Package: SRscore (via r-universe)

October 31, 2026

Type Package

Title Simple Transcriptome Meta-Analysis for Identifying Stress-Responsive Genes

Version 0.1.2

Author Yusuke Fukuda [aut, cre], Atsushi Fukushima [aut]

Maintainer Yusuke Fukuda <ysk420140@gmail.com>

Description Stress Response score (SRscore) is a stress responsiveness measure for transcriptome datasets and is based on the vote-counting method. The SRscore is determined to evaluate and score genes on the basis of the consistency of the direction of their regulation (Up-regulation, Down-regulation, or No change) under stress conditions across multiple analyzed research projects. This package is based on the HN-score (score based on the ratio of gene expression between hypoxic and normoxic conditions) proposed by Tamura and Bono (2022) <doi:10.3390/life12071079>, and can calculate both the original method and an extended calculation method described in Fukuda et al. (2025) <doi:10.1093/plphys/kiaf105>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Roxygen list(markdown = TRUE)

Imports dplyr, tidyr, utils, rlang

Suggests knitr, rmarkdown, testthat (>= 3.0.0), ggplot2, tibble, ComplexHeatmap, clusterProfiler, org.At.tair.db, BiocStyle, RColorBrewer, genefilter, DT

Config/testthat/edition 3

VignetteBuilder knitr

Depends R (>= 3.5.0)

Config/pak/sysreqs libicu-dev

Repository https://fusk-kpu.r-universe.dev

Date/Publication 2026-03-24 11:20:50 UTC

RemoteUrl https://github.com/fusk-kpu/srscore

RemoteRef HEAD

RemoteSha c47cbea2c16b503c7887c5fe118ba6ca2977f04f

Contents

calcSRratio 2
calcSRscore 3
directly_calcSRscore 4
expand_by_group 5
find_diffexp 6
HNscore 7
logHNratioHypoxia 8
MetadataABA 9
MetadataHypoxia 9
plot_SRscore_distr 10
plot_SRscore_rank 11
plot_SRscore_top 12
sample_pair_test 13
SRGA 13
SRratio_test 14
SRscore_test 15
TPMHypoxia 15
TranscriptomeABA 16

Index **17**

`calcSRratio` *Calculate the Stress Response ratio (SRratio)*

Description

This function computes the Stress Response ratio (SR ratio) for paired variables in a dataset. The function supports both log2-transformed and non-log2-transformed data and calculates the mean SRratio for grouped variables.

Usage

```
calcSRratio(.data, var1, var2, pair, is.log2 = NA)
```

Arguments

.data	A data frame containing expression values for a series of arrays, with rows corresponding to genes and columns to samples.
var1	A character vector containing column names of control samples.
var2	A character vector containing column names of treatment samples.
pair	A data frame with control samples and treatment samples.
is.log2	A logical value (TRUE, FALSE) or NA indicating whether the data in .data is log2-transformed: • If TRUE, the SR ratio is calculated as the difference between the target and reference variables. • If FALSE, the SR ratio is calculated as the log2-transformed ratio: log2((target + 1) / (reference + 1)). • If NA (default), the user will be prompted interactively to confirm whether the data is log-transformed.

Value

A data frame containing:

- Character columns from the original .data.
- Mean SRratio values for each unique target variable.

Examples

```
var1 <- "control_sample"
var2 <- "treated_sample"
grp <- "Series"

ebg <- expand_by_group(MetadataABA, grp, var1, var2)

SRratio <- calcSRratio(TranscriptomeABA, var1, var2, ebg, is.log2 = TRUE)
```

---

calcSRscore *Create a Data Frame of SRscore*

Description

SRscore is score value of genes based expression profiles across different research projects. SRratio is required to calculate SRscore.

Usage

```
calcSRscore(srratio, threshold = c(-1, 1))
```

Arguments

srratio	A data frame of SRratio.
threshold	A vector of length 2 (x, y) indicating threshold values. c(-1, 1) is default.

Value

A data frame containing results.

Examples

```
grp <- "Series"
var1 <- "control_sample"
var2 <- "treated_sample"

ebg <- expand_by_group(MetadataABA,
                       grp,
                       var1,
                       var2)

SRratio <- calcSRratio(TranscriptomeABA,
                       var1,
                       var2,
                       ebg,
                       is.log2 = TRUE)

head(calcSRscore(SRratio, threshold = c(-1, 1)))
```

---

directly_calcSRscore *Aggregate the results of the three functions into a single list*

Description

The SRscore calculation process is divided into three major processes, and functions are provided for each process (see the respective function documents for details). directly_calcSRscore() aggregates the results of the three functions into a single list.

Usage

```
directly_calcSRscore(
  .data1,
  grp,
  var1,
  var2,
  .data2,
  is.log2 = NA,
  threshold = c(-1, 1)
)
```

Arguments

.data1	A data frame containing the two variables you want to compare, as well as the variables of the group to which they belong.
grp	Column name of groups.
var1	Column name of first variable.
var2	Column name of second variable.
.data2	A data frame containing expression values for a series of arrays, with rows corresponding to genes and columns to samples.
is.log2	A logical specifying if .data2 is log-2transformed.
threshold	A vector of length 2 (x, y) indicating threshold values. c(-1, 1) is default.

Value

A data frame containing results.

See Also

expand_by_group()

calcSRratio()

calcSRscore()

Examples

```
grp <- "Series"
var1 <- "control_sample"
var2 <- "treated_sample"

ls <- directly_calcSRscore(MetadataABA,
                           grp,
                           var1,
                           var2,
                           TranscriptomeABA,
                           is.log2 = TRUE,
                           threshold = c(-1, 1))
lapply(ls, head)
```

---

expand_by_group *Create a data frame from all combinations between two specified variables within each group*

Description

expand_by_group() generates all combinations (Cartesian product) of two specified variables within each group in your dataframe.

Usage

```
expand_by_group(.data, grp, var1, var2)
```

Arguments

.data	A data frame.
grp	A column name indicating the group.
var1	A column name indicating the control.
var2	A column name indicating the treatment.

Value

Returns a data frame containing all combinations of the specified variables for each group. The structure of the returned data frame includes:

- All combinations of var1 and var2 within each group.
- The group column (grp).
- Rows with NA values removed.

Examples

```
grp <- "Series"
var1 <- "control_sample"
var2 <- "treated_sample"

ebg <- expand_by_group(MetadataABA,
                       grp,
                       var1,
                       var2)

unique_series <- unique(MetadataABA$Series)

lapply(unique_series,
       function(x) subset(ebg, Series == x))
```

---

find_diffexp *Find the expression ratio for each experimental sample for the specified gene.*

Description

Find the expression ratio for each experimental sample for the specified gene.

Usage

```
find_diffexp(genes, srratio, srscore, metadata)
```

Arguments

genes	character vector that can consist of gene IDs
srratio	A dataframe of srratio
srscore	A dataframe of srratio
metadata	A dataframe of metadata

Value

Data frame of metadata with SRratio corresponding to the specified gene ID in the back row

Examples

```
vr1 <- "control_sample"
vr2 <- "treated_sample"
grp <- "Series"

ebg <- expand_by_group(MetadataABA,
                       vr1,
                       vr2,
                       grp)

SRratio <- calcSRratio(TranscriptomeABA,
                              vr1,
                              vr2,
                              ebg,
                              is.log = 1)

SRscore <- calcSRscore(SRratio)

set.seed(1)
find_diffexp(sample(SRratio$ensembl_gene_id, 1), SRratio, SRscore, MetadataABA)
```

HNscore *Reproducing HN-scores from HN-ratios Using the SRscore Package*

Description

The HN-score is a scoring metric derived from the HN-ratio, which represents the gene expression ratio between hypoxic and normoxic conditions, and was originally proposed by Tamura and Bono (2022) doi:10.3390/life12071079. It is publicly available on figshare doi:10.6084/m9.figshare.20055086. HNscore is provided as a data frame containing HN-scores calculated from logHNratioHypoxia and is implemented as test data in the SRscore package. To reduce data size, HNscore includes HN-scores for a subset of 1,000 genes extracted from the original dataset.

Usage

```
HNscore
```

Format

A data frame with 1000 rows and 11 variables:

Transcript_id_At Transcript ID in *Arabidopsis thaliana*

Upregulated Total number of times HNratio exceeds 2

Downregulated Total number of times HNratio is below 0.5

Unchanged Total number of times SRratio is between 0.5 and 2

All Maximum possible HNscore

HN.score HN-score

Gene_name_At Gene name in *Arabidopsis thaliana*

Gene_description_At Gene description in *Arabidopsis thaliana*

Protein_id_Hs Transcript ID in *Homo Sapiens*

Gene_name_Hs Gene name in *Homo Sapiens*

Gene_description_Hs Gene name in *Homo Sapiens*

Source

Tamura, Keita, and Hidemasa Bono. 2022. "Meta-Analysis of RNA Sequencing Data of Arabidopsis and Rice Under Hypoxia." Life 12 (7).

`logHNratioHypoxia` *Reproducing HN-scores from HN-ratios Using the SRscore Package*

Description

The HN-ratio, which quantifies gene expression changes between hypoxic and normoxic conditions across multiple experiments, was originally proposed by Tamura and Bono (2022) doi:10.3390/life12071079. It is publicly available on figshare doi:10.6084/m9.figshare.20055086. In the SRscore package, the HN-ratio is introduced solely as an intermediate quantity required to compute HN-scores. `logHNratioHypoxia` is a data frame containing log2-transformed HN-ratios. To reduce data size, `logHNratioHypoxia` includes HN-ratios for a subset of 1,000 genes extracted from the original dataset.

Usage

```
logHNratioHypoxia
```

Format

An object of class `data.frame` with 1000 rows and 30 columns.

Details

Column components :

Ensembl gene id + 29 treatment sample id

Source

Tamura, Keita, and Hidemasa Bono. 2022. "Meta-Analysis of RNA Sequencing Data of Arabidopsis and Rice Under Hypoxia." Life 12 (7).

`MetadataABA` *Metadata of abscisic acid stress microarray dataset in* Arabidopsis thaliana

---

Description

MetadataABA is the metadata for the experimental dataset related to Arabidopsis thaliana under ABA stress conditions. Metadata are used to define pairs for comparison between the target sample group and the experimental sample group.

Usage

```
MetadataABA
```

Format

A data frame with 19 rows and 4 variables:

Series Research project ID

control_sample control sample ID

treated_sample treatment sample ID

treatment treatment condition

tissue tissue name

`MetadataHypoxia` *Metadata of Hypoxia stress RNA-Seq dataset in* Arabidopsis thaliana

---

Description

This is metadata of RNA-Seq data that is used in the study by Tamura and Bono.

Usage

```
MetadataHypoxia
```

Format

A data frame with 29 rows and 4 variables:

study_accession Research project ID

run_accession..hypoxia. RNA-Seq run accession ID

run_accession..normoxia. RNA-Seq run accession ID

Treatment treatment condition

`plot_SRscore_distr` *Plot the Distribution of SRscore Values*

Description

This function visualizes the distribution of SRscore values using a barplot. Values equal to 0 are excluded from the plot by design because they typically represent genes without detectable stress response activity.

Usage

```
plot_SRscore_distr(srscore, log = FALSE)
```

Arguments

`srscore`	A data.frame containing at least one column named `score`, which represents the SRscore values to be plotted.
`log`	Logical (default: `FALSE`). If `TRUE`, the y-axis of the barplot is shown in logarithmic scale (`log = "y"`).

Details

The function provides both a linear-scale plot and a log-scale version, which is particularly useful when the frequency of SRscore values spans a wide range.

The function performs the following steps:

- Validates that `srscore` is a data.frame and contains a `score` column.
- Removes SRscore values equal to 0.
- Produces a barplot of the frequency of SRscore values.
- Optionally draws the plot on a logarithmic y-axis.

Value

This function returns `NULL` invisibly and produces a barplot as a side effect.

Examples

```
# Example SRscore data
df <- data.frame(score = c(-5, -3, -3, 1, 2, 2, 2, 4, 5, 5, 0))

# Linear-scale plot
plot_SRscore_distr(df)

# Log-scale plot
plot_SRscore_distr(df, log = TRUE)
```

`plot_SRscore_rank` *Plot Ranked SRscore Values with Threshold-Based Highlighting*

Description

This function visualizes SRscore values sorted in descending order and colors each point based on user-defined thresholds. Genes with SRscore above the upper threshold are colored red (up-regulated), those below the lower threshold are colored blue (down-regulated), and values within the range are shown in black.

Usage

```
plot_SRscore_rank(srscore, threshold = c(1, -1))
```

Arguments

`srscore`	A data.frame containing at least a column named `score`, representing SRscore values for genes.
`threshold`	A numeric vector of length 2 specifying `c(upper_threshold, lower_threshold)`. Default is `c(1, -1)`.

Details

The function performs the following:

- Validates input data.
- Sorts SRscore values in descending order.
- Colors each point based on whether its value is:
  - greater than or equal to the upper threshold (red)
  - less than or equal to the lower threshold (blue)
  - between the thresholds (black)
- Produces a rank plot with a legend explaining the color mapping.

Value

Invisibly returns the sorted SRscore vector. The function produces a scatter plot as a side effect.

Examples

```
df <- data.frame(
  gene = paste0("Gene", 1:10),
  score = c(-5, -3, -1, 0, 0.5, 1.2, 2, 3, 4, 5)
)

# Basic usage
plot_SRscore_rank(df)
```

```
# Custom thresholds
plot_SRscore_rank(df, threshold = c(2, -2))
```

`plot_SRscore_top` *Barplot of Top Genes Ranked by Absolute SRscore*

Description

This function selects the top `top_n` genes with the largest absolute SRscore values and visualizes their SRscores using a barplot. The function is useful for quickly identifying genes with the strongest positive or negative stress responses.

Usage

```
plot_SRscore_top(srscore, top_n = 20)
```

Arguments

`srscore`	A data.frame containing at least a column named `score`, representing the SRscore values for genes.
`top_n`	Integer (default: 20). The number of top genes to plot, ranked by `|SRscore|`.

Details

The function performs the following steps:

- Validates the input data structure.
- Computes absolute SRscore via `abs(score)`.
- Selects the top `top_n` genes with the largest absolute score.
- Re-sorts the selected genes by actual SRscore (to separate up/down).
- Produces a barplot in which gene names (character columns) are used as labels.

The barplot displays:

- Positive SRscore (upregulated genes) as upward bars.
- Negative SRscore (downregulated genes) as downward bars.
- Genes ordered from lowest to highest SRscore for visual clarity.

Graphical parameters are temporarily modified, and restored automatically using `on.exit()` to avoid affecting the user's plotting environment.

Value

Invisibly returns the data.frame of selected top genes (after sorting). A barplot is produced as a side effect.

Examples

```
# Example data.frame of SRscore
df <- data.frame(
  gene = paste0("Gene", 1:10),
  score = c(-12, -6, -3, 1, 2, 3, 5, 8, 10, 11)
)

# Plot top 5 genes by |SRscore|
plot_SRscore_top(df, top_n = 5)
```

`sample_pair_test` *Test data to create data frames from all combinations between two specified variables within each group using sample data*

Description

Test data to create data frames from all combinations between two specified variables within each group using sample data

Usage

```
sample_pair_test
```

Format

A data frame with 71 rows and 2 variables:

control_sample Control Sample ID

treated_sample Treated Sample ID

SRGA *Test data for the SRscore package*

Description

SRGA is a reference test dataset that integrates standardized SRscores across 11 stress conditions as reported in Fukuda et al. (2025) doi:10.1093/plphys/kiaf105. Because SRscore scales differ by stress type, SRscores were standardized using z-scores. This dataset is provided solely for demonstrating and testing template matching (Pavlidis and Noble, 2001) doi:10.1186/gb-2001-2-10-research0042 workflows implemented in the SRscore package and is not intended to introduce a new analysis method. To reduce file size, the dataset includes SRscores for a subset of 1,000 genes.

Usage

```
SRGA
```

Format

A data frame with 1000 rows and 13 variables:

ensembl_gene_id Ensembl gene ID

ABA SRscore derived from ABA dataset

Cold SRscore derived from cold dataset

DC3000 SRscore derived from DC3000 dataset

Drought SRscore derived from drought dataset

Heat SRscore derived from heat dataset

High-light SRscore derived from highlight dataset

Hypoxia SRscore derived from hypoxia dataset

Osmotic SRscore derived from osmotic dataset

Oxidation SRscore derived from oxidation dataset

Salt SRscore derived from salt dataset

Wound SRscore derived from wound dataset

SYMBOL Gene symbol

SRratio_test *Test data for calculating SRratio using sample data*

Description

A dataframe containing SRratio calculated from `TranscriptomeABA`

Usage

```
SRratio_test
```

Format

An object of class `data.frame` with 1000 rows and 20 columns.

Details

Column components :

Ensembl gene id + 19 treatment sample id

SRscore_test *Test data for calculating SRscore using sample data*

Description

A dataframe containing SRscore calculated from `SRratio_test`

Usage

```
SRscore_test
```

Format

A data frame with 1000 rows and 6 variables:

ensembl_gene_id Ensembl gene ID

up Total number of times SRratio exceeds 2

dn Total number of times SRratio is below 2

unchange Total number of times SRratio is between -2 and 2

all Maximum possible SRscore

score SRscore with absolute value 2 as threshold

TPMHypoxia *Hypoxia stress RNA-Seq data in* Arabidopsis thaliana

Description

This is RNA-Seq data that is used in the study by Tamura and Bono (2022) doi:10.3390/life12071079. The quantitative RNA-Seq data, which were calculated as transcripts per million (TPM), are available at figshare doi:10.6084/m9.figshare.20055086.

Usage

```
TPMHypoxia
```

Format

An object of class `data.frame` with 1000 rows and 59 columns.

Details

Column components :

Ensembl gene id + 58 sample id (control : 29, treatment : 29)

`TranscriptomeABA` *Abscisic acid stress microarray dataset in* Arabidopsis thaliana

Description

This is a gene expression matrix for Arabidopsis under ABA stress conditions. The first column is the gene ID column, all others are sample ID columns. The expression data are read as raw data (CEL files) and summarized and normalized by Robust Multi-array Average (RMA). To keep the file size small, the data is limited to 1,000 genes.

Usage

```
TranscriptomeABA
```

Format

An object of class `data.frame` with 1000 rows and 39 columns.

Details

Column components :

Ensembl gene id + 38 sample id (control : 19, treatment : 19)

Index

∗ **datasets**
HNscore, 7
logHNratioHypoxia, 8
MetadataABA, 9
MetadataHypoxia, 9
sample_pair_test, 13
SRGA, 13
SRratio_test, 14
SRscore_test, 15
TPMHypoxia, 15
TranscriptomeABA, 16

calcSRratio, 2
calcSRratio(), *5*
calcSRscore, 3
calcSRscore(), *5*

directly_calcSRscore, 4

expand_by_group, 5
expand_by_group(), *5*

find_diffexp, 6

HNscore, 7

logHNratioHypoxia, 8

MetadataABA, 9
MetadataHypoxia, 9

plot_SRscore_distr, 10
plot_SRscore_rank, 11
plot_SRscore_top, 12

sample_pair_test, 13
SRGA, 13
SRratio_test, 14
SRscore_test, 15

TPMHypoxia, 15
TranscriptomeABA, 16