> mutate(b_0 = 0, b_1 = 0)
# Simulate data
simulate_data(counts_obj, estimate, ~type, ~1, sample, cell_group, c(b_0, b_1))
```



---

test_contrasts	<i>test_contrasts</i>
----------------	-----------------------

---

### Description

This function test ocntrasts from a sccomp result.

### Usage

```
test_contrasts(
  .data,
  contrasts = NULL,
  percent_false_positive = 5,
  test_composition_above_logit_fold_change = 0.2,
  pass_fit = TRUE
)
```

### Arguments

<code>.data</code>	A tibble. The result of <code>sccomp_glm</code> .
<code>contrasts</code>	A vector of character strings. For example if your formula is <code>~ 0 + treatment</code> and the factor <code>treatment</code> has values <code>yes</code> and <code>no</code> , your contrast could be <code>"contrasts = c(treatmentyes - treatmentno)"</code> .
<code>percent_false_positive</code>	A real between 0 and 100 non included. This used to identify outliers with a specific false positive rate.
<code>test_composition_above_logit_fold_change</code>	A positive integer. It is the effect threshold used for the hypothesis test. A value of 0.2 correspond to a change in cell proportion of 10% for a cell type with baseline proportion of 50%. That is, a cell type goes from 45% to 50%. When the baseline proportion is closer to 0 or 1 this effect thrshold has consistent value in the logit unconstrained scale.
<code>pass_fit</code>	A boolean. Whether to pass the Stan fit as attribute in the output. Because the Stan fit can be very large, setting this to <code>FALSE</code> can be used to lower the memory imprint to save the output.

### Value

A nested tibble `tbl` with `cell_group`-wise statistics

### Examples

```
data("counts_obj")

estimates =
  sccomp_glm(
    counts_obj ,
```

```
~ 0 + type, ~1, sample, cell_group, count,  
  check_outliers = FALSE,  
  cores = 1  
) |>  
  
test_contrasts("typecancer - typebenign")
```

# Index

## \* datasets

counts\_obj, 3  
sce\_obj, 14  
seurat\_obj, 15

## \* internal

glm\_dirichlet\_multinomial, 3  
glm\_dirichlet\_multinomial\_generate\_quantities,  
4  
glm\_dirichlet\_multinomial\_imputation,  
4  
glm\_multi\_beta, 5  
glm\_multi\_beta\_generate\_data, 5

counts\_obj, 3

glm\_dirichlet\_multinomial, 3  
glm\_dirichlet\_multinomial\_generate\_quantities,  
4  
glm\_dirichlet\_multinomial\_imputation,  
4  
glm\_multi\_beta, 5  
glm\_multi\_beta\_generate\_data, 5

multi\_beta\_glm, 6

plot\_summary, 7

remove\_unwanted\_variation, 7

sccomp (sccomp-package), 3  
sccomp-package, 3  
sccomp\_glm, 8  
sccomp\_predict, 12  
sccomp\_replicate, 13  
sce\_obj, 14  
seurat\_obj, 15  
simulate\_data, 15

test\_contrasts, 17