Package 'projectR'

May 8, 2024

Type Package

Title Functions for the projection of weights from PCA, CoGAPS, NMF, correlation, and clustering

Version 1.20.0

Author Gaurav Sharma, Charles Shin, Jared Slosberg, Loyal Goff, Genevieve Stein-O'Brien

Description Functions for the projection of data into the spaces defined by PCA, CoGAPS, NMF, correlation, and clustering.

License GPL (==2)

Imports methods, cluster, stats, limma, NMF, ROCR, ggalluvial, RColorBrewer, dplyr, fgsea, reshape2, viridis, scales, Matrix, MatrixModels, msigdbr, ggplot2, cowplot, ggrepel, umap, tsne

Suggests BiocStyle, CoGAPS, gridExtra, grid, testthat, devtools, knitr, rmarkdown, ComplexHeatmap, gplots, SeuratObject

LazyData TRUE

LazyDataCompression gzip

RoxygenNote 7.3.1

Encoding UTF-8

VignetteBuilder knitr

biocViews FunctionalPrediction, GeneRegulation, BiologicalQuestion, Software

BugReports https://support.bioconductor.org/t/projectR/

URL https://github.com/genesofeve/projectR/

Depends R (>= 4.0.0)

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

git_branch RELEASE_3_19

git_last_commit d729872

git_last_commit_date 2024-04-30

Repository Bioconductor 3.19

Date/Publication 2024-05-07

Maintainer Genevieve Stein-O'Brien <gsteinobrien@gmail.com>

Contents

alluvialMat *alluvialMat*

Description

Function to provide alluvial matrix for generating alluvial plot

Usage

```
alluvialMat(
  projection,
  annotations,
  annotationName = "Cell type",
  annotationType = "Cell",
  plot = TRUE,
  minPropExplained = 0.75,
  pvalThreshold = 0.05,
  qvalThreshold = 0.05
)
```

Arguments

projection	a projection generated from projectR, ensure that full = TRUE while generating projection
annotations	a character vector of annotations for the data
annotationName	a charcter fof collective name of the annotations, default is "Cell type"
annotationType	a character indicating the type of data annotated, default is "Cell"
plot	logical indicating whether to return the alluvial plot, default is TRUE
minPropExplained	threshold for minimum proportion of samples that correspond to a pattern to be used for plotting
pvalThreshold	theshold level of significance for p-value
qvalThreshold	theshold level of significance for Benjamini-Hochberg corrected p-value

Value

A matrix to generate alluvial plots

Examples

```
projection <- projectR(data=p.ESepiGen4c1l$mRNA.Seq,loadings=AP.RNAseq6l3c3t$Amean,
dataNames = map.ESepiGen4c1l[["GeneSymbols"]], full = TRUE)
alluvialMat(projection,pd.ESepiGen4c1l$Condition)
```

AP.RNAseq6l3c3t *CoGAPS patterns and genes weights for p.RNAseq6l3c3t*

Description

AP.RNAseq6l3c3t contains the output of the gapsRun function in the CoGAPS package for data = p.RNAseq6l3c3t

Usage

```
AP.RNAseq6l3c3t
```

Format

A list of 12 items

aucMat *aucMat*

Description

Calculates AUC values for each set of weights for each label and outputs the results as a matrix

Usage

```
aucMat(labels, weights)
```

Arguments

labels	a vector of labels whose length is equal to the number of columns in the weight matrix
weights	a matrix of weights from projection analysis

Value

A matrix of AUC values for each set of weights classifying each label.

Examples

```
projectR(data=p.ESepiGen4c1l$mRNA.Seq,loadings=AP.RNAseq6l3c3t$Amean,
dataNames = map.ESepiGen4c1l[["GeneSymbols"]]) -> projection
aucMat(pd.ESepiGen4c1l$Condition,projection)
```

bonferroniCorrectedDifferences *bonferroniCorrectedDifferences*

Description

Calculate weighted/unweighted mean difference for each gene between 2 groups

Usage

```
bonferroniCorrectedDifferences(
  group1,
  group2,
  pvalue,
  diff_weights = NULL,
  mode = "CI"
)
```

Arguments

group1	count matrix 1
group2	count matrix 2
pvalue	significance value to threshold
diff_weights	loadings to weight the differential expression
mode	statistical approach, confidence intervals(CI) or pvalues(PV)

cluster2pattern *Generic cluster2pattern function*

Description

Function to make patterns of continuous weights from clusters.

Usage

```
cluster2pattern(clusters, NP, data, ...)

## S4 method for signature 'character'
cluster2pattern(clusters, data)

## S4 method for signature 'numeric'
cluster2pattern(clusters, data)

## S4 method for signature 'kmeans'
```

```
cluster2pattern(clusters, data)

## S4 method for signature 'hclust'
cluster2pattern(clusters, NP, data = NA)
```

Arguments

clusters	a cluster object which could be either an hclust or a kmeans object
NP	number of desired patterns
data	data used to make clusters object
...	Additional arguments to cluster2pattern

Value

An object of class pclust containing pattern weights corresponding for each cluster.

Examples

```
k.RNAseq6l3c3t<-kmeans(t(p.RNAseq6l3c3t),3)
cluster2pattern(clusters=k.RNAseq6l3c3t,data=p.RNAseq6l3c3t)

distp <- dist(t(p.RNAseq6l3c3t))
hc.RNAseq6l3c3t <- hclust(distp)
cluster2pattern(clusters=hc.RNAseq6l3c3t,NP=3,data=p.RNAseq6l3c3t)
```

---

cluster2pattern-class *cluster2pattern*

Description

class of cluster2pattern output.

Slots

clusterMatrix matrix of continous values for projection that is output of cluster2pattern function

clusterPlotR *Generic clusterPlotR function*

Description

plotting function for clustering objects

Usage

```
clusterPlotR(cData, cls, x, NC, ...)

## S4 method for signature 'ANY,kmeans'
clusterPlotR(
  cData = NA,
  cls = NA,
  x = NA,
  NC = NA,
  annoIndx = NA,
  label = NULL,
  ...
)

## S4 method for signature 'ANY,hclust'
clusterPlotR(
  cData = NA,
  cls = NA,
  x = NA,
  NC = NA,
  annoIndx = NA,
  label = NULL,
  ...
)
```

Arguments

cData	data used to get clusters
cls	a cluster (kmeans or hclust) object
x	a vector of length equal to number of samples to use for plotting
NC	vector of integers indicating which clusters to use
...	additional parameters for plotting. ex. pch,cex,col,labels, xlab, etc.
annoIndx	vector indexing into subsets for plotting
label	character vector to use for plotting text, defaults is NULL

Value

A plot of the mean behavior for each cluster

Examples

```
## Not run:
 k.RNAseq6l3c3t<-kmeans(p.RNAseq6l3c3t,22)
 clusterPlotR(p.RNAseq6l3c3t, cls=k.RNAseq6l3c3t,NC=1,x=pd.RNAseq6l3c3t$days,
col=pd.RNAseq6l3c3t$color)

## End(Not run)
```

`correlateR` *correlateR*

Description

Function to extract genes highly correlated with a gene or reference expression pattern.

Usage

```
correlateR(genes, dat, threshtype = "R", threshold = 0.7, absR = FALSE, ...)
```

Arguments

`genes`	gene or character vector of genes for reference expression pattern
`dat`	matrix or data frame with genes to be used for to calculate correlation
`threshtype`	Default "R" indicates thresholding by R value or equivalent. Alternatively, "N" indicates a numerical cut off.
`threshold`	numeric indicating value at which to make threshold.
`absR`	logical indicating where to include both positive and negatively correlated genes
`...`	addtion inputs to cor, such as method

Details

If threshtype is "R" than threshold must be between -1 and 1. Otherwise if top N correlated genes are required, set `threshtype` as "N" and set `threshold` = N, i.e, the number of correlated genes required.

Value

A correlation matrix

Examples

```
cor2T<-correlateR(genes="T", dat=p.RNAseq6l3c3t, threshtype="N", threshold=10, absR=TRUE)
```

correlateR-class *correlateR*

Description

class of correlateR output.

Slots

`corM` correlation matrix obtained from correlateR

CR.RNAseq6l3c3t *CogapsResult object for p.RNAseq6l3c3t*

Description

CR.RNAseq6l3c3t contains the output of the CoGAPS function in the CoGAPS package for data = p.RNAseq6l3c3t

Usage

```
CR.RNAseq6l3c3t
```

Format

A CogapsResult object

cr_microglial *CogapsResult object for microglial_counts*

Description

cr_microglia contains the output of the CoGAPS function in the CoGAPS package for data = microglial_counts

Usage

```
cr_microglial
```

Format

A CogapsResult object

geneMatchR *Generic geneMatchR function*

Description

Matches genes accross datasets

Usage

```
geneMatchR(
  data1,
  data2,
  data1Names = NULL,
  data2Names = NULL,
  merge = FALSE,
  ...
)
```

Arguments

data1	a data matrix, typically genes by samples
data2	an amplitude matrix, typically genes by factors
data1Names	rownames of data matrix, for eg genenames
data2Names	rownames of amplitude matrix to be matched to rownames of datamatrix
merge	logical indicating wether or not to merged data sets
...	Additional arguments to geneMatchR

Value

A list of genes (intersection) in both datasets. (if merge = TRUE, Also returns merged data.)

Examples

```
geneMatchR(data1=p.ESepiGen4c1l$mRNA.Seq,data2=p.RNAseq6l3c3t,
data1Names=map.ESepiGen4c1l[["GeneSymbols"]])
```

getTSNE *getTSNE*

Description

Function to provide tSNE of projection

Usage

```
getTSNE(projection, axis = 2, ...)
```

Arguments

projection	martrix, a projection generated from projectR
axis	integer, either 1 umap of projection or 2 for umap of transpose of projection
...	addtional arguments passed to tsne

Examples

```
projection <- projectR(data=p.ESepiGen4c1l$mRNA.Seq,loadings=AP.RNAseq6l3c3t$Amean,
dataNames = map.ESepiGen4c1l[["GeneSymbols"]], full = TRUE)
projectionTSNE <- getTSNE(projection)
```

getUMAP *getUMAP*

Description

Function to provide umap of projection

Usage

```
getUMAP(projection, axis = 2, umapMethod = "naive", umapConfig = umap.defaults)
```

Arguments

projection	martrix, a projection generated from projectR
axis	integer, either 1 umap of projection or 2 for umap of transpose of projection
umapMethod	character, implementation. Available methods are 'naive' (an implementation written in pure R) and 'umap-learn' (requires python package 'umap-learn')
umapConfig	umap.config, a list of parameters to customize umap embedding

Value

A umap of projection

Examples

```
library(umap)
projection <- projectR(data=p.ESepiGen4c1l$mRNA.Seq,loadings=AP.RNAseq6l3c3t$Amean,
dataNames = map.ESepiGen4c1l[["GeneSymbols"]], full = TRUE)
umapConfig = umap.defaults
umapConfig$n_neighbors = 3
projectionUMAP <- getUMAP(projection,umapConfig = umapConfig)
```

---

glial_counts *log-normalized count data from astrocytes and oligodendrocytes in the p6 mouse cortex.*

Description

log-normalized count data from astrocytes and oligodendrocytes in the p6 mouse cortex.

Usage

```
glial_counts
```

Format

A gene (rows) by cell (column) matrix

---

initialize,cluster2pattern-method *Constructor for cluster2pattern*

Description

Constructor for cluster2pattern

Usage

```
## S4 method for signature 'cluster2pattern'
initialize(.Object, clusterMatrix, ...)
```

Arguments

.Object	clusterMatrix object
clusterMatrix	matrix of continous values for projection that is output of cluster2pattern function
...	additional arguments to intialize cluster2pattern

Value

initialized cluster2pattern object

initialize,correlateR-method *Constructor for correlateR*

Description

Constructor for correlateR

Usage

```
## S4 method for signature 'correlateR'
initialize(.Object, corM, ...)
```

Arguments

.Object	correlateR object
corM	correlation matrix obtained from correlateR
...	additional arguments to intialize correlateR

Value

initialized correlateR object

initialize,rotatoR-method *Constructor for rotatoR*

Description

Constructor for rotatoR

Usage

```
## S4 method for signature 'rotatoR'
initialize(.Object, rotatedM, ...)
```

Arguments

.Object	rotatoR object
rotatedM	rotated matrix from rotatoR function
...	additional arguments to intialize rotatoR

Value

initialized rotatoR object

`intersectoR` *Generic intersectoR function*

Description

A function to find and test the intersecting values of two sets of objects, presumably the genes associated with patterns in two different datasets. Both the input objects need to be of the same type either kmeans or hclust.

Usage

```
intersectoR(pSet1, pSet2, pval, ...)

## S4 method for signature 'kmeans,kmeans'
intersectoR(pSet1 = NA, pSet2 = NA, pval = 0.05, full = FALSE)

## S4 method for signature 'hclust,hclust'
intersectoR(pSet1 = NA, pSet2 = NA, pval = 0.05, full = FALSE, k = NULL)
```

Arguments

`pSet1`	an object for a set of patterns where each entry is a set of genes associated with a single pattern
`pSet2`	an object for a second set of patterns where each entry is a set of genes associated with a single pattern
`pval`	the maximum p-value considered significant
`...`	additional parameters depending on input object
`full`	logical indicating whether to return full data frame of signigicantly overlapping sets. Default is false will return summary matrix.
`k`	Numeric giving cut height for hclust objects, if a vector is given arguments will be applied to pSet1 and pSet2 in that order

Value

A list containing: Overlap matrix, overlap index, and overlapping sets.

Examples

```
ESepiGen4c1lmRNASeq <- p.ESepiGen4c1l$mRNA.Seq
rownames(ESepiGen4c1lmRNASeq) <- map.ESepiGen4c1l$GeneSymbols

k.RNAseq6l3c3t<-kmeans(p.RNAseq6l3c3t,22)
k.ESepiGen4c1l<-kmeans(ESepiGen4c1lmRNASeq,10)
intersectoR(k.RNAseq6l3c3t, k.ESepiGen4c1l, pval=.05)

h.RNAseq6l3c3t<-hclust(as.dist(1-(cor(t(p.RNAseq6l3c3t)))))
```

```
h.ESepiGen4c1l<-hclust(as.dist(1-(cor(t(ESepiGen4c1lmRNASeq)))))
intersectoR(pSet1=h.ESepiGen4c1l, pSet2=h.RNAseq6l3c3t, pval=.05, k=c(3,4))
```

`map.ESepiGen4c1l` *RNAseqing and ChIPSeq of matched genes in differentiated human iPS cells*

Description

map.ESepiGen4c1l contains gene annotations

Usage

```
map.ESepiGen4c1l
```

Format

A data frames with 93 rows and 9 variables:

References

1. Gifford, C. A. et al. Transcriptional and epigenetic dynamics during specification of human embryonic stem cells. Cell 153, 1149-1163 (2013).

`map.RNAseq6l3c3t` *RNAseqing from human 3 iPS & 3 ES cell lines in 3 experimental condition at 3 time points*

Description

map.RNAseq6l3c3 contains gene annotations for polyA bulk sequencing of 6 cell lines in 3 experimental condition at 3 time points.

Usage

```
map.RNAseq6l3c3t
```

Format

A data frames with 108 rows and 54 variables:

microglial_counts *log-normalized count data from microglial cells in the p6 mouse cortex.*

Description

log-normalized count data from microglial cells in the p6 mouse cortex.

Usage

```
microglial_counts
```

Format

A gene (rows) by cell (column) matrix

multivariateAnalysisR *Generic multivariateAnalysisR function*

Description

Performs multivariate analysis across specified clusters in datasets

Usage

```
multivariateAnalysisR(
  significanceLevel = 0.05,
  patternKeys,
  seuratobj,
  dictionaries,
  customNames = NULL,
  exclusive = TRUE,
  exportFolder = "",
  ANOVAwidth = 1000,
  ANOVAheight = 1000,
  CIwidth = 1000,
  CIheight = 1000,
  CIspacing = 1
)
```

Arguments

significanceLevel	double value for testing significance in ANOVA test
patternKeys	list of strings indicating pattern subsets from seuratobj to be analyzed
seuratobj	Seurat Object Data containing patternKeys in meta.data
dictionaries	list of dictionaries indicating clusters to be compared
customNames	list of custom names for clusters in corresponding order
exclusive	boolean value for determining interpolation between params in clusters
exportFolder	name of folder to store exported graphs and CSV files
ANOVAwidth	width of ANOVA png
ANOVAheight	height of ANOVA png
CIwidth	width of CI png
CIheight	height of CI png
CIspacing	spacing between each CI in CI graph

Value

a sorted list of ANOVA and CI results; ANOVA and Confidence Intervals are visualized and exported in both PNG and CSV

---

multivariateAnalysisR_seurat_test

Truncated Seurat Object with latent space projection done to unspecified cells in different stages for multivariateAnalysisR analysis

---

Description

Truncated Seurat Object with latent space projection done to unspecified cells in different stages for multivariateAnalysisR analysis

Usage

```
multivariateAnalysisR_seurat_test
```

Format

A Seurat Object with 31034 observations of 4 variables in meta.data:

`p.ESepiGen4c1l` *RNAseqing and ChIPSeq of matched genes in differentiated human iPS cells*

Description

p.ESepiGen4c1l contains log2(RPKM + 1) values for polyA bulk sequencing and log2 counts of normalized ChIPSeq reads of 1 cell lines with 2 replicates in 4 experimental conditions at a single time point.

Usage

```
p.ESepiGen4c1l
```

Format

p.ESepiGen4c1l is a list of 6 data frames each with with 93 rows and between 4 and 9 variables:

References

1. Gifford, C. A. et al. Transcriptional and epigenetic dynamics during specification of human embryonic stem cells. Cell 153, 1149-1163 (2013).

`p.RNAseq6l3c3t` *RNAseqing from human 3 iPS & 3 ES cell lines in 3 experimental condition at 3 time points*

Description

p.RNAseq6l3c3 contains log2(RPKM + 1) values for polyA bulk sequencing of 6 cell lines in 3 experimental condition at 3 time points.

Usage

```
p.RNAseq6l3c3t
```

Format

A data frames with 108 rows and 54 variables:

pd.ESepiGen4c1l — *RNAseqing and ChIPSeq of matched genes in differentiated human iPS cells*

Description

pd.ESepiGen4c1l.4cond contains sample phenotype and experimental information

Usage

```
pd.ESepiGen4c1l
```

Format

A data frames with 9 rows and 2 variables:

References

1. Gifford, C. A. et al. Transcriptional and epigenetic dynamics during specification of human embryonic stem cells. Cell 153, 1149-1163 (2013).

pd.RNAseq6l3c3t — *RNAseqing from human 3 iPS & 3 ES cell lines in 3 experimental condition at 3 time points*

Description

pd.RNAseq6l3c3t contains sample phenotype and experimental information for polyA bulk sequencing of 6 cell lines in 3 experimental condition at 3 time points.

Usage

```
pd.RNAseq6l3c3t
```

Format

A data frames with 54 rows and 38 variables:

pdVolcano *pdVolcano*

Description

Generate volcano plot and gate genes based on fold change and pvalue, includes vectors that can be used with fast gene set enrichment (fgsea)

Usage

```
pdVolcano(
  result,
  FC = 0.2,
  pvalue = NULL,
  subset = NULL,
  filter.inf = FALSE,
  label.num = 5L,
  display = TRUE
)
```

Arguments

result	result output from projectionDriveR function in PV mode
FC	fold change threshold, default at 0.2
pvalue	significance threshold, default set stored pvalue
subset	vector of gene names to subset the plot by
filter.inf	remove genes that have pvalues below machine double minimum value
label.num	Number of genes to label on either side of the volcano plot, default 5
display	boolean. Whether or not to plot and display volcano plots

Value

A list with weighted and unweighted differential expression metrics

plotConfidenceIntervals *plotConfidenceIntervals*

Description

Generate point and line confidence intervals from provided estimates.

Usage

```
plotConfidenceIntervals(
  confidence_intervals,
  interval_name = c("low", "high"),
  pattern_name = NULL,
  sort = TRUE,
  genes = NULL,
  weights = NULL,
  weights_clip = 0.99,
  weights_vis_norm = "none",
  weighted = FALSE
)
```

Arguments

confidence_intervals	A dataframe of features x estimates.
interval_name	Estimate column names. Default: c("low","high")
pattern_name	string to use as the title for plots.
sort	Boolean. Sort genes by their estimates (default = TRUE)
genes	a vector with names of genes to include in plot. If sort=F, estimates will be plotted in this order.
weights	optional. weights of features to include as annotation.
weights_clip	optional. quantile of data to clip color scale for improved visualization. Default: 0.99
weights_vis_norm	Which version of weights to visualize as a heatmap. Options are "none" (uses provided weights) or "quantiles". Default: none
weighted	specifies whether the confidence intervals in use are weighted by the pattern and labels plots accordingly

Value

A list with pointrange estimates and a heatmap of pattern weights.

---

plotVolcano *plotVolcano*

Description

Volcano plotting function

Usage

```
plotVolcano(stats, metadata, FC, pvalue, title)
```

Arguments

stats	data frame with differential expression statistics
metadata	#metadata from pdVolcano
FC	Fold change threshold
pvalue	p value threshold
title	plot title

---

projectionDriveR *projectionDriveR*

Description

Calculate weighted expression difference between two groups (group1 - group2)

Usage

```
projectionDriveR(
  cellgroup1,
  cellgroup2,
  loadings,
  pattern_name,
  loadingsNames = NULL,
  pvalue = 1e-05,
  display = TRUE,
  normalize_pattern = TRUE,
  mode = "CI"
)
```

Arguments

cellgroup1	gene x cell count matrix for cell group 1
cellgroup2	gene x cell count matrix for cell group 2
loadings	A matrix of continuous values defining the features
pattern_name	column of loadings for which drivers will be calculated
loadingsNames	a vector with names of loading rows defaults to rownames
pvalue	confidence level. Default 1e-5
display	boolean. Whether or not to display confidence intervals
normalize_pattern	Boolean. Whether or not to normalize pattern weights
mode	statistical approach, confidence intervals or pvalues. default CI

Value

A list with unweighted/weighted mean differences and differential genes that meet the provided signficance threshold.

projectR *Generic projectR function*

Description

A function for the projection of new data into a previously defined feature space.

Usage

```
projectR(data, loadings, dataNames = NULL, loadingsNames = NULL, ...)

## S4 method for signature 'matrix,matrix'
projectR(
  data,
  loadings,
  dataNames = NULL,
  loadingsNames = NULL,
  NP = NA,
  full = FALSE,
  family = "gaussianff",
  bootstrapPval = FALSE,
  bootIter = 1000
)

## S4 method for signature 'dgCMatrix,matrix'
projectR(
  data,
  loadings,
  dataNames = NULL,
  loadingsNames = NULL,
  NP = NA,
  full = FALSE,
  family = "gaussianff"
)

## S4 method for signature 'matrix,LinearEmbeddingMatrix'
projectR(
  data,
  loadings,
  dataNames = NULL,
  loadingsNames = NULL,
  NP = NA,
  full = FALSE,
  model = NA,
  family = "gaussianff",
  bootstrapPval = FALSE,
  bootIter = 1000
```

```
)

## S4 method for signature 'matrix,prcomp'
projectR(
  data,
  loadings,
  dataNames = NULL,
  loadingsNames = NULL,
  NP = NA,
  full = FALSE
)

## S4 method for signature 'matrix,rotatoR'
projectR(
  data,
  loadings,
  dataNames = NULL,
  loadingsNames = NULL,
  NP = NA,
  full = FALSE
)

## S4 method for signature 'matrix,correlateR'
projectR(
  data,
  loadings,
  dataNames = NULL,
  loadingsNames = NULL,
  NP = NA,
  full = FALSE,
  bootstrapPval = FALSE,
  bootIter = 1000
)

## S4 method for signature 'matrix,hclust'
projectR(
  data,
  loadings,
  dataNames = NULL,
  loadingsNames = NULL,
  full = FALSE,
  targetNumPatterns,
  sourceData,
  bootstrapPval = FALSE,
  bootIter = 1000
)

## S4 method for signature 'matrix,kmeans'
```

```
projectR(
  data,
  loadings,
  dataNames = NULL,
  loadingsNames = NULL,
  full = FALSE,
  sourceData,
  bootstrapPval = FALSE,
  bootIter = 1000
)

## S4 method for signature 'matrix,cluster2pattern'
projectR(
  data,
  loadings,
  dataNames = NULL,
  loadingsNames = NULL,
  full = FALSE,
  sourceData,
  bootstrapPval = FALSE,
  bootIter = 1000
)
```

Arguments

data	Target dataset into which you will project. It must of type matrix.
loadings	loadings learned from source dataset.
dataNames	a vector containing unique name, i.e. gene names, for the rows of the target dataset to be used to match features with the loadings, if not provided by `rownames(data)`. Order of names in vector must match order of rows in data.
loadingsNames	a vector containing unique names, i.e. gene names, for the rows ofloadings to be used to match features with the data, if not provided by `rownames(loadings)`. Order of names in vector must match order of rows in loadings.
...	Additional arguments to projectR
NP	vector of integers indicating which columns of loadings object to use. The default of NP=NA will use entire matrix.
full	logical indicating whether to return the full model solution. By default only the new pattern object is returned.
family	VGAM family function for model fitting (default: "gaussianff")
bootstrapPval	logical to indicate whether to generate p-values using bootstrap, not available for prcomp and rotatoR objects
bootIter	number of bootstrap iterations, default = 1000
model	Optional arguements to choose method for projection
targetNumPatterns	desired number of patterns with hclust
sourceData	data used to create cluster object

Details

loadings can belong to one of several classes depending on upstream analysis. Currently permitted classes are matrix, CogapsResult, CoGAPS, pclust, prcomp, rotatoR, and correlateR. Please note that loadings should not contain NA.

Value

A matrix of sample weights for each input basis in the loadings matrix (if full=TRUE, full model solution is returned).

Examples

```
projectR(data=p.ESepiGen4c1l$mRNA.Seq,loadings=AP.RNAseq6l3c3t$Amean,
dataNames = map.ESepiGen4c1l[["GeneSymbols"]])

library("CoGAPS")
# CR.RNAseq6l3c3t <- CoGAPS(p.RNAseq6l3c3t, params = new("CogapsParams", nPatterns=5))
projectR(data=p.ESepiGen4c1l$mRNA.Seq,loadings=CR.RNAseq6l3c3t,
dataNames = map.ESepiGen4c1l[["GeneSymbols"]])

pca.RNAseq6l3c3t<-prcomp(t(p.RNAseq6l3c3t))
pca.ESepiGen4c1l<-projectR(data=p.ESepiGen4c1l$mRNA.Seq,
loadings=pca.RNAseq6l3c3t, dataNames = map.ESepiGen4c1l[["GeneSymbols"]])

pca.RNAseq6l3c3t<-prcomp(t(p.RNAseq6l3c3t))
r.RNAseq6l3c3t<-rotatoR(1,1,-1,-1,pca.RNAseq6l3c3t$rotation[,1:2])
pca.ESepiGen4c1l<-projectR(data=p.ESepiGen4c1l$mRNA.Seq,
loadings=r.RNAseq6l3c3t, dataNames = map.ESepiGen4c1l[["GeneSymbols"]])

c.RNAseq6l3c3t<-correlateR(genes="T", dat=p.RNAseq6l3c3t, threshtype="N",
threshold=10, absR=TRUE)
cor.ESepiGen4c1l<-projectR(data=p.ESepiGen4c1l$mRNA.Seq, loadings=c.RNAseq6l3c3t,
NP="PositiveCOR", dataNames = map.ESepiGen4c1l[["GeneSymbols"]])

library("projectR")
data(p.RNAseq6l3c3t)
nP<-3
kClust<-kmeans(t(p.RNAseq6l3c3t),centers=nP)
kpattern<-cluster2pattern(clusters = kClust, NP = nP, data = p.RNAseq6l3c3t)
p<-as.matrix(p.RNAseq6l3c3t)
projectR(p,kpattern)
```

---

retinal_patterns *CoGAPS patterns learned from the developing mouse retina.*

Description

CoGAPS patterns learned from the developing mouse retina.

Usage

```
retinal_patterns
```

Format

A gene (rows) by pattern (column) matrix

References

1. Clark, B.S., & Stein-O'Brien G.L., et al. Single-Cell RNA-Seq Analysis of Development Identifies NFI Factors as Regulating Mitotic Exit and Late-Born Cell Specification. Cell 102, 1111-1126 (2019).

`rotatoR` *rotatoR*

Description

a function for rotating two basis about a point or line in that plane

Usage

```
rotatoR(x1, y1, x2, y2, basisSET)
```

Arguments

`x1`	a value describing a the coordinate of a point in the first basis. If no values are provided for x2
`y1`	a value describing a the coordinate of a point in the second basis
`x2`	a value describing a the coordinate of the second point in the second basis
`y2`	a value describing a the coordinate of the second point in the second basis
`basisSET`	the basis to be rotated

Value

An object of class rotatoR.

Examples

```
pca.RNAseq6l3c3t<-prcomp(t(p.RNAseq6l3c3t))
r.RNAseq6l3c3t<-rotatoR(1,1,-1,-1,pca.RNAseq6l3c3t$rotation[,1:2])
```

rotatoR-class *rotatoR*

Description

class of rotatoR output.

Slots

`rotatedM` rotated basis set (matrix) that is output of rotatoR function

Index