Package 'csdR'

November 11, 2026

Title Differential gene co-expression

Version 1.18.0

Date 07-11-2024

Description This package contains functionality to run differential gene co-expression across two different conditions.
The algorithm is inspired by Voigt et al. 2017 and finds Conserved, Specific and Differentiated genes (hence the name CSD).
This package include efficient and variance calculation by bootstrapping and Welford's algorithm.

Depends R (>= 4.1.0)

Imports WGCNA, glue, RhpcBLASctl, matrixStats, Rcpp

License GPL-3

Encoding UTF-8

NeedsCompilation yes

BugReports https://github.com/AlmaasLab/csdR/issues

URL https://almaaslab.github.io/csdR,
https://github.com/AlmaasLab/csdR

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

LinkingTo Rcpp

biocViews DifferentialExpression, GraphAndNetwork, GeneExpression, Network

Suggests rmarkdown, knitr, testthat (>= 3.0.0), BiocStyle, magrittr, igraph, dplyr

Config/testthat/edition 3

VignetteBuilder knitr

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

git_branch RELEASE_3_23

git_last_commit 8b0fec6

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Jakob Peder Pettersen [aut, cre] (ORCID:
<https://orcid.org/0000-0002-3485-1634>)

Maintainer Jakob Peder Pettersen <jakobpeder.pettersen@gmail.com>

Contents

csdR 2
data-expression 3
partial_argsort 4
run_cor_bootstrap 4
run_csd 5

Index **8**

---

csdR *Perform differential co-expression analysis*

Description

This package provides functionality to perform differential co-expression analysis by the CSD algorithm presented by Voigt et al. 2017. The main function of the package is `run_csd` which compares gene expression for two conditions and returns values for C, S and D for each gene pair. Also, the underlying function `run_cor_bootstrap` is provided for doing correlation analysis *within* a condition, but is mostly for experimental and expect use only.

The function `partial_argsort` is aimed specifically at the task of selecting the highest values of C, S and D of the results with minimal computational overhead, but is written also to be useable for other problems of the same type.

For more details on how to approach, please read the vignette for the package or the individual manpages for the functions.

Author(s)

Maintainer: Jakob Peder Pettersen <jakobpeder.pettersen@gmail.com> (ORCID)

References

Voigt A, Nowick K and Almaas E 'A composite network of conserved and tissue specific gene interactions reveals possible genetic interactions in glioma' In: *PLOS Computational Biology* 13(9): e1005739. (doi: https://doi.org/10.1371/journal.pcbi.1005739)

See Also

Useful links:

- https://almaaslab.github.io/csdR
- https://github.com/AlmaasLab/csdR
- Report bugs at https://github.com/AlmaasLab/csdR/issues

`data-expression` *Sample expression matrices for CSD*

Description

Sample expression matrices of thyroid gland tissue for thyroid cancer patients and healthy individuals. These datasets were pre-processed by Gulla et al. (2019). Due to size requirements, only 1000 randomly selected genes are provided in the dataset. Number of samples are 399 and 504 in the healthy controls and the sick samples, respectively.

Usage

```
data(normal_expression)
data(sick_expression)
```

Format

Numeric matrices of normalized gene expression. Genes are in columns, whereas samples are in rows.

Source

For the expression matrix for healthy individuals, GenotypeTissue Expression (GTEx) V7. For the thyroid cancer patients, the data are obtained for the Thyroid Cancer project (THCA) from The Cancer Genome Atlas (TCGA).

References

Gulla, Almaas, Eivind, & Voigt, André (2019). An integrated systems biology approach to investigate transcriptomic data of thyroid carcinoma. NTNU. http://hdl.handle.net/11250/2621725

partial_argsort *Extract indecies corresponding to the largest elements*

Description

Extracts the indecies of the n largest elements of the input. This procedure is equivalent to `order(x, decreasing = TRUE)[1:n_elements]`, but is much faster and avoids the overhead of sorting discarded elements. This function is useful for extracting the rows in a data frame having the largest values in one of the columns.

Usage

```
partial_argsort(x, n_elements)
```

Arguments

`x`	Numeric vector, the vector containing the numbers to sort.
`n_elements`	Integer scalar, the number of indecies to return.

Value

Numeric vector, the indecies of the largest elements (in sorted order) in `x`.

Examples

```
x <- c(10L,5L,-2L,12L,15L)
max_indecies <- partial_argsort(x,3L)
max_indecies
x[max_indecies]
order(x)[1:3]
mtcars[partial_argsort(mtcars$hp,5L),]
```

run_cor_bootstrap *Run bootstrapping of Spearman correlations within a dataset*

Description

This function provides the more low-level functionality of bootstrapping the Spearman correlations of the columns within a dataset. Only use this function if you want a low-level interface, else `run_csd` provides a more streamlined approach if you want to do a CSD analysis.

Usage

```
run_cor_bootstrap(
  x,
  n_it = 20L,
  nThreads = 1L,
  verbose = TRUE,
  iterations_gap = 1L
)
```

Arguments

x	Numeric matrix, the gene expression matrix to analyse. Genes are in columns, samples are in rows.
n_it	Integer, number of bootstrap iterations
nThreads	Integer, number of threads to use for computations
verbose	Logical, should progress be printed?
iterations_gap	If output is verbose - Number of iterations between each status message (Default=1 - Displayed only if verbose=TRUE)

Value

A list with two fields

rho Numeric matrix constaining the bootstrapped mean of the Spearman correlation between each column

var Numeric matrix constaining the bootstrapped variance of the Spearman correlation between each column

Examples

```
data("normal_expression")
cor_res <- run_cor_bootstrap(
    x = normal_expression,
    n_it = 100, nThreads = 2L
)
```

---

run_csd *Run CSD analysis*

Description

This function implements the CSD algorithm based on the one presented by Voigt et al. 2017. All pairs of genes are first compared within each condition by the Spearman correlation and the correlation and its variance are estimated by bootstrapping. Finally, the results for the two conditions are compared and C-, S- and D-values are computed and returned.

Usage

```
run_csd(
  x_1,
  x_2,
  n_it = 20L,
  nThreads = 1L,
  verbose = TRUE,
  iterations_gap = 1L
)
```

Arguments

x_1	Numeric matrix, the gene expression matrix for the first condition. Genes are in columns, samples are in rows. The columns must be named with the name of the genes. Missing values are not allowed.
x_2	Numeric matrix, the gene expression matrix for the second condition.
n_it	Integer, number of bootstrap iterations
nThreads	Integer, number of threads to use for computations
verbose	Logical, should progress be printed?
iterations_gap	If output is verbose - Number of iterations between each status message (Default=1 - Displayed only if verbose=TRUE)

Details

The gene names in x_1 and x_2 do not need to be in the same order, but must be in the same namespace. Only genes present in both datasets will be considered for the analysis. The parallelism gained by nThreads applies to the computations within a single iteration. The iterations are run is serial in order to reduce the memory footprint.

Value

A data.frame with the additional class attribute csd_res with the results of the CSD analysis. This frame has a row for each pair of genes and has the following columns:

Gene1 Character, the name of the first gene

Gene2 Character, the name of the second gene

rho1 Mean correlation of the two genes in the first condition

rho2 Mean correlation of the two genes in the second condition

var1 The estimated variance of rho1 determined by bootstrapping

var2 The estimated variance of rho2 determined by bootstrapping

cVal Numeric, the conserved score. A high value indicates that the co-expression of the two genes have the same sign in both conditions

sVal Numeric, the specific score. A high value indicates that the co-expression of the two genes have a high degree of co-expression in one condition, but not the other.

dVal Numeric, the differentiated score. A high value indicates that the co-expression of the two genes have a high degree of co-expression in both condition, but the sign of co-expression is different.

References

Voigt A, Nowick K and Almaas E 'A composite network of conserved and tissue specific gene interactions reveals possible genetic interactions in glioma' In: *PLOS Computational Biology* 13(9): e1005739. (doi: https://doi.org/10.1371/journal.pcbi.1005739)

Examples

```
data("sick_expression")
data("normal_expression")
cor_res <- run_csd(
    x_1 = sick_expression, x_2 = normal_expression,
    n_it = 100, nThreads = 2L
)
c_max <- max(cor_res$cVal)
```

Index

∗ **datasets**
data-expression, 3
∗ **internal**
csdR, 2

csdR, 2
csdR-package *(*csdR*)*, 2

data-expression, 3

normal_expression *(*data-expression*)*, 3

partial_argsort, *2*, 4

run_cor_bootstrap, *2*, 4
run_csd, *2*, *4*, 5

sick_expression *(*data-expression*)*, 3