Package 'sigvar'

September 11, 2026

Title Quantify and visualize variability of mutational signatures within and across samples

Version 0.99.8

Description This package allows users to import mutational signature attribution (a.k.a. exposure) matrices and compute, visualize, and test their variabilities within and across samples.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

URL https://github.com/MaikeMorrison/sigvar

BugReports https://github.com/MaikeMorrison/sigvar/issues

Imports dplyr, readr, ggplot2, rlang, tidyr, stringr, ggh4x, glue, ggtext, ggforce, scales, GenomicFeatures, GenomeInfoDb, BSgenome, Biostrings, rtracklayer, TxDb.Hsapiens.UCSC.hg38.knownGene, TxDb.Mmusculus.UCSC.mm10.knownGene, lifecycle, withr

Suggests knitr, rmarkdown, testthat (>= 3.0.0), magick, BiocStyle, BSgenome.Hsapiens.UCSC.hg38, PNWColors, cowplot, ggpubr, ggrepel, kableExtra, lsa, patchwork, tidyverse

biocViews Software, WholeGenome, SomaticMutation, StructuralVariation, StatisticalMethod, Visualization, DriverMutation, DataImport

Config/testthat/edition 3

Depends R (>= 4.5)

VignetteBuilder knitr

LazyData false

Config/roxygen2/version 8.0.0

RoxygenNote 8.0.0

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

git_branch devel

git_last_commit df8143d

git_last_commit_date 2026-07-17

Repository Bioconductor 3.24

Date/Publication 2026-09-10

Author Maike Morrison [aut] (ORCID: <https://orcid.org/0000-0003-0430-1401>),
Nicolas Alcala [aut, cre] (ORCID:
<https://orcid.org/0000-0002-5961-5064>),
Worldwide Cancer Research [fnd] (Grant 24-0106),
French Ligue Nationale Contre le Cancer [fnd]

Maintainer Nicolas Alcala <alcalan@iarc.who.int>

Contents

`sigvar-package` *sigvar: Quantify and visualize variability of mutational signatures within and across samples*

Description

This package allows users to import mutational signature attribution (a.k.a. exposure) matrices and compute, visualize, and test their variabilities within and across samples.

Author(s)

Maintainer: Nicolas Alcala <alcalan@iarc.who.int> (ORCID)

Authors:

- Nicolas Alcala <alcalan@iarc.who.int> (ORCID)
- Maike Morrison <maike.morrison@gmail.com> (ORCID)

Other contributors:

- Worldwide Cancer Research (Grant 24-0106) [funder]
- French Ligue Nationale Contre le Cancer [funder]

See Also

Useful links:

- https://github.com/MaikeMorrison/sigvar
- Report bugs at https://github.com/MaikeMorrison/sigvar/issues

all_sig_pal — *Color palette for mutational signatures analyzed in the study by Morrison et al.*

Description

Vector of named colors for use in plotting the relative activities of mutational signatures.

Usage

```
data(all_sig_pal)
```

Format

`all_sig_pal`:
A vector of HEX color codes for mutational signature activities.

carcinogens_mice_SBS — *Single base substitution signature attributions found in mice exposed to 20 known or suspected carcinogens.*

Description

Data from: Riva et al. Nat Genet (2020)

Usage

```
data(carcinogens_mice_SBS)
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 181 rows and 47 columns.

carcinogens_mice_SBS.refs — *Reference single base substitution signatures found in mice exposed to 20 known or suspected carcinogens.*

Description

Data from: Riva et al. Nat Genet (2020)

Usage

```
data(carcinogens_mice_SBS.refs)
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 96 rows and 13 columns.

COSMIC3.0_ID *Indel reference signatures for COSMIC3.0*

Description

COSMIC 3.0 Indel mutational signatures from the COSMIC website

Usage

```
data(COSMIC3.0_ID)
```

Format

`COSMIC3.0_ID`:
A tibble with 83 rows and 18 columns:

Type Type of ID
ID1,...,ID17 Proportion of ID in each class for each signature

Source

https://cancer.sanger.ac.uk/signatures/downloads/

COSMIC3.0_SBS *Single base substitution reference signatures for COSMIC3.0*

Description

COSMIC 3.0 Single base substitutions mutational signatures from the COSMIC website

Usage

```
data(COSMIC3.0_SBS)
```

Format

`COSMIC3.0_SBS`:
A tibble with 96 rows and 68 columns:

Type Type of SBS
SBS1,...,SBS85 Proportion of SBS in each class for each signature

Source

https://cancer.sanger.ac.uk/signatures/downloads/

COSMIC3.1_CN *Copy Number reference signatures for COSMIC3.1*

Description

COSMIC 3.1 Copy Number mutational signatures from Mangiante et al. Nat Genet 55, 607–618 (2023). https://doi.org/10.1038/s41588-023-01321-1

Usage

```
data(COSMIC3.1_CN)
```

Format

`COSMIC3.1_CN`**:**

A tibble with 48 rows and 8 columns:

MutationsType Type of segment

CN1,...,CN19 Proportion of Copy number segments in each category

Source

https://static-content.springer.com/esm/art%3A10.1038%2Fs41588-023-01321-1/MediaObjects/41588_2023_1321_MOESM4_ESM.xlsx

COSMIC3.3.1_SBS *Single base substitution reference signatures for COSMIC3.3.1*

Description

COSMIC 3.3.1 Single base substitutions mutational signatures from the COSMIC website

Usage

```
data(COSMIC3.3.1_SBS)
```

Format

`COSMIC3.3.1_SBS`**:**

A tibble with 96 rows and 80 columns:

Type Type of SBS

SBS1,...,SBS95 Proportion of SBS in each class for each signature

Source

https://cancer.sanger.ac.uk/signatures/downloads/

COSMIC3.3.1_SBS_mm10 *Mice Single base substitution reference signatures for COSMIC3.3.1*

Description

Mice COSMIC 3.3.1 Single base substitutions mutational signatures from the COSMIC website

Usage

```
data(COSMIC3.3.1_SBS_mm10)
```

Format

`COSMIC3.3.1_SBS_mm10`**:**
A tibble with 96 rows and 80 columns:

Type Type of SBS
SBS1,...,SBS95 Proportion of SBS in each class for each signature

Source

https://cancer.sanger.ac.uk/signatures/downloads/

COSMIC3.3_CN *Copy Number reference signatures for COSMIC3.3*

Description

COSMIC 3.3 Copy Number mutational signatures from the COSMIC website

Usage

```
data(COSMIC3.3_CN)
```

Format

`COSMIC3.3_CN`**:**
A tibble with 48 rows and 8 columns:

MutationsType Type of segment
CN1,...,CN19 Proportion of Copy number segments in each category

Source

https://cancer.sanger.ac.uk/signatures/downloads/

COSMIC3.3_DBS *Double base reference signatures for COSMIC3.0*

Description

COSMIC 3.3 Double base mutational signatures from the COSMIC website

Usage

```
data(COSMIC3.3_DBS)
```

Format

`COSMIC3.3_DBS`**:**
A tibble with 78 rows and 12 columns:

Type Type of ID
DBS1,...,DBS11 Proportion of DBS in each class for each signature

Source

https://cancer.sanger.ac.uk/signatures/downloads/

COSMIC3.3_DBS_mm10 *Mice Double base reference signatures for COSMIC3.0*

Description

Mice COSMIC 3.3 Double base mutational signatures from the COSMIC website

Usage

```
data(COSMIC3.3_DBS_mm10)
```

Format

`COSMIC3.3_DBS_mm10`**:**
A tibble with 78 rows and 12 columns:

Type Type of ID
DBS1,...,DBS11 Proportion of DBS in each class for each signature

Source

https://cancer.sanger.ac.uk/signatures/downloads/

COSMIC3.3_ID *Indel reference signatures for COSMIC3.3.1*

Description

COSMIC 3.3 Indel mutational signatures from the COSMIC website

Usage

```
data(COSMIC3.3_ID)
```

Format

COSMIC3.3_ID:
A tibble with 83 rows and 19 columns:

Type Type of ID
ID1,...,ID18 Proportion of ID in each class for each signature

Source

https://cancer.sanger.ac.uk/signatures/downloads/

cossim *Compute the cosine similarity matrix for a set of mutational signatures.*

Description

Compute the cosine similarity matrix for a set of mutational signatures.

Usage

```
cossim(ref_sigs)
```

Arguments

ref_sigs	A matrix with K columns containing non-negative entries that sum to 1. Each column represents a signature (e.g., SBS1, SBS2), each row represents a mutation type (e.g., A[C>A]A), and each entry represents the abundance of that type in the signature.

Value

A K x K similarity matrix with diagonal elements equal to 1 and off-diagonal elements between 0 and 1. Entry S[j,k] is the similarity between signature j and signature k, equaling 1 if the categories are to be treated as identical and equaling 0 if they are to be treated as totally dissimilar. The default value is S = diag(K).

Examples

```
# Compute the cosine similarity matrix for the lung cancer
# in never smoker datasets from Zhang et al. 2021
data(Sherlock_LCINS_SBS.refs, package = "sigvar")
cossim(ref_sigs = as.matrix(Sherlock_LCINS_SBS.refs[, seq_len(14)]))
```

EGFR_context_counts_ordered

Counts of each context class for the main transcript of gene EGFR

Description

Data from: Ensembl

Usage

```
data(EGFR_context_counts_ordered)
```

Format

An object of class `table` of length 96.

EGFR_drivers_intogen_SBS96

SBS96 spectrum from the intogen website for gene EGFR in lung adenocarcinomas

---

Description

Data from: Intogen database

Usage

```
data(EGFR_drivers_intogen_SBS96)
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 96 rows and 4 columns.

ESCC_countries_SBS *Single base substitution signature attributions found in esophageal squamous cell carcinoma across countries with varying incidence.*

Description

Data from: Moody et al. Nat Genet (2021)

Usage

```
data(ESCC_countries_SBS)
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 552 rows and 31 columns.

ESCC_countries_SBS.refs *Reference COSMIC Single base substitution signatures found in esophageal squamous cell carcinoma across countries with varying incidence.*

Description

Data from: Moody et al. Nat Genet (2021)

Usage

```
data(ESCC_countries_SBS.refs)
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 96 rows and 30 columns.

ESCC_sig_activity *Activity of SBS, DBS, and ID mutational signatures in a global sample of esophageal squamous cell carcinoma cases.*

Description

Data from: Moody, S., Senkin, S., Islam, S.M.A. et al. Mutational signatures in esophageal squamous cell carcinoma from eight countries with varying incidence. Nat Genet 53, 1553–1563 (2021).

Usage

```
data(ESCC_sig_activity)
```

Format

`ESCC_sig_activity`:

A dataframe with 552 rows and 46 columns:

Country Country where each sample was collected

Incidence_Level Incidence level (High or Low) of ESCC in the region where the sample was collected

Sample Unique sample ID

SBS1, ..., ID17 Relative abundance of mutational signatures in each sample. These columns sum to 1 for each row.

Source

https://doi.org/10.1038/s41588-021-00928-6

`ESCC_sig_similarity` *Cosine similarity of SBS, DBS, and ID mutational signatures found in a global sample of esophageal squamous cell carcinoma cases.*

Description

Data from: Moody, S., Senkin, S., Islam, S.M.A. et al. Mutational signatures in esophageal squamous cell carcinoma from eight countries with varying incidence. Nat Genet 53, 1553–1563 (2021).

Usage

```
data(ESCC_sig_similarity)
```

Format

`ESCC_sig_similarity`:

A dataframe with 43 rows and 43 columns:

SBS1, ..., ID17 Pairwise cosine similarities between mutational signatures.

Source

https://doi.org/10.1038/s41588-021-00928-6

`fst` *Compute the Fst of a matrix of compositional vectors*

Description

This function computes the population genetic statistic Fst on any matrix with rows that sum to 1. Values of 1 are achieved when each row is a permutation of (1,0,..., 0) and at least two categories have non-zero abundance across all rows. The value equals 0 when each row is identical.

Usage

```
fst(
  relab_matrix,
  K = NULL,
  S = NULL,
  w = NULL,
  time = NULL,
  group = NULL,
  normalized = FALSE
)
```

Arguments

`relab_matrix`	A matrix or data frame with rows containing non-negative entries that sum to 1. Each row represents a sample, each column represents a category, and each entry represents the abundance of that category in the sample. If `relab_matrix` contains any metadata, it must be on the left-hand side of the matrix, the right `K` entries of each row must sum to 1, and `K` must be specified. Otherwise, all entries of each row must sum to 1.
`K`	Optional; an integer specifying the number of categories in the data. Default is `K=ncol(relab_matrix)`.
`S`	Optional; a K x K similarity matrix with diagonal elements equal to 1 and off-diagonal elements between 0 and 1. Entry `S[i,k]` for `i!=k` is the similarity between category and `i` and category `k`, equalling 1 if the categories are to be treated as identical and equaling 0 if they are to be treated as totally dissimilar. The default value is `S = diag(ncol(relab_matrix))`.
`w`	Optional; a vector of length `I` with non-negative entries that sum to 1. Entry `w[i]` represents the weight placed on row `i` in the computation of the mean abundance of each category across rows. The default value is `w = rep(1/nrow(relab_matrix), nrow(relab_matrix))`.
`time`	Optional; a string specifying the name of the column that describes the sampling time for each row. Include if you wish to weight Fst by the distance between samples.
`group`	Optional; a string (or vector of strings) specifying the name(s) of the column(s) that describes which group(s) each row (sample) belongs to. Use if `relab_matrix` is a single matrix containing multiple groups of samples you wish to compare.
`normalized`	Optional; should normalized Fst be used? Default is `normalized = FALSE`; use `normalized = TRUE` to compute normalized Fst. Fst can only be normalized if no weighting/similarity options are used (i.e., `w`, `S`, or `time` cannot be specified).

Value

A numeric value between 0 and 1.

Examples

```
# Compute the Fst of
# the following compositional vectors:
q1 <- c(1, 0, 0, 0)
q2 <- c(0.5, 0.5, 0, 0)
q3 <- c(1 / 4, 1 / 4, 1 / 4, 1 / 4)
q4 <- c(0, 0, 1, 0)
#
relative_abundances <- matrix(c(q1, q2, q3, q4),
  byrow = TRUE, nrow = 4
)

fst(relative_abundances)

# Incorporating weights:

# Compute fst ignoring
# rows 2 and 3
row_weights <- c(0.5, 0, 0, 0.5)
fst(relative_abundances, w = row_weights)

# Compute fst assuming that
# categories 1 and 2 are identical:
similarity_matrix <- diag(4)
similarity_matrix[1, 2] <- 1
similarity_matrix[2, 1] <- 1
fst(relative_abundances, S = similarity_matrix)

# Assume categories 1 and 2 are identical AND
# ignore rows 2 and 4:
row_weights <- c(0.5, 0, 0.5, 0)
fst(relative_abundances, w = row_weights, S = similarity_matrix)
```

`fst_norm` *Compute the normalized Fst of a matrix of compositional vectors*

Description

This function computes the normalized Fst given the number of rows and the mean abundance of the most abundant category. We employ the normalization employed in the FSTruct package by Morrison, Alcala, and Rosenberg (2020).

Usage

```
fst_norm(relab_matrix, K = ncol(relab_matrix))
```

Arguments

relab_matrix	A matrix with I=nrow(relab_matrix) rows, each containing K=ncol(relab_matrix) non-negative entries that sum to 1. If relab_matrix contains any metadata, it must be on the left-hand side of the matrix and the number of entries that sum to 1 (K) must be specified.
K	Optional; an integer specifying the number of categories in the data. Default is K=ncol(relab_matrix).

Value

A numeric value between 0 and 1.

Examples

```
# Compute the normalized fst of
# the following compositional vectors:
q1 <- c(1, 0, 0, 0)
q2 <- c(0.5, 0.5, 0, 0)
q3 <- c(1 / 4, 1 / 4, 1 / 4, 1 / 4)
q4 <- c(0, 0, 1, 0)
#
relative_abundances <- matrix(c(q1, q2, q3, q4),
  byrow = TRUE, nrow = 4
)

fst_norm(relative_abundances)
```

---

get_SBS96_driver_spectrum
get_SBS96_driver_spectrum

Description

get_SBS96_driver_spectrum

Usage

```
get_SBS96_driver_spectrum(driverlist, ref_genome)
```

Arguments

driverlist	a table containing a list of drivers alterations, with columns chr, pos, and alt
ref_genome	BSgenome reference genome

Value

A a named integer table (length 96) containing the SBS spectrum of the transcript

Examples

```
data(TP53_drivers_intogen_LUAD, package = "sigvar")
ref_genome <- "BSgenome.Hsapiens.UCSC.hg38"
library(ref_genome, character.only = TRUE)
TP53_LUAD.driver.spectrum <- get_SBS96_driver_spectrum(TP53_drivers_intogen_LUAD,ref_genome)
print(TP53_LUAD.driver.spectrum)
```

---

get_SBS96_spectrum *get_SBS96_spectrum*

Description

get_SBS96_spectrum

Usage

```
get_SBS96_spectrum(
  transcript = "ENST00000269305.9",
  organism = "Homo sapiens",
  ref_genome
)
```

Arguments

transcript	the ensembl ID of the transcript
organism	the name of the organism associated with the transcript (species)
ref_genome	BSgenome reference genome

Value

A 96 x 1 matrix containing the SBS spectrum of the transcript

Examples

```
# Run on
ref_genome <- "BSgenome.Hsapiens.UCSC.hg38"
library(ref_genome, character.only = TRUE)
spectrum <- get_SBS96_spectrum(transcript = "ENST00000269305.9", ref_genome=ref_genome)
print(spectrum)
```

het *Compute the Gini-Simpson index of a compositional vector*

Description

This function computes the Gini-Simpson index, a statistical measure of variability also known as the heterozygosity, of a vector of non-negative entries which sum to 1. The function returns a number between 0 and 1 which quantifies the variability of the vector. Values of 0 are achieved when the vector is a permutation of (1,0,..., 0). The value approaches 1 as the number of categories K increases when the vector is equal to (1/K, 1/K, ..., 1/K).

Usage

```
het(q, K = length(q), S = diag(K))
```

Arguments

q	A vector with K=length(q) non-negative entries that sum to 1.
K	Optional; an integer specifying the number of categories in the data. Default is K=length(q).
S	Optional; a K x K similarity matrix with diagonal elements equal to 1 and off-diagonal elements between 0 and 1. Entry S[i,k] for i!=k is the similarity between category and i and category k, equalling 1 if the categories are to be treated as identical and equaling 0 if they are to be treated as totally dissimilar. The default value is S = diag(ncol(q)).

Value

A numeric value between 0 and 1.

Examples

```
# Compute unweighted Gini-Simpson index:
het(q = c(0.4, 0.3, 0.3))

# Compute Gini-Simpson index assuming that
# categories 1 and 2 are identical:
similarity_matrix <- diag(3)
similarity_matrix[1, 2] <- 1
similarity_matrix[2, 1] <- 1
het(q = c(0.4, 0.3, 0.3), S = similarity_matrix)
```

het_mean *Compute the mean Gini-Simpson index of the rows in a matrix of compositional vectors*

Description

This function computes the mean Gini-Simpson index, a statistical measure of variability also known as the heterozygosity, of a set of vectors of non-negative entries which sum to 1. The function returns a number between 0 and 1 which quantifies the mean variability of the vectors. Values of 0 are achieved when each vector is a permutation of (1,0,..., 0). The value approaches 1 as the number of categories K increases when the vectors are equal to (1/K, 1/K, ..., 1/K).

Usage

```
het_mean(relab_matrix, K = NULL, S = NULL, w = NULL, time = NULL, group = NULL)
```

Arguments

relab_matrix	A matrix or data frame with rows containing non-negative entries that sum to 1. Each row represents a sample, each column represents a category, and each entry represents the abundance of that category in the sample. If relab_matrix contains any metadata, it must be on the left-hand side of the matrix, the right K entries of each row must sum to 1, and K must be specified. Otherwise, all entries of each row must sum to 1.
K	Optional; an integer specifying the number of categories in the data. Default is K=ncol(relab_matrix).
S	Optional; a K x K similarity matrix with diagonal elements equal to 1 and off-diagonal elements between 0 and 1. Entry S[i,k] for i!=k is the similarity between category and i and category k, equalling 1 if the categories are to be treated as identical and equaling 0 if they are to be treated as totally dissimilar. The default value is S = diag(ncol(relab_matrix)).
w	Optional; a vector of length I with non-negative entries that sum to 1. Entry w[i] represents the weight placed on row i in the computation of the mean abundance of each category across rows. The default value is w = rep(1/nrow(relab_matrix), nrow(relab_matrix)).
time	Optional; a string specifying the name of the column that describes the sampling time for each row. Include if you wish to weight Fst by the distance between samples.
group	Optional; a string (or vector of strings) specifying the name(s) of the column(s) that describes which group(s) each row (sample) belongs to. Use if relab_matrix is a single matrix containing multiple groups of samples you wish to compare.

Value

A numeric value between 0 and 1.

Examples

```
# To compute the mean Gini-Simpson index of
# the following compositional vectors...
q1 <- c(1, 0, 0, 0)
q2 <- c(0.5, 0.5, 0, 0)
q3 <- c(1 / 4, 1 / 4, 1 / 4, 1 / 4)
q4 <- c(0, 0, 1, 0)

# we could compute the mean manually:
mean(vapply(X = list(q1, q2, q3, q4), FUN = het, FUN.VALUE = numeric(1)))

# Or we could use het_mean:
relative_abundances <- matrix(c(q1, q2, q3, q4),
  byrow = TRUE, nrow = 4
)

het_mean(relative_abundances)

# Incorporating weights:

# Compute mean Gini-Simpson index ignoring
# rows 2 and 3
row_weights <- c(0.5, 0, 0, 0.5)
het_mean(relative_abundances, w = row_weights)

# Compute mean Gini-Simpson index assuming that
# categories 1 and 2 are identical:
similarity_matrix <- diag(4)
similarity_matrix[1, 2] <- 1
similarity_matrix[2, 1] <- 1
het_mean(relative_abundances, S = similarity_matrix)

# Assume categories 1 and 2 are identical AND
# ignore rows 2 and 4:
row_weights <- c(0.5, 0, 0.5, 0)
het_mean(relative_abundances, w = row_weights, S = similarity_matrix)
```

`het_pooled` *Compute the pooled Gini-Simpson index of the rows in a matrix of compositional vectors*

Description

This function computes the Gini-Simpson index of a "pooled" vector equal to `colMeans(relab_matrix)`. Values of 0 are achieved when this pooled vector is a permutation of (1,0,..., 0). The value approaches 1 as the number of categories K increases when this pooled vector is equal to (1/K, 1/K, ..., 1/K).

Usage

```
het_pooled(
  relab_matrix,
  K = NULL,
```

```
  S = NULL,
  w = NULL,
  time = NULL,
  group = NULL
)
```

Arguments

relab_matrix	A matrix or data frame with rows containing non-negative entries that sum to 1. Each row represents a sample, each column represents a category, and each entry represents the abundance of that category in the sample. If relab_matrix contains any metadata, it must be on the left-hand side of the matrix, the right K entries of each row must sum to 1, and K must be specified. Otherwise, all entries of each row must sum to 1.
K	Optional; an integer specifying the number of categories in the data. Default is K=ncol(relab_matrix).
S	Optional; a K x K similarity matrix with diagonal elements equal to 1 and off-diagonal elements between 0 and 1. Entry S[i,k] for i!=k is the similarity between category and i and category k, equalling 1 if the categories are to be treated as identical and equaling 0 if they are to be treated as totally dissimilar. The default value is S = diag(ncol(relab_matrix)).
w	Optional; a vector of length I with non-negative entries that sum to 1. Entry w[i] represents the weight placed on row i in the computation of the mean abundance of each category across rows. The default value is w = rep(1/nrow(relab_matrix), nrow(relab_matrix)).
time	Optional; a string specifying the name of the column that describes the sampling time for each row. Include if you wish to weight Fst by the distance between samples.
group	Optional; a string (or vector of strings) specifying the name(s) of the column(s) that describes which group(s) each row (sample) belongs to. Use if relab_matrix is a single matrix containing multiple groups of samples you wish to compare.

Value

A numeric value between 0 and 1.

Examples

```
# To compute the pooled Gini-Simpson index of
# the following compositional vectors...
q1 <- c(1, 0, 0, 0)
q2 <- c(0.5, 0.5, 0, 0)
q3 <- c(1 / 4, 1 / 4, 1 / 4, 1 / 4)
q4 <- c(0, 0, 1, 0)

# we could compute the mean manually:
qPooled <- (q1 + q2 + q3 + q4) / 4
het(qPooled)

# Or we could use het_pooled:
relative_abundances <- matrix(c(q1, q2, q3, q4),
  byrow = TRUE, nrow = 4
)
```

```
het_pooled(relative_abundances)

# Incorporating weights:

# Compute pooled Gini-Simpson index ignoring
# rows 2 and 3
row_weights <- c(0.5, 0, 0, 0.5)
het_pooled(relative_abundances, w = row_weights)

# Compute pooled Gini-Simpson index assuming that
# categories 1 and 2 are identical:
similarity_matrix <- diag(4)
similarity_matrix[1, 2] <- 1
similarity_matrix[2, 1] <- 1
het_pooled(relative_abundances, S = similarity_matrix)

# Assume categories 1 and 2 are identical AND
# ignore rows 2 and 4:
row_weights <- c(0.5, 0, 0.5, 0)
het_pooled(relative_abundances, w = row_weights, S = similarity_matrix)
```

`import_SigProfiler` *import_SigProfiler*

Description

import_SigProfiler

Usage

```
import_SigProfiler(folder = ".")
```

Arguments

`folder`	a folder containing SigProfilerExtractor results

Value

A list of tibbles containing the SigProfiler activity matrices

Examples

```
# Import dummy SigProfilerExtractor output results
SPfolder <- system.file("extdata", "SP", package = "sigvar")
Qlist <- import_SigProfiler(SPfolder)
print(Qlist)

# Plot imported signatures
plot_dots(Qlist[[1]], normalized = FALSE)
```

Intogen_EGFR_LUAD — *Driver mutations from the intogen website for gene EGFR in lung adenocarcinomas*

Description

Data from: Intogen database

Usage

```
data(Intogen_EGFR_LUAD)
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 25 rows and 13 columns.

MESOMICS_CN_SBS — *Copy Number and Single Base Substitution mutational signatures for 120 malignant pleural mesothelioma from the MESOMICS cohort*

Description

Copy Number and Single Base Substitution mutational signature attributions from Mangiante et al. Nat Genet 55, 607–618 (2023). https://doi.org/10.1038/s41588-023-01321-1

Usage

```
data(MESOMICS_CN_SBS)
```

Format

`MESOMICS_CN_SBS`**:**
A tibble with 120 rows and 41 columns:

ID_MESOMICS Identifier of the patient

Sample Identifier of the tumor sample

Sex Patient sex

Age Patient age at diagnosis

Smoking, Professional.Asbestos, ... Patient exposure status

Survival.Censor, Survival.Time Survival status (dead or alive) and time in months

Asbestos.Exposure.probability, Asbestos.Exposure.frequency, ... Quantitative asbestos exposure variables

Type Tumor histological type (WHO)

CNB Copy number burden

CN1,...,CN19 Copy number signature attributions

SBS1,...,SBS40 Single Base Substitution signature attributions

Source

https://static-content.springer.com/esm/art%3A10.1038%2Fs41588-023-01321-1/MediaObjects/41588_2023_1321_MOESM4_ESM.xlsx

mutsig_carcinogens_mice_bootstrap_p_vals

Bootstrap p-values comparing SVA results for carcinogen-exposed murine tumors to spontaneous tumors

Description

Data on single Base Substitution mutational signature attributions from Riva et al. Nat Genet 52, 1189–1197 (2020). https://doi.org/10.1038/s41588-020-0692-4

Usage

```
data(mutsig_carcinogens_mice_bootstrap_p_vals)
```

Format

`mutsig_carcinogens_mice_bootstrap_p_vals`**:**

A tibble with 29 rows and 7 columns:

group Organ of tumor and exposure status

chemical Exposure substance

Tissue Tissue of origin

group_2 Tumor/exposure reference each tumor/exposure group was compared to (either "Spontaneous_liver" or "Spontaneous_lung")

across_sample_heterogeneity P-values comparing values of across-sample heterogeneity between group and group_2

mean_within_sample_diversity P-values comparing values of within-sample diversity between group and group_2

pooled_diversity P-values comparing the pooled diversity between group and group_2

mutsig_carcinogens_mice_SBS

Single Base Substitution mutational signatures for 181 mice exposed to known or suspected carcinogens

Description

Single Base Substitution mutational signature attributions from Riva et al. Nat Genet 52, 1189–1197 (2020). https://doi.org/10.1038/s41588-020-0692-4

Usage

```
data(mutsig_carcinogens_mice_SBS)
```

Format

`mutsig_carcinogens_mice_SBS`:

A tibble with 181 rows and 46 columns:

Sample Identifier of the tumor

mSBS1, mSBS5, ..., mSBS_N3 Number of mutations for each signature

Tissue Tissue of origin

DIAGNOSIS Type of cancer

dose Treatment dose

chemical Known or suspected carcinogen chemical administered

Source

https://github.com/team113sanger/mouse-mutatation-signatures/blob/6a00d910df40d178c373ac4d57849
figure1/mexposure.rds

---

mutsig_carcinogens_mice_SBS.refs

Definitions of murine single base substitution mutational signatures

---

Description

Single Base Substitution mutational signature definitions from Riva et al. Nat Genet 52, 1189–1197 (2020). https://doi.org/10.1038/s41588-020-0692-4

Usage

```
data(mutsig_carcinogens_mice_SBS.refs)
```

Format

`mutsig_carcinogens_mice_SBS.refs`:

A tibble with 96 rows and 13 columns:

Type Substitution type (e.g., C>A, T>G)

Subtype Substitution type with 5' and 3' context (e.g., ACA)

mSBS5, ..., mSBS_N3 Proportion of mutations in each class for each signature

PCAWG_SigProfiler_COSMIC_DBS

Double Base Substitution COSMIC mutational signatures for 2780 tumors

Description

Double Base Substitution mutational signature attributions to COSMIC signatures from the Pan Cancer Whole-Genome consortium (PCAWG). Oct 19 2019 version

Usage

```
data(PCAWG_SigProfiler_COSMIC_DBS)
```

Format

`PCAWG_SigProfiler_COSMIC_DBS`:

A tibble with 2,670 rows and 14 columns:

Cancer Types Type of cancer

Sample Names Identifier of the tumor

DBS1, DBS2 Number of mutations for each signature ...

Source

https://dcc.icgc.org/releases/PCAWG/mutational_signatures/Signatures_in_Samples/SP_Signatures_in_Samples

PCAWG_SigProfiler_COSMIC_ID

Indel COSMIC mutational signatures for 2780 tumors

Description

Indel mutational signature attributions to COSMIC signatures from the Pan Cancer Whole-Genome consortium (PCAWG). Oct 19 2019 version

Usage

```
data(PCAWG_SigProfiler_COSMIC_ID)
```

Format

`PCAWG_SigProfiler_COSMIC_ID`:

A tibble with 2,780 rows and 20 columns:

Cancer Types Type of cancer

Sample Names Identifier of the tumor

ID1, ID2 Number of mutations for each signature ...

Source

https://dcc.icgc.org/releases/PCAWG/mutational_signatures/Signatures_in_Samples/SP_Signatures_in_Samples

---

PCAWG_SigProfiler_COSMIC_SBS

Single Base Substitution COSMIC mutational signatures for 2780 tumors

---

Description

Single Base Substitution mutational signature attributions to COSMIC signatures from the Pan Cancer Whole-Genome consortium (PCAWG). Oct 19 2019 version

Usage

```
data(PCAWG_SigProfiler_COSMIC_SBS)
```

Format

`PCAWG_SigProfiler_COSMIC_SBS`**:**

A tibble with 2,780 rows and 68 columns:

Cancer Types Type of cancer

Sample Names Identifier of the tumor

SBS1, SBS2 Number of mutations for each signature ...

Source

https://dcc.icgc.org/releases/PCAWG/mutational_signatures/Signatures_in_Samples/SP_Signatures_in_Samples

---

plot_dots

Plot the mean mutational signature activities of each group of samples

---

Description

This function generates a dot plot with signatures along the y-axis, group/populations along the x-axis, dot size corresponding to the fraction of samples in each population containing a signature, and dot color corresponding to the mean relative activity of mutations to that signature.

Usage

```
plot_dots(
  sig_activity,
  group = colnames(sig_activity)[1],
  K = ncol(sig_activity) - 1 - as.numeric(!missing(facet)),
  max_dotsize = 5,
  pivot = FALSE,
  median = FALSE,
  normalized = TRUE,
  facet,
  threshold = 0
)
```

Arguments

sig_activity	A matrix or data frame with rows containing non-negative entries that sum to 1. Each row represents a sample, each column represents a mutational signature, and each entry represents the abundance of that signature in the sample. If sig_activity contains any metadata, it must be on the left-hand side of the matrix, the right K entries of each row must sum to 1, and K must be specified. Otherwise, all entries of each row must sum to 1.
group	A string (or vector of strings) specifying the name(s) of the column(s) that describes which group(s) each sample belongs to. Default is the first column of sig_activity: colnames(sig_activity)[1].
K	Optional; the number of signatures in the matrix. Default is K=ncol(sig_activity)-1-as.numeric(the number of columns in sig_activity minus either 1 (since one column must specify the group) or 2 if facet is provided, since one column must specify the variable on which to facet.
max_dotsize	Optional; a number specifying the maximum size for each dot.
pivot	Optional; set pivot=TRUE if you would like to plot groups on the y-axis and signatures on the x-axis. Default is pivot=FALSE.
median	Optional; specify median = TRUE if you would like to color points by the median signature activity across samples, rather than the mean. Default value is median = FALSE.
normalized	Optional; specify normalized = FALSE if you would like to plot absolute signature activities rather than relative signature activities. I.e., if your sig_activity matrix contains rows representing absolute signature activities (i.e., they do not necessarily sum to 1) and you wish that these activities are not automatically normalized. This determines the color scale for the resulting plot. Default is normalized = TRUE.
facet	Optional; a string specifying the name of the column by which you would like to facet your plot.
threshold	Optional; a number between 0 and 1. For each signature, the size of the associated dot corresponds to the proportion of tumors with that signature at an abundance exceeding this threshold. Default is threshold = 0.

Value

A ggplot object containing a dot plot visualization of the mean mutational signature activities

Examples

```
# Make an example matrix.
# Each row is a sample. Rows sum to 1.
sig_activity <- matrix(
  c(
    "A", .4, .2, .4,
    "A", .5, .3, .2,
    "A", .5, .4, .1,
    "B", .6, .1, .3,
    "B", .6, .3, .1
  ),
  nrow = 5,
  byrow = TRUE
)

colnames(sig_activity) <- c("pop", "SBS1", "SBS2", "SBS3")
sig_activity <- dplyr::mutate(data.frame(sig_activity), dplyr::across(
  c(SBS1, SBS2, SBS3),
  as.numeric
))

plot_dots(sig_activity,
  K = 3, # How many categories per vector?
  group = "pop"
)
```

---

plot_SBS_spectrum *Plot an SBS mutational spectrum*

Description

Plot an SBS mutational spectrum

Usage

```
plot_SBS_spectrum(SBS_table)
```

Arguments

SBS_table	A matrix with rows corresponding to the 96 single-base substitutions and columns corresponding to distinct mutational spectra you wish to plot.

Value

A ggplot2 bar chart depicting SBS mutational spectra

Examples

```
data(COSMIC3.3.1_SBS, package = "sigvar")
SBS_table <- dplyr::select(COSMIC3.3.1_SBS, SBS1, SBS5)
plot_SBS_spectrum(SBS_table)
```

plot_signature_prop *Visualize a relative abundance matrix as a stacked bar plot.*

Description

This function enables graphical visualization of a matrix of compositional data. In the output plot, each vertical bar represents a single vector; the height of each color in the bar corresponds to the abundance of each category in that vector. Because this function produces a ggplot object, its output can be modified using standard ggplot2 syntax.

Usage

```
plot_signature_prop(
  relab_matrix,
  group = NULL,
  time = NULL,
  w = NULL,
  K = NULL,
  arrange = FALSE
)
```

Arguments

relab_matrix	A matrix with I=nrow(relab_matrix) rows, each containing K=ncol(relab_matrix) non-negative entries that sum to 1. If relab_matrix contains any metadata, it must be on the left-hand side of the matrix and the number of entries that sum to 1 (K) must be specified.
group	Optional; a string specifying the name of the column that describes which group each row (sample) belongs to. Use if relab_matrix is a single matrix containing multiple groups of samples you wish to compare.
time	Optional; a string specifying the name of the column that describes the sampling time for each row. Include if you wish to weight the samples by the distance between them in the visualization.
w	Optional; a vector of length I with non-negative entries that sum to 1. Entry w[i] represents the weight placed on row i in the computation of the mean abundance of each category across rows. The default value is w = rep(1/nrow(relab_matrix), nrow(relab_matrix)).
K	Optional; an integer specifying the number of categories in the data. Default is K=ncol(relab_matrix).
arrange	Optional; controls horizontal ordering of samples and vertical ordering of categories. If arrange = TRUE or arrange = "both", samples are ordered by the categories of greatest abundance and categories are ordered in decreasing abundance. If arrange = "vertical", sample order is unchanged but categories are ordered in decreasing abundance. If arrange = "horizontal", samples are ordered by the most abundant categories, but category order is unchanged. If arrange is missing or arrange = FALSE, neither order is changed.

Value

A ggplot object containing a bar plot visualization of the relative abundance matrix.

Examples

```
# Make an example matrix of compositional data
# Each row is an individual. Rows sum to 1.
population_A <- matrix(
  c(
    .5, .3, .2,
    .4, .2, .4,
    .5, .4, .1,
    .6, .1, .3,
    .2, 0, .8
  ),
  nrow = 5,
  byrow = TRUE
)

plot_signature_prop(
  relab_matrix = population_A,
  K = 3, # How many categories per vector?
  arrange = FALSE
)
plot_signature_prop(
  relab_matrix = population_A,
  K = 3, # How many categories per vector?
  arrange = "horizontal"
)
plot_signature_prop(
  relab_matrix = population_A,
  K = 3, # How many categories per vector?
  arrange = "vertical"
)
plot_signature_prop(
  relab_matrix = population_A,
  K = 3, # How many categories per vector?
  arrange = TRUE # could also be "both"
)


# You can modify the plot as you would any ggplot2 object
plot_signature_prop(
  relab_matrix = population_A,
  K = 3, # How many categories per vector?
  arrange = TRUE
) +
  # Below are example, optional modifications to the default plot
  ggplot2::ggtitle("Population A") +
  ggplot2::scale_fill_brewer("Blues") +
  ggplot2::scale_color_brewer("Blues") +
  ggplot2::xlab("Individuals")
# Note that both scale_fill and scale_color are needed to change the color of the bars.


# Plot a dataset which has 2 populations

population_B <- matrix(
  c(
    .9, 0, .1,
```

```
    .6, .4, 0,
    .7, 0, .3,
    .3, .4, .3,
    .5, .3, .2
  ),
  nrow = 5,
  byrow = TRUE
)


populations_AB <- cbind(
  data.frame(c(
    "A", "A", "A", "A", "A",
    "B", "B", "B", "B", "B"
  )),
  rbind(population_A, population_B)
)
colnames(populations_AB) <- c("population", "category_1", "category_2", "category_3")


plot_signature_prop(relab_matrix = populations_AB, group = "population")
plot_signature_prop(relab_matrix = populations_AB, group = "population", arrange = "vertical")
plot_signature_prop(relab_matrix = populations_AB, group = "population", arrange = "horizontal")
plot_signature_prop(relab_matrix = populations_AB, group = "population", arrange = "both")
```

`radiation_sigs_morton` *Single base substitution signature attributions found in papillary thyroid carcinoma from Chernobyl incident survivors.*

Description

Data from: Morton et al. Science (2021)

Usage

```
data(radiation_sigs_morton)
```

Format

An object of class `data.frame` with 356 rows and 16 columns.

`radiation_sigs_morton_cossim` *Cosine similarity matrix of Single base substitution signatures found in papillary thyroid carcinoma from Chernobyl incident survivors.*

Description

Data from: Morton et al. Science (2021)

Usage

```
data(radiation_sigs_morton_cossim)
```

Format

An object of class `matrix` (inherits from `array`) with 13 rows and 13 columns.

---

rcc_sig_activity — *Activity of SBS, DBS, and ID mutational signatures in a global sample of renal cell carcinoma cases.*

Description

Data from: Senkin et al. (2023). Geographic variation of mutagenic exposures in kidney cancer genomes. MedRxiv, 14, 2023.06.20.23291538. https://doi.org/10.1101/2023.06.20.23291538

Usage

```
data(rcc_sig_activity)
```

Format

`rcc_sig_activity`**:**

A dataframe with 962 rows and 31 columns:

Sample Unique sample ID

Country Country where each sample was collected

ASR Age-standardized rate (ASR) of RCC incidence in each country

SBS1, ..., ID_C Relative abundance of mutational signatures in each sample. These columns sum to 1 for each row.

Source

https://doi.org/10.1101/2023.06.20.23291538

---

rcc_sim — *Activity of SBS, DBS, and ID mutational signatures in a global sample of renal cell carcinoma cases.*

Description

Data from: Senkin et al. (2023). Geographic variation of mutagenic exposures in kidney cancer genomes. MedRxiv, 14, 2023.06.20.23291538. https://doi.org/10.1101/2023.06.20.23291538

Usage

```
data(rcc_sim)
```

Format

`rcc_sim`:

A dataframe with 28 rows and 28 columns:

SBS1, ..., ID_C Cosine similarity between mutational signatures.

Source

https://doi.org/10.1101/2023.06.20.23291538

---

sbs_palette *Single Base Substitution COSMIC mutational signatures for 2780 tumors*

Description

Single Base Substitution mutational signature attributions to COSMIC signatures from the Pan Cancer Whole-Genome consortium (PCAWG). Oct 19 2019 version

Usage

```
data(sbs_palette)
```

Format

`sbs_palette`:

A vector of RGB color codes for COSMIC SBS signatures.

Source

https://dcc.icgc.org/releases/PCAWG/mutational_signatures/Signatures_in_Samples/SP_Signatures_in_Samples

---

Sherlock_LCINS.metadata *Metadata accompanying mutational signatures for lung cancer in never smokers.*

Description

Data from: Zhang et al. Nat Genet (2021)

Usage

```
data(Sherlock_LCINS.metadata)
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 232 rows and 24 columns.

Sherlock_LCINS_SBS *Single base substitution signature attributions found in Lung cancer in never smokers.*

Description

Data from: Zhang et al. Nat Genet (2021)

Usage

```
data(Sherlock_LCINS_SBS)
```

Format

An object of class `spec_tbl_df` (inherits from `tbl_df`, `tbl`, `data.frame`) with 232 rows and 15 columns.

Sherlock_LCINS_SBS.refs *Reference COSMIC Single base substitution signatures found in Lung cancer in never smokers.*

Description

Data from: Zhang et al. Nat Genet (2021)

Usage

```
data(Sherlock_LCINS_SBS.refs)
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 96 rows and 16 columns.

sigboot *Statistically compare the within-sample diversity and across-sample heterogeneity of the mutational signature activity in two or more groups of tumor samples.*

Description

`sigboot` uses bootstrapping to evaluate differences in within-sample diversity and across-sample heterogeneity of mutational signature activity between pairs of groups of tumor samples. `sigboot` takes the same options as `sigvar`, so, as with `sigvar`, you can separately analyze multiple populations or groups of samples (specify `group`), and account for cosine similarity among signatures (specify `S`). `sigboot` follows the bootstrapping procedure defined by Efron and Tibshirani (1993). Details on the bootstrapping procedure are available in the Methods section of the accompanying paper.

Usage

```
sigboot(
  sig_activity,
  n_replicates,
  group,
  K = NULL,
  S = NULL,
  w = NULL,
  time = NULL,
  normalized = FALSE,
  seed = NULL,
  alternative = "two.sided"
)
```

Arguments

sig_activity	A matrix or data frame with I rows. Each row represents a sample, each column represents a mutational signature, and each entry represents the abundance of that signature in the sample. If sig_activity contains any metadata, it must be on the left-hand side of the matrix, the right K entries of each row must sum to 1, and K must be specified. Otherwise, all entries of each row must sum to 1.
n_replicates	The number of bootstrap replicate matrices to generate for each provided relative abundance matrix.
group	A string (or vector of strings) specifying the name(s) of the column(s) that describes which group(s) each sample belongs to.
K	Optional; an integer specifying the number of mutational signatures included in sig_activity. Default is K=ncol(sig_activity)-length(group).
S	Optional; a K x K similarity matrix with diagonal elements equal to 1 and off-diagonal elements between 0 and 1. Entry S[j,k] is the similarity between signature j and signature k, equaling 1 if the categories are to be treated as identical and equaling 0 if they are to be treated as totally dissimilar. The default value is S = diag(K).
w	Optional; a vector of length I with non-negative entries that sum to 1. Entry w[i] represents the weight placed on row i of sig_activity in the computation of the mean abundance of each category across rows. The default value is w = rep(1/nrow(sig_activity), nrow(sig_activity)).
time	Optional; a string specifying the name of the column that describes the sampling time for each row (or another continuous, increasing variable). Include if you wish to weight SVA by the temporal distance between samples.
normalized	Optional; should the across-sample variability (Fst) be normalized by its upper bound conditional on the mean activity of the most abundant signature be used as the measure of across-sample variability? Default is normalized = FALSE; use normalized = TRUE to compute normalized Fst. Fst can only be normalized if it is not weighted.
seed	Optional; an integer to be used as a random seed for the simulations.
alternative	Optional; do you want to do a one- or two-sided test? Default is alternative = "two.sided". If you wish to do a one-sided test, specify either alternative = "less" or alternative = "greater".

Value

A named list containing the following entries:

- `P_values`: The probability of observing the observed difference in variability between each pair of groups if there were no difference between groups. Computed as the fraction of bootstrap differences greater than or equal to the observed difference. Depends on what `alternative` is specified ("greater", "less", or "two.sided").
- `bootstrap_distribution_plot`: The distribution of bootstrap replicate differences in each variability value. The observed differences are shown in red. The further the red points are from 0, the more significant the statistical difference between groups.
- `observed_difference`: The observed difference in diversity statistics between the groups.
- `bootstrap_difference`: The bootstrap replicate difference in diversity statistics between the groups.

Examples

```
# Estimate the uncertainty in the across-sample and mean within-sample variability of
# mutational signatures in ESCC samples grouped by country
# We provide a cosine similarity matrix in order to account for cosine similarity among signatures
data(smoker_sigs_chen, package = "sigvar")
data(smoker_sigs_chen_cossim, package = "sigvar")
smoker_boot <- sigboot(
  sig_activity = smoker_sigs_chen, K = 3, n_replicates = 500,
  group = "Smoker", S = smoker_sigs_chen_cossim,
  seed = 1
)

# We find that there is a strong, significant difference between smokers and non-smokers
# in their mean within-sample diversity, but that the difference in across-sample heterogeneity
# is less strong
smoker_boot$P_values

# We can visualize this result by looking at the observed differences between
# smokers and non-smokers and the bootstrap replicates, which assume no difference
# between these two groups
smoker_boot$bootstrap_distribution_plot
```

`sigvar` *Compute the within-sample diversity and across-sample heterogeneity of mutational signature activity in one or multiple populations of samples.*

Description

Across-sample variability is quantified by computing the population-genetic statistic Fst across the rows of the signature activity table. This method is based on the R package FAVA, which implements an Fst-based assessment of Variability across vectors of relative Abundances. Mean within-sample variability is quantified by computing the mean Gini-Simpson index across all samples. Specify `group` if your table of signature activities contains multiple populations of samples which you wish to compare. Specifying `S` allows `sigvar` to account for the pairwise cosine similarity among signatures. Specifying `w` or `time` allows for uneven weighting of the samples.

Usage

```
sigvar(
  sig_activity,
  K = NULL,
  group = NULL,
  S = NULL,
  w = NULL,
  time = NULL,
  normalized = FALSE
)
```

Arguments

sig_activity	A matrix or data frame with rows containing non-negative entries that sum to 1. Each row represents a sample, each column represents a mutational signature, and each entry represents the abundance of that signature in the sample. If sig_activity contains any metadata, it must be on the left-hand side of the matrix, the right K entries of each row must sum to 1, and K must be specified. Otherwise, all entries of each row must sum to 1.
K	Optional; an integer specifying the number of mutational signatures included in sig_activity. Default is K=ncol(sig_activity).
group	Optional; a string (or vector of strings) specifying the name(s) of the column(s) that describes which group(s) each sample belongs to. Use if sig_activity is a single matrix containing multiple groups of samples you wish to compare.
S	Optional; a K x K similarity matrix with diagonal elements equal to 1 and off-diagonal elements between 0 and 1. Entry S[j,k] is the similarity between signature j and signature k, equaling 1 if the categories are to be treated as identical and equaling 0 if they are to be treated as totally dissimilar. The default value is S = diag(K).
w	Optional; a vector of length nrow(sig_activity) with non-negative entries that sum to 1. Entry w[i] represents the weight placed on row i in the computation of the mean abundance of each category across rows. The default value is w = rep(1/nrow(sig_activity), nrow(sig_activity)).
time	Optional; a string specifying the name of the column that describes the sampling time for each row. Include if you wish to weight the variability measures by the distance between adjacent samples in the matrix.
normalized	Optional; should normalized Fst be used as the measure of across-sample variability? Default is normalized = FALSE; use normalized = TRUE to compute normalized Fst. Fst can only be normalized if it is not weighted.

Value

A data frame with columns corresponding to the across-sample variability and the mean within-sample variability for the provided samples.

Examples

```
# Compute the across-sample and mean within-sample variability of  mutational signatures
# in ESCC samples grouped by country
# We provide a cosine similarity matrix in order to account for cosine similarity among signatures
data(smoker_sigs_chen, package = "sigvar")
```

```
data(smoker_sigs_chen_cossim, package = "sigvar")
sigvar(sig_activity = smoker_sigs_chen, K = 3, group = "Smoker", S = smoker_sigs_chen_cossim)
```

`smoker_sigs_chen` *Activity of Smoking, APOBEC, and Ageing mutational signatures in a sample of East Asian lung adenocarcinoma patients.*

Description

Data from: Chen, J., Yang, H., Teo, A.S.M. et al. Genomic landscape of lung adenocarcinoma in East Asians. Nat Genet 52, 177–186 (2020).

Usage

```
data(smoker_sigs_chen)
```

Format

`smoker_sigs_chen`**:**

A dataframe with 88 rows and 20 columns:

Patient.ID Unique identifier for each patient

Smoker Smoking status for each patient: "Smoker" or "Non-smoker"

Cohort, Stage, Age, Gender Patient metadata

Smoking Activity of smoking mutational signature

APOBEC Activity of APOBEC mutational signature

Ageing Activity of ageing mutational signature

Source

https://doi.org/10.1038/s41588-019-0569-6

`smoker_sigs_chen_cossim` *Cosine similarity of smoking, APOBEC, and ageing mutational signatures found in a sample of East Asian lung adenocarcinoma patients.*

Description

Data from: Chen, J., Yang, H., Teo, A.S.M. et al. Genomic landscape of lung adenocarcinoma in East Asians. Nat Genet 52, 177–186 (2020).

Usage

```
data(smoker_sigs_chen_cossim)
```

Format

`smoker_sigs_chen_cossim`**:**

A dataframe with 3 rows and 3 columns:

Smoking, ..., Ageing Pairwise cosine similarities between mutational signatures.

Source

https://doi.org/10.1038/s41588-019-0569-6

`time_weights` *Compute a normalized weighting vector based on a vector of sampling times.*

Description

This function takes a vector of sampling times, $t = (t_1, t_2, \ldots, t_I)$ and computes a normalized vector which can be used to weight each sample based on the time between the subsequent and the preceding samples. The weighting vector w is defined such that each entry, $w_i = d_i/2T$, where $T = t_I - t_1$ and $d_i = t_{i+1} - t_{i-1}$ for i not equal to 1 or I. $d_1 = t_2 - t_1$ and $d_I = t_I - t_{I-1}$.

Usage

```
time_weights(times, group = NULL)
```

Arguments

`times`	A numeric vector of sampling times. Each entry must be non-negative and greater than the previous entry.
`group`	Optional; a character vector specifying the group identity of each sampling time. Use if there are samples from multiple replicates or subjects in one data set.

Value

A numeric vector of weights with length equal to the length of `times`. Each weight represents the normalized temporal distance between adjacent samples. Weights sum to 1 when `group` is NULL, or sum to the number of unique groups when `group` is specified.

Examples

```
time_vector = c(1, 8, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31,
                32, 33, 34, 35, 36, 37, 38, 39, 44, 50, 57, 64)

time_weights(times = time_vector)
```

TP53_context_counts_ordered

Counts of each context class for the main transcript of gene TP53

Description

Data from: Ensembl

Usage

```
data(TP53_context_counts_ordered)
```

Format

An object of class `table` of length 96.

TP53_drivers_intogen_LUAD

Driver mutations from the intogen website for gene TP53 in lung adenocarcinomas

Description

Data from: Intogen database

Usage

```
data(TP53_drivers_intogen_LUAD)
```

Format

An object of class `spec_tbl_df` (inherits from `tbl_df`, `tbl`, `data.frame`) with 202 rows and 10 columns.

TP53_drivers_intogen_SBS96

SBS96 spectrum from the intogen website for gene TP53 in lung adenocarcinomas

Description

Data from: Intogen database

Usage

```
data(TP53_drivers_intogen_SBS96)
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 96 rows and 4 columns.

zhang_sig_activity *Activity of SBS mutational signatures in samples of lung cancer in never smokers*

Description

Data from: Zhang et al. (2021). Genomic and evolutionary classification of lung cancer in never smokers. Nature Genetics, 53(9), 1348–1359. https://doi.org/10.1038/s41588-021-00920-0

Usage

```
data(zhang_sig_activity)
```

Format

`zhang_sig_activity`:

A dataframe with 39 rows and 213 columns:

Subject Unique ID for each subject ...

passive_smoking Subject smoking status ("Non-smoker" or "Passive smoker") ...

SBS1, ..., SBS40 Relative abundance of mutational signatures in each sample. These columns sum to 1 for each row.

Source

https://doi.org/10.1038/s41588-021-00920-0

zhang_sig_similarity *Cosine similarity of SBS mutational signatures found in samples of lung cancer in never smokers*

Description

Data from: Zhang et al. (2021). Genomic and evolutionary classification of lung cancer in never smokers. Nature Genetics, 53(9), 1348–1359. https://doi.org/10.1038/s41588-021-00920-0

Usage

```
data(zhang_sig_similarity)
```

Format

`zhang_sig_similarity`:

A dataframe with 14 rows and 14 columns:

SBS1, ..., SBS40 Pairwise cosine similarities between mutational signatures.

Source

https://doi.org/10.1038/s41588-021-00920-0

Index