Package: CONSTANd (via r-universe)

November 2, 2026

Type Package

Title Data normalization by matrix raking

Version 1.20.0

Date 2020-10-30

Description Normalizes a data matrix `data` by raking (using the RAS method by Bacharach, see references) the Nrows by Ncols matrix such that the row means and column means equal 1. The result is a normalized data matrix `K=RAS`, a product of row mulipliers `R` and column multipliers `S` with the original matrix `A`. Missing information needs to be presented as `NA` values and not as zero values, because CONSTANd is able to ignore missing values when calculating the mean. Using CONSTANd normalization allows for the direct comparison of values between samples within the same and even across different CONSTANd-normalized data matrices.

License file LICENSE

Encoding UTF-8

LazyData true

biocViews MassSpectrometry, Cheminformatics, Normalization, Preprocessing, DifferentialExpression, Genetics, Transcriptomics, Proteomics

NeedsCompilation no

Depends R (>= 4.1)

Suggests BiocStyle, knitr, rmarkdown, tidyr, ggplot2, gridExtra, magick, Cairo, limma

BugReports https://github.com/PDiracDelta/CONSTANd/issues

URL qcquan.net/constand

VignetteBuilder knitr

Repository Bioconductor 3.23

Date/Publication 2026-04-28 12:54:46 UTC

RemoteUrl https://github.com/bioc/CONSTANd

RemoteRef RELEASE_3_23

RemoteSha 400ef7bcace0eb05560481b179f780fd60156326

Contents

CONSTANd 2

Index **4**

`CONSTANd` *Data normalization by matrix raking*

Description

Normalizes the data matrix by raking the Nrows by Ncols matrix such that the row means and column means equal Ncols and Nrows, respectively.

Usage

```
CONSTANd(data, precision=1e-5, maxIterations=50, target=1)
```

Arguments

`data`	Nrows by Ncols matrix.
`precision`	Combined allowed deviation (residual error) of col and row means from target value.
`maxIterations`	Maximum amount of iterations (1x row and 1x col per iteration).
`target`	The mean value of quantifications in each row and column after normalization.

Details

Normalizes the data matrix <data> by raking (using the RAS method by Bacharach, see references) the Nrows by Ncols matrix such that the row means and column means equal 1. The result is a normalized data matrix K=RAS, a product of row mulipliers R and column multipliers S with the original matrix A. Missing information needs to be presented as nan values and not as zero values, because CONSTANd is able to ignore nan-values when calculating the mean. The variable <maxIterations> is an integer value that denotes the number of raking cycles. The variable <precision> defines the stopping criteria based on the L1-norm as defined by Friedrich Pukelsheim, Bruno Simeone in "On the Iterative Proportional Fitting Procedure: Structure of Accumulation Points and L1-Error Analysis".

Value

`normalized_data`
: Normalized data matrix 'K=RAS' in the RAS-formulation of the problem.

`convergence_trail`
: Precision acquired after each raking iteration (last value is the final precision).

`R`
: Row multipliers in the 'K=RAS' formulation of the problem.

`S`
: Column multipliers in the 'K=RAS' formulation of the problem.

Author(s)

Joris Van Houtven (joris.vanhoutven@uhasselt.be), Geert Jan Bex <geertjan.bex@uhasselt.be>, Dirk Valkenborg <dirk.valkenborg@uhasselt.be>

References

Maes, Evelyne, et al. "CONSTANd: A normalization method for isobaric labeled spectra by constrained optimization." Molecular & Cellular Proteomics 15.8 (2016): 2779-2790. https://doi.org/10.1074/mcp.M115.056911. Accessed 18 Oct. 2020.

Bacharach, Michael. "Estimating Nonnegative Matrices from Marginal Data." International Economic Review, vol. 6, no. 3, 1965, pp. 294–310. JSTOR, https://doi.org/10.2307%2F2525582. Accessed 18 Oct. 2020.

Examples

```
# generic use (mock data)
data_matrix <- matrix(runif(20), c(5,4))
normalized_matrix <- CONSTANd(data_matrix)$normalized_data

# customize parameters
result <- CONSTANd(data_matrix, precision=1e-3, maxIterations=30)

# explore parts of the result object
normalized_matrix <- result$normalized_data
num_iterations_performed <- length(result$convergence_trail)
attained_precision <- result$convergence_trail[num_iterations_performed]
```

Index

* **manip**
    CONSTANd, 2

CONSTANd, 2