Package 'marr'

November 12, 2026

Type Package

Title Maximum rank reproducibility

Version 1.22.0

Date 2021-05-10

Imports Rcpp, SummarizedExperiment, utils, methods, ggplot2, dplyr, magrittr, rlang, S4Vectors

LinkingTo Rcpp

Depends R (>= 4.0)

Suggests knitr, rmarkdown, BiocStyle, testthat, covr

VignetteBuilder knitr

Encoding UTF-8

License GPL (>= 3)

Description marr (Maximum Rank Reproducibility) is a nonparametric approach that detects reproducible signals using a maximal rank statistic for high-dimensional biological data. In this R package, we implement functions that measures the reproducibility of features per sample pair and sample pairs per feature in high-dimensional biological replicate experiments. The user-friendly plot functions in this package also plot histograms of the reproducibility of features per sample pair and sample pairs per feature. Furthermore, our approach also allows the users to select optimal filtering threshold values for the identification of reproducible features and sample pairs based on output visualization checks (histograms). This package also provides the subset of data filtered by reproducible features and/or sample pairs.

biocViews QualityControl, Metabolomics, MassSpectrometry, RNASeq, ChIPSeq

BugReports https://github.com/Ghoshlab/marr/issues

RoxygenNote 7.1.1

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

git_branch RELEASE_3_23

git_last_commit afc0fdb

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Tusharkanti Ghosh [aut, cre],
Max McGrath [aut],
Daisy Philtron [aut],
Katerina Kechris [aut],
Debashis Ghosh [aut, cph]

Maintainer Tusharkanti Ghosh <tusharkantighosh30@gmail.com>

Contents

data.frameORSummarizedExperiment-class 2
Marr 3
Marr-class 4
MarrAlpha 5
MarrData 6
MarrFeatures 6
MarrFeaturesfiltered 7
MarrFeatureVars 8
MarrFilterData 8
MarrPFeatures 9
MarrPlotFeatures 10
MarrPlotSamplepairs 11
MarrProc 11
MarrPSamplepairs 12
MarrSamplepairs 13
MarrSamplepairsfiltered 14
msprepCOPD 14
vectorORNull-class 15

Index **16**

`data.frameORSummarizedExperiment-class`
S4 Class union

Description

Class union allowing `MarrData` slot to be a data.frame or Summarized Experiment

Marr *Marr*

Description

This function applies an Rcpp-based implementation of a computationally efficient method for assessing reproducibility in high-throughput experiments, called the the Marr procedure. This function also defines the Marr class and constructor.

Usage

```
Marr(
  object,
  pSamplepairs = 0.75,
  pFeatures = 0.75,
  alpha = 0.05,
  featureVars = NULL
)
```

Arguments

`object`	an object which is a `matrix` or `data.frame` with features (e.g. metabolites or genes) on the rows and samples as the columns. Alternatively, a user can provide a `SummarizedExperiment` object and the `assay(object)` will be used as input for the Marr procedure.
`pSamplepairs`	(Optional) a threshold value that lies between 0 and 1, used to assign a feature to be reproducible based on the reproducibility output of the sample pairs per feature. Default is 0.75.
`pFeatures`	(Optional) a threshold value that lies between 0 and 1, used to assign a sample pair to be reproducible based on the reproducibility output of the features per sample pair. Default is 0.75.
`alpha`	(Optional) level of significance to control the False Discovery Rate (FDR). Default is 0.05.
`featureVars`	(Optional) Vector of the columns which identify features. If a 'SummarizedExperiment' is used for 'data', row variables will be used.

Details

marr (Maximum Rank Reproducibility) is a nonparametric approach, which assesses reproducibility in high-dimensional biological replicate experiments. Although it was originally developed for RNASeq data it can be applied across many different high-dimensional biological data including MassSpectrometry based Metabolomics and ChIPSeq. The Marr procedure uses a maximum rank statistic to identify reproducible signals from noise without making any distributional assumptions of reproducible signals. This procedure can be easily applied to a variety of measurement types since it employs a rank scale.

This function computes the distributions of percent reproducible sample pairs (row-wise) per feature and percent reproducible features (column-wise) per sample pair, respectively. Additionally, it also computes the percent of reproducible sample pairs and features based on a threshold value. See the vignette for more details.

Value

A object of the class `Marr` that contains a numeric vector of the Marr sample pairs in the `MarrSamplepairs` slot, a numeric vector of the Marr features in the `MarrFeatures` slot, a numeric value of the Marr filtered features in the `MarrSamplepairsfiltered` slot, and a numeric value of the Marr filtered sample pairs in the `MarrFeaturesfiltered` slot.

References

Philtron, D., Lyu, Y., Li, Q. and Ghosh, D., 2018. Maximum Rank Reproducibility: A Nonparametric Approach to Assessing Reproducibility in Replicate Experiments. Journal of the American Statistical Association, 113(523), pp.1028-1039.

Examples

```
data <- matrix(rnorm(2400), nrow=200, ncol=12)
data_Marr <- Marr(object = data, pSamplepairs=0.75,
                  pFeatures=0.75, alpha=0.05)
data("msprepCOPD")
data_Marr_COPD <- Marr(object = msprepCOPD, pSamplepairs=0.75,
                       pFeatures=0.75, alpha=0.05)
```

`Marr-class` *the Marr class*

Description

Objects of this class store needed information to work with a Marr object

Value

`MarrSamplepairs` returns the distribution of percent reproducible features (column-wise) per sample pair, `MarrFeatures` returns the distribution of percent reproducible sample pairs (row-wise) per feature, `MarrSamplepairsfiltered` returns the percent of reproducible features based on a threshold value and `MarrFeaturesfiltered` returns the percent of reproducible sample pairs based on a threshold value

Slots

`MarrSamplepairs` Marr sample pairs

`MarrFeatures` Marr features

`MarrSamplepairsfiltered` Marr sample pairs post filtering

MarrFeaturesfiltered Marr metabolites post filtering

MarrData Original data object passed to Marr

MarrPSamplepairs Value of pSamplepairs argument passed to Marr

MarrPFeatures Value of pFeatures argument passed to Marr

MarrAlpha Value of alpha argument passed to Marr

MarrFeatureVars Value of featureVars passed to Marr. NULL if featureVars was left blank

Examples

```
data <- matrix(rnorm(2400), nrow=200, ncol=12)
data_Marr <- Marr(object = data, pSamplepairs=0.75,
                  pFeatures=0.75, alpha=0.05)
```

MarrAlpha *Generic function that returns the Marr Alpha*

Description

Given a Marr object, this function returns the Marr Alpha

Accessors for the 'MarrAlpha' slot of a Marr object.

Usage

```
MarrAlpha(object)

## S4 method for signature 'Marr'
MarrAlpha(object)
```

Arguments

object	an object of class Marr.

Value

Value of alpha argument passed to Marr

Examples

```
data <- matrix(rnorm(2400), nrow=200, ncol=12)
data_Marr <- Marr(object = data, pSamplepairs=0.75,
                  pFeatures=0.75, alpha=0.05)
MarrAlpha(data_Marr)
```

MarrData *Generic function that returns the Marr Data*

Description

Given a Marr object, this function returns the Marr Data

Accessors for the 'MarrData' slot of a Marr object.

Usage

```
MarrData(object)

## S4 method for signature 'Marr'
MarrData(object)
```

Arguments

object	an object of class Marr.

Value

Original data object passed to Marr

Examples

```
data <- matrix(rnorm(2400), nrow=200, ncol=12)
data_Marr <- Marr(object = data, pSamplepairs=0.75,
                  pFeatures=0.75, alpha=0.05)
MarrData(data_Marr)
```

MarrFeatures *Generic function that returns the Marr features*

Description

Given a Marr object, this function returns the Marr features

Accessors for the 'MarrFeatures' slot of a Marr object.

Usage

```
MarrFeatures(object)

## S4 method for signature 'Marr'
MarrFeatures(object)
```

Arguments

object	an object of class Marr.

Value

The distribution of percent reproducible sample pairs (row-wise) per feature after applying the maximum rank reproducibility.

Examples

```
data <- matrix(rnorm(2400), nrow=200, ncol=12)
data_Marr <- Marr(object = data, pSamplepairs=0.75,
                  pFeatures=0.75, alpha=0.05)
MarrFeatures(data_Marr)
```

MarrFeaturesfiltered *Generic function that returns the Marr filtered features*

Description

Given a Marr object, this function returns the Marr filtered features

Accessors for the 'MarrFeaturesfiltered' slot of a Marr object.

Usage

```
MarrFeaturesfiltered(object)

## S4 method for signature 'Marr'
MarrFeaturesfiltered(object)
```

Arguments

object	an object of class Marr.

Value

The percent of reproducible sample pairs based on a threshold value after applying maximum rank reproducibility.

Examples

```
data <- matrix(rnorm(2400), nrow=200, ncol=12)
data_Marr <- Marr(object = data, pSamplepairs=0.75,
                  pFeatures=0.75, alpha=0.05)
MarrFeaturesfiltered(data_Marr)
```

MarrFeatureVars *Generic function that returns the Marr Feature Vars*

Description

Given a Marr object, this function returns the Marr Feature Vars

Accessors for the 'MarrFeatureVars' slot of a Marr object.

Usage

```
MarrFeatureVars(object)

## S4 method for signature 'Marr'
MarrFeatureVars(object)
```

Arguments

object	an object of class Marr.

Value

Value of featureVars passed to Marr. NULL if featureVars was left blank

Examples

```
data <- matrix(rnorm(2400), nrow=200, ncol=12)
data_Marr <- Marr(object = data, pSamplepairs=0.75,
                  pFeatures=0.75, alpha=0.05)
MarrFeatureVars(data_Marr)
```

MarrFilterData *Filter by Maximum Rank Reproducibility*

Description

Filters Marr object according to the Maximum Rank Reproducibility of the features, samples pairs, or both. Features are removed if their reproducibility per sample pair is less than pFeatures. Samples are removed if their sample pair reproducibility per feature is less than pSamplepairs for all pairings of that sample and the other samples in the set.

Usage

```
MarrFilterData(object, by = c("both", "features", "samplePairs"))
```

Arguments

object	a Marr object from Marr
by	String specifying which reproducibility values to filter by. Options include "features" to filter features according to their reproducibility, "samplePairs" to filter samples according to the reproducibility of sample pairs, or "both" to filter both features and sample pairs according to their respective reproducibility. Default is "both".

Value

A list of data.frame's or a SummarizedExperiment. If a data.frame was originally input into the Marr function, a list with three elements, filteredData, removedSamples, and removedFeatures, will be returned. If a SummarizedExperiment was originally input, output will be a SummarizedExperiment with the assay filtered and with two metadata objects, removedSamples and removedFeatures added.

Examples

```
data <- matrix(rnorm(2400), nrow=200, ncol=12)
data_Marr <- Marr(object = data, pSamplepairs=0.75,
                  pFeatures=0.75, alpha=0.05)
MarrFilterData(data_Marr, by = "both")
```

MarrPFeatures *Generic function that returns the Marr P Features*

Description

Given a Marr object, this function returns the Marr P Features

Accessors for the 'MarrPFeatures' slot of a Marr object.

Usage

```
MarrPFeatures(object)

## S4 method for signature 'Marr'
MarrPFeatures(object)
```

Arguments

object	an object of class Marr.

Value

Value of MarrPFeatures argument passed to Marr

Examples

```
data <- matrix(rnorm(2400), nrow=200, ncol=12)
data_Marr <- Marr(object = data, pSamplepairs=0.75,
                  pFeatures=0.75, alpha=0.05)
MarrPFeatures(data_Marr)
```

MarrPlotFeatures *Plot percent reproducible sample pairs per feature for pairwise replicates from* Marr *function.*

Description

This function plots a histogram showing the features along the y-axis and percent reproducible sample pairs per feature on the x-axis.

Usage

```
MarrPlotFeatures(
  object,
  xLab = "Percent reproducible sample pairs per feature",
  yLab = "Feature"
)
```

Arguments

object	a Marr object from Marr
xLab	label for x-axis. Default is 'Percent reproducible sample pairs per feature for pairwise replicates'.
yLab	label for y-axis. Default is 'Feature'

Value

A histogram will be created showing the features along the y-axis and percent reproducible sample pairs per feature on the x-axis.

Examples

```
data <- matrix(rnorm(2400), nrow=200, ncol=12)
data_Marr <- Marr(object = data, pSamplepairs=0.75,
                  pFeatures=0.75, alpha=0.05)
MarrPlotFeatures(data_Marr)
```

`MarrPlotSamplepairs` *Plot percent reproducible features per sample pair for pairwise replicates from* `Marr` *function.*

Description

This function plots a histogram showing the sample pairs along the y-axis and percent reproducible features per sample pair on the x-axis.

Usage

```
MarrPlotSamplepairs(
  object,
  xLab = "Percent reproducible features per sample pair",
  yLab = "Sample pair"
)
```

Arguments

`object`	a Marr object from `Marr`
`xLab`	label for x-axis. Default is 'Percent reproducible features per sample pair for pairwise replicates'.
`yLab`	label for y-axis. Default is 'Sample pair'

Value

A histogram will be created showing the sample pairs along the y-axis and percent reproducible features per sample pair on the x-axis.

Examples

```
data <- matrix(rnorm(2400), nrow=200, ncol=12)
data_Marr <- Marr(object = data, pSamplepairs=0.75,
                  pFeatures=0.75, alpha=0.05)
MarrPlotSamplepairs(data_Marr)
```

`MarrProc` *MarrProc*

Description

This function is a helper function that computes distributions of reproducible sample pairs per feature and reproducible features per sample pair for the function `Marr`.

Usage

```
MarrProc(object, alpha = 0.05)
```

Arguments

object	an object which is a matrix or data.frame with features (e.g. metabolites or genes) on the rows and samples as the columns. Alternatively, a user can provide a SummarizedExperiment object and the assay(object) will be used as input for the Marr procedure.
alpha	(Optional) level of significance to control the False Discovery Rate (FDR). Default is 0.05.

Value

A list of percent reproducible statistics including

samplepairs	the distribution of percent reproducible features (column-wise) per sample pair
features	the distribution of percent reproducible sample pairs (row-wise) per feature

Examples

```
data <- matrix(rnorm(2400), nrow=200, ncol=12)
data_MarrProc <- MarrProc(object=data, alpha = 0.05)
```

MarrPSamplepairs *Generic function that returns the Marr P Sample Pairs*

Description

Given a Marr object, this function returns the Marr P Sample Pairs

Accessors for the 'MarrPSamplepairs' slot of a Marr object.

Usage

```
MarrPSamplepairs(object)

## S4 method for signature 'Marr'
MarrPSamplepairs(object)
```

Arguments

object	an object of class Marr.

Value

Value of pSamplepairs argument passed to Marr

Examples

```
data <- matrix(rnorm(2400), nrow=200, ncol=12)
data_Marr <- Marr(object = data, pSamplepairs=0.75,
                  pFeatures=0.75, alpha=0.05)
MarrPSamplepairs(data_Marr)
```

`MarrSamplepairs` *Generic function that returns the Marr sample pairs*

Description

Given a Marr object, this function returns the Marr sample pairs

Accessors for the 'MarrSamplepairs' slot of a Marr object.

Usage

```
MarrSamplepairs(object)

## S4 method for signature 'Marr'
MarrSamplepairs(object)
```

Arguments

`object`	an object of class `Marr`.

Value

The distribution of percent reproducible features (column-wise) per sample pair after applying the maximum rank reproducibility.

Examples

```
data <- matrix(rnorm(2400), nrow=200, ncol=12)
data_Marr <- Marr(object = data, pSamplepairs=0.75,
                  pFeatures=0.75, alpha=0.05)
MarrSamplepairs(data_Marr)
```

MarrSamplepairsfiltered *Generic function that returns the Marr filtered sample pairs*

Description

Given a Marr object, this function returns the Marr filtered sample pairs

Accessors for the 'MarrSamplepairsfiltered' slot of a Marr object.

Usage

```
MarrSamplepairsfiltered(object)

## S4 method for signature 'Marr'
MarrSamplepairsfiltered(object)
```

Arguments

object	an object of class Marr.

Value

The percent of reproducible features based on a threshold value after applying maximum rank reproducibility.

Examples

```
data <- matrix(rnorm(2400), nrow=200, ncol=12)
data_Marr <- Marr(object = data, pSamplepairs=0.75,
                  pFeatures=0.75, alpha=0.05)
MarrSamplepairsfiltered(data_Marr)
```

msprepCOPD *Example of processed mass spectrometry dataset*

Description

Data contains LC-MS metabolite analysis for samples from 20 subjects. and 662 metabolites. The raw data was pre-processed using MSPrep method. The raw data pre- processing include 3 steps- Filtering, Missing Value Imputation and Normalization. Filtering- the metabolites(columns) in the raw data were removed if they were missing more than 80 percent of the samples. Missing Value Imputation- The Bayesian Principal Component Analysis (BPCA) was applied to impute the missing values. Normalization- median normalization was applied to remove unwanted variation appears from various sources in metabolomics studies. The first three columns indicate "Mass" indicating the mass-to-charge ratio, "Retention.Time", and "Compound.Name" for each present metabolite. The remaining columns indicate abundance for each of the 645 mass/retention-time combination for each subject combination.

Usage

```
data(msprepCOPD)
```

Format

SummarizedExperiment assay object containing 645 metabolites (features) of 20 subjects (samples).

Mass Mass-to-charge ratio

Retention.Time Retention-time

Compound.Name Compound name for each mass/retention time combination

X10062C The columns indicate metabolite abundances found in each subject combination. Each column begins with an 'X', followed by the subject ID.

Source

https://www.metabolomicsworkbench.org/data/DRCCMetadata.php?Mode=Project&ProjectID=PR000438

The raw data is available at the NIH Common Fund's National Metabolomics Data Repository (NMDR) website, the Metabolomics Workbench, https://www.metabolomicsworkbench.org, where it has been assigned Project ID PR000438. The raw data can be accessed directly via it's Project DOI: 10.21228/M8FC7C This work is supported by NIH grant, U2C- DK119886.

References

Nichole Reisdorph. Untargeted LC-MS metabolomics analysis of human COPD plasma, HILIC & C18, metabolomics_workbench, V1.

Hughes, G., Cruickshank-Quinn, C., Reisdorph, R., Lutz, S., Petrache, I., Reisdorph, N., Bowler, R. and Kechris, K., 2014. MSPrep—Summarization, normalization and diagnostics for processing of mass spectrometry–based metabolomic data. Bioinformatics, 30(1), pp.133-134.

Examples

```
data(msprepCOPD)
```

---

vectorORNull-class *S4 Class union*

Description

Class union allowing `MarrFeatureVars` slot to be a vector or NULL

Index

∗ **datasets**
msprepCOPD, 14

data.frameORSummarizedExperiment-class, 2

Marr, 3
Marr-class, 4
MarrAlpha, 5
MarrAlpha,Marr-method *(*MarrAlpha*)*, 5
MarrData, 6
MarrData,Marr-method *(*MarrData*)*, 6
MarrFeatures, 6
MarrFeatures,Marr-method *(*MarrFeatures*)*, 6
MarrFeaturesfiltered, 7
MarrFeaturesfiltered,Marr-method *(*MarrFeaturesfiltered*)*, 7
MarrFeatureVars, 8
MarrFeatureVars,Marr-method *(*MarrFeatureVars*)*, 8
MarrFilterData, 8
MarrPFeatures, 9
MarrPFeatures,Marr-method *(*MarrPFeatures*)*, 9
MarrPlotFeatures, 10
MarrPlotSamplepairs, 11
MarrProc, 11
MarrPSamplepairs, 12
MarrPSamplepairs,Marr-method *(*MarrPSamplepairs*)*, 12
MarrSamplepairs, 13
MarrSamplepairs,Marr-method *(*MarrSamplepairs*)*, 13
MarrSamplepairsfiltered, 14
MarrSamplepairsfiltered,Marr-method *(*MarrSamplepairsfiltered*)*, 14
msprepCOPD, 14

vectorORNull-class, 15