Package: mdp (via r-universe)

November 25, 2026

Title Molecular Degree of Perturbation calculates scores for transcriptome data samples based on their perturbation from controls

Version 1.34.0

Description The Molecular Degree of Perturbation webtool quantifies the heterogeneity of samples. It takes a data.frame of omic data that contains at least two classes (control and test) and assigns a score to all samples based on how perturbed they are compared to the controls. It is based on the Molecular Distance to Health (Pankla et al. 2009), and expands on this algorithm by adding the options to calculate the z-score using the modified z-score (using median absolute deviation), change the z-score zeroing threshold, and look at genes that are most perturbed in the test versus control classes.

URL https://mdp.sysbio.tools/

biocViews BiomedicalInformatics, QualityControl, Transcriptomics, SystemsBiology, Microarray, QualityControl

Depends R (>= 4.0)

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Imports ggplot2, gridExtra, grid, stats, utils

Suggests testthat, knitr, rmarkdown, fgsea, BiocManager

VignetteBuilder knitr

Repository Bioconductor 3.24

Date/Publication 2026-11-23 22:40:47 UTC

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

RemoteRef RELEASE_3_24

RemoteSha 3d02b1dacda769f1d0003af96ca50c3931ad2e59

Contents

compute_gene_score 2
compute_perturbed_genes 3
compute_sample_scores 3
compute_zscore 4
example_data 5
example_pheno 5
mdp 6
pathway_summary 7
sample_data 8
sample_plot 8

Index **10**

`compute_gene_score` *Compute gene score Computes gene scores for each gene within each class and perturbation freq*

Description

Compute gene score Computes gene scores for each gene within each class and perturbation freq

Usage

```
compute_gene_score(zscore, pdata, control_lab,
  score_type = c("gene_score", "gene_freq"))
```

Arguments

`zscore`	zscore data frame
`pdata`	phenotypic data with Class and Sample columns
`control_lab`	character specifying control class
`score_type`	set to 'gene_score' or 'gene_freq' to compute gene scores or frequencies

Value

data frame of gene scores or gene frequencies

compute_perturbed_genes

Compute perturbed genes Find the top fraction of genes that are more perturbed in test versus controls

Description

Compute perturbed genes Find the top fraction of genes that are more perturbed in test versus controls

Usage

```
compute_perturbed_genes(gmdp_results, control_lab, fraction_genes)
```

Arguments

gmdp_results	results table of gene scores
control_lab	label specificying control class
fraction_genes	fraction of top perturbed genes that will make the set of perturbed genes

Value

vector of perturbed genes

compute_sample_scores *Compute sample scores for each pathway*

Description

Compute sample scores for each pathway

Usage

```
compute_sample_scores(zscore, perturbed_genes, control_samples,
  test_samples, pathways, pdata)
```

Arguments

zscore	zscore data frame
perturbed_genes	list of pertured genes
control_samples	vector of control sample names
test_samples	vector of test sample names
pathways	list of pathways
pdata	phenotypic data with Sample and Class columns

Value

data frame of sample scores

`compute_zscore` *Computes the thresholded Z score Plots the Z score using control samples to compute the average and standard deviation*

Description

Computes the thresholded Z score Plots the Z score using control samples to compute the average and standard deviation

Usage

```
compute_zscore(data, control_samples, measure = c("mean", "median"),
  std = 2)
```

Arguments

`data`	Gene expression data with gene symbols in rows, sample names in columns
`control_samples`	Character vector specifying the control sample names
`measure`	Either 'mean' or 'median'. 'mean' uses mean and standard deviation. 'median' uses the median and the median absolute deviation to estimate the standard deviation (modified z-score).
`std`	Set as default to 2. This controls the standard deviation threshold for the Z-score calculation. #' Normalised expression values less than 'std' will be set to 0.

Value

zscore data frame

Examples

```
control_samples <- example_pheno$Sample[example_pheno$Class == 'baseline']
compute_zscore(example_data, control_samples,'median',2)
```

example_data *Expression data example*

Description

rownames HGNC gene names

colnames sample expression data ...

Usage

```
example_data
```

Format

A data frame with 13838 rows and 40 variables:

Details

Author expression data for GEO dataset GSE17156 of transcriptome blood samples from patients that were inoculated with the RSV virus that has been altered by collapsing for HGNC gene symbols.

Source

https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE17156

example_pheno *Phenotypic data example*

Description

Subset of the annotation data for GEO dataset GSE17156, using only patients that have been inoculated with the RSV virus

Usage

```
example_pheno
```

Format

A data frame with 40 rows and 2 variables:

Sample GSM identified

Class Symtpomatic state ...

Source

https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE17156

mdp *Molecular Degree of Perturbation*

Description

Based on the Molecular Distance to Health, this function calculates scores to each sample based on their perturbation from healthy

Usage

```
mdp(data, pdata, control_lab, directory = "", pathways, print = TRUE,
  measure = c("mean", "median"), std = 2, fraction_genes = 0.25,
  save_tables = TRUE, file_name = "")
```

Arguments

data	data frame of gene expression data with the gene symbols in the row names
pdata	data frame of phenodata with a column headed Class and the other headed Sample.
control_lab	character vector specifying the control class
directory	(optional) character string of output directory
pathways	(optional) list whose names are pathways and elements are genes in the pathway. see details section for more information
print	set as default to TRUE for pdfs of the sample scores to be saved
measure	'medan' as default, can change to 'median'. mean will select for z-score and median will select for modified z-score. (see details)
std	numeric set as default to 2, this governs the thresholding of expression data. z-scored expression values with absolute value less than 'std' will be set to 0.
fraction_genes	numeric fraction of genes that will contribute to the top perturbed genes. Set as default to 0.25
save_tables	Set as default to TRUE. Tables of zscore and gene and sample scores will be saved.
file_name	(optional) character string that will be added to the saved file names

Value

A list: zscore, gene_scores, gene_freq, sample_scores, perturbed_genes

- Z-score - z-score is calculated using the control samples to compute the average and the standard deviation. The absolute value of this matrix is taken and values less than the std are set to zero. This z-score data frame is used to compute the gene and sample scores.
- Gene scores - mean z-score value for each gene in each class
- Gene frequency - frequency with which a gene has a non zero z-score value in each class

- Sample scores - list containing sample scores for different genesets. Sample scores are the sum of the z-scored gene values for each sample, averaged for the number of genes in that geneset.
- Perturbed genes - vector of the top fraction of genes that have higher gene scores in the test classes compared to the control.
- Pathways - if genesets are provided, they are ranked according to the signal-to-noise #' ratio of test sample scores versus control sample scores calculated using that geneset.

Loading pathways

a `list` of pathways can be loaded from a .gmt file using the `fgsea` function using `fgsea::gmtPathways('gmt.file.locati`

Selecting mean or median

if `median` is selected, the z-score will be calculated using the median, and the standard deviation will be estimated using the median absolute deviation, utilising the `mad` function.

Examples

```
# basic run
mdp(example_data,example_pheno,'baseline')
# run with pathways
pathway_file <- system.file('extdata', 'ReactomePathways.gmt',
package = 'mdp')
mypathway <- fgsea::gmtPathways(pathway_file) # load a gmt file
mdp(data=example_data,pdata=example_pheno,control_lab='baseline',
pathways=mypathway)
```

---

pathway_summary *print pathways generates a summary plot for pathways and sample score plot of best gene set*

Description

print pathways generates a summary plot for pathways and sample score plot of best gene set

Usage

```
pathway_summary(sample_results, path, file_name, control_samples,
  control_lab)
```

Arguments

`sample_results`	list of sample scores for each geneset
`path`	directory to save images
`file_name`	name of saved imaged
`control_samples`	list of control sample names
`control_lab`	label that specifies control class

Value

data frame of signal to noise ratio of control vc test sample scores for each pathway

---

`sample_data` *Sample score results*

Description

Resultant sample scores when the mdp is applied to example_data and example_pheno

Usage

```
sample_data
```

Format

A data frame with 40 rows and 3 variables:

Sample GSM identified

Score Sample score

Class Symtpomatic state ...

`sample_plot` *Plot sample scores Plots the sample scores data.frame for a given geneset. Data frame must have Score, Sample and Class columns*

Description

Plot sample scores Plots the sample scores data.frame for a given geneset. Data frame must have Score, Sample and Class columns

Usage

```
sample_plot(sample_data, filename = "", directory = "", title = "",
  print = TRUE, display = TRUE, control_lab)
```

Arguments

`sample_data`	`data frame` of sample score information for a geneset. Must have columns 'Sample', 'Score' and 'Class'
`filename`	(optional) character string that will be added to the saved pdf filename
`directory`	(optional) character string of directory to save file
`title`	(optional) character string of title name for graph
`print`	(default TRUE) Save as a pdf file
`display`	(default TRUE) Display plot
`control_lab`	(optional) character string Specifying control_lab will set the control class as light blue as a default

Value

generates a plot of the sample scores

Examples

```
sample_plot(sample_data = sample_data, control_lab = 'baseline')
```

Index

∗ **datasets**
    example_data, 5
    example_pheno, 5
    sample_data, 8

compute_gene_score, 2
compute_perturbed_genes, 3
compute_sample_scores, 3
compute_zscore, 4

example_data, 5
example_pheno, 5

mdp, 6

pathway_summary, 7

sample_data, 8
sample_plot, 8