Package ‘SpatialDecon’

May 4, 2024

Title  Deconvolution of mixed cells from spatial and/or bulk gene expression data

Version  1.14.0

Description  Using spatial or bulk gene expression data, estimates abundance of mixed cell types within each observation. Based on `Advances in mixed cell deconvolution enable quantification of cell types in spatial transcriptomic data``, Danaher (2022). Designed for use with the NanoString GeoMx platform, but applicable to any gene expression data.

Depends  R (>= 4.0.0)

License  MIT + file LICENSE

Encoding  UTF-8

LazyData  TRUE

RoxygenNote  7.2.3

Imports  grDevices, stats, utils, graphics, SeuratObject, Biobase, GeomxTools, repmis, methods, Matrix, logNormReg (>= 0.4)

Suggests  testthat, knitr, rmarkdown, qpdf, Seurat

biocViews  ImmunoOncology, FeatureExtraction, GeneExpression, Transcriptomics, Spatial

VignetteBuilder  knitr

BugReports  https://github.com/Nanostring-Biostats/SpatialDecon/issues

git_url  https://git.bioconductor.org/packages/SpatialDecon

git_branch  RELEASE_3_19

git_last_commit  6c0eda1

git_last_commit_date  2024-04-30

Repository  Bioconductor 3.19

Date/Publication  2024-05-03

Author  Maddy Griswold [cre, aut],
        Patrick Danaher [aut]

Maintainer  Maddy Griswold <mgriswold@nanostring.com>
SpatialDecon-package

SpatialDecon-package

SpatialDecon: A package for estimating mixed cell type abundance in the regions of spatially-resolved gene expression studies

Description

The SpatialDecon package estimates mixed cell type abundance in the regions of spatially-resolved gene expression studies, using the method of Danaher & Kim (2020), "Advances in mixed cell deconvolution enable quantification of cell types in spatially-resolved gene expression data." It is also appropriate to apply to bulk gene expression data.

functions

Functions to help set up deconvolution:

- derive_GeoMx_background Estimates the background levels from GeoMx experiments
- collapseCellTypes reformats deconvolution results to merge closely-related cell types
- download_profile_matrix Downloads a cell profile matrix.
- safeTME: a data object, a matrix of immune cell profiles for use in tumor-immune deconvolution.

Index

SpatialDecon-package  SpatialDecon: A package for estimating mixed cell type abundance in the regions of spatially-resolved gene expression studies
Deconvolution functions:

- `spatialdecon` runs the core deconvolution function
- `reverseDecon` runs a transposed/reverse deconvolution problem, fitting the data as a function of cell abundance estimates. Used to measure genes’ dependency on cell mixing and to calculate gene residuals from cell mixing.

Plotting functions:

- `florets` Plot cell abundance on a specified x-y space, with each point a cockscomb plot showing the cell abundances of that region/sample.
- `TIL_barplot` Plot abundances of tumor infiltrating lymphocytes (TILs) estimated from the safeTME cell profile matrix

Examples

```r
data(mini_geomx_dataset)
data(safeTME)
data(safeTME.matches)
# estimate background:
mini_geomx_dataset$bg <- derive_GeoMx_background(
  norm = mini_geomx_dataset$normalized,
  probepool = rep(1, nrow(mini_geomx_dataset$normalized)),
  negnames = "NegProbe"
)
# run basic decon:
res0 <- spatialdecon(
  norm = mini_geomx_dataset$normalized,
  bg = mini_geomx_dataset$bg,
  X = safeTME
)
# run decon with bells and whistles:
res <- spatialdecon(
  norm = mini_geomx_dataset$normalized,
  bg = mini_geomx_dataset$bg,
  X = safeTME,
  cellmerges = safeTME.matches,
  cell_counts = mini_geomx_dataset$annot$nuclei,
  is_pure_tumor = mini_geomx_dataset$annot$AOI.name == "Tumor"
)
```

---

### cellcols

**Default colors for the cell types in the safeTME matrix**

**Description**

A named vector of colors, giving colors for the cell types of the safeTME matrix.

**Usage**

`cellcols`
collapseCellTypes

Description

Given the input of an SpatialDecon result output and a list of which cell types to combine, returns a reshaped deconvolution result object with the specified cell types merged.

Usage

collapseCellTypes(fit, matching)

Arguments

fit The object (a list) returned by the SpatialDecon algorithm
matching A list object holding the mapping from beta’s cell names to official cell names. See str(safeTME.matches) for an example.

Value

A reshaped deconvolution result object

Examples

data(mini_geomx_dataset)
data(safeTME)
data(safeTME.matches)
# estimate background:
mini_geomx_dataset$bg <- derive_GeoMx_background(
   norm = mini_geomx_dataset$normalized,
   probepool = rep(1, nrow(mini_geomx_dataset$normalized)),
   negnames = "NegProbe"
)
# run basic decon:
res0 <- spatialdecon(
   norm = mini_geomx_dataset$normalized,
   bg = mini_geomx_dataset$bg,
   X = safeTME
)
res1 <- collapseCellTypes(
   fit = res0,
   matching = safeTME.matches
)
**create_profile_matrix**  
*Create Custom Cell Profile Matrix*

**Description**

Create custom cell profile matrix using single cell data. The average gene expression for each cell type is returned.

**Usage**

```r
create_profile_matrix(
  mtx,
  cellAnnots,
  cellTypeCol,
  cellNameCol,
  matrixName = "Custom",
  outDir = "./",
  geneList = NULL,
  normalize = FALSE,
  scalingFactor = 5,
  minCellNum = 15,
  minGenes = 100,
  discardCellTypes = FALSE
)
```

**Arguments**

- **mtx**
  - gene x cell count matrix
- **cellAnnots**
  - cell annotations with cell type and cell name as columns
- **cellTypeCol**
  - column containing cell type
- **cellNameCol**
  - column containing cell ID/name
- **matrixName**
  - name of final profile matrix
- **outDir**
  - path to desired output directory, set to NULL if matrix should not be written
- **geneList**
  - gene list to filter profile matrix to
- **normalize**
  - Should data be normalized? (TRUE/FALSE) if TRUE data will be normalized using total gene count
- **scalingFactor**
  - what should all values be multiplied by for final matrix, set to 1 if no scaling is wanted
- **minCellNum**
  - minimum number of cells of one type needed to create profile, exclusive
- **minGenes**
  - minimum number of genes expressed in a cell, exclusive
- **discardCellTypes**
  - should cell types be filtered for types like mitotic, doublet, low quality, unknown, etc.
Value
A custom cell profile matrix genes (rows) by cell types (columns), matrix gets written to disk and outDir

Examples

cellNames <- paste0("Cell", seq_len(1500))
geneNames <- paste0("Gene", seq_len(1500))
mtx <- matrix(data=sample(size = length(cellNames)*length(geneNames),
replace = TRUE,
x = c(0, seq_len(100)),
prob = c(0.6784, rep(0.0075, 15), rep(0.005, 25),
rep(0.002, 25), rep(0.001, 35))),
ncol = length(cellNames), nrow = length(geneNames),
dimnames = list(geneNames, cellNames))

cellAnnots <- as.data.frame(cbind(CellID=cellNames,
cellType=sample(size = length(cellNames),
replace = TRUE,
x = c("A", "B", "C", "D"),
prob = c(0.1, 0.4, 0.3, 0.2))))
table(cellAnnots$cellType)
profile_matrix <- create_profile_matrix(mtx = mtx,
cellAnnots = cellAnnots,
cellTypeCol = "cellType",
cellNameCol = "CellID",
minGenes = 10,
scalingFactor = 1)

head(profile_matrix)

---

**derive_GeoMx_background**

*Derive background at the scale of the normalized data for GeoMx data*

**Description**

Estimates per-datapoint background levels from a GeoMx experiment. In studies with two or more probe pools, different probes will have different background levels. This function provides a convenient way to account for this phenomenon.

**Usage**

```
derive_GeoMx_background(norm, probepool, negnames)
```

**Arguments**

- **norm**: Matrix of normalized data, genes in rows and segments in columns. Must include negprobes, and must have rownames.
- **probepool**: Vector of probe pool names for each gene, aligned to the rows of "norm".
- **negnames**: Names of all negProbes in the dataset. Must be at least one neg.name within each probe pool.
Value

A matrix of expected background values, in the same scale and dimensions as the "norm" argument.

Examples

data(mini_geomx_dataset)
# estimate background:
mini_geomx_dataset$bg <- derive_GeoMx_background(
  norm = mini_geomx_dataset$normalized,
  probepool = rep(1, nrow(mini_geomx_dataset$normalized)),
  negnames = "NegProbe"
)

Description

Download a cell profile matrix from the online library

Usage

download_profile_matrix(species, age_group, matrixname)

Arguments

species species of profile matrix
age_group age_group of profile matrix, if fetal mouse please add the developmental stage separated with /, i.e. Fetal/E14.5
matrixname name of profile matrix

Details

Valid matrices can be found on the github site https://github.com/Nanostring-Biostats/CellProfileLibrary/tree/master

Value

A cell profile matrix, suggested cell groups, and paper metadata

Examples

download_profile_matrix(species = "Human", age_group = "Adult", matrixname = "Colon_HCA")
head(profile_matrix)
print(cellGroups)
print(metadata)
florets  
\textit{Draw coxcomb plots as points in a graphics window}

\section*{Description}

Draws a scatterplot where each point is a circular barplot, intended to show decon results.

\section*{Usage}

\begin{verbatim}
florets(
  x,
  y,
  b,
  col = NULL,
  legendwindow = FALSE,
  rescale.by.sqrt = TRUE,
  border = NA,
  add = FALSE,
  cex = 1,
  bty = "n",
  xaxt = "n",
  yaxt = "n",
  xlab = "",
  ylab = "",
  ...
)
\end{verbatim}

\section*{Arguments}

- **x**: Vector of x coordinates
- **y**: Vector of y coordinates
- **b**: matrix or cell abundances, with columns aligned with the elements of x and y
- **col**: vector of colors, aligned to the rows of b.
- **legendwindow**: Logical. If TRUE, the function draws a color legend in a new window
- **rescale.by.sqrt**: Logical, for whether to rescale b by its square root to make value proportional to shape area, not shape length.
- **border**: Color of pie segment border, defaults to NA/none
- **add**: Logical. If TRUE, the function draws florets atop an existing graphics device (TRUE) or call a new device (FALSE).
- **cex**: Floret size. Florets are scaled relative to the range of x and y; this further scales up or down.
- **bty**: bty argument passed to plot()
- **xaxt**: xaxt argument passed to plot()
mean.resid.sd

yaxt
yaxt argument passed to plot()
xlab
xlab, defaults to ""
ylab
ylab, defaults to ""
...
additional arguments passed to plot()

Value
Draws a coxcomb plot, returns no data.

Examples

data(mini_geomx_dataset)
data(safeTME)
# estimate background:
mini_geomx_dataset$bg <- derive_GeoMx_background(
  norm = mini_geomx_dataset$normalized,
  probepool = rep(1, nrow(mini_geomx_dataset$normalized)),
  negnames = "NegProbe"
)
# run basic decon:
res0 <- spatialdecon(
  norm = mini_geomx_dataset$normalized,
  bg = mini_geomx_dataset$bg,
  X = safeTME
)
# draw florets:
florets(
  x = mini_geomx_dataset$annot$x,
  y = mini_geomx_dataset$annot$y,
  b = res0$beta, cex = 2
)

mean.resid.sd

Genes’ biological variability in immune deconvolution from TCGA.

Description
Genes’ biological SDs, as estimated from immune deconvolution from TCGA. Used to weight genes in spatialdecon.

Usage
mean.resid.sd

Format
A named vector giving SDs of 1179 genes.
mergeTumorIntoX  Estimate a tumor-specific profile and merge it with the pre-specified cell profile matrix (X)

Description

Given the input of "tumor-only" AOI's, estimates an collection of tumor-specific expression profiles and merges them with the immune cell expression training matrix. The process:

1. log2/normalized data from tumor-only AOIs is clustered with hclust, and cutree() is used to define clusters.
2. Each cluster's geomean profile is merged into the immune cell profile matrix.

Usage

mergeTumorIntoX(norm, bg, pure_tumor_ids, X, K = 10)

Arguments

norm  matrix of normalized data
bg  matrix of expected background, on the scale of norm.
pure_tumor_ids  Vector identifying columns of norm that are pure tumor. Can be indices, logicals or column names.
X  The training matrix
K  the number of clusters to fit

Value

an updated X matrix with new columns, "tumor.1", "tumor.2", ...

Examples

data(mini_geomx_dataset)
data(safeTME)
mini_geomx_dataset$bg <- derive_GeoMx_background(
  norm = mini_geomx_dataset$normalized,
  probepool = rep(1, nrow(mini_geomx_dataset$normalized)),
  negnames = "NegProbe"
)
safeTME.with.tumor <- mergeTumorIntoX(
  norm = mini_geomx_dataset$norm,
  bg = mini_geomx_dataset$bg,
  pure_tumor_ids = mini_geomx_dataset$annot$AOI.name == "Tumor",
  X = safeTME,
  K = 3
)
**mini_geomx_dataset**  
*Small example GeoMx data*

**Description**

A miniature GeoMx dataset used by the spatialdecon examples.

**Usage**

mini_geomx_dataset

**Format**

A list with the following elements:

- normalized: normalized data matrix
- raw: raw data matrix
- annot: AOI annotation data frame

---

**mini_singleCell_dataset**  
*Mini human colon single cell dataset*

**Description**


**Usage**

mini_singleCell_dataset

**Format**

A list with the following elements:

- mtx: sparse count matrix
- annots: cell type annotation data frame
**nsclc**  
*Large example GeoMx data*

**Description**
A GeoMx dataset with dense AOIs gridded over a NSCLC tumor. Each AOI is split into tumor and microenvironment segments.

**Usage**
nsclc

**Format**
GeoMxSet Object

---

**reverseDecon**  
*Reverse deconvolution*

**Description**
Performs "reverse deconvolution", modelling each gene expression’s ~ cell scores. Returns a matrix of "fitted" expression values, a matrix of residuals, a matrix of reverse decon coefficients for genes * cells.

**Usage**
reverseDecon(norm, beta, epsilon = NULL)

**Arguments**
- **norm**: Matrix of normalized data, with genes in rows and observations in columns
- **beta**: Matrix of cell abundance estimates, with cells in rows and observations in columns. Columns are aligned to "norm".
- **epsilon**: All y and yhat values are thresholded up to this point when performing decon. Essentially says, "ignore variability in counts below this threshold."

**Value**
A list:
- coefs, a matrix of coefficients for genes * cells, where element i,j is interpreted as "every unit increase in cell score j is expected to increase expression of gene i by _".
- yhat, a matrix of fitted values, in the same dimension as norm
runCollapseCellTypes

• resids, a matrix of log2-scale residuals from the reverse decon fit, in the same dimension as norm
• cors, a vector giving each gene’s correlation between fitted and observed expression
• resid.sd, a vector of each gene’s residual SD, a metric of how much variability genes have independently of cell mixing.

Examples

data(mini_geomx_dataset)
data(safeTME)
# estimate background:
mini_geomx_dataset$bg <- derive_GeoMx_background(
  norm = mini_geomx_dataset$normalized,
  probepool = rep(1, nrow(mini_geomx_dataset$normalized)),
  negnames = "Neg Probe"
)
# run basic decon:
res0 <- spatialdecon(
  norm = mini_geomx_dataset$normalized,
  bg = mini_geomx_dataset$bg,
  X = safeTME
)
# run reverse decon:
rdecon <- reverseDecon(
  norm = mini_geomx_dataset$norm,
  beta = res0$beta
)

runCollapseCellTypes Run collapseCellTypes

Description

Runs collapseCellTypes from an S4 object

Given the input of an SpatialDecon result output and a list of which cell types to combine, returns a reshaped deconvolution result object with the specified cell types merged.

Usage

runCollapseCellTypes(object, ...)

## S4 method for signature 'NanoStringGeoMxSet'
runCollapseCellTypes(object, matching = NULL)
runErrorModel

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>object</td>
<td>An S4 object such as a GeoMxSet object</td>
</tr>
<tr>
<td>...</td>
<td>Arguments passed to collapseCellTypes</td>
</tr>
<tr>
<td>matching</td>
<td>A list object holding the mapping from beta’s cell names to official cell names. See str(safeTME.matches) for an example.</td>
</tr>
</tbody>
</table>

Value

A reshaped deconvolution result object

Examples

```r
library(GeomxTools)
datadir <- system.file("extdata", "DSP_NGS_Example_Data", package = "GeomxTools")
demoData <- readRDS(file.path(datadir, "/demoData.rds"))
demoData <- shiftCountsOne(demoData)
target_demoData <- aggregateCounts(demoData)
target_demoData <- normalize(target_demoData, "quant")

# run basic decon:
res0 <- runspatialdecon(object = target_demoData,
                         norm_elt = "exprs_norm",
                         raw_elt = "exprs")

# run reverse decon:
target_demoData <- runReverseDecon(object = target_demoData,
                                    norm_elt = "exprs_norm",
                                    beta = pData(res0)$beta)
```

**runErrorModel**

Apply error model to estimate technical SD from raw counts

Description

Based on raw counts, uses past data to estimate each raw count’s log-scale SD from technical noise. Specifies different error models for different platforms.

Usage

```r
runErrorModel(counts, platform = "general")
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>counts</td>
<td>vector or matrix of raw counts</td>
</tr>
<tr>
<td>platform</td>
<td>String specifying which platform was used to create &quot;rawCounts&quot;. Default to &quot;dsp&quot;, for digital spatial profiler/aka GeoMx. Other options include &quot;ncounter&quot;, &quot;rsem&quot;, &quot;quantile&quot;, and &quot;st&quot; for spatial transcriptomics/visium.</td>
</tr>
</tbody>
</table>
runMergeTumorIntoX

Value

a matrix of log2-scale SDs

Examples

library(GeomxTools)
datadir <- system.file("extdata", "DSP_NGS_Example_Data", package = "GeomxTools")
demoData <- readRDS(file.path(datadir, "/demoData.rds"))

demoData <- shiftCountsOne(demoData)
target_demoData <- aggregateCounts(demoData)

sd_from_noise <- runErrorModel(counts = exprs(target_demoData), platform = "dsp")
wts <- 1 / sd_from_noise

runMergeTumorIntoX  Run MergeTumorIntoX

Description

Runs mergeTumorIntoX from an S4 object
A wrapper for applying mergeTumorIntoX to a NanostringGeomxSet object.

Usage

runMergeTumorIntoX(object, ...)

## S4 method for signature 'NanoStringGeoMxSet'
runMergeTumorIntoX(object, X, K = 10, pure_tumor_ids = NULL, norm_elt = NULL)

Arguments

object  An S4 object such as a GeoMxSet object
...
Arguments passed to mergeTumorIntoX
X  The training matrix
K  the number of clusters to fit
pure_tumor_ids  Vector identifying columns of norm that are pure tumor. Can be indices, logicals or column names.
norm_elt  norm data element in assayData

Value

updated X matrix with new columns, "tumor.1", "tumor.2", ...
Examples

```r
library(GeomxTools)
datadir <- system.file("extdata", "DSP_NGS_Example_Data", package = "GeomxTools")
demoData <- readRDS(file.path(datadir, "/demoData.rds"))

demoData <- shiftCountsOne(demoData)
target_demoData <- aggregateCounts(demoData)

target_demoData <- normalize(target_demoData, "quant")

data(safeTME)
tumor.ids <- as.logical(sample(x = c("TRUE","FALSE"), size = 88, replace = TRUE))
safeTME.with.tumor <- runMergeTumorIntoX(object = target_demoData,
X = safeTME,
K = 3,
pure_tumor_ids = tumor.ids,
norm_elt = "exprs_norm")
```

---

runReverseDecon  Run Reversedecon

Description

Runs reversedecon from an S4 object

A wrapper for applying reversedecon to a NanostringGeomxSet object.

Usage

```r
runReverseDecon(object, ...)
```

## S4 method for signature 'NanoStringGeoMxSet'
runReverseDecon(object, norm_elt = NULL, beta, epsilon = NULL)

Arguments

- **object**: An S4 object such as a GeoMxSet object
- **...**: Arguments passed to reversedecon
- **norm_elt**: normalized data element in assayData.
- **beta**: Matrix of cell abundance estimates, with cells in columns and observations in rows. Columns are aligned to "norm".
- **epsilon**: All y and yhat values are thresholded up to this point when performing decon. Essentially says, "ignore variability in counts below this threshold."
runspatialdecon

Value

a valid GeoMx S4 object including the following items:

- in fData
  - coefs, a matrix of coefficients for genes * cells, where element i,j is interpreted as "every unit increase in cell score j is expected to increase expression of gene i by _".
  - cors, a vector giving each gene's correlation between fitted and observed expression
  - resid.sd, a vector of each gene's residual SD, a metric of how much variability genes have independent of cell mixing.

- in assayData
  - yhat, a matrix of fitted values, in the same dimension as norm
  - resids, a matrix of log2-scale residuals from the reverse decon fit, in the same dimension as norm

Examples

library(GeomxTools)
datadir <- system.file("extdata", "DSP_NGS_Example_Data", package = "GeomxTools")
demoData <- readRDS(file.path(datadir, "/demoData.rds"))
demoData <- shiftCountsOne(demoData)
target_demoData <- aggregateCounts(demoData)
target_demoData <- normalize(target_demoData, "quant")

# run basic decon:
res0 <- runspatialdecon(object = target_demoData,
norm_elt = "exprs_norm",
raw_elt = "exprs")

# run reverse decon:
target_demoData <- runReverseDecon(object = target_demoData,
norm_elt = "exprs_norm",
beta = pData(res0)$beta)

runspatialdecon  Run spatialdecon

Description

Runs spatialdecon from an S4 object

A wrapper for applying spatialdecon to a NanostringGeomxSet object.

A wrapper for applying spatialdecon to the Spatial data element in a Seurat object. Unlike spatialdecon, which expects a normalized data matrix, this function operates on raw counts. Scaling for total cells
Usage

runspatialdecon(object, ...)

## S4 method for signature 'NanoStringGeoMxSet'
runspatialdecon(
  object,
  X = NULL,
  norm_elt = NULL,
  raw_elt = NULL,
  wts = NULL,
  resid_thres = 3,
  lower_thres = 0.5,
  align_genes = TRUE,
  is_pure_tumor = NULL,
  n_tumor_clusters = 10,
  cell_counts = NULL,
  cellmerges = NULL,
  maxit = 1000
)

## S4 method for signature 'Seurat'
runspatialdecon(
  object,
  X = NULL,
  bg = 0.1,
  wts = NULL,
  resid_thres = 3,
  lower_thres = 0.5,
  align_genes = TRUE,
  is_pure_tumor = NULL,
  n_tumor_clusters = 10,
  cell_counts = NULL,
  cellmerges = NULL,
  maxit = 1000
)

Arguments

object An S4 object such as a Seurat object that include a "Spatial" element in the "assays" slot or a GeoMxSet object
...
Arguments passed to spatialdecon
X Cell profile matrix. If NULL, the safeTME matrix is used.

norm_elt normalized data element in assayData in NanostringGeomxSet object
raw_elt raw data element in assayData in NanostringGeomxSet object
wts Optional, a matrix of weights.
resid_thres A scalar, sets a threshold on how extreme individual data points' values can be (in log2 units) before getting flagged as outliers and set to NA.
**lower_thresh**  A scalar. Before log2-scale residuals are calculated, both observed and fitted values get thresholded up to this value. Prevents log2-scale residuals from becoming extreme in points near zero.

**align_genes**  Logical. If TRUE, then Y, X, bg, and wts are row-aligned by shared genes.

**is_pure_tumor**  A logical vector denoting whether each AOI consists of pure tumor. If specified, then the algorithm will derive a tumor expression profile and merge it with the immune profiles matrix.

**n_tumor_clusters**  Number of tumor-specific columns to merge into the cell profile matrix. Has an impact only when is_pure_tumor argument is used to indicate pure tumor AOIs. Takes this many clusters from the pure-tumor AOI data and gets the average expression profile in each cluster. Default 10.

**cell_counts**  Number of cells estimated to be within each sample. If provided alongside norm_factors, then the algorithm will additionally output cell abundance estimates on the scale of cell counts.

**cellmerges**  A list object holding the mapping from beta’s cell names to combined cell names. If left NULL, then defaults to a mapping of granular immune cell definitions to broader categories.

**maxit**  Maximum number of iterations. Default 1000.

**bg**  Expected background counts. Either a scalar applied equally to all points in the count matrix, or a matrix with the same dimensions as the count matrix in GetAssayData(object, assay = "Spatial"). Recommended to use a small non-zero value, default of 0.1.

**Value**

decn results in list or in GeoMxSet object.

For GeoMxSet object, if not given cellmerges and cell_counts, a valid GeoMx S4 object including the following items:

- In pData
  - beta: matrix of cell abundance estimates, cells in rows and observations in columns
  - p: matrix of p-values for H0: beta == 0
  - t: matrix of t-statistics for H0: beta == 0
  - se: matrix of standard errors of beta values
  - prop_of_all: rescaling of beta to sum to 1 in each observation
  - prop_of_nontumor: rescaling of beta to sum to 1 in each observation, excluding tumor abundance estimates
  - sigmas: covariance matrices of each observation’s beta estimates

- In assayData
  - yhat: a matrix of fitted values
  - resid: a matrix of residuals from the model fit. (log2(pmax(y, lower_thresh))) - log2(pmax(xb, lower_thresh))).

- In experimentData
SpatialDeconMatrix: the cell profile matrix used in the decon fit.

if given cellmerges, the valid GeoMx S4 object will additionally include the following items

- In pData
  - beta.granular: cell abundances prior to combining closely-related cell types
  - sigma.granular: sigmas prior to combining closely-related cell types

if given cell_counts, the valid GeoMx S4 object will additionally include the following items

- In pData
  - cell.counts: beta rescaled to estimate cell numbers, based on prop_of_all and nuclei count

if given both cellmerges and cell_counts, the valid GeoMx S4 object will additionally include the following items

- In pData
  - cell.counts.granular: cell.counts prior to combining closely-related cell types

For Seurat Object, if not given cellmerges and cell_counts, a list including the following items:

- beta: matrix of cell abundance estimates, cells in rows and observations in columns
- p: matrix of p-values for H0: beta == 0
- t: matrix of t-statistics for H0: beta == 0
- se: matrix of standard errors of beta values
- prop_of_all: rescaling of beta to sum to 1 in each observation
- prop_of_nontumor: rescaling of beta to sum to 1 in each observation, excluding tumor abundance estimates
- yhat: a matrix of fitted values
- resids: a matrix of residuals from the model fit. (log2(pmax(y, lower_thresh)) - log2(pmax(xb, lower_thresh))).
- X: the cell profile matrix used in the decon fit.
- sigmas: covariance matrices of each observation’s beta estimates

if given cellmerges, the list will additionally include the following items

- beta.granular: cell abundances prior to combining closely-related cell types
- sigma.granular: sigmas prior to combining closely-related cell types

if given cell_counts, the list will additionally include the following items

- cell.counts: beta rescaled to estimate cell numbers, based on prop_of_all and nuclei count

if given both cellmerges and cell_counts, the list will additionally include the following items

- cell.counts.granular: cell.counts prior to combining closely-related cell types
## safeTME

**SafeTME matrix**

### Description

A matrix of expression profiles of 906 genes over 18 cell types.

### Usage

```
safeTME
```

### Format

A matrix with 906 genes (rows) and 18 cell types (columns)
safeTME.matches

Mapping from granularly-defined cell populations to broaded cell populations

Description
Mapping from granularly-defined cell populations to broaded cell populations, for use by the convertCellTypes function.

Usage
safeTME.matches

Format
A list. Each element of the list contains the granular cell types that roll up to a single coarse cell type.

spatialdecon

Mixed cell deconvolution of spatially-resolved gene expression data

Description
Runs the spatialdecon algorithm with added optional functionalities. Workflow is:

1. compute weights from raw data
2. Estimate a tumor profile and merge it into the cell profiles matrix
3. run deconvolution once
4. remove poorly-fit genes from first round of decon
5. re-run decon with cleaned-up gene set
6. combine closely-related cell types
7. compute p-values
8. rescale abundance estimates, to proportions of total, proportions of immune, cell counts

Usage
spatialdecon(
norm,
bg,
X = NULL,
raw = NULL,
wts = NULL,
resid_thresh = 3,
lower_thresh = 0.5,
spatialdecon

```r
align_genes = TRUE,
is_pure_tumor = NULL,
n_tumor_clusters = 10,
cell_counts = NULL,
cellmerges = NULL,
maxit = 1000
```

Arguments

- **norm**: p-length expression vector or p * N expression matrix - the actual (linear-scale) data
- **bg**: Same dimension as norm: the background expected at each data point.
- **X**: Cell profile matrix. If NULL, the safeTME matrix is used.
- **raw**: Optional for using an error model to weight the data points. p-length expression vector or p * N expression matrix - the raw (linear-scale) data
- **wts**: Optional, a matrix of weights.
- **resid_thresh**: A scalar, sets a threshold on how extreme individual data points’ values can be (in log2 units) before getting flagged as outliers and set to NA.
- **lower_thresh**: A scalar. Before log2-scale residuals are calculated, both observed and fitted values get thresholded up to this value. Prevents log2-scale residuals from becoming extreme in points near zero.
- **align_genes**: Logical. If TRUE, then Y, X, bg, and wts are row-aligned by shared genes.
- **is_pure_tumor**: A logical vector denoting whether each AOI consists of pure tumor. If specified, then the algorithm will derive a tumor expression profile and merge it with the immune profiles matrix.
- **n_tumor_clusters**: Number of tumor-specific columns to merge into the cell profile matrix. Has an impact only when is_pure_tumor argument is used to indicate pure tumor AOIs. Takes this many clusters from the pure-tumor AOI data and gets the average expression profile in each cluster. Default 10.
- **cell_counts**: Number of cells estimated to be within each sample. If provided alongside norm_factors, then the algorithm will additionally output cell abundance estimates on the scale of cell counts.
- **cellmerges**: A list object holding the mapping from beta's cell names to combined cell names. If left NULL, then defaults to a mapping of granular immune cell definitions to broader categories.
- **maxit**: Maximum number of iterations. Default 1000.

Value

- A list:
  - **beta**: matrix of cell abundance estimates, cells in rows and observations in columns
  - **sigmas**: covariance matrices of each observation's beta estimates
• p: matrix of p-values for H0: beta == 0
• t: matrix of t-statistics for H0: beta == 0
• se: matrix of standard errors of beta values
• prop_of_all: rescaling of beta to sum to 1 in each observation
• prop_of_nontumor: rescaling of beta to sum to 1 in each observation, excluding tumor abundance estimates
• cell.counts: beta rescaled to estimate cell numbers, based on prop_of_all and nuclei count
• beta.granular: cell abundances prior to combining closely-related cell types
• sigma.granular: sigmas prior to combining closely-related cell types
• cell.counts.granular: cell.counts prior to combining closely-related cell types
• resids: a matrix of residuals from the model fit. \( \log_2(p_{\max}(y, \text{lower\_thresh})) - \log_2(p_{\max}(xb, \text{lower\_thresh})) \).
• X: the cell profile matrix used in the decon fit.

Examples

data(mini_geomx_dataset)
data(safeTME)
data(safeTME.matches)
# estimate background:
mini_geomx_dataset$bg <- derive_GeoMx_background(
    norm = mini_geomx_dataset$normalized, 
    probepool = rep(1, nrow(mini_geomx_dataset$normalized)),
    negnames = "NegProbe"
)
# run basic decon:
res0 <- spatialdecon(
    norm = mini_geomx_dataset$normalized,
    bg = mini_geomx_dataset$bg,
    X = safeTME
)
# run decon with bells and whistles:
res <- spatialdecon(
    norm = mini_geomx_dataset$normalized,
    bg = mini_geomx_dataset$bg,
    X = safeTME,
    cellmerges = safeTME.matches,
    cell_counts = mini_geomx_dataset$annot$nuclei,
    is_pure_tumor = mini_geomx_dataset$annot$AOI.name == "Tumor"
)
TIL_barplot

Barplot of abundance estimates

Description

Draw barplot of the "betas" from a decon fit

Usage

TIL_barplot(mat, draw_legend = FALSE, main = "", col = NULL, ...)

Arguments

mat
Matrix of cell proportions or abundances, in the same dimensions output by
spatialdecon (cells in rows, observations in columns). User is free to re-order
columns/observations in whatever order is best for display.
draw_legend
Logical. If TRUE, the function draws a legend in a new plot frame.
main
Title for barplot
col
Vector of colors for cell types. Defaults to pre-set colors for the safeTME cell
types.
...Arguments passed to barplot()

Value

Draws a barplot.

Examples

data(mini_geomx_dataset)
data(safeTME)
# estimate background:
mini_geomx_dataset$bg <- derive_GeoMx_background(
  norm = mini_geomx_dataset$normalized,
  probepool = rep(1, nrow(mini_geomx_dataset$normalized)),
  negnames = "NegProbe"
)
# run basic decon:
res0 <- spatialdecon(
  norm = mini_geomx_dataset$normalized,
  bg = mini_geomx_dataset$bg,
  X = safeTME
)
# run barplot:
TIL_barplot(mat = res0$beta)
# run barplot and draw a color legend
TIL_barplot(mat = res0$beta, draw_legend = TRUE)
Index

* datasets
  - cellcols, 3
  - mean.resid.sd, 9
  - mini_geomx_dataset, 11
  - mini_singleCell_dataset, 11
  - nsclc, 12
  - safeTME, 21
  - safeTME.matches, 22

  cellcols, 3
  collapseCellTypes, 4
  create_profile_matrix, 5

  derive_GeoMx_background, 6
  download_profile_matrix, 7

  florets, 8

  mean.resid.sd, 9
  mergeTumorIntoX, 10
  mini_geomx_dataset, 11
  mini_singleCell_dataset, 11

  nsclc, 12

  reverseDecon, 12
  runCollapseCellTypes, 13
  runCollapseCellTypes,NanoStringGeoMxSet-method
    (runCollapseCellTypes), 13
  runErrorModel, 14
  runMergeTumorIntoX, 15
  runMergeTumorIntoX,NanoStringGeoMxSet-method
    (runMergeTumorIntoX), 15
  runReverseDecon, 16
  runReverseDecon,NanoStringGeoMxSet-method
    (runReverseDecon), 16
  runspatialdecon, 17
  runspatialdecon,NanoStringGeoMxSet-method
    (runspatialdecon), 17
  runspatialdecon,Seurat-method
    (runspatialdecon), 17

  safeTME, 21
  safeTME.matches, 22
  spatialdecon, 22
  SpatialDecon-package, 2
  TIL_barplot, 25