Package 'tidybulk'

April 28, 2020

Type Package

Title Friendly tidy wrappers for streamlined bulk transcriptional analysis

Version 1.0.0

Author person(``Stefano'', ``Mangiola'', email=mangiolastefano@gmail.com, role=c(``cre'', aut``), comment = c(ORCID = "0000-0001-7474-836X")).

Maintainer Stefano Mangiola <mangiolastefano@gmail.com>

Description This is a collection of utility functions that allow to perform exploration of and calculations to RNA sequencing data, in a modular, pipe-friendly and tidy fashion.

License GPL-3

Depends R (>= 4.0)

Imports tibble,
readr,
dplyr,
magrittr,
tidyr,
rlang,
purrr,
preprocessCore,
stats,
parallel,
utils,
lifecycle

Suggests testthat,
AnnotationDbi,
BiocManager,
Rsubread,
e1071,
edgeR,
limma,
org.Hs.eg.db,
sva,
GGally,
knitr,
qpdf,

covr,
Seurat,
KernSmooth,
Rtsne,
EGSEA,
SummarizedExperiment,
S4Vectors,
ggplot2,
widyr

VignetteBuilder knitr

RdMacros lifecycle

Biarch true

biocViews AssayDomain, Infrastructure

Encoding UTF-8

LazyData true

RoxygenNote 7.0.2

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

git_branch RELEASE_3_11

git_last_commit b7e0663

git_last_commit_date 2020-04-27

Date/Publication 2020-04-27

R topics documented:

add_attr *Add attribute to abject*

Description

Add attribute to abject

Usage

```
add_attr(var, attribute, name)
```

Arguments

var	A tibble
attribute	An object
name	A character name of the attribute

Value

A tibble with an additional attribute

add_class *Add class to abject*

Description

Add class to abject

Usage

```
add_class(var, name)
```

Arguments

var	A tibble
name	A character name of the attribute

Value

A tibble with an additional attribute

`add_scaled_counts_bulk.calcNormFactor` *Calculate the norm factor with calcNormFactor from limma*

Description

Calculate the norm factor with calcNormFactor from limma

Usage

```
add_scaled_counts_bulk.calcNormFactor(
  .data,
  reference = NULL,
  factor_of_interest = NULL,
  minimum_counts = 10,
  minimum_proportion = 0.7,
  .sample = sample,
  .transcript = transcript,
  .abundance = count,
  method
)
```

Arguments

`.data`	A tibble
`reference`	A reference matrix, not sure if used anymore
`factor_of_interest`	The name of the column of the factor of interest
`minimum_counts`	A positive integer. Minimum counts required for at least some samples.
`minimum_proportion`	A real positive number between 0 and 1. It is the threshold of proportion of samples for each transcripts/genes that have to be characterised by a cmp bigger than the threshold to be included for scaling procedure.
`.sample`	The name of the sample column
`.transcript`	The name of the transcript/gene column
`.abundance`	The name of the transcript/gene abundance column
`method`	A string character. The scaling method passed to the backend function (i.e., edgeR::calcNormFactors; "TMM","TMMwsp","RLE","upperquartile")

Value

A list including the filtered data frame and the normalization factors

add_scaled_counts_bulk.get_low_expressed *Drop lowly tanscribed genes for TMM normalization*

Description

Drop lowly tanscribed genes for TMM normalization

Usage

```
add_scaled_counts_bulk.get_low_expressed(
  .data,
  .sample = sample,
  .transcript = transcript,
  .abundance = count,
  factor_of_interest = NULL,
  minimum_counts = 10,
  minimum_proportion = 0.7
)
```

Arguments

.data	A tibble
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
factor_of_interest	The name of the column of the factor of interest
minimum_counts	A positive integer. Minimum counts required for at least some samples.
minimum_proportion	A real positive number between 0 and 1. It is the threshold of proportion of samples for each transcripts/genes that have to be characterised by a cmp bigger than the threshold to be included for scaling procedure.

Value

A tibble filtered

adjust_abundance *Adjust transcript abundance for unwanted variation*

Description

adjust_abundance() takes as imput a 'tbl' formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> | and returns a 'tbl' with an edditional adjusted abundance column. This method uses scaled counts if present.

Usage

```
adjust_abundance(
  .data,
  .formula,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  log_transform = TRUE,
  action = "add",
  ...
)
```

Arguments

.data	A 'tbl' formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.formula	A formula with no response variable, representing the desired linear model where the first covariate is the factor of interest and the second covariate is the unwanted variation (of the kind ~ factor_of_intrest + batch)
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
log_transform	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)
action	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).
...	Further parameters passed to the function sva::ComBat

Details

Maturing

This function adjusts the abundance for (known) unwanted variation. At the moment just an unwanted covariated is allowed at a time.

Value

A 'tbl' with additional columns for the adjusted counts as '<COUNT COLUMN>_adjusted'

Examples

```
cm = tidybulk::counts_mini
cm$batch = 0
cm$batch[cm$sample %in% c("SRR1740035", "SRR1740043")] = 1

res =
adjust_abundance(
cm,
~ condition + batch,
.sample = sample,
```

```
.transcript = transcript,
.abundance = count
)
```

---

adjust_abundance,RangedSummarizedExperiment-method
adjust_abundance

Description

adjust_abundance

Usage

```
## S4 method for signature 'RangedSummarizedExperiment'
adjust_abundance(
  .data,
  .formula,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  log_transform = TRUE,
  action = "add",
  ...
)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.formula	A formula with no response variable, representing the desired linear model where the first covariate is the factor of interest and the second covariate is the unwanted variation (of the kind ~ factor_of_intrest + batch)
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
log_transform	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)
action	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).
...	Further parameters passed to the function sva::ComBat

Value

A ‘SummarizedExperiment‘ object

`adjust_abundance,spec_tbl_df-method`
adjust_abundance

Description

adjust_abundance

Usage

```
## S4 method for signature 'spec_tbl_df'
adjust_abundance(
  .data,
  .formula,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  log_transform = TRUE,
  action = "add",
  ...
)
```

Arguments

`.data`	A ‘tbl‘ formatted as	<SAMPLE>	<TRANSCRIPT>	<COUNT>	<...>	
`.formula`	A formula with no response variable, representing the desired linear model where the first covariate is the factor of interest and the second covariate is the unwanted variation (of the kind ~ factor_of_intrest + batch)
`.sample`	The name of the sample column
`.transcript`	The name of the transcript/gene column
`.abundance`	The name of the transcript/gene abundance column
`log_transform`	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)
`action`	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).
`...`	Further parameters passed to the function sva::ComBat

Value

A ‘tbl‘ with additional columns for the adjusted counts as ‘<COUNT COLUMN>_adjusted‘

adjust_abundance,SummarizedExperiment-method
adjust_abundance

Description

adjust_abundance

Usage

```
## S4 method for signature 'SummarizedExperiment'
adjust_abundance(
  .data,
  .formula,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  log_transform = TRUE,
  action = "add",
  ...
)
```

Arguments

.data	A ‘tbl‘ formatted as	<SAMPLE>	<TRANSCRIPT>	<COUNT>	<...>	
.formula	A formula with no response variable, representing the desired linear model where the first covariate is the factor of interest and the second covariate is the unwanted variation (of the kind ~ factor_of_intrest + batch)
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
log_transform	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)
action	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).
...	Further parameters passed to the function sva::ComBat

Value

A ‘SummarizedExperiment‘ object

adjust_abundance,tbl_df-method
adjust_abundance

Description

adjust_abundance

Usage

```
## S4 method for signature 'tbl_df'
adjust_abundance(
  .data,
  .formula,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  log_transform = TRUE,
  action = "add",
  ...
)
```

Arguments

.data	A ‘tbl‘ formatted as	<SAMPLE>	<TRANSCRIPT>	<COUNT>	<...>	
.formula	A formula with no response variable, representing the desired linear model where the first covariate is the factor of interest and the second covariate is the unwanted variation (of the kind ~ factor_of_intrest + batch)
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
log_transform	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)
action	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).
...	Further parameters passed to the function sva::ComBat

Value

A ‘tbl‘ with additional columns for the adjusted counts as ‘<COUNT COLUMN>_adjusted‘

adjust_abundance,tidybulk-method
adjust_abundance

Description

adjust_abundance

Usage

```
## S4 method for signature 'tidybulk'
adjust_abundance(
  .data,
  .formula,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  log_transform = TRUE,
  action = "add",
  ...
)
```

Arguments

.data	A ‘tbl‘ formatted as	<SAMPLE>	<TRANSCRIPT>	<COUNT>	<...>	
.formula	A formula with no response variable, representing the desired linear model where the first covariate is the factor of interest and the second covariate is the unwanted variation (of the kind ~ factor_of_intrest + batch)
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
log_transform	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)
action	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).
...	Further parameters passed to the function sva::ComBat

Value

A ‘tbl‘ with additional columns for the adjusted counts as ‘<COUNT COLUMN>_adjusted‘

`aggregate_duplicated_transcripts_bulk` *Aggregates multiple counts from the same samples (e.g., from isoforms) This function aggregates counts over samples, concatenates other character columns, and averages other numeric columns*

Description

Aggregates multiple counts from the same samples (e.g., from isoforms) This function aggregates counts over samples, concatenates other character columns, and averages other numeric columns

Usage

```
aggregate_duplicated_transcripts_bulk(
  .data,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  aggregation_function = sum,
  keep_integer = TRUE
)
```

Arguments

`.data`	A tibble
`.sample`	The name of the sample column
`.transcript`	The name of the transcript/gene column
`.abundance`	The name of the transcript/gene abundance column
`aggregation_function`	A function for counts aggregation (e.g., sum)
`keep_integer`	A boolean

Value

A tibble with aggregated genes and annotation

`aggregate_duplicates` *Aggregates multiple counts from the same samples (e.g., from isoforms), concatenates other character columns, and averages other numeric columns*

Description

aggregate_duplicates() takes as imput a ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> | and returns a ‘tbl‘ with aggregated transcripts that were duplicated.

Usage

```
aggregate_duplicates(
  .data,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  aggregation_function = sum,
  keep_integer = TRUE
)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
aggregation_function	A function for counts aggregation (e.g., sum, median, or mean)
keep_integer	A boolean. Whether to force the aggregated counts to integer

Details

Maturing

This function aggregates duplicated transcripts (e.g., isoforms, ensembl). For example, we often have to convert ensembl symbols to gene/transcript symbol, but in doing so we have to deal with duplicates. ‘aggregate_duplicates‘ takes a tibble and column names (as symbols; for ‘sample‘, ‘transcript‘ and ‘count‘) as arguments and returns a tibble with aggregate transcript with the same name. All the rest of the column are appended, and factors and boolean are appended as characters.

Value

A ‘tbl‘ object with aggregated transcript abundance and annotation

Examples

```
aggregate_duplicates(
tidybulk::counts_mini,
sample,
transcript,
`count`,
aggregation_function = sum
)
```

aggregate_duplicates,RangedSummarizedExperiment-method *aggregate_duplicates*

Description

aggregate_duplicates

Usage

```
## S4 method for signature 'RangedSummarizedExperiment'
aggregate_duplicates(
  .data,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  aggregation_function = sum,
  keep_integer = TRUE
)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
aggregation_function	A function for counts aggregation (e.g., sum, median, or mean)
keep_integer	A boolean. Whether to force the aggregated counts to integer

Value

A ‘SummarizedExperiment‘ object

aggregate_duplicates,spec_tbl_df-method *aggregate_duplicates*

Description

aggregate_duplicates

Usage

```
## S4 method for signature 'spec_tbl_df'
aggregate_duplicates(
  .data,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  aggregation_function = sum,
  keep_integer = TRUE
)
```

Arguments

.data	A ‘tbl‘ formatted as	<SAMPLE>	<TRANSCRIPT>	<COUNT>	<...>	
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
aggregation_function	A function for counts aggregation (e.g., sum, median, or mean)
keep_integer	A boolean. Whether to force the aggregated counts to integer

Value

A ‘tbl‘ object with aggregated transcript abundance and annotation

---

aggregate_duplicates,SummarizedExperiment-method *aggregate_duplicates*

Description

aggregate_duplicates

Usage

```
## S4 method for signature 'SummarizedExperiment'
aggregate_duplicates(
  .data,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  aggregation_function = sum,
  keep_integer = TRUE
)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
aggregation_function	A function for counts aggregation (e.g., sum, median, or mean)
keep_integer	A boolean. Whether to force the aggregated counts to integer

Value

A ‘SummarizedExperiment‘ object

---

aggregate_duplicates,tbl_df-method *aggregate_duplicates*

Description

aggregate_duplicates

Usage

```
## S4 method for signature 'tbl_df'
aggregate_duplicates(
  .data,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  aggregation_function = sum,
  keep_integer = TRUE
)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
aggregation_function	A function for counts aggregation (e.g., sum, median, or mean)
keep_integer	A boolean. Whether to force the aggregated counts to integer

Value

A ‘tbl‘ object with aggregated transcript abundance and annotation

aggregate_duplicates,tidybulk-method *aggregate_duplicates*

Description

aggregate_duplicates

Usage

```
## S4 method for signature 'tidybulk'
aggregate_duplicates(
  .data,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  aggregation_function = sum,
  keep_integer = TRUE
)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
aggregation_function	A function for counts aggregation (e.g., sum, median, or mean)
keep_integer	A boolean. Whether to force the aggregated counts to integer

Value

A ‘tbl‘ object with aggregated transcript abundance and annotation

arrange *Arrange rows by column values*

Description

‘arrange()‘ order the rows of a data frame rows by the values of selected columns.

Unlike other dplyr verbs, ‘arrange()‘ largely ignores grouping; you need to explicit mention grouping variables (or use ‘by_group = TRUE‘) in order to group by them, and functions of variables are evaluated once per data frame, not once per group.

Usage

```
arrange(.data, ..., .by_group = FALSE)

## Default S3 method:
arrange(.data, ..., .by_group = FALSE)
```

Arguments

.data	A data frame, data frame extension (e.g. a tibble), or a lazy data frame (e.g. from dbplyr or dtplyr). See *Methods*, below, for more details.
...	<['tidy-eval'][dplyr_tidy_eval]> Variables, or functions or variables. Use [desc()] to sort a variable in descending order.
.by_group	If 'TRUE', will sort first by grouping variable. Applies to grouped data frames only.

Details

Locales The sort order for character vectors will depend on the collating sequence of the locale in use: see [locales()].

Missing values Unlike base sorting with 'sort()', 'NA' are: * always sorted to the end for local data, even when wrapped with 'desc()'. * treated differently for remote data, depending on the backend.

Value

An object of the same type as '.data'.

* All rows appear in the output, but (usually) in a different place. * Columns are not modified. * Groups are not modified. * Data frame attributes are preserved.

Methods

This function is a **generic**, which means that packages can provide implementations (methods) for other classes. See the documentation of individual methods for extra arguments and differences in behaviour.

The following methods are currently available in loaded packages:

See Also

Other single table verbs: `filter()`, `mutate()`, `rename()`, `summarise()`

Examples

```
`%>%` = magrittr::`%>%`
arrange(mtcars, cyl, disp)
arrange(mtcars, desc(disp))

# grouped arrange ignores groups
by_cyl <- mtcars %>% group_by(cyl)
by_cyl %>% arrange(desc(wt))
# Unless you specifically ask:
by_cyl %>% arrange(desc(wt), .by_group = TRUE)
```

as_matrix *Get matrix from tibble*

Description

Get matrix from tibble

Usage

```
as_matrix(tbl, rownames = NULL, do_check = TRUE)
```

Arguments

tbl	A tibble
rownames	A character string of the rownames
do_check	A boolean

Value

A matrix

Examples

```
as_matrix(head(dplyr::select(tidybulk::counts_mini, transcript, count)), rownames=transcript)
```

bind *Efficiently bind multiple data frames by row and column*

Description

This is an efficient implementation of the common pattern of ‘do.call(rbind, dfs)‘ or ‘do.call(cbind, dfs)‘ for binding many data frames into one.

Usage

```
bind_rows(..., .id = NULL)

bind_cols(..., .id = NULL)
```

Arguments

...	Data frames to combine. Each argument can either be a data frame, a list that could be a data frame, or a list of data frames. When row-binding, columns are matched by name, and any missing columns will be filled with NA. When column-binding, rows are matched by position, so all data frames must have the same number of rows. To match by value, not position, see [mutate-joins].
.id	Data frame identifier. When '.id' is supplied, a new column of identifiers is created to link each row to its original data frame. The labels are taken from the named arguments to 'bind_rows()'. When a list of data frames is supplied, the labels are taken from the names of the list. If no names are found a numeric sequence is used instead.

Details

The output of 'bind_rows()' will contain a column if that column appears in any of the inputs.

Value

'bind_rows()' and 'bind_cols()' return the same type as the first input, either a data frame, 'tbl_df', or 'grouped_df'.

Examples

```
`%>%` = magrittr::`%>%`
one <- mtcars[1:4, ]
two <- mtcars[11:14, ]

# You can supply data frames as arguments:
bind_rows(one, two)
```

---

breast_tcga_mini *Data set*

Description

Data set

Usage

```
breast_tcga_mini
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 125500 rows and 5 columns.

check_if_counts_is_na *Check whether there are NA counts*

Description

Check whether there are NA counts

Usage

```
check_if_counts_is_na(.data, .abundance)
```

Arguments

.data	A tibble of read counts
.abundance	A character name of the read count column

Value

A tbl

check_if_duplicated_genes *Check whether there are duplicated genes/transcripts*

Description

Check whether there are duplicated genes/transcripts

Usage

```
check_if_duplicated_genes(
  .data,
  .sample = sample,
  .transcript = transcript,
  .abundance = `read count`
)
```

Arguments

.data	A tibble of read counts
.sample	A character name of the sample column
.transcript	A character name of the transcript/gene column
.abundance	A character name of the read count column

Value

A tbl

check_if_wrong_input *Check whether there are NA counts*

Description

Check whether there are NA counts

Usage

```
check_if_wrong_input(.data, list_input, expected_type)
```

Arguments

.data	A tibble of read counts
list_input	A list
expected_type	A character string

Value

A tbl

cluster_elements *Get clusters of elements (e.g., samples or transcripts)*

Description

cluster_elements() takes as imput a 'tbl' formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> | and identify clusters in the data.

Usage

```
cluster_elements(
  .data,
  .element = NULL,
  .feature = NULL,
  .abundance = NULL,
  method,
  of_samples = TRUE,
  log_transform = TRUE,
  action = "add",
  ...
)
```

Arguments

.data	A 'tbl' formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.element	The name of the element column (normally samples).
.feature	The name of the feature column (normally transcripts/genes)
.abundance	The name of the column including the numerical value the clustering is based on (normally transcript abundance)
method	A character string. The cluster algorithm to use, ay the moment k-means is the only algorithm included.
of_samples	A boolean. In case the input is a tidybulk object, it indicates Whether the element column will be sample or transcript column
log_transform	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)
action	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).
...	Further parameters passed to the function kmeans

Details

Maturing

identifies clusters in the data, normally of samples. This function returns a tibble with additional columns for the cluster annotation. At the moment only k-means clustering is supported, the plan is to introduce more clustering methods.

Value

A tbl object with additional columns with cluster labels

Examples

```
cluster_elements(tidybulk::counts_mini, sample, transcript, count,centers = 2, method="kmeans")
```

---

cluster_elements,RangedSummarizedExperiment-method
cluster_elements

Description

cluster_elements

Usage

```
## S4 method for signature 'RangedSummarizedExperiment'
cluster_elements(
  .data,
  .element = NULL,
  .feature = NULL,
  .abundance = NULL,
  method,
  of_samples = TRUE,
  log_transform = TRUE,
  action = "add",
  ...
)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.element	The name of the element column (normally samples).
.feature	The name of the feature column (normally transcripts/genes)
.abundance	The name of the column including the numerical value the clustering is based on (normally transcript abundance)
method	A character string. The cluster algorithm to use, ay the moment k-means is the only algorithm included.
of_samples	A boolean. In case the input is a tidybulk object, it indicates Whether the element column will be sample or transcript column
log_transform	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)
action	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).
...	Further parameters passed to the function kmeans

Value

A ‘SummarizedExperiment‘ object

---

cluster_elements,spec_tbl_df-method *cluster_elements*

Description

cluster_elements

Usage

```
## S4 method for signature 'spec_tbl_df'
cluster_elements(
  .data,
  .element = NULL,
  .feature = NULL,
  .abundance = NULL,
  method,
  of_samples = TRUE,
  log_transform = TRUE,
  action = "add",
  ...
)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.element	The name of the element column (normally samples).
.feature	The name of the feature column (normally transcripts/genes)
.abundance	The name of the column including the numerical value the clustering is based on (normally transcript abundance)
method	A character string. The cluster algorithm to use, ay the moment k-means is the only algorithm included.
of_samples	A boolean. In case the input is a tidybulk object, it indicates Whether the element column will be sample or transcript column
log_transform	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)
action	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).
...	Further parameters passed to the function kmeans

Value

A tbl object with additional columns with cluster labels

---

cluster_elements,SummarizedExperiment-method
cluster_elements

Description

cluster_elements

Usage

```
## S4 method for signature 'SummarizedExperiment'
cluster_elements(
  .data,
  .element = NULL,
  .feature = NULL,
  .abundance = NULL,
  method,
  of_samples = TRUE,
  log_transform = TRUE,
  action = "add",
  ...
)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.element	The name of the element column (normally samples).
.feature	The name of the feature column (normally transcripts/genes)
.abundance	The name of the column including the numerical value the clustering is based on (normally transcript abundance)
method	A character string. The cluster algorithm to use, ay the moment k-means is the only algorithm included.
of_samples	A boolean. In case the input is a tidybulk object, it indicates Whether the element column will be sample or transcript column
log_transform	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)
action	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).
...	Further parameters passed to the function kmeans

Value

A ‘SummarizedExperiment‘ object

---

cluster_elements,tbl_df-method
cluster_elements

Description

cluster_elements

Usage

```
## S4 method for signature 'tbl_df'
cluster_elements(
  .data,
  .element = NULL,
  .feature = NULL,
  .abundance = NULL,
  method,
  of_samples = TRUE,
  log_transform = TRUE,
  action = "add",
  ...
)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.element	The name of the element column (normally samples).
.feature	The name of the feature column (normally transcripts/genes)
.abundance	The name of the column including the numerical value the clustering is based on (normally transcript abundance)
method	A character string. The cluster algorithm to use, ay the moment k-means is the only algorithm included.
of_samples	A boolean. In case the input is a tidybulk object, it indicates Whether the element column will be sample or transcript column
log_transform	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)
action	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).
...	Further parameters passed to the function kmeans

Value

A tbl object with additional columns with cluster labels

---

cluster_elements,tidybulk-method *cluster_elements*

Description

cluster_elements

Usage

```
## S4 method for signature 'tidybulk'
cluster_elements(
  .data,
  .element = NULL,
  .feature = NULL,
  .abundance = NULL,
  method,
  of_samples = TRUE,
  log_transform = TRUE,
  action = "add",
  ...
)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.element	The name of the element column (normally samples).
.feature	The name of the feature column (normally transcripts/genes)
.abundance	The name of the column including the numerical value the clustering is based on (normally transcript abundance)
method	A character string. The cluster algorithm to use, ay the moment k-means is the only algorithm included.
of_samples	A boolean. In case the input is a tidybulk object, it indicates Whether the element column will be sample or transcript column
log_transform	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)
action	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).
...	Further parameters passed to the function kmeans

Value

A tbl object with additional columns with cluster labels

---

counts *Example data set*

Description

Example data set

Usage

```
counts
```

Format

An object of class `spec_tbl_df` (inherits from `tbl_df`, `tbl`, `data.frame`) with 938112 rows and 8 columns.

counts_ensembl *Counts with ensembl annotation*

Description

Counts with ensembl annotation

Usage

```
counts_ensembl
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 119 rows and 6 columns.

counts_mini *Example data set reduced*

Description

Example data set reduced

Usage

```
counts_mini
```

Format

An object of class `spec_tbl_df` (inherits from `tbl_df`, `tbl`, `data.frame`) with 2635 rows and 6 columns.

create_tt_from_bam_sam_bulk *Convert bam/sam files to a tidy gene transcript counts data frame*

Description

Convert bam/sam files to a tidy gene transcript counts data frame

Usage

```
create_tt_from_bam_sam_bulk(file_names, genome = "hg38", ...)
```

Arguments

`file_names`	A character vector
`genome`	A character string
`...`	Further parameters passed to the function Rsubread::featureCounts

Value

A tibble of gene counts

`create_tt_from_tibble_bulk` *Create tt object from tibble*

Description

Create tt object from tibble

Usage

```
create_tt_from_tibble_bulk(
  .data,
  .sample,
  .transcript,
  .abundance,
  .abundance_scaled = NULL
)
```

Arguments

`.data`	A tibble
`.sample`	The name of the sample column
`.transcript`	The name of the transcript/gene column
`.abundance`	The name of the transcript/gene abundance column
`.abundance_scaled`	The name of the transcript/gene scaled abundance column

Value

A tibble with an additional column

`deconvolve_cellularity` *Get cell type proportions from samples*

Description

deconvolve_cellularity() takes as imput a ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> | and returns a ‘tbl‘ with the estimated cell type abundance for each sample

Usage

```
deconvolve_cellularity(
  .data,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  reference = X_cibersort,
  method = "cibersort",
  action = "add",
  ...
)
```

Arguments

.data	A 'tbl' formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
reference	A data frame. The transcript/cell_type data frame of integer transcript abundance
method	A character string. The method to be used. At the moment Cibersort (default) and llsr (linear least squares regression) are available.
action	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).
...	Further parameters passed to the function Cibersort

Details

Maturing

This function infers the cell type composition of our samples (with the algorithm Cibersort; Newman et al., 10.1038/nmeth.3337).

Value

A 'tbl' object including additional columns for each cell type estimated

Examples

```
deconvolve_cellularity(tidybulk::counts, sample, transcript, `count`, cores = 2)
```

deconvolve_cellularity,RangedSummarizedExperiment-method
deconvolve_cellularity

Description

deconvolve_cellularity

Usage

```
## S4 method for signature 'RangedSummarizedExperiment'
deconvolve_cellularity(
  .data,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  reference = X_cibersort,
  method = "cibersort",
  action = "add",
  ...
)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
reference	A data frame. The transcript/cell_type data frame of integer transcript abundance
method	A character string. The method to be used. At the moment Cibersort (default) and llsr (linear least squares regression) are available.
action	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).
...	Further parameters passed to the function Cibersort

Value

A ‘SummarizedExperiment‘ object

deconvolve_cellularity,spec_tbl_df-method
deconvolve_cellularity

Description

deconvolve_cellularity

Usage

```
## S4 method for signature 'spec_tbl_df'
deconvolve_cellularity(
  .data,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  reference = X_cibersort,
  method = "cibersort",
  action = "add",
  ...
)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
reference	A data frame. The transcript/cell_type data frame of integer transcript abundance
method	A character string. The method to be used. At the moment Cibersort (default) and llsr (linear least squares regression) are available.
action	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).
...	Further parameters passed to the function Cibersort

Value

A ‘tbl‘ object including additional columns for each cell type estimated

`deconvolve_cellularity,SummarizedExperiment-method`
deconvolve_cellularity

Description

deconvolve_cellularity

Usage

```
## S4 method for signature 'SummarizedExperiment'
deconvolve_cellularity(
  .data,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  reference = X_cibersort,
  method = "cibersort",
  action = "add",
  ...
)
```

Arguments

`.data`	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
`.sample`	The name of the sample column
`.transcript`	The name of the transcript/gene column
`.abundance`	The name of the transcript/gene abundance column
`reference`	A data frame. The transcript/cell_type data frame of integer transcript abundance
`method`	A character string. The method to be used. At the moment Cibersort (default) and llsr (linear least squares regression) are available.
`action`	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).
`...`	Further parameters passed to the function Cibersort

Value

A ‘SummarizedExperiment‘ object

`deconvolve_cellularity,tbl_df-method` *deconvolve_cellularity*

Description

deconvolve_cellularity

Usage

```
## S4 method for signature 'tbl_df'
deconvolve_cellularity(
  .data,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  reference = X_cibersort,
  method = "cibersort",
  action = "add",
  ...
)
```

Arguments

`.data`	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
`.sample`	The name of the sample column
`.transcript`	The name of the transcript/gene column
`.abundance`	The name of the transcript/gene abundance column
`reference`	A data frame. The transcript/cell_type data frame of integer transcript abundance
`method`	A character string. The method to be used. At the moment Cibersort (default) and llsr (linear least squares regression) are available.
`action`	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).
`...`	Further parameters passed to the function Cibersort

Value

A ‘tbl‘ object including additional columns for each cell type estimated

deconvolve_cellularity,tidybulk-method
deconvolve_cellularity

Description

deconvolve_cellularity

Usage

```
## S4 method for signature 'tidybulk'
deconvolve_cellularity(
  .data,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  reference = X_cibersort,
  method = "cibersort",
  action = "add",
  ...
)
```

Arguments

.data	A 'tbl' formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
reference	A data frame. The transcript/cell_type data frame of integer transcript abundance
method	A character string. The method to be used. At the moment Cibersort (default) and llsr (linear least squares regression) are available.
action	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).
...	Further parameters passed to the function Cibersort

Value

A 'tbl' object including additional columns for each cell type estimated

distinct *distinct*

Description

distinct

Usage

```
distinct(.data, ..., .keep_all = FALSE)
```

Arguments

.data	A tbl. (See dplyr)
...	Data frames to combine (See dplyr)
.keep_all	If TRUE, keep all variables in .data. If a combination of ... is not distinct, this keeps the first row of values. (See dplyr)

Value

A tt object

Examples

```
distinct(tidybulk::counts_mini)
```

drop_attr *Drop attribute to abject*

Description

Drop attribute to abject

Usage

```
drop_attr(var, name)
```

Arguments

var	A tibble
name	A character name of the attribute

Value

A tibble with an additional attribute

drop_class *Remove class to abject*

Description

Remove class to abject

Usage

```
drop_class(var, name)
```

Arguments

var	A tibble
name	A character name of the class

Value

A tibble with an additional attribute

ensembl_symbol_mapping *Data set*

Description

Data set

Usage

```
ensembl_symbol_mapping
```

Format

An object of class spec_tbl_df (inherits from tbl_df, tbl, data.frame) with 291249 rows and 3 columns.

`ensembl_to_symbol` *Add transcript symbol column from ensembl id for human and mouse data*

Description

ensembl_to_symbol() takes as imput a ‘tbl‘ formatted as | <SAMPLE> | <ENSEMBL_ID> | <COUNT> | <...> | and returns a ‘tbl‘ with the additional transcript symbol column

Usage

```
ensembl_to_symbol(.data, .ensembl, action = "add")
```

Arguments

`.data`	A ‘tbl‘ formatted as	<SAMPLE>	<ENSEMBL_ID>	<COUNT>	<...>	
`.ensembl`	A character string. The column that is represents ensembl gene id
`action`	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).

Details

Maturing

This is useful since different resources use ensembl IDs while others use gene symbol IDs. At the moment this work for human (genes and transcripts) and mouse (genes) data.

Value

A ‘tbl‘ object including additional columns for transcript symbol

Examples

```
ensembl_to_symbol(tidybulk::counts_ensembl, ens)
```

`ensembl_to_symbol,spec_tbl_df-method` *ensembl_to_symbol*

Description

ensembl_to_symbol

Usage

```
## S4 method for signature 'spec_tbl_df'
ensembl_to_symbol(.data, .ensembl, action = "add")
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <ENSEMBL_ID> | <COUNT> | <...> |
.ensembl	A character string. The column that is represents ensembl gene id
action	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).

Value

A ‘tbl‘ object including additional columns for transcript symbol

---

ensembl_to_symbol,tbl_df-method

ensembl_to_symbol

Description

ensembl_to_symbol

Usage

```
## S4 method for signature 'tbl_df'
ensembl_to_symbol(.data, .ensembl, action = "add")
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <ENSEMBL_ID> | <COUNT> | <...> |
.ensembl	A character string. The column that is represents ensembl gene id
action	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).

Value

A ‘tbl‘ object including additional columns for transcript symbol

---

ensembl_to_symbol,tidybulk-method

ensembl_to_symbol

Description

ensembl_to_symbol

Usage

```
## S4 method for signature 'tidybulk'
ensembl_to_symbol(.data, .ensembl, action = "add")
```

Arguments

.data	A ‘tbl‘ formatted as	<SAMPLE>	<ENSEMBL_ID>	<COUNT>	<...>	
.ensembl	A character string. The column that is represents ensembl gene id
action	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).

Value

A ‘tbl‘ object including additional columns for transcript symbol

---

error_if_counts_is_na *Check whether there are NA counts*

Description

Check whether there are NA counts

Usage

```
error_if_counts_is_na(.data, .abundance)
```

Arguments

.data	A tibble of read counts
.abundance	A character name of the read count column

Value

A tbl

---

error_if_duplicated_genes *Check whether there are duplicated genes/transcripts*

Description

Check whether there are duplicated genes/transcripts

Usage

```
error_if_duplicated_genes(
  .data,
  .sample = sample,
  .transcript = transcript,
  .abundance = `read count`
)
```

Arguments

.data	A tibble of read counts
.sample	A character name of the sample column
.transcript	A character name of the transcript/gene column
.abundance	A character name of the read count column

Value

A tbl

---

error_if_log_transformed *Check whether a numeric vector has been log transformed*

Description

Check whether a numeric vector has been log transformed

Usage

```
error_if_log_transformed(x, .abundance)
```

Arguments

x	A numeric vector
.abundance	A character name of the transcript/gene abundance column

Value

NA

---

error_if_wrong_input *Check whether there are NA counts*

Description

Check whether there are NA counts

Usage

```
error_if_wrong_input(.data, list_input, expected_type)
```

Arguments

.data	A tibble of read counts
list_input	A list
expected_type	A character string

Value

A tbl

`fill_NA_using_formula` *This function is needed for DE in case the matrix is not rectangular, but includes NA*

Description

This function is needed for DE in case the matrix is not rectangular, but includes NA

Usage

```
fill_NA_using_formula(
  .data,
  .formula,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  .abundance_scaled = NULL
)
```

Arguments

`.data`	A tibble
`.formula`	a formula with no response variable, of the kind ~ factor_of_intrest + batch
`.sample`	The name of the sample column
`.transcript`	The name of the transcript/gene column
`.abundance`	The name of the transcript/gene abundance column
`.abundance_scaled`	The name of the transcript/gene scaled abundance column

Value

A tibble with adjusted counts

`fill_NA_with_row_median` *This function is needed for DE in case the matrix is not rectangular, but includes NA*

Description

This function is needed for DE in case the matrix is not rectangular, but includes NA

Usage

```
fill_NA_with_row_median(.matrix)
```

Arguments

`.matrix`	A matrix

Value

A matrix

`filter` *Subset rows using column values*

Description

‘filter()‘ retains the rows where the conditions you provide a ‘TRUE‘. Note that, unlike base subsetting with ‘[‘, rows where the condition evaluates to ‘NA‘ are dropped.

Most data operations are done on groups defined by variables. ‘group_by()‘ takes an existing tbl and converts it into a grouped tbl where operations are performed "by group". ‘ungroup()‘ removes grouping.

Usage

```
filter(.data, ..., .preserve = FALSE)

filter(.data, ..., .preserve = FALSE)
```

Arguments

`.data`	A data frame, data frame extension (e.g. a tibble), or a lazy data frame (e.g. from dbplyr or dtplyr). See *Methods*, below, for more details.
`...`	In ‘group_by()‘, variables or computations to group by. In ‘ungroup()‘, variables to remove from the grouping.
`.preserve`	when ‘FALSE‘ (the default), the grouping structure is recalculated based on the resulting data, otherwise it is kept as is.

Details

dplyr is not yet smart enough to optimise filtering optimisation on grouped datasets that don’t need grouped calculations. For this reason, filtering is often considerably faster on [ungroup()]ed data.

Value

An object of the same type as ‘.data‘.

* Rows are a subset of the input, but appear in the same order. * Columns are not modified. * The number of groups may be reduced (if ‘.preserve‘ is not ‘TRUE‘). * Data frame attributes are preserved.

A [grouped data frame][grouped_df()], unless the combination of ‘...‘ and ‘add‘ yields a non empty set of grouping columns, a regular (ungrouped) data frame otherwise.

Useful filter functions

* [‘==‘], [‘>‘], [‘>=‘] etc * [‘&‘], [‘|‘], [‘¡], [xor()] * [is.na()] * [between()], [near()]

Grouped tibbles

Because filtering expressions are computed within groups, they may yield different results on grouped tibbles. This will be the case as soon as an aggregating, lagging, or ranking function is involved. Compare this ungrouped filtering:

The former keeps rows with ‘mass‘ greater than the global average whereas the latter keeps rows with ‘mass‘ greater than the gender

average.

Methods

This function is a **generic**, which means that packages can provide implementations (methods) for other classes. See the documentation of individual methods for extra arguments and differences in behaviour.

The following methods are currently available in loaded packages:

These function are **generic**s, which means that packages can provide implementations (methods) for other classes. See the documentation of individual methods for extra arguments and differences in behaviour.

Methods available in currently loaded packages:

See Also

[filter_all()], [filter_if()] and [filter_at()].

Other single table verbs: `arrange()`, `mutate()`, `rename()`, `summarise()`

Other grouping functions: `group_by()`

Examples

```
# Learn more in ?dplyr_tidy_eval
`%>%` = magrittr::`%>%`
by_cyl <- mtcars %>% group_by(cyl)

# grouping doesn't change how the data looks (apart from listing
# how it's grouped):
by_cyl

# It changes how it acts with the other dplyr verbs:
by_cyl %>% summarise(
  disp = mean(disp),
  hp = mean(hp)
)
by_cyl %>% filter(disp == max(disp))

# Each call to summarise() removes a layer of grouping
`%>%` = magrittr::`%>%`
by_vs_am <- mtcars %>% group_by(vs, am)
by_vs <- by_vs_am %>% summarise(n = n())
by_vs
by_vs %>% summarise(n = sum(n))

# To removing grouping, use ungroup
by_vs %>%
```

```
  ungroup() %>%
  summarise(n = sum(n))

# You can group by expressions: this is just short-hand for
# a mutate() followed by a group_by()
mtcars %>% group_by(vsam = vs + am)



# when factors are involved, groups can be empty
tbl <- tibble(
  x = 1:10,
  y = factor(rep(c("a", "c"), each  = 5), levels = c("a", "b", "c"))
)
```

`flybaseIDs` *flybaseIDs*

Description

flybaseIDs

Usage

```
flybaseIDs
```

Format

An object of class `character` of length 14599.

`full_join` *Full join datasets*

Description

Full join datasets

Usage

```
full_join(x, y, by = NULL, copy = FALSE, suffix = c(".x", ".y"), ...)
```

Arguments

x	tbls to join. (See dplyr)
y	tbls to join. (See dplyr)
by	A character vector of variables to join by. (See dplyr)
copy	If x and y are not from the same data source, and copy is TRUE, then y will be copied into the same src as x. (See dplyr)

suffix	If there are non-joined duplicate variables in x and y, these suffixes will be added to the output to disambiguate them. Should be a character vector of length 2. (See dplyr)
...	Data frames to combine (See dplyr)

Value

A tt object

Examples

```
`%>%` = magrittr::`%>%`
annotation = tidybulk::counts %>% distinct(sample) %>% mutate(source = "AU")
tidybulk::counts %>% full_join(annotation)
```

---

get_abundance_norm_if_exists

Get column names either from user or from attributes

Description

Get column names either from user or from attributes

Usage

```
get_abundance_norm_if_exists(.data, .abundance)
```

Arguments

.data	A tibble
.abundance	A character name of the abundance column

Value

A list of column enquo or error

---

get_adjusted_counts_for_unwanted_variation_bulk

Get adjusted count for some batch effect

Description

Get adjusted count for some batch effect

Usage

```
get_adjusted_counts_for_unwanted_variation_bulk(
  .data,
  .formula,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  log_transform = TRUE,
  ...
)
```

Arguments

.data	A tibble
.formula	a formula with no response variable, of the kind ~ factor_of_intrest + batch
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
log_transform	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)
...	Further parameters passed to the function sva::ComBat

Value

A tibble with adjusted counts

---

get_cell_type_proportions

Get cell type proportions from cibersort

Description

Get cell type proportions from cibersort

Usage

```
get_cell_type_proportions(
  .data,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  reference = X_cibersort,
  method = "cibersort",
  ...
)
```

Arguments

.data	A tibble
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
reference	A data frame. The transcript/cell_type data frame of integer transcript abundance
method	A character string. The method to be used. At the moment Cibersort (default) and llsr (linear least squares regression) are available.
...	Further parameters passed to the function Cibersort

Value

A tibble including additional columns

---

get_clusters_kmeans_bulk

Get K-mean clusters to a tibble

Description

Get K-mean clusters to a tibble

Usage

```
get_clusters_kmeans_bulk(
  .data,
  .element = NULL,
  .feature = NULL,
  .abundance = NULL,
  of_samples = TRUE,
  log_transform = TRUE,
  ...
)
```

Arguments

.data	A tibble
.element	A column symbol. The column that is used to calculate distance (i.e., normally genes)
.feature	A column symbol. The column that is represents entities to cluster (i.e., normally samples)
.abundance	A column symbol with the value the clustering is based on (e.g., 'count')
of_samples	A boolean
log_transform	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)
...	Further parameters passed to the function kmeans

Value

A tibble with additional columns

`get_clusters_SNN_bulk` *Get SNN shared nearest neighbour clusters to a tibble*

Description

Get SNN shared nearest neighbour clusters to a tibble

Usage

```
get_clusters_SNN_bulk(
  .data,
  .element = NULL,
  .feature = NULL,
  .abundance,
  of_samples = TRUE,
  log_transform = TRUE,
  ...
)
```

Arguments

`.data`	A tibble
`.element`	A column symbol. The column that is used to calculate distance (i.e., normally genes)
`.feature`	A column symbol. The column that is represents entities to cluster (i.e., normally samples)
`.abundance`	A column symbol with the value the clustering is based on (e.g., 'count')
`of_samples`	A boolean
`log_transform`	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)
`...`	Further parameters passed to the function kmeans

Value

A tibble with additional columns

`get_differential_transcript_abundance_bulk` *Get differential transcription information to a tibble using edgeR.*

Description

Get differential transcription information to a tibble using edgeR.

Usage

```
get_differential_transcript_abundance_bulk(
  .data,
  .formula,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  .contrasts = NULL,
  significance_threshold = 0.05,
  minimum_counts = 10,
  minimum_proportion = 0.7,
  fill_missing_values = FALSE,
  scaling_method = "TMM",
  omit_contrast_in_colnames = FALSE
)
```

Arguments

`.data`	A tibble
`.formula`	a formula with no response variable, referring only to numeric variables
`.sample`	The name of the sample column
`.transcript`	The name of the transcript/gene column
`.abundance`	The name of the transcript/gene abundance column
`.contrasts`	A character vector. See edgeR makeContrasts specification for the parameter 'contrasts'. If contrasts are not present the first covariate is the one the model is tested against (e.g., ~ factor_of_interest)
`significance_threshold`	A real between 0 and 1
`minimum_counts`	A positive integer. Minimum counts required for at least some samples.
`minimum_proportion`	A real positive number between 0 and 1. It is the threshold of proportion of samples for each transcripts/genes that have to be characterised by a cmp bigger than the threshold to be included for scaling procedure.
`fill_missing_values`	A boolean. Whether to fill missing sample/transcript values with the median of the transcript. This is rarely needed.
`scaling_method`	A character string. The scaling method passed to the backend function (i.e., edgeR::calcNormFactors; "TMM","TMMwsp","RLE","upperquartile")
`omit_contrast_in_colnames`	If just one contrast is specified you can choose to omit the contrast label in the colnames.

Value

A tibble with edgeR results

`get_elements` *Get column names either from user or from attributes*

Description

Get column names either from user or from attributes

Usage

```
get_elements(.data, .element, of_samples = TRUE)
```

Arguments

`.data`	A tibble
`.element`	A character name of the sample column
`of_samples`	A boolean

Value

A list of column enquo or error

`get_elements_features` *Get column names either from user or from attributes*

Description

Get column names either from user or from attributes

Usage

```
get_elements_features(.data, .element, .feature, of_samples = TRUE)
```

Arguments

`.data`	A tibble
`.element`	A character name of the sample column
`.feature`	A character name of the transcript/gene column
`of_samples`	A boolean

Value

A list of column enquo or error

`get_elements_features_abundance`
Get column names either from user or from attributes

Description

Get column names either from user or from attributes

Usage

```
get_elements_features_abundance(
  .data,
  .element,
  .feature,
  .abundance,
  of_samples = TRUE
)
```

Arguments

`.data`	A tibble
`.element`	A character name of the sample column
`.feature`	A character name of the transcript/gene column
`.abundance`	A character name of the read count column
`of_samples`	A boolean

Value

A list of column enquo or error

`get_reduced_dimensions_MDS_bulk`
Get dimensionality information to a tibble using MDS

Description

Get dimensionality information to a tibble using MDS

Usage

```
get_reduced_dimensions_MDS_bulk(
  .data,
  .element = NULL,
  .feature = NULL,
  .abundance = NULL,
  .dims = 2,
  top = 500,
  of_samples = TRUE,
  log_transform = TRUE
)
```

Arguments

.data	A tibble
.element	A column symbol. The column that is used to calculate distance (i.e., normally samples)
.feature	A column symbol. The column that is represents entities to cluster (i.e., normally genes)
.abundance	A column symbol with the value the clustering is based on (e.g., 'count')
.dims	A integer vector corresponding to principal components of interest (e.g., 1:6)
top	An integer. How many top genes to select
of_samples	A boolean
log_transform	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)

Value

A tibble with additional columns

---

get_reduced_dimensions_PCA_bulk

Get principal component information to a tibble using PCA

Description

Get principal component information to a tibble using PCA

Usage

```
get_reduced_dimensions_PCA_bulk(
  .data,
  .element = NULL,
  .feature = NULL,
  .abundance = NULL,
  .dims = 2,
  top = 500,
  of_samples = TRUE,
  log_transform = TRUE,
  scale = FALSE,
  ...
)
```

Arguments

.data	A tibble
.element	A column symbol. The column that is used to calculate distance (i.e., normally samples)
.feature	A column symbol. The column that is represents entities to cluster (i.e., normally genes)
.abundance	A column symbol with the value the clustering is based on (e.g., 'count')

.dims	A integer vector corresponding to principal components of interest (e.g., 1:6)
top	An integer. How many top genes to select
of_samples	A boolean
log_transform	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)
scale	A boolean
...	Further parameters passed to the function prcomp

Value

A tibble with additional columns

`get_reduced_dimensions_TSNE_bulk` *Get principal component information to a tibble using tSNE*

Description

Get principal component information to a tibble using tSNE

Usage

```
get_reduced_dimensions_TSNE_bulk(
  .data,
  .element = NULL,
  .feature = NULL,
  .abundance = NULL,
  .dims = 2,
  top = 500,
  of_samples = TRUE,
  log_transform = TRUE,
  ...
)
```

Arguments

.data	A tibble
.element	A column symbol. The column that is used to calculate distance (i.e., normally samples)
.feature	A column symbol. The column that is represents entities to cluster (i.e., normally genes)
.abundance	A column symbol with the value the clustering is based on (e.g., 'count')
.dims	A integer vector corresponding to principal components of interest (e.g., 1:6)
top	An integer. How many top genes to select
of_samples	A boolean
log_transform	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)
...	Further parameters passed to the function Rtsne

Value

A tibble with additional columns

`get_rotated_dimensions` *Get rotated dimensions of two principal components or MDS dimension of choice, of an angle*

Description

Get rotated dimensions of two principal components or MDS dimension of choice, of an angle

Usage

```
get_rotated_dimensions(
  .data,
  dimension_1_column,
  dimension_2_column,
  rotation_degrees,
  .element = NULL,
  of_samples = TRUE,
  dimension_1_column_rotated = NULL,
  dimension_2_column_rotated = NULL
)
```

Arguments

`.data`	A tibble
`dimension_1_column`	A column symbol. The column of the dimension 1
`dimension_2_column`	A column symbol. The column of the dimension 2
`rotation_degrees`	A real number between 0 and 360
`.element`	A column symbol. The column that is used to calculate distance (i.e., normally samples)
`of_samples`	A boolean
`dimension_1_column_rotated`	A column symbol. The column of the dimension 1 rotated
`dimension_2_column_rotated`	A column symbol. The column of the dimension 2 rotated

Value

A tibble with additional rotated columns

get_sample — *Get column names either from user or from attributes*

Description

Get column names either from user or from attributes

Usage

```
get_sample(.data, .sample)
```

Arguments

.data	A tibble
.sample	A character name of the sample column

Value

A list of column enquo or error

get_sample_counts — *Get column names either from user or from attributes*

Description

Get column names either from user or from attributes

Usage

```
get_sample_counts(.data, .sample, .abundance)
```

Arguments

.data	A tibble
.sample	A character name of the sample column
.abundance	A character name of the read count column

Value

A list of column enquo or error

get_sample_transcript *Get column names either from user or from attributes*

Description

Get column names either from user or from attributes

Usage

```
get_sample_transcript(.data, .sample, .transcript)
```

Arguments

.data	A tibble
.sample	A character name of the sample column
.transcript	A character name of the transcript/gene column

Value

A list of column enquo or error

get_sample_transcript_counts *Get column names either from user or from attributes*

Description

Get column names either from user or from attributes

Usage

```
get_sample_transcript_counts(.data, .sample, .transcript, .abundance)
```

Arguments

.data	A tibble
.sample	A character name of the sample column
.transcript	A character name of the transcript/gene column
.abundance	A character name of the read count column

Value

A list of column enquo or error

`get_scaled_counts_bulk` *Get a tibble with scaled counts using TMM*

Description

Get a tibble with scaled counts using TMM

Usage

```
get_scaled_counts_bulk(
  .data,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  factor_of_interest = NULL,
  minimum_counts = 10,
  minimum_proportion = 0.7,
  method = "TMM",
  reference_selection_function = median
)
```

Arguments

`.data`	A tibble
`.sample`	The name of the sample column
`.transcript`	The name of the transcript/gene column
`.abundance`	The name of the transcript/gene abundance column
`factor_of_interest`	The name of the column of the factor of interest. This is used for identifying lowly abundant transcript, to be ignored for calculating scaling fators.
`minimum_counts`	A positive integer. Minimum counts required for at least some samples.
`minimum_proportion`	A real positive number between 0 and 1. It is the threshold of proportion of samples for each transcripts/genes that have to be characterised by a cmp bigger than the threshold to be included for scaling procedure.
`method`	A character string. The scaling method passed to the backend function (i.e., edgeR::calcNormFactors; "TMM","TMMwsp","RLE","upperquartile")
`reference_selection_function`	A function between median, mean and max

Value

A tibble including additional columns

`get_symbol_from_ensembl` *after wget, this function merges hg37 and hg38 mapping data bases - Do not execute!*

Description

Get transcript column from ensembl gene id

Usage

```
get_symbol_from_ensembl(.data, .ensembl)
```

Arguments

`.data`	A tibble
`.ensembl`	A column symbol. The column that is represents ensembl gene id

Value

A tibble with ensembl-transcript mapping

after wget, this function merges hg37 and hg38 mapping data bases - Do not execute!

A tibble with ensembl-transcript mapping

get_symbol_from_ensembl

A tibble with added annotation

`get_transcript` *Get column names either from user or from attributes*

Description

Get column names either from user or from attributes

Usage

```
get_transcript(.data, .transcript)
```

Arguments

`.data`	A tibble
`.transcript`	A character name of the transcript column

Value

A list of column enquo or error

get_x_y_annotation_columns *get_x_y_annotation_columns*

Description

This function recognise what are the sample-wise columns and transcrip-wise columns

Usage

```
get_x_y_annotation_columns(
  .data,
  .horizontal,
  .vertical,
  .abundance,
  .abundance_scaled
)
```

Arguments

.data	A ‘tbl‘ formatted as	<SAMPLE>	<TRANSCRIPT>	<COUNT>	<...>	
.horizontal	The name of the column horizontally presented in the heatmap
.vertical	The name of the column vertically presented in the heatmap
.abundance	The name of the transcript/gene abundance column
.abundance_scaled	The name of the transcript/gene scaled abundance column

Value

A list

group_by *Group by one or more variables*

Description

Most data operations are done on groups defined by variables. ‘group_by()‘ takes an existing tbl and converts it into a grouped tbl where operations are performed "by group". ‘ungroup()‘ removes grouping.

Usage

```
group_by(.data, ..., .add = FALSE, .drop = group_by_drop_default(.data))

ungroup(x, ...)
```

Arguments

.data	A data frame, data frame extension (e.g. a tibble), or a lazy data frame (e.g. from dbplyr or dtplyr). See *Methods*, below, for more details.
...	In 'group_by()', variables or computations to group by. In 'ungroup()', variables to remove from the grouping.
.add	When 'FALSE', the default, 'group_by()' will override existing groups. To add to the existing groups, use '.add = TRUE'. This argument was previously called 'add', but that prevented creating a new grouping variable called 'add', and conflicts with our naming conventions.
.drop	When '.drop = TRUE', empty groups are dropped. See [group_by_drop_default()] for what the default value is for this argument.
x	A [tbl()]

Value

A [grouped data frame][grouped_df()], unless the combination of '...' and 'add' yields a non empty set of grouping columns, a regular (ungrouped) data frame otherwise.

Methods

These function are **generic**s, which means that packages can provide implementations (methods) for other classes. See the documentation of individual methods for extra arguments and differences in behaviour.

Methods available in currently loaded packages:

See Also

Other grouping functions: `filter()`

Examples

```
`%>%` = magrittr::`%>%`
by_cyl <- mtcars %>% group_by(cyl)

# grouping doesn't change how the data looks (apart from listing
# how it's grouped):
by_cyl

# It changes how it acts with the other dplyr verbs:
by_cyl %>% summarise(
  disp = mean(disp),
  hp = mean(hp)
)
by_cyl %>% filter(disp == max(disp))

# Each call to summarise() removes a layer of grouping
by_vs_am <- mtcars %>% group_by(vs, am)
by_vs <- by_vs_am %>% summarise(n = n())
by_vs
by_vs %>% summarise(n = sum(n))

# To removing grouping, use ungroup
by_vs %>%
```

```
  ungroup() %>%
  summarise(n = sum(n))

# You can group by expressions: this is just short-hand for
# a mutate() followed by a group_by()
mtcars %>% group_by(vsam = vs + am)



# when factors are involved, groups can be empty
tbl <- tibble(
  x = 1:10,
  y = factor(rep(c("a", "c"), each  = 5), levels = c("a", "b", "c"))
)
```

`ifelse2_pipe` *This is a generalisation of ifelse that acceots an object and return an objects*

Description

This is a generalisation of ifelse that acceots an object and return an objects

Usage

```
ifelse2_pipe(.x, .p1, .p2, .f1, .f2, .f3 = NULL)
```

Arguments

`.x`	A tibble
`.p1`	A boolean
`.p2`	ELSE IF condition
`.f1`	A function
`.f2`	A function
`.f3`	A function

Value

A tibble

`ifelse_pipe` *This is a generalisation of ifelse that acceots an object and return an objects*

Description

This is a generalisation of ifelse that acceots an object and return an objects

Usage

```
ifelse_pipe(.x, .p, .f1, .f2 = NULL)
```

Arguments

`.x`	A tibble
`.p`	A boolean
`.f1`	A function
`.f2`	A function

Value

A tibble

`impute_abundance` *Impute transcript abundance if missing from sample-transcript pairs*

Description

impute_abundance() takes as imput a 'tbl' formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> | and returns a 'tbl' with an edditional adjusted abundance column. This method uses scaled counts if present.

Usage

```
impute_abundance(
  .data,
  .formula,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL
)
```

Arguments

`.data`	A 'tbl' formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
`.formula`	A formula with no response variable, representing the desired linear model where the first covariate is the factor of interest and the second covariate is the unwanted variation (of the kind ~ factor_of_intrest + batch)
`.sample`	The name of the sample column
`.transcript`	The name of the transcript/gene column
`.abundance`	The name of the transcript/gene abundance column

Details

Maturing

This function imputes the abundance of missing sample-transcript pair using the median of the sample group defined by the formula

Value

A ‘tbl‘ non-sparse abundance

Examples

```
res =
impute_abundance(
tidybulk::counts_mini,
~ condition,
.sample = sample,
.transcript = transcript,
.abundance = count
)
```

---

impute_abundance,RangedSummarizedExperiment-method
impute_abundance

Description

impute_abundance

Usage

```
## S4 method for signature 'RangedSummarizedExperiment'
impute_abundance(
  .data,
  .formula,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL
)
```

Arguments

.data	A ‘tbl‘ formatted as	<SAMPLE>	<TRANSCRIPT>	<COUNT>	<...>	
.formula	A formula with no response variable, representing the desired linear model where the first covariate is the factor of interest and the second covariate is the unwanted variation (of the kind ~ factor_of_intrest + batch)
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column

Value

A 'SummarizedExperiment' object

`impute_abundance,spec_tbl_df-method`
impute_abundance

Description

impute_abundance

Usage

```
## S4 method for signature 'spec_tbl_df'
impute_abundance(
  .data,
  .formula,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL
)
```

Arguments

`.data`	A 'tbl' formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
`.formula`	A formula with no response variable, representing the desired linear model where the first covariate is the factor of interest and the second covariate is the unwanted variation (of the kind ~ factor_of_intrest + batch)
`.sample`	The name of the sample column
`.transcript`	The name of the transcript/gene column
`.abundance`	The name of the transcript/gene abundance column

Value

A 'tbl' with imputed abundnce

`impute_abundance,SummarizedExperiment-method`
impute_abundance

Description

impute_abundance

Usage

```
## S4 method for signature 'SummarizedExperiment'
impute_abundance(
  .data,
  .formula,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL
)
```

Arguments

.data	A ‘tbl‘ formatted as	<SAMPLE>	<TRANSCRIPT>	<COUNT>	<...>	
.formula	A formula with no response variable, representing the desired linear model where the first covariate is the factor of interest and the second covariate is the unwanted variation (of the kind ~ factor_of_intrest + batch)
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column

Value

A ‘SummarizedExperiment‘ object

---

impute_abundance,tbl_df-method
impute_abundance

Description

impute_abundance

Usage

```
## S4 method for signature 'tbl_df'
impute_abundance(
  .data,
  .formula,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL
)
```

Arguments

.data	A ‘tbl‘ formatted as	<SAMPLE>	<TRANSCRIPT>	<COUNT>	<...>	
.formula	A formula with no response variable, representing the desired linear model where the first covariate is the factor of interest and the second covariate is the unwanted variation (of the kind ~ factor_of_intrest + batch)

.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column

Value

A ‘tbl‘ with imputed abundnce

`impute_abundance,tidybulk-method`
impute_abundance

Description

impute_abundance

Usage

```
## S4 method for signature 'tidybulk'
impute_abundance(
  .data,
  .formula,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL
)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.formula	A formula with no response variable, representing the desired linear model where the first covariate is the factor of interest and the second covariate is the unwanted variation (of the kind ~ factor_of_intrest + batch)
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column

Value

A ‘tbl‘ with imputed abundnce

inner_join *Inner join datasets*

Description

Inner join datasets

Usage

```
inner_join(x, y, by = NULL, copy = FALSE, suffix = c(".x", ".y"), ...)
```

Arguments

x	tbls to join. (See dplyr)
y	tbls to join. (See dplyr)
by	A character vector of variables to join by. (See dplyr)
copy	If x and y are not from the same data source, and copy is TRUE, then y will be copied into the same src as x. (See dplyr)
suffix	If there are non-joined duplicate variables in x and y, these suffixes will be added to the output to disambiguate them. Should be a character vector of length 2. (See dplyr)
...	Data frames to combine (See dplyr)

Value

A tt object

Examples

```
`%>%` = magrittr::`%>%`
annotation = tidybulk::counts %>% distinct(sample) %>% mutate(source = "AU")
tidybulk::counts %>% inner_join(annotation)
```

keep_abundant *Filter abundant transcripts*

Description

keep_abundant() takes as imput a ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> | and returns a ‘tbl‘ with additional columns for the statistics from the hypothesis test.

Usage

```
keep_abundant(
  .data,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  factor_of_interest = NULL,
  minimum_counts = 10,
  minimum_proportion = 0.7
)
```

Arguments

.data	A 'tbl' formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
factor_of_interest	The name of the column of the factor of interest. This is used for defining sample groups for the filtering process.
minimum_counts	A real positive number. It is the threshold of count per million that is used to filter transcripts/genes out from the scaling procedure.
minimum_proportion	A real positive number between 0 and 1. It is the threshold of proportion of samples for each transcripts/genes that have to be characterised by a cmp bigger than the threshold to be included for scaling procedure.

Details

Maturing

At the moment this function uses edgeR only, but other inference algorithms will be added in the near future.

Value

A 'tbl' with additional columns for the statistics from the hypothesis test (e.g., log fold change, p-value and false discovery rate).

Examples

```
keep_abundant(
tidybulk::counts_mini,
    sample,
    transcript,
    `count`
)
```

keep_abundant,RangedSummarizedExperiment-method
keep_abundant

Description

keep_abundant

Usage

```
## S4 method for signature 'RangedSummarizedExperiment'
keep_abundant(
  .data,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  factor_of_interest = NULL,
  minimum_counts = 10,
  minimum_proportion = 0.7
)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
factor_of_interest	The name of the column of the factor of interest. This is used for defining sample groups for the filtering process.
minimum_counts	A real positive number. It is the threshold of count per million that is used to filter transcripts/genes out from the scaling procedure.
minimum_proportion	A real positive number between 0 and 1. It is the threshold of proportion of samples for each transcripts/genes that have to be characterised by a cmp bigger than the threshold to be included for scaling procedure.

Value

A ‘SummarizedExperiment‘ object

keep_abundant,spec_tbl_df-method
keep_abundant

Description

keep_abundant

Usage

```
## S4 method for signature 'spec_tbl_df'
keep_abundant(
  .data,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  factor_of_interest = NULL,
  minimum_counts = 10,
  minimum_proportion = 0.7
)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
factor_of_interest	The name of the column of the factor of interest. This is used for defining sample groups for the filtering process.
minimum_counts	A real positive number. It is the threshold of count per million that is used to filter transcripts/genes out from the scaling procedure.
minimum_proportion	A real positive number between 0 and 1. It is the threshold of proportion of samples for each transcripts/genes that have to be characterised by a cmp bigger than the threshold to be included for scaling procedure.

Value

A ‘tbl‘ with additional columns for the statistics from the hypothesis test (e.g., log fold change, p-value and false discovery rate).

keep_abundant,SummarizedExperiment-method
keep_abundant

Description

keep_abundant

Usage

```
## S4 method for signature 'SummarizedExperiment'
keep_abundant(
  .data,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  factor_of_interest = NULL,
  minimum_counts = 10,
  minimum_proportion = 0.7
)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
factor_of_interest	The name of the column of the factor of interest. This is used for defining sample groups for the filtering process.
minimum_counts	A real positive number. It is the threshold of count per million that is used to filter transcripts/genes out from the scaling procedure.
minimum_proportion	A real positive number between 0 and 1. It is the threshold of proportion of samples for each transcripts/genes that have to be characterised by a cmp bigger than the threshold to be included for scaling procedure.

Value

A ‘SummarizedExperiment‘ object

keep_abundant,tbl_df-method
keep_abundant

Description

keep_abundant

Usage

```
## S4 method for signature 'tbl_df'
keep_abundant(
  .data,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  factor_of_interest = NULL,
  minimum_counts = 10,
  minimum_proportion = 0.7
)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
factor_of_interest	The name of the column of the factor of interest. This is used for defining sample groups for the filtering process.
minimum_counts	A real positive number. It is the threshold of count per million that is used to filter transcripts/genes out from the scaling procedure.
minimum_proportion	A real positive number between 0 and 1. It is the threshold of proportion of samples for each transcripts/genes that have to be characterised by a cmp bigger than the threshold to be included for scaling procedure.

Value

A ‘tbl‘ with additional columns for the statistics from the hypothesis test (e.g., log fold change, p-value and false discovery rate).

keep_abundant,tidybulk-method
keep_abundant

Description

keep_abundant

Usage

```
## S4 method for signature 'tidybulk'
keep_abundant(
  .data,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  factor_of_interest = NULL,
  minimum_counts = 10,
  minimum_proportion = 0.7
)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
factor_of_interest	The name of the column of the factor of interest. This is used for defining sample groups for the filtering process.
minimum_counts	A real positive number. It is the threshold of count per million that is used to filter transcripts/genes out from the scaling procedure.
minimum_proportion	A real positive number between 0 and 1. It is the threshold of proportion of samples for each transcripts/genes that have to be characterised by a cmp bigger than the threshold to be included for scaling procedure.

Value

A ‘tbl‘ with additional columns for the statistics from the hypothesis test (e.g., log fold change, p-value and false discovery rate).

keep_variable *Filter variable transcripts*

Description

keep_variable() takes as imput a ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> | and returns a ‘tbl‘ with additional columns for the statistics from the hypothesis test.

Usage

```
keep_variable(
  .data,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  top = 500,
  log_transform = TRUE
)
```

Arguments

.data	A ‘tbl‘ formatted as	<SAMPLE>	<TRANSCRIPT>	<COUNT>	<...>	
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
top	Integer. Number of top transcript to consider
log_transform	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)

Details

Maturing

At the moment this function uses edgeR only, but other inference algorithms will be added in the near future.

Value

A ‘tbl‘ with additional columns for the statistics from the hypothesis test (e.g., log fold change, p-value and false discovery rate).

Examples

```
keep_variable(
tidybulk::counts_mini,
    sample,
    transcript,
    `count`,
```

```
    top = 500
)
```

---

keep_variable,RangedSummarizedExperiment-method
keep_variable

Description

keep_variable

Usage

```
## S4 method for signature 'RangedSummarizedExperiment'
keep_variable(
  .data,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  top = 500,
  log_transform = TRUE
)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
top	Integer. Number of top transcript to consider
log_transform	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)

Value

A ‘SummarizedExperiment‘ object

keep_variable,spec_tbl_df-method
keep_variable

Description

keep_variable

Usage

```
## S4 method for signature 'spec_tbl_df'
keep_variable(
  .data,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  top = 500,
  log_transform = TRUE
)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
top	Integer. Number of top transcript to consider
log_transform	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)

Value

A ‘tbl‘ with additional columns for the statistics from the hypothesis test (e.g., log fold change, p-value and false discovery rate).

keep_variable,SummarizedExperiment-method
keep_variable

Description

keep_variable

Usage

```
## S4 method for signature 'SummarizedExperiment'
keep_variable(
  .data,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  top = 500,
  log_transform = TRUE
)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
top	Integer. Number of top transcript to consider
log_transform	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)

Value

A ‘SummarizedExperiment‘ object

---

keep_variable,tbl_df-method

keep_variable

Description

keep_variable

Usage

```
## S4 method for signature 'tbl_df'
keep_variable(
  .data,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  top = 500,
  log_transform = TRUE
)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
top	Integer. Number of top transcript to consider
log_transform	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)

Value

A ‘tbl‘ with additional columns for the statistics from the hypothesis test (e.g., log fold change, p-value and false discovery rate).

---

keep_variable,tidybulk-method *keep_variable*

Description

keep_variable

Usage

```
## S4 method for signature 'tidybulk'
keep_variable(
  .data,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  top = 500,
  log_transform = TRUE
)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
top	Integer. Number of top transcript to consider
log_transform	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)

Value

A ‘tbl‘ with additional columns for the statistics from the hypothesis test (e.g., log fold change, p-value and false discovery rate).

keep_variable_transcripts *Identify variable genes for dimensionality reduction*

Description

Identify variable genes for dimensionality reduction

Usage

```
keep_variable_transcripts(
  .data,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  top = 500,
  log_transform = TRUE
)
```

Arguments

.data	A tibble
.sample	A character name of the sample column
.transcript	A character name of the transcript/gene column
.abundance	A character name of the read count column
top	An integer. How many top genes to select
log_transform	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)

Value

A tibble filtered genes

left_join *Left join datasets*

Description

Left join datasets

Usage

```
left_join(x, y, by = NULL, copy = FALSE, suffix = c(".x", ".y"), ...)
```

Arguments

x	tbls to join. (See dplyr)
y	tbls to join. (See dplyr)
by	A character vector of variables to join by. (See dplyr)
copy	If x and y are not from the same data source, and copy is TRUE, then y will be copied into the same src as x. (See dplyr)
suffix	If there are non-joined duplicate variables in x and y, these suffixes will be added to the output to disambiguate them. Should be a character vector of length 2. (See dplyr)
...	Data frames to combine (See dplyr)

Value

A tt object

Examples

```
`%>%` = magrittr::`%>%`
annotation = tidybulk::counts %>% distinct(sample) %>% mutate(source = "AU")
tidybulk::counts %>% left_join(annotation)
```

---

mutate *Create, modify, and delete columns*

Description

'mutate()' adds new variables and preserves existing ones; 'transmute()' adds new variables and drops existing ones. New variables overwrite existing variables of the same name. Variables can be removed by setting their value to 'NULL'.

Usage

```
mutate(.data, ...)
```

Arguments

.data	A data frame, data frame extension (e.g. a tibble), or a lazy data frame (e.g. from dbplyr or dtplyr). See *Methods*, below, for more details.
...	<['tidy-eval'][dplyr_tidy_eval]> Name-value pairs. The name gives the name of the column in the output. The value can be: * A vector of length 1, which will be recycled to the correct length. * A vector the same length as the current group (or the whole data frame if ungrouped). * 'NULL', to remove the column. * A data frame or tibble, to create multiple columns in the output.

Value

An object of the same type as ‘.data‘.

For ‘mutate()‘:

* Rows are not affected. * Existing columns will be preserved unless explicitly modified. * New columns will be added to the right of existing columns. * Columns given value ‘NULL‘ will be removed * Groups will be recomputed if a grouping variable is mutated. * Data frame attributes are preserved.

For ‘transmute()‘:

* Rows are not affected. * Apart from grouping variables, existing columns will be remove unless explicitly kept. * Column order matches order of expressions. * Groups will be recomputed if a grouping variable is mutated. * Data frame attributes are preserved.

Useful mutate functions

* [‘+‘], [‘-‘], [log()], etc., for their usual mathematical meanings

* [lead()], [lag()]

* [dense_rank()], [min_rank()], [percent_rank()], [row_number()], [cume_dist()], [ntile()]

* [cumsum()], [cummean()], [cummin()], [cummax()], [cumany()], [cumall()]

* [na_if()], [coalesce()]

* [if_else()], [recode()], [case_when()]

Grouped tibbles

Because mutating expressions are computed within groups, they may yield different results on grouped tibbles. This will be the case as soon as an aggregating, lagging, or ranking function is involved. Compare this ungrouped mutate:

With the grouped equivalent:

The former normalises ‘mass‘ by the global average whereas the latter normalises by the averages within gender levels.

Methods

These function are **generic**s, which means that packages can provide implementations (methods) for other classes. See the documentation of individual methods for extra arguments and differences in behaviour.

Methods available in currently loaded packages:

See Also

Other single table verbs: `arrange()`, `filter()`, `rename()`, `summarise()`

Examples

```
`%>%` = magrittr::`%>%`
# Newly created variables are available immediately
mtcars %>% as_tibble() %>% mutate(
  cyl2 = cyl * 2,
  cyl4 = cyl2 * 2
)
```

nest *nest*

Description

nest

Usage

```
nest(.data, ...)
```

Arguments

.data	A tbl. (See tidyr)
...	Name-variable pairs of the form new_col = c(col1, col2, col3) (See tidyr)

Value

A tt object

Examples

```
nest(tidybulk(tidybulk::counts_mini, sample, transcript, count), data = -transcript)
```

parse_formula *.formula parser*

Description

.formula parser

Usage

```
parse_formula(fm)
```

Arguments

fm	a formula

Value

A character vector

`pivot_sample` *Extract sampe-wise information*

Description

pivot_sample() takes as imput a ‘tbl‘ formatted as | <SAMPLE> | <ENSEMBL_ID> | <COUNT> | <...> | and returns a ‘tbl‘ with only sampe-related columns

Usage

```
pivot_sample(.data, .sample = NULL)
```

Arguments

`.data`	A ‘tbl‘ formatted as	<SAMPLE>	<TRANSCRIPT>	<COUNT>	<...>	
`.sample`	The name of the sample column

Details

Maturing

This functon extracts only sample-related information for downstream analysis (e.g., visualisation). It is disruptive in the sense that it cannot be passed anymore to tidybulk function.

Value

A ‘tbl‘ object

Examples

```
pivot_sample(
tidybulk::counts_mini,
.sample = sample
)
```

`pivot_sample,spec_tbl_df-method` *pivot_sample*

Description

pivot_sample

Usage

```
## S4 method for signature 'spec_tbl_df'
pivot_sample(.data, .sample = NULL)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.sample	The name of the sample column

Value

A ‘tbl‘ object

---

pivot_sample,tbl_df-method *pivot_sample*

Description

pivot_sample

Usage

```
## S4 method for signature 'tbl_df'
pivot_sample(.data, .sample = NULL)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.sample	The name of the sample column

Value

A ‘tbl‘ object

---

pivot_sample,tidybulk-method *pivot_sample*

Description

pivot_sample

Usage

```
## S4 method for signature 'tidybulk'
pivot_sample(.data, .sample = NULL)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.sample	The name of the sample column

Value

A ‘tbl‘ object

`pivot_transcript` *Extract transcript-wise information*

Description

pivot_transcript() takes as imput a ‘tbl‘ formatted as | <SAMPLE> | <ENSEMBL_ID> | <COUNT> | <...> | and returns a ‘tbl‘ with only sampe-related columns

Usage

```
pivot_transcript(.data, .transcript = NULL)
```

Arguments

`.data`	A ‘tbl‘ formatted as	<SAMPLE>	<TRANSCRIPT>	<COUNT>	<...>	
`.transcript`	The name of the transcript column

Details

Maturing

This functon extracts only transcript-related information for downstream analysis (e.g., visualisation). It is disruptive in the sense that it cannot be passed anymore to tidybulk function.

Value

A ‘tbl‘ object

Examples

```
pivot_transcript(
tidybulk::counts_mini,
.transcript = transcript
)
```

`pivot_transcript,spec_tbl_df-method` *pivot_transcript*

Description

pivot_transcript

Usage

```
## S4 method for signature 'spec_tbl_df'
pivot_transcript(.data, .transcript = NULL)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.transcript	The name of the transcript column

Value

A ‘tbl‘ object

---

pivot_transcript,tbl_df-method
pivot_transcript

Description

pivot_transcript

Usage

```
## S4 method for signature 'tbl_df'
pivot_transcript(.data, .transcript = NULL)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.transcript	The name of the transcript column

Value

A ‘tbl‘ object

---

pivot_transcript,tidybulk-method
pivot_transcript

Description

pivot_transcript

Usage

```
## S4 method for signature 'tidybulk'
pivot_transcript(.data, .transcript = NULL)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.transcript	The name of the transcript column

Value

A ‘tbl‘ object

prepend *From rlang deprecated*

Description

From rlang deprecated

Usage

```
prepend(x, values, before = 1)
```

Arguments

x	An array
values	An array
before	A boolean

Value

An array

reduce_dimensions *Dimension reduction of the transcript abundance data*

Description

reduce_dimensions() takes as imput a 'tbl' formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> | and calculates the reduced dimensional space of the transcript abundance.

Usage

```
reduce_dimensions(
  .data,
  .element = NULL,
  .feature = NULL,
  .abundance = NULL,
  method,
  .dims = 2,
  top = 500,
  of_samples = TRUE,
  log_transform = TRUE,
  scale = TRUE,
  action = "add",
  ...
)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.element	The name of the element column (normally samples).
.feature	The name of the feature column (normally transcripts/genes)
.abundance	The name of the column including the numerical value the clustering is based on (normally transcript abundance)
method	A character string. The dimension reduction algorithm to use (PCA, MDS, tSNE).
.dims	A list of integer vectors corresponding to principal components of interest (e.g., list(1:2, 3:4, 5:6))
top	An integer. How many top genes to select for dimensionality reduction
of_samples	A boolean. In case the input is a tidybulk object, it indicates Whether the element column will be sample or transcript column
log_transform	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)
scale	A boolean for method="PCA", this will be passed to the ‘prcomp‘ function. It is not included in the ... argument because although the default for ‘prcomp‘ if FALSE, it is advisable to set it as TRUE.
action	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).
...	Further parameters passed to the function prcomp if you choose method="PCA" or Rtsne if you choose method="tSNE"

Details

Maturing

This function reduces the dimensions of the transcript abundances. It can use multi-dimensional scaling (MDS) of principal component analysis (PCA).

Value

A tbl object with additional columns for the reduced dimensions

Examples

```
counts.MDS = reduce_dimensions(tidybulk::counts_mini, sample, transcript, count, method="MDS", .dims = 3)

counts.PCA = reduce_dimensions(tidybulk::counts_mini, sample, transcript, count, method="PCA", .dims = 3)
```

`reduce_dimensions,RangedSummarizedExperiment-method`
reduce_dimensions

Description

reduce_dimensions

Usage

```
## S4 method for signature 'RangedSummarizedExperiment'
reduce_dimensions(
  .data,
  .element = NULL,
  .feature = NULL,
  .abundance = NULL,
  method,
  .dims = 2,
  top = 500,
  of_samples = TRUE,
  log_transform = TRUE,
  scale = TRUE,
  action = "add",
  ...
)
```

Arguments

`.data`	A ‘tbl‘ formatted as	<SAMPLE>	<TRANSCRIPT>	<COUNT>	<...>	
`.element`	The name of the element column (normally samples).
`.feature`	The name of the feature column (normally transcripts/genes)
`.abundance`	The name of the column including the numerical value the clustering is based on (normally transcript abundance)
`method`	A character string. The dimension reduction algorithm to use (PCA, MDS, tSNE).
`.dims`	A list of integer vectors corresponding to principal components of interest (e.g., list(1:2, 3:4, 5:6))
`top`	An integer. How many top genes to select for dimensionality reduction
`of_samples`	A boolean. In case the input is a tidybulk object, it indicates Whether the element column will be sample or transcript column
`log_transform`	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)
`scale`	A boolean for method="PCA", this will be passed to the ‘prcomp‘ function. It is not included in the ... argument because although the default for ‘prcomp‘ if FALSE, it is advisable to set it as TRUE.
`action`	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).
`...`	Further parameters passed to the function prcomp if you choose method="PCA" or Rtsne if you choose method="tSNE"

Value

A 'SummarizedExperiment' object

`reduce_dimensions,spec_tbl_df-method`
reduce_dimensions

Description

reduce_dimensions

Usage

```
## S4 method for signature 'spec_tbl_df'
reduce_dimensions(
  .data,
  .element = NULL,
  .feature = NULL,
  .abundance = NULL,
  method,
  .dims = 2,
  top = 500,
  of_samples = TRUE,
  log_transform = TRUE,
  scale = TRUE,
  action = "add",
  ...
)
```

Arguments

`.data`	A 'tbl' formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
`.element`	The name of the element column (normally samples).
`.feature`	The name of the feature column (normally transcripts/genes)
`.abundance`	The name of the column including the numerical value the clustering is based on (normally transcript abundance)
`method`	A character string. The dimension reduction algorithm to use (PCA, MDS, tSNE).
`.dims`	A list of integer vectors corresponding to principal components of interest (e.g., list(1:2, 3:4, 5:6))
`top`	An integer. How many top genes to select for dimensionality reduction
`of_samples`	A boolean. In case the input is a tidybulk object, it indicates Whether the element column will be sample or transcript column
`log_transform`	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)
`scale`	A boolean for method="PCA", this will be passed to the 'prcomp' function. It is not included in the ... argument because although the default for 'prcomp' if FALSE, it is advisable to set it as TRUE.

action	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).
...	Further parameters passed to the function prcomp if you choose method="PCA" or Rtsne if you choose method="tSNE"

Value

A tbl object with additional columns for the reduced dimensions

---

`reduce_dimensions,SummarizedExperiment-method`
reduce_dimensions

Description

reduce_dimensions

Usage

```
## S4 method for signature 'SummarizedExperiment'
reduce_dimensions(
  .data,
  .element = NULL,
  .feature = NULL,
  .abundance = NULL,
  method,
  .dims = 2,
  top = 500,
  of_samples = TRUE,
  log_transform = TRUE,
  scale = TRUE,
  action = "add",
  ...
)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.element	The name of the element column (normally samples).
.feature	The name of the feature column (normally transcripts/genes)
.abundance	The name of the column including the numerical value the clustering is based on (normally transcript abundance)
method	A character string. The dimension reduction algorithm to use (PCA, MDS, tSNE).
.dims	A list of integer vectors corresponding to principal components of interest (e.g., list(1:2, 3:4, 5:6))
top	An integer. How many top genes to select for dimensionality reduction
of_samples	A boolean. In case the input is a tidybulk object, it indicates Whether the element column will be sample or transcript column

log_transform	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)
scale	A boolean for method="PCA", this will be passed to the ‘prcomp‘ function. It is not included in the ... argument because although the default for ‘prcomp‘ if FALSE, it is advisable to set it as TRUE.
action	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).
...	Further parameters passed to the function prcomp if you choose method="PCA" or Rtsne if you choose method="tSNE"

Value

A ‘SummarizedExperiment‘ object

`reduce_dimensions,tbl_df-method`
reduce_dimensions

Description

reduce_dimensions

Usage

```
## S4 method for signature 'tbl_df'
reduce_dimensions(
  .data,
  .element = NULL,
  .feature = NULL,
  .abundance = NULL,
  method,
  .dims = 2,
  top = 500,
  of_samples = TRUE,
  log_transform = TRUE,
  scale = TRUE,
  action = "add",
  ...
)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.element	The name of the element column (normally samples).
.feature	The name of the feature column (normally transcripts/genes)
.abundance	The name of the column including the numerical value the clustering is based on (normally transcript abundance)
method	A character string. The dimension reduction algorithm to use (PCA, MDS, tSNE).

.dims	A list of integer vectors corresponding to principal components of interest (e.g., list(1:2, 3:4, 5:6))
top	An integer. How many top genes to select for dimensionality reduction
of_samples	A boolean. In case the input is a tidybulk object, it indicates Whether the element column will be sample or transcript column
log_transform	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)
scale	A boolean for method="PCA", this will be passed to the ‘prcomp‘ function. It is not included in the ... argument because although the default for ‘prcomp‘ if FALSE, it is advisable to set it as TRUE.
action	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).
...	Further parameters passed to the function prcomp if you choose method="PCA" or Rtsne if you choose method="tSNE"

Value

A tbl object with additional columns for the reduced dimensions

---

`reduce_dimensions,tidybulk-method`
reduce_dimensions

Description

reduce_dimensions

Usage

```
## S4 method for signature 'tidybulk'
reduce_dimensions(
  .data,
  .element = NULL,
  .feature = NULL,
  .abundance = NULL,
  method,
  .dims = 2,
  top = 500,
  of_samples = TRUE,
  log_transform = TRUE,
  scale = TRUE,
  action = "add",
  ...
)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.element	The name of the element column (normally samples).
.feature	The name of the feature column (normally transcripts/genes)
.abundance	The name of the column including the numerical value the clustering is based on (normally transcript abundance)
method	A character string. The dimension reduction algorithm to use (PCA, MDS, tSNE).
.dims	A list of integer vectors corresponding to principal components of interest (e.g., list(1:2, 3:4, 5:6))
top	An integer. How many top genes to select for dimensionality reduction
of_samples	A boolean. In case the input is a tidybulk object, it indicates Whether the element column will be sample or transcript column
log_transform	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)
scale	A boolean for method="PCA", this will be passed to the ‘prcomp‘ function. It is not included in the ... argument because although the default for ‘prcomp‘ if FALSE, it is advisable to set it as TRUE.
action	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).
...	Further parameters passed to the function prcomp if you choose method="PCA" or Rtsne if you choose method="tSNE"

Value

A tbl object with additional columns for the reduced dimensions

---

remove_redundancy *Drop redundant elements (e.g., samples) for which feature (e.g., transcript/gene) aboundances are correlated*

Description

remove_redundancy() takes as imput a ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> | for correlation method or | <DIMENSION 1> | <DIMENSION 2> | <...> | for reduced_dimensions method, and returns a ‘tbl‘ with dropped elements (e.g., samples).

Usage

```
remove_redundancy(
  .data,
  .element = NULL,
  .feature = NULL,
  .abundance = NULL,
  method,
  of_samples = TRUE,
  correlation_threshold = 0.9,
```

```
  top = Inf,
  log_transform = FALSE,
  Dim_a_column,
  Dim_b_column
)
```

Arguments

.data	A 'tbl' formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.element	The name of the element column (normally samples).
.feature	The name of the feature column (normally transcripts/genes)
.abundance	The name of the column including the numerical value the clustering is based on (normally transcript abundance)
method	A character string. The cluster algorithm to use, ay the moment k-means is the only algorithm included.
of_samples	A boolean. In case the input is a tidybulk object, it indicates Whether the element column will be sample or transcript column
correlation_threshold	A real number between 0 and 1. For correlation based calculation.
top	An integer. How many top genes to select for correlation based method
log_transform	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)
Dim_a_column	A character string. For reduced_dimension based calculation. The column of one principal component
Dim_b_column	A character string. For reduced_dimension based calculation. The column of another principal component

Details

Maturing

This function removes redundant elements from the original data set (e.g., samples or transcripts). For example, if we want to define cell-type specific signatures with low sample redundancy. This function returns a tibble with dropped recundant elements (e.g., samples). Two redundancy estimation approaches are supported: (i) removal of highly correlated clusters of elements (keeping a representative) with method="correlation"; (ii) removal of most proximal element pairs in a reduced dimensional space.

Value

A tbl object with with dropped recundant elements (e.g., samples).

Examples

```
remove_redundancy(
 tidybulk::counts_mini,
.element = sample,
.feature = transcript,
 .abundance =  count,
```

```
    method = "correlation"
    )

counts.MDS =  reduce_dimensions(tidybulk::counts_mini, sample, transcript, count, method="MDS", .dims = 3)

remove_redundancy(
counts.MDS,
Dim_a_column = `Dim1`,
Dim_b_column = `Dim2`,
.element = sample,
  method = "reduced_dimensions"
)
```

`remove_redundancy,RangedSummarizedExperiment-method`
remove_redundancy

Description

remove_redundancy

Usage

```
## S4 method for signature 'RangedSummarizedExperiment'
remove_redundancy(
  .data,
  .element = NULL,
  .feature = NULL,
  .abundance = NULL,
  method,
  of_samples = TRUE,
  correlation_threshold = 0.9,
  top = Inf,
  log_transform = FALSE,
  Dim_a_column = NULL,
  Dim_b_column = NULL
)
```

Arguments

`.data`	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
`.element`	The name of the element column (normally samples).
`.feature`	The name of the feature column (normally transcripts/genes)
`.abundance`	The name of the column including the numerical value the clustering is based on (normally transcript abundance)
`method`	A character string. The cluster algorithm to use, ay the moment k-means is the only algorithm included.
`of_samples`	A boolean. In case the input is a tidybulk object, it indicates Whether the element column will be sample or transcript column

`correlation_threshold`
: A real number between 0 and 1. For correlation based calculation.

`top`
: An integer. How many top genes to select for correlation based method

`log_transform`
: A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)

`Dim_a_column`
: A character string. For reduced_dimension based calculation. The column of one principal component

`Dim_b_column`
: A character string. For reduced_dimension based calculation. The column of another principal component

Value

A 'SummarizedExperiment' object

`remove_redundancy,spec_tbl_df-method`
remove_redundancy

Description

remove_redundancy

Usage

```
## S4 method for signature 'spec_tbl_df'
remove_redundancy(
  .data,
  .element = NULL,
  .feature = NULL,
  .abundance = NULL,
  method,
  of_samples = TRUE,
  correlation_threshold = 0.9,
  top = Inf,
  log_transform = FALSE,
  Dim_a_column = NULL,
  Dim_b_column = NULL
)
```

Arguments

`.data`
: A 'tbl' formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |

`.element`
: The name of the element column (normally samples).

`.feature`
: The name of the feature column (normally transcripts/genes)

`.abundance`
: The name of the column including the numerical value the clustering is based on (normally transcript abundance)

`method`
: A character string. The cluster algorithm to use, ay the moment k-means is the only algorithm included.

of_samples	A boolean. In case the input is a tidybulk object, it indicates Whether the element column will be sample or transcript column
correlation_threshold	A real number between 0 and 1. For correlation based calculation.
top	An integer. How many top genes to select for correlation based method
log_transform	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)
Dim_a_column	A character string. For reduced_dimension based calculation. The column of one principal component
Dim_b_column	A character string. For reduced_dimension based calculation. The column of another principal component

Value

A tbl object with with dropped recundant elements (e.g., samples).

`remove_redundancy,SummarizedExperiment-method`
remove_redundancy

Description

remove_redundancy

Usage

```
## S4 method for signature 'SummarizedExperiment'
remove_redundancy(
  .data,
  .element = NULL,
  .feature = NULL,
  .abundance = NULL,
  method,
  of_samples = TRUE,
  correlation_threshold = 0.9,
  top = Inf,
  log_transform = FALSE,
  Dim_a_column = NULL,
  Dim_b_column = NULL
)
```

Arguments

.data	A ‘tbl‘ formatted as	<SAMPLE>	<TRANSCRIPT>	<COUNT>	<...>	
.element	The name of the element column (normally samples).
.feature	The name of the feature column (normally transcripts/genes)
.abundance	The name of the column including the numerical value the clustering is based on (normally transcript abundance)

method	A character string. The cluster algorithm to use, ay the moment k-means is the only algorithm included.
of_samples	A boolean. In case the input is a tidybulk object, it indicates Whether the element column will be sample or transcript column
correlation_threshold	A real number between 0 and 1. For correlation based calculation.
top	An integer. How many top genes to select for correlation based method
log_transform	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)
Dim_a_column	A character string. For reduced_dimension based calculation. The column of one principal component
Dim_b_column	A character string. For reduced_dimension based calculation. The column of another principal component

Value

A 'SummarizedExperiment' object

`remove_redundancy,tbl_df-method`
remove_redundancy

Description

remove_redundancy

Usage

```
## S4 method for signature 'tbl_df'
remove_redundancy(
  .data,
  .element = NULL,
  .feature = NULL,
  .abundance = NULL,
  method,
  of_samples = TRUE,
  correlation_threshold = 0.9,
  top = Inf,
  log_transform = FALSE,
  Dim_a_column = NULL,
  Dim_b_column = NULL
)
```

Arguments

.data	A 'tbl' formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.element	The name of the element column (normally samples).
.feature	The name of the feature column (normally transcripts/genes)

.abundance	The name of the column including the numerical value the clustering is based on (normally transcript abundance)
method	A character string. The cluster algorithm to use, ay the moment k-means is the only algorithm included.
of_samples	A boolean. In case the input is a tidybulk object, it indicates Whether the element column will be sample or transcript column
correlation_threshold	A real number between 0 and 1. For correlation based calculation.
top	An integer. How many top genes to select for correlation based method
log_transform	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)
Dim_a_column	A character string. For reduced_dimension based calculation. The column of one principal component
Dim_b_column	A character string. For reduced_dimension based calculation. The column of another principal component

Value

A tbl object with with dropped recundant elements (e.g., samples).

`remove_redundancy,tidybulk-method`
remove_redundancy

Description

remove_redundancy

Usage

```
## S4 method for signature 'tidybulk'
remove_redundancy(
  .data,
  .element = NULL,
  .feature = NULL,
  .abundance = NULL,
  method,
  of_samples = TRUE,
  correlation_threshold = 0.9,
  top = Inf,
  log_transform = FALSE,
  Dim_a_column = NULL,
  Dim_b_column = NULL
)
```

Arguments

.data	A 'tbl' formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.element	The name of the element column (normally samples).
.feature	The name of the feature column (normally transcripts/genes)
.abundance	The name of the column including the numerical value the clustering is based on (normally transcript abundance)
method	A character string. The cluster algorithm to use, ay the moment k-means is the only algorithm included.
of_samples	A boolean. In case the input is a tidybulk object, it indicates Whether the element column will be sample or transcript column
correlation_threshold	A real number between 0 and 1. For correlation based calculation.
top	An integer. How many top genes to select for correlation based method
log_transform	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)
Dim_a_column	A character string. For reduced_dimension based calculation. The column of one principal component
Dim_b_column	A character string. For reduced_dimension based calculation. The column of another principal component

Value

A tbl object with with dropped recundant elements (e.g., samples).

---

remove_redundancy_elements_though_reduced_dimensions

Identifies the closest pairs in a MDS contaxt and return one of them

---

Description

Identifies the closest pairs in a MDS contaxt and return one of them

Usage

```
remove_redundancy_elements_though_reduced_dimensions(
  .data,
  Dim_a_column,
  Dim_b_column,
  .element = NULL,
  of_samples = TRUE
)
```

Arguments

.data	A tibble
Dim_a_column	A column symbol. The column of one principal component
Dim_b_column	A column symbol. The column of another principal component
.element	A column symbol. The column that is represents entities to cluster (i.e., normally samples)
of_samples	A boolean

Value

A tibble with pairs dropped

`remove_redundancy_elements_through_correlation`
Drop redundant elements (e.g., samples) for which feature (e.g., genes) aboundances are correlated

---

Description

Drop redundant elements (e.g., samples) for which feature (e.g., genes) aboundances are correlated

Usage

```
remove_redundancy_elements_through_correlation(
  .data,
  .element = NULL,
  .feature = NULL,
  .abundance = NULL,
  correlation_threshold = 0.9,
  top = Inf,
  of_samples = TRUE,
  log_transform = FALSE
)
```

Arguments

`.data`	A tibble
`.element`	A column symbol. The column that is used to calculate distance (i.e., normally samples)
`.feature`	A column symbol. The column that is represents entities to cluster (i.e., normally genes)
`.abundance`	A column symbol with the value the clustering is based on (e.g., ‘count‘)
`correlation_threshold`	A real number between 0 and 1
`top`	An integer. How many top genes to select
`of_samples`	A boolean
`log_transform`	A boolean, whether the value should be log-transformed (e.g., TRUE for RNA sequencing data)

Value

A tibble with redundant elemens removed

`rename` *Rename columns*

Description

Rename individual variables using ‘new_name = old_name‘ syntax.

Usage

```
rename(.data, ...)
```

Arguments

`.data`	A data frame, data frame extension (e.g. a tibble), or a lazy data frame (e.g. from dbplyr or dtplyr). See *Methods*, below, for more details.
`...`	<[‘tidy-select‘][dplyr_tidy_select]> Use ‘new_name = old_name‘ to rename selected variables.

Value

An object of the same type as ‘.data‘. * Rows are not affected. * Column names are changed; column order is preserved * Data frame attributes are preserved. * Groups are updated to reflect new names.

Scoped selection and renaming

Use the three scoped variants ([rename_all()], [rename_if()], [rename_at()]) to renaming a set of variables with a function.

Methods

This function is a **generic**, which means that packages can provide implementations (methods) for other classes. See the documentation of individual methods for extra arguments and differences in behaviour.

The following methods are currently available in loaded packages:

See Also

Other single table verbs: `arrange()`, `filter()`, `mutate()`, `summarise()`

Examples

```
`%>%` = magrittr::`%>%`
iris <- as_tibble(iris) # so it prints a little nicer
rename(iris, petal_length = Petal.Length)
```

right_join *Right join datasets*

Description

Right join datasets

Usage

```
right_join(x, y, by = NULL, copy = FALSE, suffix = c(".x", ".y"), ...)
```

Arguments

x	tbls to join. (See dplyr)
y	tbls to join. (See dplyr)
by	A character vector of variables to join by. (See dplyr)
copy	If x and y are not from the same data source, and copy is TRUE, then y will be copied into the same src as x. (See dplyr)
suffix	If there are non-joined duplicate variables in x and y, these suffixes will be added to the output to disambiguate them. Should be a character vector of length 2. (See dplyr)
...	Data frames to combine (See dplyr)

Value

A tt object

Examples

```
`%>%` = magrittr::`%>%`
annotation = tidybulk::counts %>% distinct(sample) %>% mutate(source = "AU")
tidybulk::counts %>% right_join(annotation)
```

rotate_dimensions *Rotate two dimensions (e.g., principal components) of an arbitrary angle*

Description

rotate_dimensions() takes as imput a ‘tbl‘ formatted as | <DIMENSION 1> | <DIMENSION 2> | <...> | and calculates the rotated dimensional space of the transcript abundance.

Usage

```
rotate_dimensions(
  .data,
  dimension_1_column,
  dimension_2_column,
  rotation_degrees,
  .element = NULL,
  of_samples = TRUE,
  dimension_1_column_rotated = NULL,
  dimension_2_column_rotated = NULL,
  action = "add"
)
```

Arguments

.data	A 'tbl' formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
dimension_1_column	A character string. The column of the dimension 1
dimension_2_column	A character string. The column of the dimension 2
rotation_degrees	A real number between 0 and 360
.element	The name of the element column (normally samples).
of_samples	A boolean. In case the input is a tidybulk object, it indicates Whether the element column will be sample or transcript column
dimension_1_column_rotated	A character string. The column of the rotated dimension 1 (optional)
dimension_2_column_rotated	A character string. The column of the rotated dimension 2 (optional)
action	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).

Details

Maturing

This function to rotate two dimensions such as the reduced dimensions.

Value

A tbl object with additional columns for the reduced dimensions. additional columns for the rotated dimensions. The rotated dimensions will be added to the original data set as '<NAME OF DIMENSION> rotated <ANGLE>' by default, or as specified in the input arguments.

Examples

```
counts.MDS =  reduce_dimensions(tidybulk::counts_mini, sample, transcript, count, method="MDS", .dims = 3)

counts.MDS.rotated =  rotate_dimensions(counts.MDS, `Dim1`, `Dim2`, rotation_degrees = 45, .element = sample)
```

rotate_dimensions,RangedSummarizedExperiment-method
rotate_dimensions

Description

rotate_dimensions

Usage

```
## S4 method for signature 'RangedSummarizedExperiment'
rotate_dimensions(
  .data,
  dimension_1_column,
  dimension_2_column,
  rotation_degrees,
  .element = NULL,
  of_samples = TRUE,
  dimension_1_column_rotated = NULL,
  dimension_2_column_rotated = NULL,
  action = "add"
)
```

Arguments

`.data`	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
`dimension_1_column`	A character string. The column of the dimension 1
`dimension_2_column`	A character string. The column of the dimension 2
`rotation_degrees`	A real number between 0 and 360
`.element`	The name of the element column (normally samples).
`of_samples`	A boolean. In case the input is a tidybulk object, it indicates Whether the element column will be sample or transcript column
`dimension_1_column_rotated`	A character string. The column of the rotated dimension 1 (optional)
`dimension_2_column_rotated`	A character string. The column of the rotated dimension 2 (optional)
`action`	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).

Value

A ‘SummarizedExperiment‘ object

`rotate_dimensions,spec_tbl_df-method`
rotate_dimensions

Description

rotate_dimensions

Usage

```
## S4 method for signature 'spec_tbl_df'
rotate_dimensions(
  .data,
  dimension_1_column,
  dimension_2_column,
  rotation_degrees,
  .element = NULL,
  of_samples = TRUE,
  dimension_1_column_rotated = NULL,
  dimension_2_column_rotated = NULL,
  action = "add"
)
```

Arguments

`.data`	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
`dimension_1_column`	A character string. The column of the dimension 1
`dimension_2_column`	A character string. The column of the dimension 2
`rotation_degrees`	A real number between 0 and 360
`.element`	The name of the element column (normally samples).
`of_samples`	A boolean. In case the input is a tidybulk object, it indicates Whether the element column will be sample or transcript column
`dimension_1_column_rotated`	A character string. The column of the rotated dimension 1 (optional)
`dimension_2_column_rotated`	A character string. The column of the rotated dimension 2 (optional)
`action`	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).

Value

A tbl object with additional columns for the reduced dimensions. additional columns for the rotated dimensions. The rotated dimensions will be added to the original data set as ‘<NAME OF DIMENSION> rotated <ANGLE>‘ by default, or as specified in the input arguments.

`rotate_dimensions,SummarizedExperiment-method`
rotate_dimensions

Description

rotate_dimensions

Usage

```
## S4 method for signature 'SummarizedExperiment'
rotate_dimensions(
  .data,
  dimension_1_column,
  dimension_2_column,
  rotation_degrees,
  .element = NULL,
  of_samples = TRUE,
  dimension_1_column_rotated = NULL,
  dimension_2_column_rotated = NULL,
  action = "add"
)
```

Arguments

`.data`	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
`dimension_1_column`	A character string. The column of the dimension 1
`dimension_2_column`	A character string. The column of the dimension 2
`rotation_degrees`	A real number between 0 and 360
`.element`	The name of the element column (normally samples).
`of_samples`	A boolean. In case the input is a tidybulk object, it indicates Whether the element column will be sample or transcript column
`dimension_1_column_rotated`	A character string. The column of the rotated dimension 1 (optional)
`dimension_2_column_rotated`	A character string. The column of the rotated dimension 2 (optional)
`action`	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).

Value

A ‘SummarizedExperiment‘ object

rotate_dimensions,tbl_df-method
rotate_dimensions

Description

rotate_dimensions

Usage

```
## S4 method for signature 'tbl_df'
rotate_dimensions(
  .data,
  dimension_1_column,
  dimension_2_column,
  rotation_degrees,
  .element = NULL,
  of_samples = TRUE,
  dimension_1_column_rotated = NULL,
  dimension_2_column_rotated = NULL,
  action = "add"
)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
dimension_1_column	A character string. The column of the dimension 1
dimension_2_column	A character string. The column of the dimension 2
rotation_degrees	A real number between 0 and 360
.element	The name of the element column (normally samples).
of_samples	A boolean. In case the input is a tidybulk object, it indicates Whether the element column will be sample or transcript column
dimension_1_column_rotated	A character string. The column of the rotated dimension 1 (optional)
dimension_2_column_rotated	A character string. The column of the rotated dimension 2 (optional)
action	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).

Value

A tbl object with additional columns for the reduced dimensions. additional columns for the rotated dimensions. The rotated dimensions will be added to the original data set as ‘<NAME OF DIMENSION> rotated <ANGLE>‘ by default, or as specified in the input arguments.

rotate_dimensions,tidybulk-method
rotate_dimensions

Description

rotate_dimensions

Usage

```
## S4 method for signature 'tidybulk'
rotate_dimensions(
  .data,
  dimension_1_column,
  dimension_2_column,
  rotation_degrees,
  .element = NULL,
  of_samples = TRUE,
  dimension_1_column_rotated = NULL,
  dimension_2_column_rotated = NULL,
  action = "add"
)
```

Arguments

`.data`	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
`dimension_1_column`	A character string. The column of the dimension 1
`dimension_2_column`	A character string. The column of the dimension 2
`rotation_degrees`	A real number between 0 and 360
`.element`	The name of the element column (normally samples).
`of_samples`	A boolean. In case the input is a tidybulk object, it indicates Whether the element column will be sample or transcript column
`dimension_1_column_rotated`	A character string. The column of the rotated dimension 1 (optional)
`dimension_2_column_rotated`	A character string. The column of the rotated dimension 2 (optional)
`action`	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).

Value

A tbl object with additional columns for the reduced dimensions. additional columns for the rotated dimensions. The rotated dimensions will be added to the original data set as ‘<NAME OF DIMENSION> rotated <ANGLE>‘ by default, or as specified in the input arguments.

rowwise *Group input by rows*

Description

Questioning

Usage

```
rowwise(.data)
```

Arguments

.data	Input data frame.

Details

See [this repository](https://github.com/jennybc/row-oriented-workflows) for alternative ways to perform row-wise operations.

'rowwise()' is used for the results of [do()] when you create list-variables. It is also useful to support arbitrary complex operations that need to be applied to each row.

Currently, rowwise grouping only works with data frames. Its main impact is to allow you to work with list-variables in [summarise()] and [mutate()] without having to use `[[1]]`. This makes 'summarise()' on a rowwise tbl effectively equivalent to [plyr::ldply()].

Value

A 'tbl'

A 'tbl'

Examples

```
`%>%` = magrittr::`%>%`
df <- expand.grid(x = 1:3, y = 3:1)
df_done <- df %>% rowwise() %>% do(i = seq(.$x, .$y))
df_done
df_done %>% summarise(n = length(i))
```

run_llsr *Perform linear equation system analysis through llsr*

Description

Perform linear equation system analysis through llsr

Usage

```
run_llsr(mix, reference)
```

Arguments

mix	A data frame
reference	A data frame

Value

A data frame

---

scale_abundance *Scale the counts of transcripts/genes*

Description

scale_abundance() takes as imput a ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> | and Scales transcript abundance compansating for sequencing depth (e.g., with TMM algorithm, Robinson and Oshlack doi.org/10.1186/gb-2010-11-3-r25).

Usage

```
scale_abundance(
  .data,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  factor_of_interest = NULL,
  minimum_counts = 10,
  minimum_proportion = 0.7,
  method = "TMM",
  reference_selection_function = median,
  action = "add"
)
```

Arguments

.data	A ‘tbl‘ formatted as	<SAMPLE>	<TRANSCRIPT>	<COUNT>	<...>	
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
factor_of_interest	The name of the column of the factor of interest. This is used for identifying lowly abundant transcript, to be ignored for calculating scaling fators.
minimum_counts	A real positive number. It is the threshold of count per million that is used to filter transcripts/genes out from the scaling procedure. The scaling inference is then applied back to all unfiltered data.
minimum_proportion	A real positive number between 0 and 1. It is the threshold of proportion of samples for each transcripts/genes that have to be characterised by a cmp bigger than the threshold to be included for scaling procedure.

`method` A character string. The scaling method passed to the backend function (i.e., edgeR::calcNormFactors; "TMM","TMMwsp","RLE","upperquartile")

`reference_selection_function` A fucntion that is used to selecting the reference sample for scaling. It could be max (default), which choose the sample with maximum library size; or median, which chooses the sample with median library size.

`action` A character string between "add" (default) and "only". "add" joins the new information to the input tbl (default), "only" return a non-redundant tbl with the just new information.

Details

Maturing

Scales transcript abundance compansating for sequencing depth (e.g., with TMM algorithm, Robinson and Oshlack doi.org/10.1186/gb-2010-11-3-r25). Lowly transcribed transcripts/genes (defined with minimum_counts and minimum_proportion parameters) are filtered out from the scaling procedure. The scaling inference is then applied back to all unfiltered data.

Value

A tbl object with additional columns with scaled data as ‘<NAME OF COUNT COLUMN>_scaled‘

Examples

```
scale_abundance(tidybulk::counts_mini,  sample, transcript, `count`)
```

---

scale_abundance,RangedSummarizedExperiment-method
scale_abundance

Description

scale_abundance

Usage

```
## S4 method for signature 'RangedSummarizedExperiment'
scale_abundance(
  .data,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  factor_of_interest = NULL,
  minimum_counts = 10,
  minimum_proportion = 0.7,
  method = "TMM",
```

```
  reference_selection_function = median,
  action = "add"
)
```

Arguments

.data	A 'tbl' formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
factor_of_interest	The name of the column of the factor of interest. This is used for identifying lowly abundant transcript, to be ignored for calculating scaling fators.
minimum_counts	A real positive number. It is the threshold of count per million that is used to filter transcripts/genes out from the scaling procedure. The scaling inference is then applied back to all unfiltered data.
minimum_proportion	A real positive number between 0 and 1. It is the threshold of proportion of samples for each transcripts/genes that have to be characterised by a cmp bigger than the threshold to be included for scaling procedure.
method	A character string. The scaling method passed to the backend function (i.e., edgeR::calcNormFactors; "TMM","TMMwsp","RLE","upperquartile")
reference_selection_function	A fucntion that is used to selecting the reference sample for scaling. It could be max (default), which choose the sample with maximum library size; or median, which chooses the sample with median library size.
action	A character string between "add" (default) and "only". "add" joins the new information to the input tbl (default), "only" return a non-redundant tbl with the just new information.

Value

A 'SummarizedExperiment' object

---

scale_abundance,spec_tbl_df-method *scale_abundance*

Description

scale_abundance

Usage

```
## S4 method for signature 'spec_tbl_df'
scale_abundance(
  .data,
  .sample = NULL,
  .transcript = NULL,
```

```
  .abundance = NULL,
  factor_of_interest = NULL,
  minimum_counts = 10,
  minimum_proportion = 0.7,
  method = "TMM",
  reference_selection_function = median,
  action = "add"
)
```

Arguments

.data	A 'tbl' formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
factor_of_interest	The name of the column of the factor of interest. This is used for identifying lowly abundant transcript, to be ignored for calculating scaling fators.
minimum_counts	A real positive number. It is the threshold of count per million that is used to filter transcripts/genes out from the scaling procedure. The scaling inference is then applied back to all unfiltered data.
minimum_proportion	A real positive number between 0 and 1. It is the threshold of proportion of samples for each transcripts/genes that have to be characterised by a cmp bigger than the threshold to be included for scaling procedure.
method	A character string. The scaling method passed to the backend function (i.e., edgeR::calcNormFactors; "TMM","TMMwsp","RLE","upperquartile")
reference_selection_function	A fucntion that is used to selecting the reference sample for scaling. It could be max (default), which choose the sample with maximum library size; or median, which chooses the sample with median library size.
action	A character string between "add" (default) and "only". "add" joins the new information to the input tbl (default), "only" return a non-redundant tbl with the just new information.

Value

A tbl object with additional columns with scaled data as '<NAME OF COUNT COLUMN>_scaled'

---

scale_abundance,SummarizedExperiment-method *scale_abundance*

Description

scale_abundance

Usage

```
## S4 method for signature 'SummarizedExperiment'
scale_abundance(
  .data,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  factor_of_interest = NULL,
  minimum_counts = 10,
  minimum_proportion = 0.7,
  method = "TMM",
  reference_selection_function = median,
  action = "add"
)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
factor_of_interest	The name of the column of the factor of interest. This is used for identifying lowly abundant transcript, to be ignored for calculating scaling fators.
minimum_counts	A real positive number. It is the threshold of count per million that is used to filter transcripts/genes out from the scaling procedure. The scaling inference is then applied back to all unfiltered data.
minimum_proportion	A real positive number between 0 and 1. It is the threshold of proportion of samples for each transcripts/genes that have to be characterised by a cmp bigger than the threshold to be included for scaling procedure.
method	A character string. The scaling method passed to the backend function (i.e., edgeR::calcNormFactors; "TMM","TMMwsp","RLE","upperquartile")
reference_selection_function	A fucntion that is used to selecting the reference sample for scaling. It could be max (default), which choose the sample with maximum library size; or median, which chooses the sample with median library size.
action	A character string between "add" (default) and "only". "add" joins the new information to the input tbl (default), "only" return a non-redundant tbl with the just new information.

Value

A ‘SummarizedExperiment‘ object

scale_abundance,tbl_df-method
scale_abundance

Description

scale_abundance

Usage

```
## S4 method for signature 'tbl_df'
scale_abundance(
  .data,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  factor_of_interest = NULL,
  minimum_counts = 10,
  minimum_proportion = 0.7,
  method = "TMM",
  reference_selection_function = median,
  action = "add"
)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
factor_of_interest	The name of the column of the factor of interest. This is used for identifying lowly abundant transcript, to be ignored for calculating scaling fators.
minimum_counts	A real positive number. It is the threshold of count per million that is used to filter transcripts/genes out from the scaling procedure. The scaling inference is then applied back to all unfiltered data.
minimum_proportion	A real positive number between 0 and 1. It is the threshold of proportion of samples for each transcripts/genes that have to be characterised by a cmp bigger than the threshold to be included for scaling procedure.
method	A character string. The scaling method passed to the backend function (i.e., edgeR::calcNormFactors; "TMM","TMMwsp","RLE","upperquartile")
reference_selection_function	A fucntion that is used to selecting the reference sample for scaling. It could be max (default), which choose the sample with maximum library size; or median, which chooses the sample with median library size.
action	A character string between "add" (default) and "only". "add" joins the new information to the input tbl (default), "only" return a non-redundant tbl with the just new information.

Value

A tbl object with additional columns with scaled data as '<NAME OF COUNT COLUMN>_scaled'

`scale_abundance,tidybulk-method`
scale_abundance

Description

scale_abundance

Usage

```
## S4 method for signature 'tidybulk'
scale_abundance(
  .data,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  factor_of_interest = NULL,
  minimum_counts = 10,
  minimum_proportion = 0.7,
  method = "TMM",
  reference_selection_function = median,
  action = "add"
)
```

Arguments

`.data`	A 'tbl' formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
`.sample`	The name of the sample column
`.transcript`	The name of the transcript/gene column
`.abundance`	The name of the transcript/gene abundance column
`factor_of_interest`	The name of the column of the factor of interest. This is used for identifying lowly abundant transcript, to be ignored for calculating scaling fators.
`minimum_counts`	A real positive number. It is the threshold of count per million that is used to filter transcripts/genes out from the scaling procedure. The scaling inference is then applied back to all unfiltered data.
`minimum_proportion`	A real positive number between 0 and 1. It is the threshold of proportion of samples for each transcripts/genes that have to be characterised by a cmp bigger than the threshold to be included for scaling procedure.
`method`	A character string. The scaling method passed to the backend function (i.e., edgeR::calcNormFactors; "TMM","TMMwsp","RLE","upperquartile")
`reference_selection_function`	A fucntion that is used to selecting the reference sample for scaling. It could be max (default), which choose the sample with maximum library size; or median, which chooses the sample with median library size.

action	A character string between "add" (default) and "only". "add" joins the new information to the input tbl (default), "only" return a non-redundant tbl with the just new information.

Value

A tbl object with additional columns with scaled data as '<NAME OF COUNT COLUMN>_scaled'

---

scale_design *Scale design matrix*

Description

Scale design matrix

Usage

```
scale_design(df, .formula)
```

Arguments

df	A tibble
.formula	a formula

Value

A tibble

---

se *SummarizedExperiment*

Description

SummarizedExperiment

Usage

```
se
```

Format

An object of class `RangedSummarizedExperiment` with 100 rows and 8 columns.

select_closest_pairs *Sub function of remove_redundancy_elements_though_reduced_dimensions*

Description

Sub function of remove_redundancy_elements_though_reduced_dimensions

Usage

```
select_closest_pairs(df)
```

Arguments

df	A tibble

Value

A tibble with pairs to drop

se_mini *SummarizedExperiment mini for vignette*

Description

SummarizedExperiment mini for vignette

Usage

```
se_mini
```

Format

An object of class `SummarizedExperiment` with 527 rows and 5 columns.

summarise *Summarise each group to fewer rows*

Description

'summarise()' creates a new data frame. It will have one (or more) rows for each combination of grouping variables; if there are no grouping variables, the output will have a single row summarising all observations in the input. It will contain one column for each grouping variable and one column for each of the summary statistics that you have specified.

'summarise()' and 'summarize()' are synonyms.

Usage

```
summarise(.data, ...)
```

Arguments

.data	A data frame, data frame extension (e.g. a tibble), or a lazy data frame (e.g. from dbplyr or dtplyr). See *Methods*, below, for more details.
...	<[‘tidy-eval‘][dplyr_tidy_eval]> Name-value pairs of summary functions. The name will be the name of the variable in the result. The value can be: * A vector of length 1, e.g. ‘min(x)‘, ‘n()‘, or ‘sum(is.na(y))‘. * A vector of length ‘n‘, e.g. ‘quantile()‘. * A data frame, to add multiple columns from a single expression.

Value

An object _usually_ of the same type as ‘.data‘.

* The rows come from the underlying ‘group_keys()‘. * The columns are a combination of the grouping keys and the summary expressions that you provide. * If ‘x‘ is grouped by more than one variable, the output will be another [grouped_df] with the right-most group removed. * If ‘x‘ is grouped by one variable, or is not grouped, the output will be a [tibble]. * Data frame attributes are **not** preserved, because ‘summarise()‘ fundamentally creates a new data frame.

Useful functions

* Center: [mean()], [median()] * Spread: [sd()], [IQR()], [mad()] * Range: [min()], [max()], [quantile()] * Position: [first()], [last()], [nth()], * Count: [n()], [n_distinct()] * Logical: [any()], [all()]

Backend variations

The data frame backend supports creating a variable and using it in the same summary. This means that previously created summary variables can be further transformed or combined within the summary, as in [mutate()]. However, it also means that summary variables with the same names as previous variables overwrite them, making those variables unavailable to later summary variables.

This behaviour may not be supported in other backends. To avoid unexpected results, consider using new names for your summary variables, especially when creating multiple summaries.

Methods

This function is a **generic**, which means that packages can provide implementations (methods) for other classes. See the documentation of individual methods for extra arguments and differences in behaviour.

The following methods are currently available in loaded packages:

See Also

Other single table verbs: `arrange()`, `filter()`, `mutate()`, `rename()`

Examples

```
`%>%` = magrittr::`%>%`
# A summary applied to ungrouped tbl returns a single row
mtcars %>%
  summarise(mean = mean(disp), n = n())

# Usually, you'll want to group first
```

```
mtcars %>%
  group_by(cyl) %>%
  summarise(mean = mean(disp), n = n())

# dplyr 1.0.0 allows to summarise to more than one value:

# You use a data frame to create multiple columns so you can wrap
# this up into a function:
my_quantile <- function(x, probs) {
  tibble(x = quantile(x, probs), probs = probs)
}



# Refer to column names stored as strings with the `.data` pronoun:
var <- "mass"
# Learn more in ?dplyr_tidy_eval
```

`symbol_to_entrez` *Get ENTREZ id from gene SYMBOL*

Description

Get ENTREZ id from gene SYMBOL

Usage

```
symbol_to_entrez(.data, .transcript = NULL, .sample = NULL)
```

Arguments

`.data`	A tt or tbl object.
`.transcript`	A character. The name of the ene symbol column.
`.sample`	The name of the sample column

Value

A tbl

Examples

```
symbol_to_entrez(tidybulk::counts_mini, .transcript = transcript, .sample = sample)
```

test_differential_abundance *Add differential transcription information to a tbl using edgeR.*

Description

test_differential_abundance() takes as imput a ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> | and returns a ‘tbl‘ with additional columns for the statistics from the hypothesis test.

Usage

```
test_differential_abundance(
  .data,
  .formula,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  .contrasts = NULL,
  significance_threshold = 0.05,
  minimum_counts = 10,
  minimum_proportion = 0.7,
  fill_missing_values = FALSE,
  scaling_method = "TMM",
  omit_contrast_in_colnames = FALSE,
  action = "add"
)
```

Arguments

.data	A ‘tbl‘ formatted as	<SAMPLE>	<TRANSCRIPT>	<COUNT>	<...>	
.formula	A formula with no response variable, representing the desired linear model
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
.contrasts	A character vector. See edgeR makeContrasts specification for the parameter ‘contrasts‘. If contrasts are not present the first covariate is the one the model is tested against (e.g., ~ factor_of_interest)
significance_threshold	A real between 0 and 1 (usually 0.05).
minimum_counts	A real positive number. It is the threshold of count per million that is used to filter transcripts/genes out from the scaling procedure.
minimum_proportion	A real positive number between 0 and 1. It is the threshold of proportion of samples for each transcripts/genes that have to be characterised by a cmp bigger than the threshold to be included for scaling procedure.
fill_missing_values	A boolean. Whether to fill missing sample/transcript values with the median of the transcript. This is rarely needed.

scaling_method A character string. The scaling method passed to the backend function (i.e., edgeR::calcNormFactors; "TMM","TMMwsp","RLE","upperquartile")

omit_contrast_in_colnames
If just one contrast is specified you can choose to omit the contrast label in the colnames.

action A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).

Details

Maturing

At the moment this function uses edgeR only, but other inference algorithms will be added in the near future.

Value

A ‘tbl‘ with additional columns for the statistics from the hypothesis test (e.g., log fold change, p-value and false discovery rate).

Examples

```
test_differential_abundance(
 tidybulk::counts_mini,
    ~ condition,
    sample,
    transcript,
    `count`
)

# The functon `test_differential_abundance` operated with contrasts too

 test_differential_abundance(
    tidybulk::counts_mini,
    ~ 0 + condition,
    sample,
    transcript,
    `count`,
    .contrasts = c( "conditionTRUE - conditionFALSE")
 )
```

---

test_differential_abundance,RangedSummarizedExperiment-method
test_differential_abundance

Description

test_differential_abundance

Usage

```
## S4 method for signature 'RangedSummarizedExperiment'
test_differential_abundance(
  .data,
  .formula,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  .contrasts = NULL,
  significance_threshold = 0.05,
  minimum_counts = 10,
  minimum_proportion = 0.7,
  fill_missing_values = FALSE,
  scaling_method = "TMM",
  omit_contrast_in_colnames = FALSE,
  action = "add"
)
```

Arguments

.data	A 'tbl' formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.formula	A formula with no response variable, representing the desired linear model
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
.contrasts	A character vector. See edgeR makeContrasts specification for the parameter 'contrasts'. If contrasts are not present the first covariate is the one the model is tested against (e.g., ~ factor_of_interest)
significance_threshold	A real between 0 and 1 (usually 0.05).
minimum_counts	A real positive number. It is the threshold of count per million that is used to filter transcripts/genes out from the scaling procedure.
minimum_proportion	A real positive number between 0 and 1. It is the threshold of proportion of samples for each transcripts/genes that have to be characterised by a cmp bigger than the threshold to be included for scaling procedure.
fill_missing_values	A boolean. Whether to fill missing sample/transcript values with the median of the transcript. This is rarely needed.
scaling_method	A character string. The scaling method passed to the backend function (i.e., edgeR::calcNormFactors; "TMM","TMMwsp","RLE","upperquartile")
omit_contrast_in_colnames	If just one contrast is specified you can choose to omit the contrast label in the colnames.
action	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).

Value

A 'SummarizedExperiment' object

`test_differential_abundance,spec_tbl_df-method`
test_differential_abundance

Description

test_differential_abundance

Usage

```
## S4 method for signature 'spec_tbl_df'
test_differential_abundance(
  .data,
  .formula,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  .contrasts = NULL,
  significance_threshold = 0.05,
  minimum_counts = 10,
  minimum_proportion = 0.7,
  fill_missing_values = FALSE,
  scaling_method = "TMM",
  omit_contrast_in_colnames = FALSE,
  action = "add"
)
```

Arguments

`.data`	A ‘tbl‘ formatted as	<SAMPLE>	<TRANSCRIPT>	<COUNT>	<...>	
`.formula`	A formula with no response variable, representing the desired linear model
`.sample`	The name of the sample column
`.transcript`	The name of the transcript/gene column
`.abundance`	The name of the transcript/gene abundance column
`.contrasts`	A character vector. See edgeR makeContrasts specification for the parameter ‘contrasts‘. If contrasts are not present the first covariate is the one the model is tested against (e.g., ~ factor_of_interest)
`significance_threshold`	A real between 0 and 1 (usually 0.05).
`minimum_counts`	A real positive number. It is the threshold of count per million that is used to filter transcripts/genes out from the scaling procedure.
`minimum_proportion`	A real positive number between 0 and 1. It is the threshold of proportion of samples for each transcripts/genes that have to be characterised by a cmp bigger than the threshold to be included for scaling procedure.
`fill_missing_values`	A boolean. Whether to fill missing sample/transcript values with the median of the transcript. This is rarely needed.

scaling_method A character string. The scaling method passed to the backend function (i.e., edgeR::calcNormFactors; "TMM","TMMwsp","RLE","upperquartile")

omit_contrast_in_colnames
If just one contrast is specified you can choose to omit the contrast label in the colnames.

action A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).

Value

A 'tbl' with additional columns for the statistics from the hypothesis test (e.g., log fold change, p-value and false discovery rate).

---

test_differential_abundance,SummarizedExperiment-method
test_differential_abundance

Description

test_differential_abundance

Usage

```
## S4 method for signature 'SummarizedExperiment'
test_differential_abundance(
  .data,
  .formula,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  .contrasts = NULL,
  significance_threshold = 0.05,
  minimum_counts = 10,
  minimum_proportion = 0.7,
  fill_missing_values = FALSE,
  scaling_method = "TMM",
  omit_contrast_in_colnames = FALSE,
  action = "add"
)
```

Arguments

.data A 'tbl' formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |

.formula A formula with no response variable, representing the desired linear model

.sample The name of the sample column

.transcript The name of the transcript/gene column

.abundance The name of the transcript/gene abundance column

.contrasts A character vector. See edgeR makeContrasts specification for the parameter 'contrasts'. If contrasts are not present the first covariate is the one the model is tested against (e.g., ~ factor_of_interest)

significance_threshold
: A real between 0 and 1 (usually 0.05).

minimum_counts
: A real positive number. It is the threshold of count per million that is used to filter transcripts/genes out from the scaling procedure.

minimum_proportion
: A real positive number between 0 and 1. It is the threshold of proportion of samples for each transcripts/genes that have to be characterised by a cmp bigger than the threshold to be included for scaling procedure.

fill_missing_values
: A boolean. Whether to fill missing sample/transcript values with the median of the transcript. This is rarely needed.

scaling_method
: A character string. The scaling method passed to the backend function (i.e., edgeR::calcNormFactors; "TMM","TMMwsp","RLE","upperquartile")

omit_contrast_in_colnames
: If just one contrast is specified you can choose to omit the contrast label in the colnames.

action
: A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).

Value

A 'SummarizedExperiment' object

---

test_differential_abundance,tbl_df-method
test_differential_abundance

Description

test_differential_abundance

Usage

```
## S4 method for signature 'tbl_df'
test_differential_abundance(
  .data,
  .formula,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  .contrasts = NULL,
  significance_threshold = 0.05,
  minimum_counts = 10,
  minimum_proportion = 0.7,
  fill_missing_values = FALSE,
  scaling_method = "TMM",
  omit_contrast_in_colnames = FALSE,
  action = "add"
)
```

Arguments

.data	A 'tbl' formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.formula	A formula with no response variable, representing the desired linear model
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
.contrasts	A character vector. See edgeR makeContrasts specification for the parameter 'contrasts'. If contrasts are not present the first covariate is the one the model is tested against (e.g., ~ factor_of_interest)
significance_threshold	A real between 0 and 1 (usually 0.05).
minimum_counts	A real positive number. It is the threshold of count per million that is used to filter transcripts/genes out from the scaling procedure.
minimum_proportion	A real positive number between 0 and 1. It is the threshold of proportion of samples for each transcripts/genes that have to be characterised by a cmp bigger than the threshold to be included for scaling procedure.
fill_missing_values	A boolean. Whether to fill missing sample/transcript values with the median of the transcript. This is rarely needed.
scaling_method	A character string. The scaling method passed to the backend function (i.e., edgeR::calcNormFactors; "TMM","TMMwsp","RLE","upperquartile")
omit_contrast_in_colnames	If just one contrast is specified you can choose to omit the contrast label in the colnames.
action	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).

Value

A 'tbl' with additional columns for the statistics from the hypothesis test (e.g., log fold change, p-value and false discovery rate).

---

test_differential_abundance,tidybulk-method
test_differential_abundance

Description

test_differential_abundance

Usage

```
## S4 method for signature 'tidybulk'
test_differential_abundance(
  .data,
  .formula,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL,
  .contrasts = NULL,
  significance_threshold = 0.05,
  minimum_counts = 10,
  minimum_proportion = 0.7,
  fill_missing_values = FALSE,
  scaling_method = "TMM",
  omit_contrast_in_colnames = FALSE,
  action = "add"
)
```

Arguments

.data	A ‘tbl‘ formatted as	<SAMPLE>	<TRANSCRIPT>	<COUNT>	<...>	
.formula	A formula with no response variable, representing the desired linear model
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
.contrasts	A character vector. See edgeR makeContrasts specification for the parameter ‘contrasts‘. If contrasts are not present the first covariate is the one the model is tested against (e.g., ~ factor_of_interest)
significance_threshold	A real between 0 and 1 (usually 0.05).
minimum_counts	A real positive number. It is the threshold of count per million that is used to filter transcripts/genes out from the scaling procedure.
minimum_proportion	A real positive number between 0 and 1. It is the threshold of proportion of samples for each transcripts/genes that have to be characterised by a cmp bigger than the threshold to be included for scaling procedure.
fill_missing_values	A boolean. Whether to fill missing sample/transcript values with the median of the transcript. This is rarely needed.
scaling_method	A character string. The scaling method passed to the backend function (i.e., edgeR::calcNormFactors; "TMM","TMMwsp","RLE","upperquartile")
omit_contrast_in_colnames	If just one contrast is specified you can choose to omit the contrast label in the colnames.
action	A character string. Whether to join the new information to the input tbl (add), or just get the non-redundant tbl with the new information (get).

Value

A ‘tbl‘ with additional columns for the statistics from the hypothesis test (e.g., log fold change, p-value and false discovery rate).

`test_gene_enrichment` *analyse gene enrichment with EGSEA*

Description

test_gene_enrichment() takes as imput a ‘tbl‘ formatted as | <SAMPLE> | <ENSEMBL_ID> | <COUNT> | <...> | and returns a ‘tbl‘ with the additional transcript symbol column

Usage

```
test_gene_enrichment(
  .data,
  .formula,
  .sample = NULL,
  .entrez,
  .abundance = NULL,
  .contrasts = NULL,
  species,
  cores = 10
)
```

Arguments

`.data`	A ‘tbl‘ formatted as	<SAMPLE>	<TRANSCRIPT>	<COUNT>	<...>	
`.formula`	A formula with no response variable, representing the desired linear model
`.sample`	The name of the sample column
`.entrez`	The ENTREZ doce of the transcripts/genes
`.abundance`	The name of the transcript/gene abundance column
`.contrasts`	= NULL,
`species`	A character. For example, human or mouse
`cores`	An integer. The number of cores available

Details

Maturing

This wrapper execute gene enrichment analyses of the dataset

Value

A ‘tbl‘ object

Examples

```
df_entrez = symbol_to_entrez(tidybulk::counts_mini, .transcript = transcript, .sample = sample)
df_entrez = aggregate_duplicates(df_entrez, aggregation_function = sum, .sample = sample, .transcript = entrez

library("EGSEA")

test_gene_enrichment(
df_entrez,
~ condition,
.sample = sample,
.entrez = entrez,
.abundance = count,
species="human"
)
```

---

test_gene_enrichment,spec_tbl_df-method
test_gene_enrichment

Description

test_gene_enrichment

Usage

```
## S4 method for signature 'spec_tbl_df'
test_gene_enrichment(
  .data,
  .formula,
  .sample = NULL,
  .entrez,
  .abundance = NULL,
  .contrasts = NULL,
  species,
  cores = 10
)
```

Arguments

.data	A ‘tbl‘ formatted as	<SAMPLE>	<TRANSCRIPT>	<COUNT>	<...>	
.formula	A formula with no response variable, representing the desired linear model
.sample	The name of the sample column
.entrez	The ENTREZ doce of the transcripts/genes
.abundance	The name of the transcript/gene abundance column
.contrasts	= NULL,
species	A character. For example, human or mouse
cores	An integer. The number of cores available

Value

A ‘tbl‘ object

`test_gene_enrichment,tbl_df-method`
test_gene_enrichment

Description

test_gene_enrichment

Usage

```
## S4 method for signature 'tbl_df'
test_gene_enrichment(
  .data,
  .formula,
  .sample = NULL,
  .entrez,
  .abundance = NULL,
  .contrasts = NULL,
  species,
  cores = 10
)
```

Arguments

`.data`	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
`.formula`	A formula with no response variable, representing the desired linear model
`.sample`	The name of the sample column
`.entrez`	The ENTREZ doce of the transcripts/genes
`.abundance`	The name of the transcript/gene abundance column
`.contrasts`	= NULL,
`species`	A character. For example, human or mouse
`cores`	An integer. The number of cores available

Value

A ‘tbl‘ object

test_gene_enrichment,tidybulk-method *test_gene_enrichment*

Description

test_gene_enrichment

Usage

```
## S4 method for signature 'tidybulk'
test_gene_enrichment(
  .data,
  .formula,
  .sample = NULL,
  .entrez,
  .abundance = NULL,
  .contrasts = NULL,
  species,
  cores = 10
)
```

Arguments

.data	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
.formula	A formula with no response variable, representing the desired linear model
.sample	The name of the sample column
.entrez	The ENTREZ doce of the transcripts/genes
.abundance	The name of the transcript/gene abundance column
.contrasts	= NULL,
species	A character. For example, human or mouse
cores	An integer. The number of cores available

Value

A ‘tbl‘ object

test_gene_enrichment_bulk_EGSEA *Get gene enrichment analyses using EGSEA*

Description

Get gene enrichment analyses using EGSEA

Usage

```
test_gene_enrichment_bulk_EGSEA(
  .data,
  .formula,
  .sample = NULL,
  .entrez,
  .abundance = NULL,
  .contrasts = NULL,
  species,
  cores = 10
)
```

Arguments

.data	A ‘tbl‘ formatted as	<SAMPLE>	<TRANSCRIPT>	<COUNT>	<...>	
.formula	A formula with no response variable, representing the desired linear model
.sample	The name of the sample column
.entrez	The ENTREZ doce of the transcripts/genes
.abundance	The name of the transcript/gene abundance column
.contrasts	= NULL,
species	A character. For example, human or mouse
cores	An integer. The number of cores available

Value

A tibble with edgeR results

---

tidybulk *Creates a ‘tt‘ object from a ‘tbl“*

Description

tidybulk() creates a ‘tt‘ object from a ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |

Usage

```
tidybulk(.data, .sample, .transcript, .abundance, .abundance_scaled = NULL)
```

Arguments

.data	A ‘tbl‘ formatted as	<SAMPLE>	<TRANSCRIPT>	<COUNT>	<...>	
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column
.abundance_scaled	The name of the transcript/gene scaled abundance column

Details

Maturing

This function created a tidybulk object and is useful if you want to avoid to specify .sample, .transcript and .abundance arguments all the times. The tidybulk object have an attribute called tt_internals where these three arguments are stored as metadata. They can be extracted as attr(<object>, "tt_internals").

Value

A 'tidybulk' object

Examples

```
my_tt =  tidybulk(tidybulk::counts_mini, sample, transcript, count)
```

`tidybulk,RangedSummarizedExperiment-method`
tidybulk

Description

tidybulk

Usage

```
## S4 method for signature 'RangedSummarizedExperiment'
tidybulk(.data, .sample, .transcript, .abundance, .abundance_scaled = NULL)
```

Arguments

`.data`	A 'tbl' formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
`.sample`	The name of the sample column
`.transcript`	The name of the transcript/gene column
`.abundance`	The name of the transcript/gene abundance column
`.abundance_scaled`	The name of the transcript/gene scaled abundance column

Value

A 'tidybulk' object

`tidybulk,spec_tbl_df-method`
tidybulk

Description

tidybulk

Usage

```
## S4 method for signature 'spec_tbl_df'
tidybulk(.data, .sample, .transcript, .abundance, .abundance_scaled = NULL)
```

Arguments

`.data`	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
`.sample`	The name of the sample column
`.transcript`	The name of the transcript/gene column
`.abundance`	The name of the transcript/gene abundance column
`.abundance_scaled`	The name of the transcript/gene scaled abundance column

Value

A ‘tidybulk‘ object

`tidybulk,SummarizedExperiment-method`
tidybulk

Description

tidybulk

Usage

```
## S4 method for signature 'SummarizedExperiment'
tidybulk(.data, .sample, .transcript, .abundance, .abundance_scaled = NULL)
```

Arguments

`.data`	A ‘tbl‘ formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
`.sample`	The name of the sample column
`.transcript`	The name of the transcript/gene column
`.abundance`	The name of the transcript/gene abundance column
`.abundance_scaled`	The name of the transcript/gene scaled abundance column

Value

A 'tidybulk' object

`tidybulk,tbl_df-method` *tidybulk*

Description

tidybulk

Usage

```
## S4 method for signature 'tbl_df'
tidybulk(.data, .sample, .transcript, .abundance, .abundance_scaled = NULL)
```

Arguments

`.data`	A 'tbl' formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |
`.sample`	The name of the sample column
`.transcript`	The name of the transcript/gene column
`.abundance`	The name of the transcript/gene abundance column
`.abundance_scaled`	The name of the transcript/gene scaled abundance column

Value

A 'tidybulk' object

`tidybulk_SAM_BAM` *Creates a 'tt' object from a list of file names of BAM/SAM*

Description

tidybulk_SAM_BAM() creates a 'tt' object from a 'tbl' formatted as | <SAMPLE> | <TRANSCRIPT> | <COUNT> | <...> |

Usage

```
tidybulk_SAM_BAM(file_names, genome = "hg38", ...)
```

Arguments

`file_names`	A character vector
`genome`	A character string
`...`	Further parameters passed to the function Rsubread::featureCounts

Details

Maturing

This function is based on FeatureCounts package. This function created a tidybulk object and is useful if you want to avoid to specify .sample, .transcript and .abundance arguments all the times. The tidybulk object have an attribute called tt_internals where these three arguments are stored as metadata. They can be extracted as attr(<object>, "tt_internals").

Value

A 'tidybulk' object

`tidybulk_SAM_BAM,character,character-method`
tidybulk_SAM_BAM

Description

tidybulk_SAM_BAM

Usage

```
## S4 method for signature 'character,character'
tidybulk_SAM_BAM(file_names, genome = "hg38", ...)
```

Arguments

`file_names`	A character vector
`genome`	A character string
`...`	Further parameters passed to the function Rsubread::featureCounts

Value

A 'tidybulk' object

`tidybulk_to_SummarizedExperiment`
tidybulk_to_SummarizedExperiment

Description

tidybulk_to_SummarizedExperiment

Usage

```
tidybulk_to_SummarizedExperiment(
  .data,
  .sample = NULL,
  .transcript = NULL,
  .abundance = NULL
)
```

Arguments

.data	A tibble
.sample	The name of the sample column
.transcript	The name of the transcript/gene column
.abundance	The name of the transcript/gene abundance column

Value

A SummarizedExperiment

X_cibersort *Cibersort reference*

Description

Cibersort reference

Usage

```
X_cibersort
```

Format

An object of class data.frame with 547 rows and 22 columns.

Index