# Package 'DamageDetective'

July 21, 2025

**Type** Package

**Title** Detecting Damaged Cells in Single-Cell RNA Sequencing Data

**Version** 1.0.0

**Description** Detects and filters damaged cells in single-cell RNA sequencing (scRNA-seq) data using a novel approach inspired by 'DoubletFinder'. Damage is detected by measuring the extent to which cells deviate from artificially damaged profiles of themselves, simulated through the probabilistic escape of cytoplasmic RNA. As output, a damage score ranging from 0 to 1 is given for each cell providing an intuitive scale for filtering that is standardised across cell types, samples, and experiments.

**License** AGPL (>= 3)

**Encoding** UTF-8

**Language** en-US

**LazyData** false

**RoxygenNote** 7.3.2

**Depends** R (>= 4.4.0)

**Imports** cowplot, dplyr, ggplot2, ggpubr, Matrix, patchwork, scales, RcppHNSW, rlang, tidyr, withr

**Suggests** Seurat, knitr, rmarkdown, spelling

**VignetteBuilder** knitr

**URL** https://alicenjoyhenning.github.io/DamageDetective/, https://github.com/alicenjoyhenning/DamageDetective

**BugReports** https://github.com/alicenjoyhenning/DamageDetective/issues

**NeedsCompilation** no

**Author** Alicen Henning [aut, cre, cph] (ORCID: <https://orcid.org/0009-0004-0535-1624>)

**Maintainer** Alicen Henning <alicen.jhb@gmail.com>

**Repository** CRAN

**Date/Publication** 2025-04-03 16:20:02 UTC

# Contents



---

`detect_damage` *detect_damage*

---

## Description

Quality control function to identify and filter damaged cells from an input count matrix, where 'damage' is defined by the loss of cytoplasmic RNA.

## Usage

```
detect_damage(
  count_matrix,
  ribosome_penalty = 0.01,
  organism = "Hsap",
  annotated_celltypes = FALSE,
  target_damage = c(0.1, 0.8),
  damage_distribution = "right_skewed",
  distribution_steepness = "moderate",
  beta_shape_parameters = NULL,
  damage_levels = 5,
  damage_proportion = 0.15,
  seed = 7,
  mito_quantile = 0.75,
  kN = NULL,
  generate_plot = TRUE,
  display_plot = TRUE,
  palette = c("grey", "#7023FD", "#E60006"),
  filter_threshold = 0.7,
  filter_counts = FALSE,
  verbose = TRUE
)
```

## Arguments

`count_matrix` Matrix or dgCMatrix containing the counts from single cell RNA sequencing data.

`ribosome_penalty` Numeric specifying the factor by which the probability of loosing a transcript from a ribosomal gene is multiplied by. Here, values closer to 0 represent a greater penalty.

- Default is 0.01.

`organism` String specifying the organism of origin of the input data where there are two standard options,

- "Hsap"
- "Mmus"

If a user wishes to use a non-standard organism they must input a list containing strings for the patterns to match mitochondrial and ribosomal genes of the organism. If available, nuclear-encoded genes that are likely retained in the nucleus, such as in nuclear speckles, must also be specified. An example for humans is below,

- organism = c(mito_pattern = "^MT-", ribo_pattern = "^(RPS|RPL)", nuclear <- c("NEAT1","XIST", "MALAT1")
- Default is "Hsap"

`annotated_celltypes`
Boolean specifying whether input matrix has cell type information stored.

- Default is FALSE

`target_damage` Numeric vector specifying the upper and lower range of the level of damage that will be introduced.

Here, damage refers to the amount of cytoplasmic RNA lost by a cell where values closer to 1 indicate more loss and therefore more heavily damaged cells.

- Default is c(0.1, 0.8)

`damage_distribution`
String specifying whether the distribution of damage levels among the damaged cells should be shifted towards the upper or lower range of damage specified in 'target_damage' or follow a symmetric distribution between them. There are three valid options:

- "right_skewed"
- "left_skewed"
- "symmetric"
- Default is "right_skewed"

`distribution_steepness`
String specifying how concentrated the spread of damaged cells are about the mean of the target distribution specified in 'target_damage'. Here, an increase in steepness manifests in a more apparent skewness.There are three valid options:

- "shallow"
- "moderate"
- "steep"
- Default is "moderate"

`beta_shape_parameters`
Numeric vector that allows for the shape parameters of the beta distribution to defined explicitly. This offers greater flexibility than allowed by the 'damage_distribution' and 'distribution_steepness' parameters and will override the defaults they offer.

- Default is 'NULL'

`damage_levels` Numeric specifying the number of distinct sets of artificial damaged cells simulated, each with a defined range of loss. Default ptions include,

- 3 : c(0.00001, 0.08), c(0.1, 0.4), c(0.5, 0.9)
- 5 : c(0.00001, 0.08), c(0.1, 0.3), c(0.3, 0.5), c(0.5, 0.7), c(0.7, 0.9)
- 7 : c(0.00001, 0.08), c(0.1, 0.3), c(0.3, 0.4), c(0.4, 0.5), c(0.5, 0.7), c(0.7, 0.9), c(0.9, 0.99999).

A user can also provide a list specifying sets with their own ranges of loss,

- damage_levels = list( pANN_50 = c(0.1, 0.5), pANN_100 = c(0.5, 1) )

By introducing more sets of damage a user can improve the accuracy of loss estimations (scaled_pANN) as they are found through scaling the pANN within each set according to the lower and upper boundary of the set's damage level. However, introducing more sets increases the computational time for the function.

- Default is 5.

`damage_proportion`
Numeric describing what proportion of the input data should be altered to resemble damaged data.

- Must range between 0 and 1.

`seed` Numeric specifying the random seed to ensure reproducibility of the function's output. Setting a seed ensures that the random sampling and perturbation processes produce the same results when the function is run multiple times with the same input data and parameters.

- Default is 7.

`mito_quantile` Numeric between 0 and 1 specifying below what level of mitochondrial proportion cells are sampled for simulations. This step is done to protect against simulating damaged cell profiles from cells that are likely damaged.

- Default is 0.75.

`kN` Numeric describing how many nearest neighbours are considered for pANN calculations. kN cannot exceed the total cell number.

- Default is one third of the total cell number.

`generate_plot` Boolean specifying whether the QC plot should be outputted. QC plots will be generated by default as we recommend verifying the perturbed data retains characteristics of true single cell data.

- Default is TRUE.

`display_plot` Boolean specifying whether the output QC plot should be displayed in the global environment. Naturally, this is only relevant when generate_plot is TRUE.

- Default is TRUE.

`palette` String specifying the three colours that will be used to create the continuous colour palette for colouring the 'damage_column'.

- Default is a range from purple to red, c("grey", "#7023FD", "#E60006").

`filter_threshold`
Numeric specifying the proportion of RNA loss above which a cell should be considered damaged.

- Default is 0.75.

`filter_counts` Boolean specifying whether the output matrix should be filtered, returned containing only cells that fall below the filter threshold. Alternatively, a data frame containing cell barcodes and their associated label as either 'damaged' or 'cell' is returned.

- Default is FALSE.

`verbose` Boolean specifying whether messages and function progress should be displayed in the console.

- Default is TRUE.

## Details

Using the simulation framework of `simulate_counts()`, `detect_damage()` generates artificially damaged cell profiles by introducing defined levels of RNA loss into the input data. True and artificial cells are then merged and pre-processed to compute the following quality control metrics:

- Log-normalized feature count
- Log-normalized total counts
- Mitochondrial proportion
- Ribosomal proportion
- Log-normalized MALAT1 gene expression

Principal component analysis (PCA) is performed on these metrics, and a Euclidean distance matrix is constructed from the PC embeddings. For each true cell, the proportion of nearest neighbours that are artificial cells (pANN) is calculated across all damage levels and the damage level with the highest pANN is assigned to the true cell. Finally, cells exceeding a specified damage threshold, `filter_threshold`, are marked as damaged.

This filtering method is inspired by approaches developed for DoubletFinder (McGinnis et al., 2019) to detect doublets in single-cell data.

## Value

Filtered matrix or data frame containing damage labels.

## References

McGinnis, C. S., Murrow, L. M., & Gartner, Z. J. (2019). DoubletFinder: Doublet Detection in Single-Cell RNA Sequencing Data Using Artificial Nearest neighbours. *Cell Systems, 8*(4), 329-337.e4. doi:10.1016/j.cels.2019.03.003

## Examples

```
data("test_counts", package = "DamageDetective")

test <- detect_damage(
  count_matrix = test_counts,
  ribosome_penalty = 0.001,
  damage_levels = 3,
```

```
  damage_proportion = 0.1,
  generate_plot = FALSE,
  seed = 7
)
```

---

`select_penalty` *select_penalty*

---

## Description

Recommended prerequisite function to detect_damage() that estimates the ideal `ribosome_penalty` value for the input data.

## Usage

```
select_penalty(
  count_matrix,
  organism = "Hsap",
  mito_quantile = 0.75,
  penalty_range = c(1e-05, 0.5),
  penalty_step = 0.005,
  max_penalty_trials = 10,
  target_damage = c(0.2, 0.99),
  damage_distribution = "right_skewed",
  distribution_steepness = "steep",
  beta_shape_parameters = NULL,
  stability_limit = 3,
  damage_proportion = 0.15,
  annotated_celltypes = FALSE,
  return_output = "penalty",
  ribosome_penalty = NULL,
  seed = NULL,
  verbose = TRUE
)
```

## Arguments

| | |
|---|---|
| `count_matrix` | Matrix or dgCMatrix containing the counts from single cell RNA sequencing data. |
| `organism` | String specifying the organism of origin of the input data where there are two standard options,<br>• "Hsap"<br>• "Mmus"<br>If a user wishes to use a non-standard organism they must input a list containing strings for the patterns to match mitochondrial and ribosomal genes of the organism. If available, nuclear-encoded genes that are likely retained in the nucleus, such as in nuclear speckles, must also be specified. An example for humans is below, |

- organism = c(mito_pattern = "^MT-", ribo_pattern = "^(RPS|RPL)", nuclear <- c("NEAT1","XIST", "MALAT1")
- Default is "Hsap"

`mito_quantile` Numeric specifying below what proportion of mitochondrial content cells are used for sampling for simulation.

- Default is 0.75, meaning only cells with less than 0.75 proportion of mitochondrial counts are sampled for simulated.

`penalty_range` Numerical vector of length 2 specifying the lower and upper limit of values tested for the ribosomal penalty.

- Default is c(0.00001, 0.5).

`penalty_step` Numeric specifying the value added to each increment of penalty tested.

- Default is 0.005.

`max_penalty_trials`
Numeric specifying the maximum number of iterations for the ribosomal penalty value.

- Default is 10.

`target_damage` Numeric vector specifying the upper and lower range of the level of damage that will be introduced.

Here, damage refers to the amount of cytoplasmic RNA lost by a cell where values closer to 1 indicate more loss and therefore more heavily damaged cells.

- Default is c(0.1, 0.8)

`damage_distribution`
String specifying whether the distribution of damage levels among the damaged cells should be shifted towards the upper or lower range of damage specified in 'target_damage' or follow a symmetric distribution between them. There are three valid options:

- "right_skewed"
- "left_skewed"
- "symmetric"
- Default is "right_skewed"

`distribution_steepness`
String specifying how concentrated the spread of damaged cells are about the mean of the target distribution specified in 'target_damage'. Here, an increase in steepness manifests in a more apparent skewness.There are three valid options:

- "shallow"
- "moderate"
- "steep"
- Default is "moderate"

`beta_shape_parameters`
Numeric vector that allows for the shape parameters of the beta distribution to defined explicitly. This offers greater flexibility than allowed by the 'damage_distribution' and 'distribution_steepness' parameters and will override the defaults they offer.

- Default is 'NULL'

`stability_limit`

Numeric specifying the number of additional iterations allotted after the median minimum distance of the artificial cells to the true cells is greater than the previous minimum distance.

The idea here is that if a higher penalty is not causing an improvement in the output, there is little need to continue testing with larger penalties.

- Default is 3.

`damage_proportion`

Numeric describing what proportion of the input data should be altered to resemble damaged data.

- Must range between 0 and 1.

`annotated_celltypes`

Boolean specifying whether input matrix has cell type information stored.

- Default is FALSE

`return_output` String specifying what form the output of the function should take where the options are either,

- "penalty"
- "full"

"Penalty" will return only the ribosomal penalty that resulted in the best performance (the smallest median distance between artificial and true cells). While "full" will return the ideal ribosomal penalty and the median distance between artificial and true cells for each penalty tested. This allows insight into how the penalty was selected.

- Default is "penalty".

`ribosome_penalty`

Numeric specifying the factor by which the probability of loosing a transcript from a ribosomal gene is multiplied by. Here, values closer to 0 represent a greater penalty.

- Default is 0.01.

`seed` Numeric specifying the random seed to ensure reproducibility of the function's output. Setting a seed ensures that the random sampling and perturbation processes produce the same results when the function is run multiple times with the same input data and parameters.

- Default is 7.

`verbose` Boolean specifying whether messages and function progress should be displayed in the console.

- Default is TRUE.

## Details

Based on observations of true single cell data, we find that ribosomal RNA loss occurs less frequently than expected based on abundance alone. To adjust for this, the probability scores of ribosomal gene loss are multiplied by a numerical value (`ribosome_penalty`) between 0 and 1. Lower

values (closer to zero) better approximate true data, with a default of 0.01, though this can often be greatly refined for the input data.

Refinement follows a similar workflow to detect_damage(), but rather than evaluating the similarity of true cells to sets of artificial cells to infer their level of damage, we evaluate the similarity of artificial cells to true cells to infer the effectiveness of their approximation to true data. This is calculated using the distance to the nearest true cell (dTNN) taken for each artificial cell found using the Euclidean distance matrix. The median dTNN is computed iteratively until stabilization or a worsening trend. The ideal `ribosomal_penalty` is then selected as that which generated the lowest dTNN.

## Value

Numeric representing the ideal ribosomal penalty for an input dataset.

## Examples

```
data("test_counts", package = "DamageDetective")

penalty <- select_penalty(
 count_matrix = test_counts,
 stability_limit = 1,
 max_penalty_trials = 1,
 seed = 7
)
```

---

`simulate_counts` *simulate_counts*

---

## Description

Function to simulate damaged cells by perturbing the gene expression of existing cells.

## Usage

```
simulate_counts(
  count_matrix,
  damage_proportion,
  annotated_celltypes = FALSE,
  target_damage = c(0.1, 0.8),
  damage_distribution = "right_skewed",
  distribution_steepness = "moderate",
  beta_shape_parameters = NULL,
  ribosome_penalty = 0.001,
  generate_plot = TRUE,
  palette = c("grey", "#7023FD", "#E60006"),
  plot_ribosomal_penalty = FALSE,
  display_plot = TRUE,
  seed = NULL,
```

```
  organism = "Hsap"
)
```

## Arguments

| | |
|---|---|
| count_matrix | Matrix or dgCMatrix containing the counts from single cell RNA sequencing data. |
| damage_proportion | Numeric describing what proportion of the input data should be altered to resemble damaged data.<br>• Must range between 0 and 1. |
| annotated_celltypes | Boolean specifying whether input matrix has cell type information stored.<br>• Default is FALSE |
| target_damage | Numeric vector specifying the upper and lower range of the level of damage that will be introduced.<br>Here, damage refers to the amount of cytoplasmic RNA lost by a cell where values closer to 1 indicate more loss and therefore more heavily damaged cells.<br>• Default is c(0.1, 0.8) |
| damage_distribution | String specifying whether the distribution of damage levels among the damaged cells should be shifted towards the upper or lower range of damage specified in 'target_damage' or follow a symmetric distribution between them. There are three valid options:<br>• "right_skewed"<br>• "left_skewed"<br>• "symmetric"<br>• Default is "right_skewed" |
| distribution_steepness | String specifying how concentrated the spread of damaged cells are about the mean of the target distribution specified in 'target_damage'. Here, an increase in steepness manifests in a more apparent skewness.There are three valid options:<br>• "shallow"<br>• "moderate"<br>• "steep"<br>• Default is "moderate" |
| beta_shape_parameters | Numeric vector that allows for the shape parameters of the beta distribution to defined explicitly. This offers greater flexibility than allowed by the 'damage_distribution' and 'distribution_steepness' parameters and will override the defaults they offer.<br>• Default is 'NULL' |
| ribosome_penalty | Numeric specifying the factor by which the probability of loosing a transcript from a ribosomal gene is multiplied by. Here, values closer to 0 represent a greater penalty. |

- Default is 0.01.

`generate_plot` Boolean specifying whether the QC plot should be outputted. QC plots will be generated by default as we recommend verifying the perturbed data retains characteristics of true single cell data.

- Default is TRUE.

`palette` Character vector containing three colours to create the continuous palette for damaged cells.

- Default is c("grey", "#7023FD", "#E60006").

`plot_ribosomal_penalty`
Boolean specifying whether the output QC plot should focus on only the ribosomal proportion or contain additional QC information. If TRUE, this can be useful for visualising the impact of the ribosomal penalty parameter.

- Default is FALSE.

`display_plot` Boolean specifying whether the output QC plot should be displayed in the global environment. Naturally, this is only relevant when generate_plot is TRUE.

- Default is TRUE.

`seed` Numeric specifying the random seed to ensure reproducibility of the function's output. Setting a seed ensures that the random sampling and perturbation processes produce the same results when the function is run multiple times with the same input data and parameters.

- Default is 7.

`organism` String specifying the organism of origin of the input data where there are two standard options,

- "Hsap"
- "Mmus"

If a user wishes to use a non-standard organism they must input a list containing strings for the patterns to match mitochondrial and ribosomal genes of the organism. If available, nuclear-encoded genes that are likely retained in the nucleus, such as in nuclear speckles, must also be specified. An example for humans is below,

- organism = c(mito_pattern = "^MT-", ribo_pattern = "^(RPS|RPL)", nuclear <- c("NEAT1","XIST", "MALAT1")
- Default is "Hsap"

## Details

'DamageDetective' models damage in single-cell RNA sequencing data as the loss of cytoplasmic RNA, where cells experiencing greater RNA loss are assumed to be more extensively damaged, while those with minimal loss are considered largely intact. The perturbation process introduces RNA loss into existing cells and is controlled by three key parameters: the **target proportion of damage**, which specifies the fraction of cells to be perturbed; the **target level of damage**, which defines the extent of RNA loss across cells; and the **target distribution of damage**, which determines how the different levels of RNA loss are distributed across cells.

Based on these parameters, cells are randomly selected and assigned a target proportion of RNA loss. The total number of transcripts to be removed is determined, and perturbation is applied through weighted sampling without replacement from cytoplasmic gene counts. Here, the probability of transcript loss is determined by gene abundance, with highly expressed genes more likely to lose transcripts. Once the target RNA loss is reached, the cell's expression profile is updated, and the process repeats for all selected cells.

## Value

A list containing the altered count matrix, a data frame with summary statistics, and, if specified, a 'ggplot2' object of the quality control metrics of the alteration.

## Examples

```
data("test_counts", package = "DamageDetective")

simulated_damage <- simulate_counts(
  count_matrix = test_counts,
  damage_proportion = 0.1,
  ribosome_penalty = 0.01,
  target_damage = c(0.5, 0.9),
  generate_plot = FALSE,
  seed = 7
)
```

---

# test_counts *Test Counts*

---

## Description

A sparse matrix of PBMC single-cell gene expression data for function testing.

## Usage

```
data(test_counts)
```

## Format

A sparse matrix with:

- 32 738 rows representing genes.
- 500 columns representing single cells.
- Stored expression values representing nonzero gene expression levels.

## Details

This dataset consists of an abridged sparse matrix originally derived from PBMC single-cell RNA sequencing data. The data was processed and stored in a sparse matrix format.

**Source**

doi:10.1038/s4159102007698

# Index