Package 'dsb'

October 13, 2022

Type Package

Title Normalize & Denoise Droplet Single Cell Protein Data (CITE-Seq)

Version 1.0.2

Description This lightweight R package provides a method for normalizing and denoising protein expression data from droplet based single cell experiments. Raw protein Unique Molecular Index (UMI) counts from sequencing DNA-conjugated antibody derived tags (ADT) in droplets (e.g. 'CITE-seq') have substantial measurement noise. Our experiments and computational modeling revealed two major components of this noise: 1) protein-specific noise originating from ambient, unbound antibody encapsulated in droplets that can be accurately inferred via the expected protein counts detected in empty droplets, and 2) droplet/cell-specific noise revealed via the shared variance component associated with isotype antibody controls and background protein counts in each cell. This package normalizes and removes both of these sources of noise from raw protein data derived from methods such as 'CITE-seq', 'REAP-seq', 'ASAP-seq', 'TEA-seq', 'proteogenomic' data from the Mission Bio platform, etc. See the vignette for tutorials on how to integrate dsb with 'Seurat' and 'Bioconductor' and how to use dsb in 'Python'. Please see our paper Mulè M.P., Martins A.J., and Tsang J.S. Nature Communications 2022 <https://www.nature.com/articles/s41467-022-29356-8> for more details on the method.

License CC0 | file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Depends R (>= 2.10)

biocViews

Imports magrittr, limma, mclust, stats

Suggests testthat, knitr, rmarkdown, ggplot2, cowplot, spelling

URL https://github.com/niaid/dsb

BugReports https://github.com/niaid/dsb/issues

VignetteBuilder knitr, rmarkdown

Language en-US

NeedsCompilation no

Author Matthew Mulè [aut, cre] (<https://orcid.org/0000-0001-8457-2716>),
Andrew Martins [aut] (<https://orcid.org/0000-0002-1832-1924>),
John Tsang [pdr] (<https://orcid.org/0000-0003-3186-3047>)

Maintainer Matthew Mulè <mattmule@gmail.com>

Repository CRAN

Date/Publication 2022-05-27 08:40:06 UTC

R topics documented:

`cells_citeseq_mtx` *small example CITE-seq protein dataset for 87 surface protein in 2872 cells*

Description

A matrix of raw UMI counts for surface proteins for surface proteins measured with CITE-seq antibodies. This data is used for example scripts in the dsb package. Raw data was processed with CITE-seq-Count.

Usage

```
cells_citeseq_mtx
```

Format

An R matrix, rows: 87 proteins, columns: 2872 cells

cells_citeseq_mtx R matrix of cells by proteins; a random distribution of a maximum of 100 cells per cluster from the 30 clusters reported in Kotliarov et. al. 2020

References

Kotliarov et. al. 2020 Nat. Medicine

DSBNormalizeProtein *DSBNormalizeProtein R function: Normalize single cell antibody derived tag (ADT) protein data. This function implements both step I (ambient protein background correction) and step II. (defining and removing cell to cell technical variation) of the dsb normalization method. See <https://www.biorxiv.org/content/10.1101/2020.02.24.963603v3> for details of the algorithm.*

Description

DSBNormalizeProtein R function: Normalize single cell antibody derived tag (ADT) protein data. This function implements both step I (ambient protein background correction) and step II. (defining and removing cell to cell technical variation) of the dsb normalization method. See <https://www.biorxiv.org/content/10.1101/ for details of the algorithm.

Usage

```
DSBNormalizeProtein(
  cell_protein_matrix,
  empty_drop_matrix,
  denoise.counts = TRUE,
  use.isotype.control = TRUE,
  isotype.control.name.vec = NULL,
  define.pseudocount = FALSE,
  pseudocount.use,
  quantile.clipping = FALSE,
  quantile.clip = c(0.001, 0.9995),
  scale.factor = c("standardize", "mean.subtract")[1],
  return.stats = FALSE
)
```

Arguments

`cell_protein_matrix`
: Raw protein ADT UMI count data to be normalized. Cells - columns Proteins (ADTs) - rows.

`empty_drop_matrix`
: Raw empty droplet / background ADT UMI count data used for background correction with Cells - columns and Proteins (ADTs) - rows. This can easily be defined from the raw_feature_bc_matrix output from Cell Ranger or other alignment tools such as kallisto and Cite-Seq-Count. See vignette.

`denoise.counts`
: Recommended function default 'denoise.counts = TRUE' and 'use.isotype.control = TRUE'. This runs step II of the dsb algorithm to define and remove cell to cell technical noise.

`use.isotype.control`
: Recommended function default 'denoise.counts = TRUE' and 'use.isotype.control = TRUE'. This includes isotype controls in defining the dsb technical component.

`isotype.control.name.vec`
: A vector of the names of the isotype control proteins in the rows of the cells and background matrix e.g. isotype.control.name.vec = c('isotype1', 'isotype2').

`define.pseudocount`
: FALSE (default) uses the value 10 optimized for protein ADT data.

`pseudocount.use`
: Must be defined if 'define.pseudocount = TRUE'. This is the pseudocount to be added to raw ADT UMI counts. Otherwise the default pseudocount used.

`quantile.clipping`
: FALSE (default), if outliers or a large range of values for some proteins are observed (e.g. -50 to 50) these are often from rare outlier cells. re-running the function with 'quantile.clipping = TRUE' will adjust by applying 0.001 and 0.998th quantile value clipping to trim values to those max and min values. If range of normalized values are still very broad and high (e.g. above 40) try setting 'scale.factor = mean.subtract'.

`quantile.clip`
: if 'quantile.clipping = TRUE', a vector of the lowest and highest quantiles to clip. These can be tuned to the dataset size. The default c(0.001, 0.9995) optimized to clip only a few of the most extreme outliers.

`scale.factor`
: one of 'standardize' or 'mean.subtract'. The recommended default 'standardize' subtracts from the cells the the background droplet matrix mean and divides by the background matrix standard deviation. Values for each protein with this method are interpretable as the number of standard deviations from the mean of the protein background distribution. If 'mean.subtract', subtract the mean without dividing by the standard deviation; can be useful if low background levels detected.

`return.stats`
: if TRUE, returns a list, element 1 $dsb_normalized_matrix is the normalized adt matrix element 2 $dsb_stats is the internal stats used by dsb during denoising (the background mean, isotype control values, and the final dsb technical component that is regressed out of the counts)

Value

Normalized ADT data are returned as a standard R "matrix" of cells (columns), proteins (rows) that can be added to Seurat, SingleCellExperiment or python anndata object - see vignette. If return.stats = TRUE, function returns a list: x$dsb_normalized_matrix normalized matrix, x$protein_stats are mean and sd of log transformed cell, background and the dsb normalized values (as list). x$technical_stats includes the dsb technical component value for each cell and each variable used to calculate the technical component.

Author(s)

Matthew P. Mulè, <mattmule@gmail.com>

References

https://doi.org/10.1101/2020.02.24.963603

Examples

```
library(dsb) # load example data cells_citeseq_mtx and empty_drop_matrix included in package

# use a subset of cells and background droplets from example data
cells_citeseq_mtx = cells_citeseq_mtx[ ,1:400]
empty_drop_matrix = empty_drop_citeseq_mtx[ ,1:400]

# example I
adt_norm = dsb::DSBNormalizeProtein(
  # step I: remove ambient protein noise reflected in counts from empty droplets
  cell_protein_matrix = cells_citeseq_mtx,
  empty_drop_matrix = empty_drop_matrix,

 # recommended step II: model and remove the technical component of each cell's protein data
  denoise.counts = TRUE,
  use.isotype.control = TRUE,
  isotype.control.name.vec = rownames(cells_citeseq_mtx)[67:70]
)

# example II - experiments without isotype controls
adt_norm = dsb::DSBNormalizeProtein(
  cell_protein_matrix = cells_citeseq_mtx,
  empty_drop_matrix = empty_drop_matrix,
  denoise.counts = FALSE
)

# example III - return dsb internal stats used during denoising for each cell
# returns a 2 element list - the normalized matrix and the internal stats
dsb_output = dsb::DSBNormalizeProtein(
   cell_protein_matrix = cells_citeseq_mtx,
   empty_drop_matrix = empty_drop_matrix,
   isotype.control.name.vec = rownames(cells_citeseq_mtx)[67:70],
   return.stats = TRUE
)

# the dsb normalized matrix to be used in downstream analysis is dsb_output$dsb_normalized_matrix
# protein level stats are in dsb_output$protein_stats
# cell-level stats are in dsb_output$technical_stats
```

`empty_drop_citeseq_mtx` *small example CITE-seq protein dataset for 87 surface protein in 8005 empty droplets*

Description

A matrix of empty background droplet counts for surface proteins measured with CITE-seq antibodies. This data is used for example scripts in the dsb package. Raw data was processed with CITE-seq-Count.

Usage

```
empty_drop_citeseq_mtx
```

Format

An R matrix, rows: 87 proteins, columns: 8005 empty droplets.

empty_drop_citeseq_mtx R matrix of empty / background droplets from a CITE-seq experiment. Negative drops were called on cell hashing data with Seurat's HTODemux function and cross referencing mRNA in droplets against patient genotypes with Demuxlet. Ambiguous drops, and with less than 80 unique mRNA were removed. This is used for robust estimation of the background distribution of each protein

References

Kotliarov et. al. 2020 Nat. Medicine

`ModelNegativeADTnorm` *ModelNegativeADTnorm R function: Normalize single cell antibody derived tag (ADT) protein data. This function defines the background level for each protein by fitting a 2 component Gaussian mixture after log transformation. Empty Droplet ADT counts are not supplied. The fitted background mean of each protein across all cells is subtracted from the log transformed counts. Note this is distinct from and unrelated to the 2 component mixture used in the second step of 'DSBNormalizeProtein' which is fitted to all proteins of each cell. After this background correction step, 'ModelNegativeADTnorm' then models and removes technical cell to cell variations using the same step II procedure as in the DSBNormalizeProtein function using identical function arguments. This is a experimental function that performs well in testing and is motivated by our observation in Supplementary Fig 1 in the dsb paper showing that the fitted background mean was concordant with the mean of ambient ADTs in both empty droplets and unstained control cells. We recommend using 'ModelNegativeADTnorm' if empty droplets are not available. See <https://www.biorxiv.org/content/10.1101/2020.02.24.963603v3> for details of the algorithm.*

Description

ModelNegativeADTnorm R function: Normalize single cell antibody derived tag (ADT) protein data. This function defines the background level for each protein by fitting a 2 component Gaussian mixture after log transformation. Empty Droplet ADT counts are not supplied. The fitted background mean of each protein across all cells is subtracted from the log transformed counts. Note this is distinct from and unrelated to the 2 component mixture used in the second step of 'DSBNormalizeProtein' which is fitted to all proteins of each cell. After this background correction step, 'ModelNegativeADTnorm' then models and removes technical cell to cell variations using the same step II procedure as in the DSBNormalizeProtein function using identical function arguments. This is a experimental function that performs well in testing and is motivated by our observation in Supplementary Fig 1 in the dsb paper showing that the fitted background mean was concordant with the mean of ambient ADTs in both empty droplets and unstained control cells. We recommend using 'ModelNegativeADTnorm' if empty droplets are not available. See <https://www.biorxiv.org/content/10.1101/2020.02.24.963603v3> for details of the algorithm.

Usage

```
ModelNegativeADTnorm(
  cell_protein_matrix,
  denoise.counts = TRUE,
  use.isotype.control = TRUE,
  isotype.control.name.vec = NULL,
  define.pseudocount = FALSE,
  pseudocount.use,
  quantile.clipping = FALSE,
  quantile.clip = c(0.001, 0.9995),
  return.stats = FALSE
)
```

Arguments

`cell_protein_matrix`
: Raw protein ADT UMI count data to be normalized. Cells - columns Proteins (ADTs) - rows.

`denoise.counts`
: Recommended function default 'denoise.counts = TRUE' and 'use.isotype.control = TRUE'. This runs step II of the dsb algorithm to define and remove cell to cell technical noise.

`use.isotype.control`
: Recommended function default 'denoise.counts = TRUE' and 'use.isotype.control = TRUE'. This includes isotype controls in defining the dsb technical component.

`isotype.control.name.vec`
: A vector of the names of the isotype control proteins in the rows of the cells and background matrix e.g. isotype.control.name.vec = c('isotype1', 'isotype2').

`define.pseudocount`
: FALSE (default) uses the value 10 optimized for protein ADT data.

`pseudocount.use`
: Must be defined if 'define.pseudocount = TRUE'. This is the pseudocount to be added to raw ADT UMI counts. Otherwise the default pseudocount used.

`quantile.clipping`
: FALSE (default), if outliers or a large range of values for some proteins are observed (e.g. -50 to 50) these are often from rare outlier cells. re-running the function with 'quantile.clipping = TRUE' will adjust by applying 0.001 and 0.998th quantile value clipping to trim values to those max and min values.

`quantile.clip`
: if 'quantile.clipping = TRUE', a vector of the lowest and highest quantiles to clip. These can be tuned to the dataset size. The default c(0.001, 0.9995) optimized to clip only a few of the most extreme outliers.

`return.stats`
: if TRUE, returns a list, element 1 $dsb_normalized_matrix is the normalized adt matrix element 2 $dsb_stats is the internal stats used by dsb during denoising (the background mean, isotype control values, and the final dsb technical component that is regressed out of the counts)

Value

Normalized ADT data are returned as a standard R "matrix" of cells (columns), proteins (rows) that can be added to Seurat, SingleCellExperiment or python anndata object - see vignette. If return.stats = TRUE, function returns a list: x$dsb_normalized_matrix normalized matrix, x$protein_stats are mean and sd of log transformed cell, background and the dsb normalized values (as list). x$technical_stats includes the dsb technical component value for each cell and each variable used to calculate the technical component.

Author(s)

Matthew P. Mulè, <mattmule@gmail.com>

References

https://doi.org/10.1101/2020.02.24.963603

Examples

```
library(dsb) # load example data cells_citeseq_mtx and empty_drop_matrix included in package

# use a subset of cells and background droplets from example data
cells_citeseq_mtx = cells_citeseq_mtx[ ,1:400]
empty_drop_matrix = empty_drop_citeseq_mtx[ ,1:400]

# example I
adt_norm = dsb::ModelNegativeADTnorm(
  # step I: remove ambient protein noise modeled by a gaussian mixture
  cell_protein_matrix = cells_citeseq_mtx,

 # recommended step II: model and remove the technical component of each cell's protein data
  denoise.counts = TRUE,
  use.isotype.control = TRUE,
  isotype.control.name.vec = rownames(cells_citeseq_mtx)[67:70]
```

)

Index