Package 'RGSEA'

May 26, 2026

Type Package

Title Random Gene Set Enrichment Analysis

Version 1.47.0

Date 2014-04-22

Author Chengcheng Ma

biocViews GeneSetEnrichment, StatisticalMethod, Classification

Maintainer Chengcheng Ma <ccma@sibs.ac.cn>

Description Combining bootstrap aggregating and Gene set enrichment analysis (GSEA), RGSEA is a classfication algorithm with high robustness and no over-fitting problem. It performs well especially for the data generated from different exprements.

License GPL(>=3)

Depends R(>= 2.10.0)

Imports BiocGenerics

VignetteBuilder knitr

Suggests BiocStyle, GEOquery, knitr, RUnit

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

git_branch devel

git_last_commit eba6181

git_last_commit_date 2026-04-28

Repository Bioconductor 3.24

Date/Publication 2026-05-25

Contents

RGSEA-package *Random Gene Set Enrichment Analysis (RGSEA)*

Description

This is the package for similarity identifucation and classification of transcriptome data

Details

Package:	RGSEA
Type:	Package
Version:	1.0
Date:	2014-04-22
License:	GPL(>=3)

~~ An overview of how to use the package, including the most important functions ~~

Author(s)

Chengcheng Ma Maintainer: tcmcc1986@gmail.com

References

~~ Literature or other references for background information ~~

See Also

Song L, Langfelder P, Horvath S. Random generalized linear model: a highly accurate and interpretable ensemble predictor[J]. BMC bioinformatics, 2013, 14(1): 5. Subramanian A, Tamayo P, Mootha V K, et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles[J]. Proceedings of the National Academy of Sciences of the United States of America, 2005, 102(43): 15545-15550.

Examples

```
if(interactive()) {
    data(e1)
    data(e2)
    RGSEAfix(e1,e2, queryclasses=colnames(e1), refclasses=colnames(e2),
random=20000, featurenum=1000, iteration=100)->test
}
```

cmap *Data from Connectivity map build 01*

Description

It is the sample data used for testing the function RGSEAsd.

Usage

```
data(cmap)
```

Format

The format is: num [1:22268, 1:6] 0.4892 -0.6137 3.5242 -0.0139 -2.0255 ... - attr(*, "dimnames")=List of 2 ..$: chr [1:22268] "1007_s_at" "1053_at" "117_at" "121_at" ... ..$: chr [1:6] "thioridazine" "tretinoin" "prochlorperazine" "chlorpromazine" ...

Details

The query data is the instance 5202764005791175120104.C08, treated with thioridazine.

The reference data is 5202764005789148112904.G05, X5202764005789148112904.F03, X5202764005789148112904.
X5202764005789148112904.E02, X5202764005789148112904.E04. They were treated with tretinoin, prochlorperazine, chlorpromazine, vorinostat, sirolimus respectively. All the data were generated by MCF7 cell line.

Source

http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE5258

References

Lamb J, Crawford ED, Peck D, Modell JW et al. The Connectivity Map: using gene-expression signatures to connect small molecules, genes, and disease. Science 2006 Sep 29;313(5795):1929-35. PMID: 17008526

Examples

```
data(cmap)
```

e1 *Data from GDS4102*

Description

This is the query data for testing the function RGSEAfix.

Usage

```
data(e1)
```

Format

The format is: num [1:54675, 1:2] 1012 44.32 43.03 36.65 4.92 ... - attr(*, "dimnames")=List of 2 ..$: chr [1:54675] "1007_s_at" "1053_at" "117_at" "121_at" ... ..$: chr [1:2] "tumor" "normal"

Details

The two data are GSM414924 GSM414975 repectively.

Source

http://www.ncbi.nlm.nih.gov/sites/GDSbrowser?acc=GDS4102

References

Pei H, Li L, Fridley BL, Jenkins GD et al. FKBP51 affects cancer cell response to chemotherapy by negatively regulating Akt. Cancer Cell 2009 Sep 8;16(3):259-66. PMID: 19732725

Examples

```
data(e1)
```

---

e2 *Data from GDS4100*

Description

This is the reference data for testing the function RGSEAfix.

Usage

```
data(e2)
```

Format

The format is: num [1:54675, 1:24] 1.48 1.19 0.67 2.75 NA 0.51 1.68 NA NA 1.99 ... - attr(*, "dimnames")=List of 2 ..$: chr [1:54675] "1007_s_at" "1053_at" "117_at" "121_at" ... ..$: chr [1:24] "tumor" "tumor" "tumor" "tumor" ...

Details

This dataset contains all the 4 data of GDS4100.

Source

http://www.ncbi.nlm.nih.gov/sites/GDSbrowser?acc=GDS4100

References

Zhang L, Farrell JJ, Zhou H, Elashoff D et al. Salivary transcriptomic biomarkers for detection of resectable pancreatic cancer. Gastroenterology 2010 Mar;138(3):949-57.e1-7. PMID: 19931263

Examples

```
data(e2)
```

RGSEAfix *Random Gene Set Enrichment Analysis with fixed number of features*

Description

This is the function for classification and feature selection with fixed number of features from top and bottom of the subtset features.

Usage

```
RGSEAfix(query, reference, queryclasses, refclasses, random = 5000, featurenum
 = 500, iteration = 100)
```

Arguments

query	A matrix, The query data. This is the data which the research wants to know the class.
reference	A matrix. The reference data. Based of the reference data, the research infer the class of query data.
queryclasses	A character vector. It contains the classes of query data. If you don't know the classes of query data, just give it a character vector equal to the number of query data.
refclasses	A character vector. It contains the classes of reference data. You must know it.
random	A numeric variable. The number of features in the subset randomly sampled from the whole features each time.
featurenum	A numeric varialbe. The number of features selected from top and bottom of the subset respectivelly.
iteration	A numeric varialbe. The times of random sampling.

Value

[1] The times of each sample in the reference dataset is the most similar to the query data. [2] The frequencey of features selected from the top and bottom of the subsets from the query data, if the query data is correcly classified.

Author(s)

Chengcheng Ma

Examples

```
if(interactive()) {
    data(e1)
    data(e2)
    RGSEAfix(e1,e2, queryclasses=colnames(e1), refclasses=colnames(e2),
random=20000, featurenum=1000, iteration=100)->test
}
```

RGSEApredict *Predict the class of the query data with the result of RGSEA functions*

Description

Predict the class of the query data with the result of RGSEA functions–RGSEAfix or RGSEAsd

Usage

```
RGSEApredict(RGSEAresult, refclasses)
```

Arguments

RGSEAresult	The first item of the results generated by RGSEA functions.
refclasses	A character vector. The classes of the reference data.

Author(s)

Chengcheng Ma

Examples

```
if(interactive()) {
    data(e1)
    data(e2)
    RGSEAfix(e1,e2, queryclasses=colnames(e1), refclasses=colnames(e2),
random=20000, featurenum=1000, iteration=100)->test
    RGSEApredict(test[[1]], colnames(e2))
}
```

RGSEAsd *Random Gene Set Enrichment Analysis features selected based on standard deviation from the mean value*

Description

This is the function for classification with features from top and bottom deviating from the mean value of the whole transcriptome for a certain standard deviations of the subtset features.

Usage

```
RGSEAsd(query, reference, queryclasses, refclasses, random = 5000, sd = 2, iteration = 100)
```

Arguments

query	A matrix, The query data. This is the data which the research wants to know the class.
reference	A matrix. The reference data. Based of the reference data, the research infer the class of query data.
queryclasses	A character vector. It contains the classes of query data. If you don't know the classes of query data, just give it a character vector equal to the number of query data.
refclasses	A character vector. It contains the classes of reference data. You must know it.
random	A numeric variable. The number of features in the subset randomly sampled from the whole features each time.
sd	number of standard deviations the features selected from the subset deviate from the mean value of the subset.
iteration	A numeric varialbe. The times of random sampling.

Value

[1] The times of each sample in the reference dataset is the most similar to the query data. [2] The frequencey of features selected from the top and bottom of the subsets from the query data, if the query data is correcly classified.

Author(s)

Chengcheng Ma

Examples

```
if(interactive()) {
    data(cmap)
    test <- RGSEAsd(cmap[,1],cmap[,2:6], queryclasses=colnames(cmap)[1],
      refclasses=colnames(cmap)[2:6], random=5000, sd=2, iteration=100)
}
```

Index