Package: fishash (via r-universe)

November 25, 2026

Title Cell Hashing with One-Sided Fisher Test

Version 1.0.0

Description Assigns guide RNAs or other genetic perturbations to cells in single-cell sequencing experiments using a one-sided Fisher's exact test. Implements an iterative refitting procedure to mitigate Simpson's paradox, supports multiple false discovery rate correction methods (Benjamini-Hochberg, Benjamini-Yekutueli, and Guo & Sarkar 2020), and provides simulation utilities for benchmarking demultiplexing methods. Results are returned as SummarizedExperiment objects.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 8.0.0

biocViews SingleCell, Sequencing, Preprocessing, StatisticalMethod

Imports dplyr, extraDistr, ggplot2, Matrix, methods, nnet, patchwork, rlang, S4Vectors, SingleCellExperiment, SummarizedExperiment, sparseMatrixStats

Depends R (>= 4.5.0)

LazyData false

Suggests BiocStyle, ComplexHeatmap, ggExtra, glmGamPoi, knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

URL https://github.com/jackkamm/fishash

BugReports https://github.com/jackkamm/fishash/issues

Config/pak/sysreqs zlib1g-dev

Repository Bioconductor 3.24

Date/Publication 2026-11-23 22:56:38 UTC

RemoteUrl https://github.com/bioc/fishash

RemoteRef RELEASE_3_24

RemoteSha a925ab1bd9c962d80659e8ee2ff065057730dab2

Contents

diagonal_heatmap 2
fishash 3
impute_masked_counts 4
nonzero_histogram_with_weighted 5
simulate_guidebender 6
simulate_guidebender2 8
sort_columns_by_top_row 9
tapseq_diffex 10

Index **11**

`diagonal_heatmap` *Plot guide count matrix as an approximately diagonal heatmap*

Description

This function plots a heatmap after permuting the matrix columns to "look diagonal" by sorting them according to their largest element. The function is geared towards sparse count matrices (it only draws the nonzero elements).

Usage

```
diagonal_heatmap(
  counts,
  subsample_cells = 1000,
  max_guides = NULL,
  rownames_size = 8,
  plot = TRUE
)
```

Arguments

`counts`	Matrix of counts, cells are columns and features (guides) are rows
`subsample_cells`	If smaller than `ncol(counts)`, subsamples the columns
`max_guides`	If non-NULL and smaller than `nrow(counts)`, subsets to the top guides by total UMI count.
`rownames_size`	Scales the font size of the row (feature) names
`plot`	If TRUE return a plot from ggplot2; if FALSE return the dataframe used to produce the plot.

Value

Either a ggplot2 object or the dataframe used to produce it, depending on the `plot` argument.

See Also

Uses `sort_columns_by_top_row()` to obtain the permuted matrix. Originally proposed by pertpy.

Examples

```
data(tapseq_diffex)
library(SingleCellExperiment)
diagonal_heatmap(counts(altExp(tapseq_diffex)))
```

---

fishash *Cell hashing by one-sided Fisher test*

Description

Cell hashing by one-sided Fisher test

Usage

```
fishash(
  counts,
  padj_cutoff = 0.05,
  padj_method = c("GS", "BY", "BH"),
  min_count = 2,
  min_frac = 0,
  refit = 10,
  exclude_empty = TRUE,
  background = NULL
)
```

Arguments

`counts`	Matrix of counts. Typically, columns are cells and rows are features (e.g. gRNAs).
`padj_cutoff`	Threshold for false discovery rate
`padj_method`	FDR correction type. "BH" for Benjamini-Hochberg "BY" for Benjamini-Yekutueli, "GS" for Guo & Sarkar 2020 assuming independence across cells (and arbitrary dependence within).
`min_count`	Minimum number of counts to call a feature present.
`min_frac`	Minimum fraction of counts within a cell to call a feature present.

refit	Whether to iteratively refit the model, re-estimating the background vs signal counts from the previous iteration. If 0, no refitting is done; if a positive integer, reruns with that many iterations (or until convergence, whichever is first). The significant entries from the previous iteration are subtracted from the counts to get a matrix of "background noise" counts, which are then used for non-cell entries of the 2x2 table. This may mitigate effects from Simpson's paradox.
exclude_empty	If TRUE, rows and columns that have 0 counts are ignored for multiple testing correction.
background	Optional matrix of background noise counts, e.g. estimated from demuxEM, which are used for non-cell entries of the 2x2 table.

Value

A SummarizedExperiment. The colData has columns for whether the cell is singlet, doublet, or unassigned ("demux_type"), and the assignment as a comma-delimited string ("assignment"). The metadata contains an entry for the p-value cutoff at the given FDR level ("log_pval_cutoff"). The SummarizedExperiment contains assays for the assignment matrix ("assigned"), the log-p-value ("log_pval"), the odds-ratio ("odds_ratio"), and a regularized version of the odds-ratio that avoids dividing by 0 by adding a pseudocount to the off-diagonal entries of the 2x2 contingency table ("odds_ratio_regularized"). Note the "odds_ratio_regularized" assay is EXPERIMENTAL and subject to change.

Examples

```
data(tapseq_diffex)
library(SingleCellExperiment)
result <- fishash(counts(altExp(tapseq_diffex)))
```

---

impute_masked_counts *For a matrix of counts, impute masked entries via an alternating algorithm*

Description

More specifically, we perform rank-1 matrix completion on the masked counts optimizing the Poisson likelihood of the unmasked values.

Usage

```
impute_masked_counts(counts, mask, eps = 1e-04, max_iter = 10, verbose = FALSE)
```

Arguments

counts	Matrix of counts
mask	Matrix of 0/1 or TRUE/FALSE indicating which entries to mask
eps	Stops when the relative change in column and row factors less than this
max_iter	Maximum iterations for the alternating algorithm
verbose	Whether to produce verbose output

Value

A matrix of the same dimensions as `counts` with masked entries replaced by their rank-1 Poisson imputed values.

Examples

```
counts <- Matrix::Matrix(c(1, 0, 2, 3, 0, 1, 0, 4, 2),
                         nrow = 3, sparse = TRUE)
mask <- Matrix::Matrix(c(0, 1, 0, 0, 0, 1, 0, 0, 1),
                       nrow = 3, sparse = TRUE)
impute_masked_counts(counts, mask)
```

---

nonzero_histogram_with_weighted

Plot histogram and weighted histogram of UMI counts

Description

Plot histogram and weighted histogram of UMI counts

Usage

```
nonzero_histogram_with_weighted(counts, bins = 50, assigned = NULL)
```

Arguments

`counts`	Matrix of counts, columns are cells and rows are features.
`bins`	Number of bins for histogram
`assigned`	Optional matrix of assigned entries, or SummarizedExperiment returned by `fishash()`. If non-NULL, the histogram is colored by guide assignment status.

Value

A patchwork object.

Examples

```
data(tapseq_diffex)
library(SingleCellExperiment)
nonzero_histogram_with_weighted(counts(altExp(tapseq_diffex)))
```

`simulate_guidebender` *Simulate guide count matrix based on a cellbender-like model*

Description

Simulate guide count matrix based on a cellbender-like model

Usage

```
simulate_guidebender(
  n_guides,
  n_cells,
  moi,
  hurdle_prob,
  guide_infection_alpha = 1,
  d_sigma_guide = 0.5,
  endo_shape_sum = 1,
  endo_shape_flat = 0,
  d_mu_drop = log(10),
  d_sigma_drop = 0.5,
  d_mu_cell = log(60),
  d_sigma_cell = 0.5,
  rho_alpha = 0.5,
  rho_beta = 9.5,
  eps_alpha = 50,
  Phi_cell = 1,
  Phi_noise = 0,
  return_sparse_only = FALSE,
  median_per_cell = NULL,
  chunk_cells = NULL
)
```

Arguments

`n_guides`	Number of guides. Must be strictly greater than 1.
`n_cells`	Number of cells. Must be strictly greater than 1.
`moi`	The pre-selection multiplicity of infection. The Poisson-rate in the hurdle Poisson for the number of guides per cell.
`hurdle_prob`	The probability a cell has zero guides; the hurdle in the hurdle-Poisson for the number of guides per cell.
`guide_infection_alpha`	The guide infection relative frequencies (`p_g`) is Dirichlet-distributed with shape `guide_infection_alpha`.
`d_sigma_guide`	The guide expression size factor (`d_g^{guide}`) is lognormal with mean 0 and SD `d_sigma_guide`

endo_shape_sum The endogenouse noise guide frequencies \chi_g^a is Dirichlet distributed; its shape parameters sum to endo_shape_sum * n_guides.

endo_shape_flat
If this is 1, then the endogenous noise guide frequencies \chi_g^a are Dirichlet distributed with constant shape parameter equal to endo_shape_sum. If this is 0, then the Dirichlet shape parameters are instead proportional to d_g^{guide} * p_g, and sum to endo_shape_sum * n_guides. If this is between 0 and 1, it interpolates between these 2 options.

d_mu_drop The droplet ambient size factor d_n^{drop} is lognormal with location d_mu_drop.

d_sigma_drop The droplet ambient size factor d_n^{drop} is lognormal with scale d_sigma_drop.

d_mu_cell The cell size factor d_n^{cell} is lognormal with location d_mu_cell.

d_sigma_cell The cell size factor d_n^{cell} is lognormal with scale d_sigma_cell.

rho_alpha Exogeneous noise frequency (i.e. from PCR chimeras) is Beta(rho_alpha, rho_beta).

rho_beta Exogeneous noise frequency (i.e. from PCR chimeras) is Beta(rho_alpha, rho_beta).

eps_alpha Droplet-specificy capture efficiency param \epsilon_n is Gamma with mean 1 and shape eps_alpha

Phi_cell The negative binomial overdispersion of the signal counts

Phi_noise The negative binomial overdispersion of the noise counts

return_sparse_only
Whether to return only the sparse matrices of counts and assignments, or whether to also return the dense matrices of Poisson rates and latent odds-ratios.

median_per_cell
If non-null, the total number of counts is rescaled so that this is the median per cell.

chunk_cells Simulate in chunks, with this many cells per chunk. This option can reduce memory usage, when used in combination with return_sparse_only=TRUE. Must perfectly divide n_cells, and be strictly greater than 1.

Value

A SummarizedExperiment, with the following assays:

ground_truth A logical matrix indicating which cells were infected with which guides

counts The observed guide counts

counts_signal The counts coming from the true signal (i.e. with noise counts subtracted).

When return_sparse_only is FALSE, the following assays are also added:

lambd_signal The Poisson rate of the signal counts

lambd_noise The Poisson rate of the noise counts

latent_odds_ratio The true odds ratio of each cell-guide pair.

Examples

```
set.seed(123)
sim <- simulate_guidebender(n_guides = 5, n_cells = 20,
                            moi = 0.2, hurdle_prob = 0.1)
```

`simulate_guidebender2` *Alternative interface to* `simulate_guidebender` *that tries to have more interpretable input parameters, in particular: Signal to Noise Ratio, count per cell, and fraction of noise that is endogeneous vs exogeneous. These replace the following parameters from* `simulate_guidebender` *which should not be used with this function:* `d_mu_drop`, `d_mu_cell`, `rho_alpha`, `rho_beta`.

Description

Alternative interface to `simulate_guidebender` that tries to have more interpretable input parameters, in particular: Signal to Noise Ratio, count per cell, and fraction of noise that is endogeneous vs exogeneous. These replace the following parameters from `simulate_guidebender` which should not be used with this function: `d_mu_drop`, `d_mu_cell`, `rho_alpha`, `rho_beta`.

Usage

```
simulate_guidebender2(
  n_guides,
  n_cells,
  moi,
  hurdle_prob,
  snr,
  count_per_cell,
  frac_noise_endo,
  rho_sum = 10,
  d_sigma_drop = 0.5,
  d_sigma_cell = 0.5,
  d_sigma_guide = 0.5,
  use_median = TRUE,
  ...
)
```

Arguments

`n_guides`	Number of guides
`n_cells`	Number of cells
`moi`	The pre-selection multiplicity of infection. The Poisson-rate in the hurdle Poisson for the number of guides per cell.
`hurdle_prob`	The probability a cell has zero guides; the hurdle in the hurdle-Poisson for the number of guides per cell.

snr	Ratio of counts from signal to counts from noise
count_per_cell	Median or mean counts per cell. use_median determines whether this specifies the median or mean.
frac_noise_endo	Fraction of the noise that is endogeneous (ie background extracellular debris) as opposed to exogeneous (ie PCR chimeras).
rho_sum	Sum of rho_alpha and rho_beta from simulate_guidebender. Smaller values imply greater variability between cells for the fraction of their counts coming from exogenous noise (PCR chimeras).
d_sigma_drop	The droplet ambient size factor d_n^{drop} is lognormal with scale d_sigma_drop.
d_sigma_cell	The cell size factor d_n^{cell} is lognormal with scale d_sigma_cell.
d_sigma_guide	The guide expression size factor (d_g^{guide}) is lognormal with mean 0 and SD d_sigma_guide
use_median	Whether counts_per_cell refers to the median (TRUE) or mean (FALSE).
...	Additional arguments to pass to simulate_guidebender. Note the following arguments must NOT be specified: d_mu_drop, d_mu_cell, rho_alpha, rho_beta.

Value

A SummarizedExperiment as returned by simulate_guidebender.

Examples

```
set.seed(123)
sim <- simulate_guidebender2(n_guides = 5, n_cells = 20,
                             moi = 0.2, hurdle_prob = 0.1,
                             snr = 5, count_per_cell = 100,
                             frac_noise_endo = 0.5)
```

---

sort_columns_by_top_row

Sort matrix columns to place maxes near diagonal

Description

This re-orders the columns of a matrix according to the row-index of its top element (breaking ties randomly). It is useful for visualizing a guide count matrix as an approximately diagonal heatmap.

Usage

```
sort_columns_by_top_row(counts)
```

Arguments

counts	Matrix of counts, columns are cells and rows are features

Value

A matrix of the same dimension as `counts`

See Also

Used by `diagonal_heatmap()`. Originally proposed by pertpy.

Examples

```
data(tapseq_diffex)
library(SingleCellExperiment)
sort_columns_by_top_row(counts(altExp(tapseq_diffex)))
```

---

tapseq_diffex *TAPseq benchmark dataset (Schraivogel et al 2020)*

Description

TAPseq ("Targeted Perturbseq") dataset for benchmarking differential expression (Schraivogel et al 2020).

Usage

```
data(tapseq_diffex)
```

Format

`tapseq_diffex`**:**
A SingleCellExperiment with 72 rows and 21977 columns.

Source

https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE135497, https://github.com/velten-group/crispat

Index

∗ **datasets**
tapseq_diffex, 10

diagonal_heatmap, 2
diagonal_heatmap(), *10*

fishash, 3

impute_masked_counts, 4

nonzero_histogram_with_weighted, 5

simulate_guidebender, 6
simulate_guidebender2, 8
sort_columns_by_top_row, 9
sort_columns_by_top_row(), *3*

tapseq_diffex, 10