Package 'GenomicSuperSignature'

January 20, 2025

Title Interpretation of RNA-seq experiments through robust, efficient comparison to public databases

Version 1.14.0

Date 2022-9-28

Description This package provides a novel method for interpreting new transcriptomic datasets through near-instantaneous comparison to public archives without high-performance computing requirements. Through the pre-computed index, users can identify public resources associated with their dataset such as gene sets, MeSH term, and publication. Functions to identify interpretable annotations and intuitive visualization options are implemented in this package.

Depends R (>= 4.1.0), SummarizedExperiment

Imports ComplexHeatmap, ggplot2, methods, S4Vectors, Biobase, ggpubr, dplyr, plotly, BiocFileCache, grid, flextable, irlba

Suggests knitr, rmarkdown, devtools, roxygen2, pkgdown, usethis, BiocStyle, testthat, forcats, stats, wordcloud, circlize, EnrichmentBrowser, clusterProfiler, msigdbr, cluster, RColorBrewer, reshape2, tibble, BiocManager, bcellViper, readr, utils

License Artistic-2.0

biocViews Transcriptomics, SystemsBiology, PrincipalComponent, RNASeq, Sequencing, Pathways, Clustering

Encoding UTF-8

LazyData true

RoxygenNote 7.2.1

VignetteBuilder knitr

URL https://github.com/shbrief/GenomicSuperSignature

BugReports https://github.com/shbrief/GenomicSuperSignature/issues

Collate 'GenomicSignatures-class.R' 'GenomicSignatures-methods.R' 'PCAGenomicSignatures-methods.R' 'annotatePC.R' 'annotateRAV.R' 'availableRAVmodel.R' 'buildAvgLoading.R' 'calculateScore.R' 'data.R' 'drawWordcloud.R' 'extractPC.R' 'findSignature.R' 'findStudiesInCluster.R' 'getMetadata.R' 'getModel.R' 'heatmapTable.R' 'plotAnnotatedPCA.R' 'plotValidate.R' 'rmNaInf.R' 'sampleScoreHeatmap.R' 'subsetEnrichedPathways.R' 'utils.R' 'validate.R' 'validatedSignatures.R'

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

git_branch RELEASE_3_20

git_last_commit 42a9dcd

git_last_commit_date 2024-10-29

Repository Bioconductor 3.20

Date/Publication 2025-01-20

Author Sehyun Oh [aut, cre],
Levi Waldron [aut],
Sean Davis [aut]

Maintainer Sehyun Oh <shbrief@gmail.com>

Contents

`.calculateSilhouetteWidth` *Calculate Silhouette Information of RAVs*

Description

The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The silhouette width ranges from -1 to +1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters.

Usage

```
.calculateSilhouetteWidth(dat, kmeansRes)
```

Arguments

`dat`	A matrix with all the top PCs from training data to be clustered.
`kmeansRes`	Output from `stats::kmeans`.

Value

Silhouette-class object, which is an n x 3 matrix with attributes.

See Also

`kmeans`

`.loadingCor` *Validating new dataset*

Description

Validating new dataset

Usage

```
.loadingCor(dataset, avgLoading, method = "pearson", scale = FALSE)
```

Arguments

`dataset`	A gene expression profile to be validated. Different classes of objects can be used including ExpressionSet, SummarizedExperiment, RangedSummarizedExperiment, or matrix. Rownames (genes) should be in human gene symbol format (HGNC). If dataset is a matrix, genes should be in rows and samples in columns. RNA-seq counts should be log(count + 1) prior to the 'validate()' call.
`avgLoading`	A matrix with genes by RAVs.
`method`	A character string indicating which correlation coefficient is to be computed. One of "pearson" (default), "kendall", or "spearman": can be abbreviated.
`scale`	Default is `FALSE`. If it is set to `TRUE`, rows will be converted to z-score prior to PCA.

Value

A matrix of Pearson correlation coefficient (default, defined through `method` argument) between RAVs (row) and the top 8 PCs from the datasets (column)

`.RAVName` *Formatting RAV name*

Description

Keep the name with 'k + cluster number + number of PCs + number of unique studies' info during the model construction to make it easy to keep track of them, but at the PCAGenomicSignatures-class object building step, covert them into 'RAV + cluster number'.

Usage

```
.RAVName(x, ...)
```

Arguments

`x`	PCAGenomicSignatures object
`...`	Additional arguments for supporting functions.

Value

a character vector

`annotatePC` *Annotate top PCs from the dataset*

Description

This function finds the RAV with the highest validation score (including RAVs with negative silhouette width) for specified PC of the dataset and returns the top enriched pathways.

Usage

```
annotatePC(
  PCnum,
  val_all,
  RAVmodel,
  n = 5,
  scoreCutoff = 0.5,
  nesCutoff = NULL,
  simplify = TRUE,
  abs = FALSE,
  trimed_pathway_len = 45
)
```

Arguments

PCnum	A numeric vector. PC number of your dataset to retrieve annotation results for. The vector can contain any integer number among 1:8.
val_all	The output from validate
RAVmodel	The RAVmodel used to generate the input for the argument, val_all.
n	An integer. Default is 5. The number of the top enriched pathways to print out. If there are fewer than n pathways passed the cutoff, it will print out NA.
scoreCutoff	A numeric value for the minimum correlation between loadings of the dataset principal component and the RAV. Default is 0.5.
nesCutoff	A numeric value for the minimum Normalized Enrichment Score (NES) for the enrichment analysis. Default is NULL The suggested value is 3.
simplify	A logical. Under default (TRUE), the output will be a data frame with the number of column same as the length of PCnum argument, and the number of row same as the n argument. If it is set to FALSE, the output will be a list with the length of PCnum argument, where each element is a data frame containing detailed GSEA output of enriched pathways.
abs	Default is FALSE. If it's set to TRUE, the enriched pathways will be listed based on absolute value of the Normalized Enrichment Score (NES).
trimed_pathway_len	Positive integer values, which is the display width of pathway names. Default is 45.

Value

A data frame of a list based on the simplify argument. Check the output detail above.

Examples

```
data(miniRAVmodel)
library(bcellViper)
data(bcellViper)
val_all <- validate(dset, miniRAVmodel)
annotatePC(2, val_all, miniRAVmodel)
```

---

annotateRAV *Search the top enriched pathways for RAV*

Description

Search the top enriched pathways for RAV

Usage

```
annotateRAV(RAVmodel, ind, n = 5, abs = FALSE)
```

Arguments

RAVmodel	PCAGenomicSignatures object.
ind	An integer for RAV you want to check the enriched pathways.
n	A number of top enriched pathways to output. Default is 5.
abs	Default is FALSE. If it's set to TRUE, the enriched pathways will be listed based on abs(NES).

Value

A data frame with n rows and 4 columns; Description, NES, pvalue, and qvalues

Examples

```
data(miniRAVmodel)
annotateRAV(miniRAVmodel, ind = 695)
```

---

availableRAVmodel *List the available RAVmodels*

Description

List the available RAVmodels

Usage

```
availableRAVmodel(simplify = TRUE)
```

Arguments

simplify	Default is TRUE. If it is set to FALSE, the additional metadata of different versions of RAVmodel

Value

Under the default, this function will return a data frame with four columns - prior, version, update, pkg_version.

- prior : Different gene sets used for RAVmodel annotation. Currently, two are available - C2 for MSigDB C2 (curated gene sets), and PLIERpriors for bloodCellMarkersIRISDMAP, svmMarkers, and canonicalPathways
- version : RAVmodel's version, which can be an input for version argument of getModel function
- update : Date the RAVmodel is updated
- pkg_version : Compatible version of GenomicSuperSignature

Examples

```
availableRAVmodel()
```

`buildAvgLoading` *Calculate average loadings of each cluster*

Description

Calculate average loadings of each cluster

Usage

```
buildAvgLoading(dat, k, n = 20, cluster = NULL, study = TRUE)
```

Arguments

`dat`	A data frame. Each row represents principle components from different training datasets. Columns are genes used for PCA analysis.
`k`	The number of clusters used for hierarchical clustering
`n`	The number of top principle components from each datasets used for model building. Default is 20.
`cluster`	Provide pre-defined cluster membership of your data.
`study`	Under default (`TRUE`), studies involved in each cluster will be added in the output.

Value

A named list of 6 elements is returned. It contains:

`cluster` A numeric vector on cluster membership of PCs

`size` A integer vector on the size of clusters

`avgLoading` A matrix of average loadings. Columns for clusters and rows for genes

`k` The number of clusters

`n` The number of top PCs used for clustering

`studies` A list of character vector containing studies in each cluster

Examples

```
data(miniAllZ)
data(res_hcut)
res <- buildAvgLoading(miniAllZ, k = 40, cluster = res_hcut$cluster)
```

calculateScore *Calculate the validation score for a new dataset*

Description

Calculate the validation score for a new dataset

Usage

```
calculateScore(dataset, RAVmodel, rescale.after = TRUE)
```

Arguments

dataset	A gene expression profile to be validated. Different classes of objects can be used including ExpressionSet, SummarizedExperiment, RangedSummarizedExperiment, or matrix. Rownames (genes) should be in symbol format. If it is a matrix, genes should be in rows and samples in columns.
RAVmodel	PCAGenomicSignatures object. A matrix of average loadings, an output from buildAvgLoading, can be directly provided.
rescale.after	Under the default (TRUE), the continuous scores are rescaled post assignment, so average loadings have the same standard deviation in different studies. If it is FALSE, the rescaling of column (= dividing by sqrt(sum(x^2)) is done before score assignment.

Value

A list containing the score matrices for input datasets. Scores are assigned to each sample (row) for each cluster (column).

Examples

```
data(miniRAVmodel)
library(bcellViper)
data(bcellViper)
score <- calculateScore(dset, miniRAVmodel)

data(miniTCGA)
score <- calculateScore(miniTCGA, miniRAVmodel)
```

drawWordcloud *Draw wordcloud using the collection of RAVs' MeSH terms*

Description

Plot a word cloud using the remaining MeSH terms in the selected RAV after user-defined filtering.

Usage

```
drawWordcloud(
  RAVmodel,
  ind,
  rm.noise = NULL,
  scale = c(3, 0.5),
  weighted = TRUE,
  drop = NULL,
  filterMessage = TRUE
)
```

Arguments

RAVmodel	PCAGenomicSignatures object
ind	An index of the RAV you want to draw wordcloud.
rm.noise	An integer. Under the default (rm.noise=NULL), if cluster size (= s) is smaller than 8, rm.noise = floor(s*0.5). For clusters with >= 8 PCs, rm.noise = 4. If rm.noise = 0, all the MeSH terms in RAV will be used to draw wordcloud.
scale	A scale argument for wordcloud function
weighted	A logical. If TRUE (default), MeSH terms from each study are weighted based on the variance explained by the principle component of the study contributing to a given RAV.
drop	A character vector containing MeSH terms to be excluded from word cloud. Under the default (NULL), manually selected non-informative MeSH terms are excluded, which can be viewed through data(droplist).
filterMessage	A logical. Under the default TRUE, any output RAV belong to the filtering list will give a message. Silence this message with filterMessage=FALSE. You can check the filter list using data("filterList").

Value

A word cloud with the MeSH terms associated with the given cluster.

Examples

```
data(miniRAVmodel)
drawWordcloud(miniRAVmodel, 1139)
```

---

droplist *MeSH terms to be excluded in drawWordcloud function*

Description

MeSH terms to be excluded in drawWordcloud function

Usage

```
droplist
```

Format

A character vector containing MeSH terms to be excluded.

Author(s)

Sehyun Oh <shbrief@gmail.com>

`extractPC` *PCA on gene expression profile*

Description

Performs a principal components analysis on the given data matrix and returns the results as an object of class `prcomp`.

Usage

```
extractPC(x)
```

Arguments

`x`	a numeric or complex matrix (or data frame) which provides the gene expression data for the principal components analysis. Genes in the rows and samples in the columns.

Value

A `prcomp` object.

See Also

`prcomp`

Examples

```
m = matrix(rnorm(100),ncol=5)
extractPC(m)
```

filterList — *RAVs that will output with quality-control messages*

Description

RAVs that will output with quality-control messages

Usage

```
filterList
```

Format

A named list with four elements - "Cluster_Size_filter", "GSEA_C2_filter", "GSEA_PLIERpriors_filter", and "Redundancy_filter".

Author(s)

Sehyun Oh <shbrief@gmail.com>

findKeywordInRAV — *Find the rank of your keyword in the RAV's GSEA annotation*

Description

Once you provide RAVmodel, keyword you're searching for, and the RAV number to this function, it will give you the abs(NES)-based rank of your keyword in the enriched pathways of the target RAV. If can be useful to find out how uniquely your keyword-containing pathways are represented.

Usage

```
findKeywordInRAV(RAVmodel, keyword, ind, n = NULL, includeTotal = TRUE)
```

Arguments

RAVmodel	PCAGenomicSignatures-object.
keyword	A character vector. If you are searching for multiple keywords at the same time, use paste with collapse="|" argument.
ind	An integer. The RAV number you want to check.
n	An integer. The number of top enriched pathways (based on abs(NES)) to search. Under default (NULL), all the enriched pathways are used.
includeTotal	Under the default condition (TRUE), the total number of enriched pathways will be also printed out as a part of the output.

Value

A character containing the rank of keyword-containing pathways (separated by |), followed by the total number of enriched pathways in parenthesis.

Examples

```
data(miniRAVmodel)
findKeywordInRAV(miniRAVmodel, "Bcell", ind = 695)
```

`findSignature` *Find the RAVs with the keyword-containing enriched pathways*

Description

This function finds RAVs containing the keyword you provide. If you provide "the number of keyword-containing pathways per RAV" in argument `k`, it will give you the RAV number.

Usage

```
findSignature(RAVmodel, keyword, n = 5, k = NULL)
```

Arguments

`RAVmodel`	PCAGenomicSignatures-object
`keyword`	A character vector. If you are searching for multiple keywords at the same time, use `paste` with `collapse="|"` argument.
`n`	The number of top ranked (based on abs(NES)) pathways you want to search your keyword
`k`	The number of keyword-containing pathways you want to get the RAV number. Under default (`NULL`), the output will be a data frame with two columns: '# of keyword-containing pathways' and 'Freq'. If you assign the value for this argument, the output will be an integer vector containing the RAV index.

Value

A data frame or integer vector depending on the parameter `k`.

Examples

```
data(miniRAVmodel)
findSignature(miniRAVmodel, "Bcell")
findSignature(miniRAVmodel, "Bcell", k = 5)
```

findStudiesInCluster *Find the studies contributing each RAV*

Description

Find the studies contributing each RAV

Usage

```
findStudiesInCluster(RAVmodel, ind = NULL, studyTitle = FALSE)
```

Arguments

RAVmodel	PCAGenomicSignatures object.
ind	A numeric vector containing the RAV indexes. Under the default (NULL), studies associated with all the RAV indexes will be returned as a list.
studyTitle	Default is FALSE. This parameter is effective only when the index value is specificed. If it's TRUE, the output will be a data frame with the study

Value

A list of character vectors. Under the default condition (ind = NULL), all the RAVs will be checked for their contributing studies and the length of the list will be same as the number of RAVs (= metadata(x)$k). If you provide the ind argument, studies associated with only the specified RAVs will be returned.

Note

Mainly used for model building, within buildAvgLoading.

Examples

```
data(miniRAVmodel)
findStudiesInCluster(miniRAVmodel, 1076)
```

GenomicSignatures-class *Virtual class inherited from SummarizedExperiment*

Description

GenomicSignatures is a virtual class inherited from SummarizedExperiment and hosts GenomicSignatures models built from different dimensional reduction methods. Currently, PCA-based model, called PCAGenomicSignatures, is available.

Arguments

x	A GenomicSignatures-class object
value	See details.

Details

GenomicSignatures

`GenomicSignatures-methods`
Methods and accesors for `GenomicSignatures` *object*

Description

The default contents of `GenomicSignatures` object, with a set of getter and setter generic functions, which extract either the `assay`, `colData`, or `metadata` slots of a `GenomicSignatures-class` object. When you create this object, `colData$studies` should be populated before adding any information in `trainingData` slot.

Usage

```
## S4 method for signature 'GenomicSignatures'
RAVindex(x)

## S4 method for signature 'GenomicSignatures'
geneSets(x)

## S4 method for signature 'GenomicSignatures'
updateNote(x)

## S4 method for signature 'GenomicSignatures'
version(x)

## S4 replacement method for signature 'GenomicSignatures'
geneSets(x) <- value

## S4 replacement method for signature 'GenomicSignatures'
updateNote(x) <- value
```

Arguments

`x`	A `GenomicSignatures` object
`value`	See details.

Details

- assay(x) : RAVindex (= avgLoadings) containing genes x RAVs
- metadata(x) : Metadata associated with RAVindex building process
- colData(x) : Information on RAVs

Value

A GenomicSignatures object for the constructor

Setters

Setter method values (i.e., `function(x) <- value`):

- metadata<- : Assign metadata
- coldata<- : Assign extra information associated with RAVs
- geneSets<- : A character vector containing the name of gene sets used to annotate average loadings
- updateNote<- : A character vector. Describes the main feature of a model construction

Getters

- RAVindex : Equivalent to `assays(x)$RAVindex`
- geneSets : Access the `metadata(x)$geneSets` slot
- updateNote : Access the `metadata(x)$updateNote` slot
- version : Access the `metadata(x)$version` slot

Examples

```
data(miniRAVmodel)
miniRAVmodel
```

`getModel` *Download a PCAGenomicSignatures model*

Description

Download a PCAGenomicSignatures model

Usage

```
getModel(prior = c("C2", "PLIERpriors"), version = "latest", load = TRUE)
```

Arguments

`prior`	The name of gene sets used to annotate PCAGenomicSignatures. Currently there are two available options. • `C2` : MSigDB C2 (curated gene sets) • `PLIERpriors` : bloodCellMarkersIRISDMAP, svmMarkers, and canonicalPathways
`version`	Default is `latest`. Available versions are listed in `version` column of `availableRAVmodel()` output.
`load`	Default is `TRUE`. If it's set to `FALSE`, the models are just downloaded to cache but not loaded into memory.

Value

File cache location or PCAGenomicSignatures object loaded from it.

Examples

```
z = getModel("C2")
```

`getRAVInfo` *Extract information on a specific RAV*

Description

Extract information on a specific RAV

Usage

```
getRAVInfo(RAVmodel, ind)
```

Arguments

`RAVmodel`	A PCAGenomicSignatures object
`ind`	An index of RAV

Value

A list with four elements: clusterSize, silhouetteWidth, enrichedPathways (the number of enriched pathways), and members. The 'members' is the summary table of PCs in RAV, containing three columns: studyName, PC, and Variance explained (

Examples

```
data(miniRAVmodel)
getRAVInfo(miniRAVmodel, ind = 438)
```

`getStudyInfo` *Extract information on a specific training dataset*

Description

Extract information on a specific training dataset

Usage

```
getStudyInfo(RAVmodel, study)
```

Arguments

`RAVmodel`	A PCAGenomicSignatures object
`study`	A character for SRA study accession.

Value

A list with three elements: studyTitle, studySize (the number of samples from this study used in the RAVmodel building), and RAVs. 'RAVs' is a data frame with three columns - PC (1 to 20), RAV (RAV that the given PC belongs to), and Variance explained (miniRAVmodel, which doesn't have all the PCA summary information, so the example will return only the two PCs of the study instead of all twenty.

Examples

```
data(miniRAVmodel)
getStudyInfo(miniRAVmodel, "SRP028155")
```

`heatmapTable` *Validation result in heatmap format*

Description

This function subsets `validate` outputs with different criteria and visualize it in a heatmap-like table.

Usage

```
heatmapTable(
  val_all,
  RAVmodel,
  ind = NULL,
  num.out = 5,
  scoreCutoff = NULL,
  swCutoff = NULL,
  clsizeCutoff = NULL,
  breaks = c(0, 0.5, 1),
  colors = c("white", "white smoke", "red"),
  column_title = NULL,
  row_title = NULL,
  whichPC = NULL,
  filterMessage = TRUE,
  ...
)
```

Arguments

`val_all`	An output matrix from `validate` function with the parameter `level = "max"`. Subset of this matrix is plotted as a heatmap using `Heatmap`
`RAVmodel`	PCAGenomicSignatures-class object. RAVmodel used to prepare `val_all` input.
`ind`	An integer vector. If this parameter is provided, the other parameters, `num.out`, `scoreCutoff`, `swCutoff`, `clsizeCutoff` will be ignored and the heatmap table containing only the provided index will be printed.

num.out	A number of highly validated RAVs to output. Default is 5. If any of the cutoff parameters are provided, num.out or the number of filtered RAVs, whichever smaller, will be chosen.
scoreCutoff	A numeric value for the minimum correlation (not include). If val_all input is from multiple studies, the default is 0.7 and this is the only cutoff criteria considered: swCutoff and clsizeCutoff will be ignored.
swCutoff	A numeric value for the minimum average silhouette width.
clsizeCutoff	A integer value for the minimum cluster size.
breaks	A numeric vector of length 3. Number represents the values assigned to three colors. Default is c(0, 0.5, 1).
colors	A character vector of length 3. Each represents the color assigned to three breaks. Default is c("white", "white smoke", "red").
column_title	A character string. Provide the column title.
row_title	A character string. Provide the row title.
whichPC	An integer value between 1 and 8. PC number of your data to check the validated signatures with. Under the default (NULL), it outputs top scored signatures with any PC of your data.
filterMessage	A logical. Under the default TRUE, any output RAV belong to the filtering list will give a message. Silence this message with filterMessage=FALSE. You can check the filter list using data("filterList").
...	any additional argument for Heatmap

Value

A heatmap displaying the subset of the validation result that met the given cutoff criteria. If val_all input is from a single dataset, the output heatmap will contain both score and average silhouette width for each cluster.

If val_all input is from multiple studies, the output heatmap's rows will represent each study and the columns will be RAVs, which meet scoreCutoff for any of the input studies.

Examples

```
data(miniRAVmodel)
library(bcellViper)
data(bcellViper)

## Single dataset
val_all <- validate(dset, miniRAVmodel)
heatmapTable(val_all, miniRAVmodel, swCutoff = 0)

## A list of datasets
val_all2 <- validate(miniTCGA, miniRAVmodel)
heatmapTable(val_all2, miniRAVmodel)
```

meshTable *Build a two-column word/frequency table*

Description

Build a two-column word/frequency table

Usage

```
meshTable(
  RAVmodel,
  ind,
  rm.noise = NULL,
  weighted = TRUE,
  filterMessage = TRUE
)
```

Arguments

RAVmodel	A PCAGenomicSignatures object
ind	An index of RAV
rm.noise	An integer. Under the default (rm.noise=NULL), if cluster size (= s) is smaller than 8, rm.noise = floor(s*0.5). For clusters with >= 8 PCs, rm.noise = 4. If rm.noise = 0, all the MeSH terms in RAV will be used to draw wordcloud.
weighted	A logical. If TRUE, MeSH terms from each study are weighted based on the variance explained by the principle component of the study contributing a give RAV. Default is TRUE.
filterMessage	A logical. Under the default TRUE, any output RAV belong to the filtering list will give a message. Silence this message with filterMessage=FALSE. You can check the filter list using data("filterList").

Value

A table with two columns, word and freq. MeSH terms in the defined RAV (by ind argument) is ordered based on their frequency.

Examples

```
data(miniRAVmodel)
meshTable(miniRAVmodel,1139)
```

miniAllZ — *Subset of allZ matrix constructed from 8 CRC training datasets*

Description

Eight colorectal cancer microarray datasets were used to build RAVmodel and the intermediate file containing genes and top PCs from each dataset is named as `allZ`. Hierarchical clustering result of `allZ` is saved as `res_hcut`. For demonstration, we subset the `allZ` matrix with the first 100 genes, which is named as `miniAllZ`.

Usage

```
miniAllZ
```

Format

A matrix with 100 genes and 160 PCs from 8 training datasets.

Author(s)

Sehyun Oh <shbrief@gmail.com>

Source

https://github.com/shbrief/model_building/tree/main/RAVmodel_8CRC

miniRAVmodel — *RAVmodel from 536 studies, annotated with MSigDB C2*

Description

A object providing a miniature version of RAVmodel_C2 (PCAGenomicSignatures object constructed from 536 studies and annotated with MSigDB C2).

Usage

```
miniRAVmodel
```

Format

PCAGenomicSignatures

Author(s)

Sehyun Oh <shbrief@gmail.com>

miniTCGA *Subset of TCGA-COAD and TCGA-BRCA RNA sequencing datasets*

Description

TCGA-COAD and TCGA-BRCA RNA sequencing data were acquired using `GSEABenchmarkeR::loadEData` and log-transformed. Conversion from EntrezID to gene symbol was done with `EnrichmentBrowser::idMap`. Only 8 samples from each dataset are kept.

Usage

```
miniTCGA
```

Format

A list containing two SummarizedExperiment objects.

Author(s)

Sehyun Oh <shbrief@gmail.com>

PCAGenomicSignatures *Construct* `PCAGenomicSignatures` *object*

Description

The default contents of `PCAGenomicSignatures` object, with a set of accessors and setter generic functions, which extract either the `assay`, `colData`, `metadata`, or `trainingData` slots of a `PCAGenomicSignatures-cla` object. When you create this object, `colData$studies` should be populated before adding any information in `trainingData` slot

Usage

```
PCAGenomicSignatures(..., trainingData)
```

Arguments

`...`	Additional arguments for supporting functions.
`trainingData`	A `DataFrame` class object for metadata associated with training data

Details

- RAVindex(x) : RAVindex (= avgLoadings) containing genes x RAVs
- metadata(x)$cluster : A vector of integers (from 1:k) indicating the cluster to which each point is allocated.
- metadata(x)$size : The number of PCs in each cluster.
- metadata(x)$k : The number of RAVs.
- metadata(x)$n : The number of top PCs from each dataset.

- metadata(x)$geneSets : Name of the prior gene sets used to annotate average loadings.
- colData(x)$studies : A list of character vectors containing studies contributing to each PC cluster.
- colData(x)$silhouetteWidth : A numeric array of average silhouette widths of each clusters
- colData(x)$gsea : A list of data frames. Each element is a subset of outputs from `clusterProfiler::GSEA` function.

Value

PCAGenomicSignatures object with multiple setters or accessors

Slots

`trainingData` A `DataFrame` class object for metadata associated with training data

Setters

Setter method values (i.e., `function(x) <- value`):

- geneSets<- : A character vector containing the name of gene sets used to annotate average loadings
- studies<- : A list of character vectors containing gene sets used to annotate average loadings
- gsea<- : A list of data frames. Each element is a subset of output from `gseaResult` objects.
- metadata<- : A `list` object of metadata
- '$<-' : A vector to replace the indicated column in `colData`

Accessors

All the accessors inherited from `SummarizedExperiment` are available and the additional accessors for `PCAGenomicSignatures` specific data are listed below.

- RAVindex : Equivalent to the `assay(x)`
- geneSets : Access the `metadata(x)$geneSets` slot
- studies : Access the `colData(x)$studies` slot
- gsea : Access the `colData(x)$gsea`
- '$' : Access a column in `colData`
- trainingData : Access the `trainingData` slot
- mesh : Access the `trainingData(x)$MeSH` slot
- PCAsummary : Access the `trainingData(x)$PCAsummary` slot

Examples

```
data(miniRAVmodel)
miniRAVmodel
```

PCAGenomicSignatures-class

PCAGenomicSignatures-class

Description

PCA-based GenomicSignatures-class.

Arguments

x	A GenomicSignatures-class object
value	See details.

Details

PCAGenomicSignatures

Slots

trainingData A DataFrame class object for metadata associated with training data

Examples

```
data(miniRAVmodel)
miniRAVmodel
```

PCAGenomicSignatures-methods

Methods and accesors for PCAGenomicSignatures *object*

Description

The default contents of PCAGenomicSignatures object, with a set of accessor and setter generic functions, which extract either the assay, colData, metadata, or trainingData slots of a PCAGenomicSignatures-cla object. When you create this object, colData$studies should be populated before adding any information in trainingData slot

Usage

```
## S4 replacement method for signature 'PCAGenomicSignatures'
studies(x) <- value

## S4 replacement method for signature 'PCAGenomicSignatures'
silhouetteWidth(x) <- value

## S4 replacement method for signature 'PCAGenomicSignatures'
gsea(x) <- value

## S4 replacement method for signature 'PCAGenomicSignatures'
```

```
trainingData(x) <- value

## S4 replacement method for signature 'PCAGenomicSignatures'
mesh(x) <- value

## S4 replacement method for signature 'PCAGenomicSignatures'
PCAsummary(x) <- value

## S4 method for signature 'PCAGenomicSignatures'
studies(x)

## S4 method for signature 'PCAGenomicSignatures'
silhouetteWidth(x)

## S4 method for signature 'PCAGenomicSignatures'
gsea(x)

## S4 method for signature 'PCAGenomicSignatures'
trainingData(x)

## S4 method for signature 'PCAGenomicSignatures'
mesh(x)

## S4 method for signature 'PCAGenomicSignatures'
PCAsummary(x)

## S4 method for signature 'PCAGenomicSignatures'
show(object)
```

Arguments

value	See details.
object, x	A PCAGenomicSignatures object

Details

- RAVindex(x) : RAVindex (= avgLoadings) containing genes x RAVs
- metadata(x)$cluster : A vector of integers (from 1:k) indicating the cluster to which each PC is allocated.
- metadata(x)$size : The number of PCs in each cluster.
- metadata(x)$k : The number of RAVs.
- metadata(x)$n : The number of top PCs from each dataset.
- metadata(x)$geneSets : Name of the prior gene sets used to annotate average loadings.
- colData(x)$studies : A list of character vectors containing studies contributing to each PC cluster.
- colData(x)$gsea : A list of data frames. Each element is a subset of outputs from `clusterProfiler::GSEA` function.

Value

PCAGenomicSignatures object with multiple setters or accessors

Slots

`trainingData` A `DataFrame` class object for metadata associated with training data

Setters

Setter method values (i.e., `function(x) <- value`):

- geneSets<- : A character vector containing the name of gene sets used to annotate average loadings
- studies<- : A list of character vectors containing gene sets used to annotate average loadings
- gsea<- : A list of `gseaResult` objects.
- metadata<- : A `list` object of metadata
- '$<-' : A vector to replace the indicated column in `colData`

Accessors

All the accessors inherited from `SummarizedExperiment` are available and the additional accessors for `PCAGenomicSignatures` specific data are listed below.

- RAVindex : Equivalent to the `assay(x)`
- geneSets : Access the `metadata(x)$geneSets` slot
- studies : Access the `colData(x)$studies` slot
- gsea : Access the `colData(x)$gsea`
- '$' : Access a column in `colData`
- trainingData : Access the `trainingData` slot
- mesh : Access the `trainingData(x)$MeSH` slot
- PCAsummary : Access the `trainingData(x)$PCAsummary` slot

Examples

```
data(miniRAVmodel)
miniRAVmodel
```

PCinRAV *Extract the list of PCs in a cluster*

Description

A RAV model contain clusters of PCs from individual studies. This function extracts the names of the original PCs from the RAV model given the index in the RAV model.

Usage

```
PCinRAV(RAVmodel, ind)
```

Arguments

`RAVmodel`	A PCAGenomicSignatures object
`ind`	An index of RAV

Value

A character vector of PC/study names

Examples

```
data(miniRAVmodel)
PCinRAV(miniRAVmodel,695)
```

`plotAnnotatedPCA` *Two-dimensional PCA plot with the PC annotation*

Description

Two-dimensional PCA plot with the PC annotation

Usage

```
plotAnnotatedPCA(
  dataset,
  RAVmodel,
  PCnum,
  val_all = NULL,
  scoreCutoff = 0.5,
  nesCutoff = NULL,
  color_by = NULL,
  color_lab = NULL,
  trimed_pathway_len = 45
)
```

Arguments

`dataset`	A gene expression profile to be validated. Different classes of objects can be used including ExpressionSet, SummarizedExperiment, RangedSummarizedExperiment, or matrix. Rownames (genes) should be in symbol format. If it is a matrix, genes should be in rows and samples in columns.
`RAVmodel`	PCAGenomicSignatures-class object
`PCnum`	A numeric vector length of 2. The values should be between 1 and 8.
`val_all`	The output from `validate`
`scoreCutoff`	A numeric value for the minimum correlation. Default 0.5.
`nesCutoff`	A numeric value for the minimum NES. Default is `NULL` and the suggested value is 3.
`color_by`	A named vector with the feature you want to color by. Name should be match with the sample names of the dataset.
`color_lab`	A name for color legend. If this argument is not provided, the color legend will be labeled as "Color By" by default.
`trimed_pathway_len`	Positive inter values, which is the display width of pathway names. Default is 45.

Value

Scatter plot and the table with annotation. If enriched pathway didn't pass the `scoreCutoff` the table will be labeled as "No significant pathways". If any enriched pathway didn't pass the `nesCutoff`, it will labeled as NA.

Examples

```
data(miniRAVmodel)
library(bcellViper)
data(bcellViper)
## Not run:
plotAnnotatedPCA(exprs(dset), miniRAVmodel, PCnum = c(1,2))

## End(Not run)
```

`plotValidate` *Plot validation results in an interactive graph*

Description

There are three main information on the graph:

- x-axis : Pearson correlation coefficient. Higher value means that test dataset and RAV is more tightly associated with.
- y-axis : Silhouette width representing the quality of RAVs.
- size : The number of studies in each RAV. (= cluster size)
- color : Test dataset's PC number that validate each RAV. Because we used top 8 PCs of the test dataset, there are 8 categories.

Usage

```
plotValidate(
  val_all,
  minClusterSize = 2,
  swFilter = FALSE,
  minSilhouetteWidth = 0,
  interactive = FALSE,
  minClSize = NULL,
  maxClSize = NULL,
  colorPalette = "Dark2"
)
```

Arguments

`val_all`	Output from validate function.
`minClusterSize`	The minimum size of clusters to be included in the plotting. Default value is 2, so any single-element clusters are excluded.
`swFilter`	If `swFilter=TRUE`, only RAV above the cutoff, defined through `minSilhouetteWidth` argument will be plotted. Default is `swFilter=FALSE`

`minSilhouetteWidth`
: A minimum average silhouette width to be plotted. Only effective under `swFilter=TRUE` condition. Default is 0.

`interactive`
: If set to `TRUE`, the output will be interactive plot. Default is `FALSE`.

`minClSize`
: The minimum number of PCs in the clusters you want.

`maxClSize`
: The maximum number of PCs in the clusters you want.

`colorPalette`
: Default is `Dark2`. For other color options, please check `scale_color_brewer`

Value

a ggplot object

Examples

```
data(miniRAVmodel)
library(bcellViper)
data(bcellViper)
val_all <- validate(dset, miniRAVmodel)
plotValidate(val_all)
```

---

res_hcut *Subset of allZ matrix constructed from 8 CRC training datasets*

Description

Eight colorectal cancer microarray datasets were used to build RAVmodel and the intermediate file containing genes and top PCs from each dataset is named as `allZ`. Hierarchical clustering result of `allZ` is saved as `res_hcut`.

Usage

```
res_hcut
```

Format

`hclust` object from `factoextra::hcut` function.

Author(s)

Sehyun Oh <shbrief@gmail.com>

rmNaInf *Remove rows with missing and Inf values from a matrix*

Description

Remove rows with missing and Inf values from a matrix

Usage

```
rmNaInf(x)
```

Arguments

x	A numeric matrix.

Value

The updated input matrix where rows with NA and Inf values are removed.

Examples

```
m = matrix(rnorm(100),ncol=10)
m[1,1] = NA

m1 = rmNaInf(m)
dim(m1)
```

sampleScoreHeatmap *Plot heatmap of the sample scores*

Description

Plot heatmap of the sample scores

Usage

```
sampleScoreHeatmap(
  score,
  dataName,
  modelName,
  cluster_rows = TRUE,
  cluster_columns = TRUE,
  show_row_names = TRUE,
  show_column_names = TRUE,
  row_names_gp = 0.7,
  column_names_gp = 5,
  ...
)
```

Arguments

score	An output from calculateScore function, which is a matrix with samples (row) and PrcompClusters (column) If it is a simple vector, it will be converted to a one-column matrix.
dataName	Title on the row. The name of the dataset to be scored.
modelName	Title on the column. The RAVmodel used for scoring.
cluster_rows	A logical. Under the default (TRUE), rows will be clustered.
cluster_columns	A logical. Under the default (TRUE), columns will be clustered.
show_row_names	Whether show row names. Default is TRUE, showing the row name.
show_column_names	Whether show column names. Default is TRUE, showing the column name.
row_names_gp	Graphic parameters for row names. The default is 0.7.
column_names_gp	Graphic parameters for column names. The default is 5.
...	Any additional argument for Heatmap

Value

A heatmap of the sample score. Rows represent samples and columns represent RAVs.

Examples

```
data(miniRAVmodel)
library(bcellViper)
data(bcellViper)
score <- calculateScore(dset, miniRAVmodel)
sampleScoreHeatmap(score, dataName="bcellViper", modelName="miniRAVmodel")
```

---

subsetEnrichedPathways

Subset enriched pathways of RAV

Description

Subset enriched pathways of RAV

Usage

```
subsetEnrichedPathways(
  RAVmodel,
  ind = NULL,
  n = 10,
  both = FALSE,
  include_nes = FALSE
)
```

Arguments

RAVmodel	PCAGenomicSignatures object. Also an output from GSEA can be used.
ind	A numeric vector containing the RAV number you want to check enriched pathways. If not specified, this function returns results from all the RAVs.
n	The number of top and bottom pathways to be selected based on normalized enrichment score (NES).
both	Default is FALSE, where only the top n pathways will be printed. If it is set to TRUE, the output will contain both top and bottom n pathways.
include_nes	Defalt is FALSE. If it set to TRUE, the output will include both description and NES of the enriched pathway.

Value

A DataFrame with top and bottom n pathways from the enrichment results.

Examples

```
data(miniRAVmodel)

# all RAVS in model
subsetEnrichedPathways(miniRAVmodel,n=5)

# only a specific RAV (note the colnames above)
subsetEnrichedPathways(miniRAVmodel,ind=695,n=5)
```

---

validate *Validate new datasets*

Description

Validate new datasets

Usage

```
validate(
  dataset,
  RAVmodel,
  method = "pearson",
  maxFrom = "PC",
  level = "max",
  scale = FALSE
)
```

Arguments

dataset	Single or a named list of SummarizedExperiment (RangedSummarizedExperiment, ExpressionSet or matrix) object(s). Gene names should be in 'symbol' format. Currently, each dataset should have at least 8 samples.
RAVmodel	PCAGenomicSignatures object.
method	A character string indicating which correlation coefficient is to be computed. One of "pearson" (default), "kendall", or "spearman": can be abbreviated.
maxFrom	Select whether to display the maximum value from dataset's PCs or avgLoadings. Under the default (maxFrom="PC"), the maximum correlation coefficient from top 8 PCs for each avgLoading will be selected as an output. If you choose (maxFrom="avgLoading"), the avgLoading with the maximum correlation coefficient with each PC will be in the output.
level	Output format of validated result. Two options are available: c("max", "all"). Default is "max", which outputs the matrix containing only the maximum coefficient. To get the coefficient of all 8 PCs, set this argument as "all". level = "all" can be used only for one dataset.
scale	Default is FALSE. If it is set to TRUE, dataset will be row normalized.

Value

A data frame containing the maximum pearson correlation coefficient between the top 8 PCs of the dataset and pre-calculated average loadings (in row) of training datasets (score column). It also contains other metadata associated with each RAV: PC for one of the top 8 PCs of the dataset that results in the given score, sw for the average silhouette width of the RAV, cl_size for the size of each RAV.

If the input for dataset argument is a list of different datasets, each row of the output represents a new dataset for test, and each column represents clusters from training datasets. If level = "all", a list containing the matrices of the pearson correlation coefficient between all top 8 PCs of the datasets and avgLoading.

Examples

```
data(miniRAVmodel)
library(bcellViper)
data(bcellViper)
validate(dset, miniRAVmodel)
validate(dset, miniRAVmodel, maxFrom = "avgLoading")
```

---

validatedSignatures *Validation result in data frame*

Description

Validation result in data frame

Usage

```
validatedSignatures(
  val_all,
  RAVmodel,
  num.out = 5,
  scoreCutoff = NULL,
  swCutoff = NULL,
  clsizeCutoff = NULL,
  indexOnly = FALSE,
  whichPC = NULL,
  filterMessage = TRUE
)
```

Arguments

val_all	An output matrix from validate function. If this input is from multiple datasets, only scoreCutoff argument will be considered and other inputs will be ignored.
RAVmodel	PCAGenomicSignatures-class object. RAVmodel used to prepare val_all input.
num.out	A number of highly validated RAVs to output. Default is 5. If any of the cutoff parameters are provided, num.out or the number of filtered RAVs, whichever smaller, will be chosen.
scoreCutoff	A numeric value for the minimum correlation. For multi-studies case, the default is 0.7.
swCutoff	A numeric value for the minimum average silhouette width.
clsizeCutoff	An integer value for the minimum cluster size.
indexOnly	A logical. Under the default (= FALSE), the detailed information on validated RAVs, such as score, average silhouette width, cluster size, is printed. If it is set TRUE, only the RAV number will be printed.
whichPC	An integer value between 1 and 8. PC number of your data to check the validated signatures with. Under the default (NULL), it outputs top scored signatures with any PC of your data.
filterMessage	A logical. Under the default TRUE, any output RAV belong to the filtering list will give a message. Silence this message with filterMessage=FALSE. You can check the filter list using data("filterList").

Value

A subset of the input matrix, which meets the given condition.

Examples

```
data(miniRAVmodel)
library(bcellViper)
data(bcellViper)
val_all <- validate(dset, miniRAVmodel)
validatedSignatures(val_all, miniRAVmodel, num.out = 3, scoreCutoff = 0)
```

Index