

Package ‘GeneScoreR’

November 26, 2025

Title Gene Scoring from Count Tables

Version 0.2.0

Description Provides methods for automatic calculation of gene scores from gene count tables, including a Z-score method that requires a table of samples being scored and a count table with control samples; a geometric mean method that does not rely on control samples; and a principal component-based method that summarizes gene expression using user-selected principal components. The Z-score and geometric mean approaches are described in Kim et al. (2018) <[doi:10.1089/jir.2017.0127](https://doi.org/10.1089/jir.2017.0127)>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Imports stats

NeedsCompilation no

Author Aris Syntakas [aut, cre]

Maintainer Aris Syntakas <sejjsyn@ucl.ac.uk>

Repository CRAN

Date/Publication 2025-11-26 16:00:02 UTC

Contents

geomean	2
pcscore	3
zscore	4
Index	5

`geomean`*Calculate Geometric Means from Count Tables*

Description

This function computes the geometric mean for each sample in the given count table. Users can choose how to handle missing values.

Usage

```
geomean(count_table, na.action = c("omit", "fail"))
```

Arguments

- | | |
|--------------------------|---|
| <code>count_table</code> | A data frame of gene count data (genes as rows, samples as columns). All columns must be numeric. |
| <code>na.action</code> | Character: how to handle NAs. Options are <ul style="list-style-type: none">• "omit" (default): ignore NAs and compute the geometric mean with remaining values• "fail": return NA for the sample if any NA is present |

Value

A data frame with the geometric means per sample and the sample IDs.

Examples

```
count_table <- data.frame(
  sample1 = c(1, 10, 100),
  sample2 = c(2, 20, 200),
  sample3 = c(3, 30, NA)
)
rownames(count_table) <- c("gene1", "gene2", "gene3")

geomean(count_table)

geomean(count_table, na.action = "fail")
```

pcscore

Calculate Principal Component Scores from Count Tables

Description

This function computes selected principal component (PC) scores for each sample in the given count table.

Usage

```
pcscore(count_table, pc = 1, stabilize_sign = FALSE)
```

Arguments

count_table A data frame of gene count data (genes as rows, samples as columns). All columns must be numeric.

pc Integer vector specifying which principal components to extract (default = 1). Use "all" to return all PCs.

stabilize_sign Logical. Should the sign of each PC be stabilized? (default = FALSE)

Details

By default, the first principal component (PC1) is returned. Users can optionally stabilize the sign of each PC.

Value

A data frame with PC scores per sample and the sample IDs.

Examples

```
count_table <- data.frame(
  sample1 = c(5, 10, 15, 20),
  sample2 = c(6, 11, 16, 21),
  sample3 = c(7, 12, 17, 22)
)
rownames(count_table) <- c("gene1", "gene2", "gene3", "gene4")

pcscore(count_table)

pcscore(count_table, pc = 2)

pcscore(count_table, pc = c(1, 2))

pcscore(count_table, pc = "all")

pcscore(count_table, pc = 1, stabilize_sign = TRUE)
```

zscore

*Calculate Z-Scores from Count Tables***Description**

This function computes a Z-score sum for each sample in the given "scored" count table, based on the means and SDs of the genes in the control table. Users can choose how to handle missing values.

Usage

```
zscore(scored_table, control_table, na.action = c("omit", "fail"))
```

Arguments

- | | |
|---------------|---|
| scored_table | Data frame of samples to be scored (genes as rows, samples as columns). All columns must be numeric. |
| control_table | Data frame of control samples (genes as rows, samples as columns). All columns must be numeric. |
| na.action | Character. How to handle NAs. Options are: <ul style="list-style-type: none"> • "omit" (default): ignore NAs and calculate using available values • "fail": return NA if any value is missing |

Value

A data frame with the sum of Z-scores per sample and the sample IDs.

Examples

```
scored_table <- data.frame(
  sample1 = c(1, 2, 3),
  sample2 = c(4, NA, 6),
  sample3 = c(7, 8, 9)
)
rownames(scored_table) <- c("gene1", "gene2", "gene3")

control_table <- data.frame(
  control1 = c(1, 1, 1),
  control2 = c(2, 2, 2),
  control3 = c(3, 3, 3)
)
rownames(control_table) <- c("gene1", "gene2", "gene3")

zscore(scored_table, control_table)

zscore(scored_table, control_table, na.action = "fail")
```

Index

geomean, [2](#)

pcscore, [3](#)

zscore, [4](#)