Package ‘projectR’

April 4, 2024

Type Package
Title Functions for the projection of weights from PCA, CoGAPS, NMF, correlation, and clustering
Version 1.18.0
Author Gaurav Sharma, Genevieve Stein-O’Brien
Maintainer Genevieve Stein-O’Brien <gsteinobrien@gmail.com>
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, reshape2, viridis, scales, Matrix, MatrixModels, ggplot2, cowplot, ggrepel, umap, tsne
Suggests BiocStyle, CoGAPS, gridExtra, grid, testthat, devtools, rmarkdown, knitr, rmarkdown, ComplexHeatmap, gplots
LazyData TRUE
RoxygenNote 7.2.3
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 (>= 4.0.1)
git_url https://git.bioconductor.org/packages/projectR
git_branch RELEASE_3_18
git_last_commit 91ca9bd
git_last_commit_date 2023-10-24
Repository Bioconductor 3.18
Date/Publication 2024-04-03
[R topics documented:]

- alluvialMat ......................................................... 2
- AP.RNAseq6l3c3t .................................................... 3
- aucMat .............................................................. 4
- bonferroniCorrectedDifferences .............................. 4
- cluster2pattern .................................................... 5
- cluster2pattern-class ............................................. 6
- clusterPlotR ........................................................ 6
- correlateR .......................................................... 7
- correlateR-class ................................................... 8
- CR.RNAseq6l3c3t .................................................... 9
- geneMatchR .......................................................... 9
- getTSNE .............................................................. 10
- getUMAP ............................................................. 10
- glial_counts ......................................................... 11
- initialize,cluster2pattern-method ......................... 12
- initialize,correlateR-method ................................. 12
- initialize,rotatoR-method ....................................... 13
- intersectoR .......................................................... 13
- map.ESepiGen4c1l .................................................. 14
- map.RNAseq6l3c3t .................................................. 15
- microglial_counts .................................................. 15
- p.ESepiGen4c1l ....................................................... 16
- p.RNAseq6l3c3t ...................................................... 16
- pd.ESepiGen4c1l ..................................................... 17
- pd.RNAseq6l3c3t ..................................................... 17
- plotConfidenceIntervals ........................................ 18
- projectionDriveR .................................................. 19
- projectR ............................................................. 20
- retinal_patterns ................................................... 23
- rotatoR .............................................................. 24
- rotatoR-class ....................................................... 25

### Index

| alluvialMat | 26 |

### Description

Function to provide alluvial matrix for generating alluvial plot
Usage

```r
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 character for 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**: threshold level of significance for p-value
- **qvalThreshold**: threshold level of significance for Benjamini-Hochberg corrected p-value

Value

A matrix to generate alluvial plots

Examples

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

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

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.ESepiGen4c11$mRNA.Seq,loadings=AP.RNAseq6l3c3t$Amean,
dataNames = map.ESepiGen4c11[["GeneSymbols")]] -> projection
aucMat(pd.ESepiGen4c11$Condition,projection)

bonferroniCorrectedDifferences

Description
Calculate the (weighted) difference in means for each measurement between two groups.

Usage
bonferroniCorrectedDifferences(group1, group2, diff_weights = NULL, pvalue)
cluster2pattern

Arguments

- group1: count matrix 1
- group2: count matrix 2
- diff_weights: loadings to weight the differential expression between the groups
- pvalue: significance value to threshold at

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

```r
k.RNAseq6l3ct <- 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

```r
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,
  ...)
```
correlateR

cData
cls
x
NC
annoIndx
label
...
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
- **...**: addition 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

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

---

**correlateR-class**

<table>
<thead>
<tr>
<th>correlateR-class</th>
<th>correlateR</th>
</tr>
</thead>
</table>

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

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
getUMAP

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

Description

Function to provide tSNE of projection

Usage

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

Arguments

projection  
matrix, a projection generated from projectR
axis  
integer, either 1 umap of projection or 2 for umap of transpose of projection
...  
additional 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

Description

Function to provide umap of projection

Usage

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

**Arguments**

- **projection**
  - matrix, 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**

```r
library(umap)
projection <- projectR(data=p.ESepiGen4c1l$mRNA.Seq,loadings=AP.RNAseq613c3t$Amean,
dataNames = map.ESepiGen4c1l[["GeneSymbols"]], full = TRUE)
umapConfig = umap.defaults
umapConfig$n_neighbors = 3
projectionUMAP <- getUMAP(projection,umapConfig = umapConfig)
```

**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**

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

**Arguments**

- `.Object`: clusterMatrix object
- `clusterMatrix`: matrix of continuous values for projection that is output of cluster2pattern function
- `...`: additional arguments to initialize cluster2pattern

**Value**

initialized cluster2pattern object

---

### initialize,correlateR-method

**Constructor for correlateR**

**Description**

Constructor for correlateR

**Usage**

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

**Arguments**

- `.Object`: correlateR object
- `corM`: correlation matrix obtained from correlateR
- `...`: additional arguments to initialize 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 initialize 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
full  logical indicating whether to return full data frame of significantly 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

```r
ESepiGen4c11mRNASeq <- p.ESepiGen4c11$mRNA.Seq
rownames(ESepiGen4c11mRNASeq) <- map.ESepiGen4c11$GeneSymbols

k.RNAseq613c3t<-kmeans(p.RNAseq613c3t,22)
k.ESepiGen4c11<-kmeans(ESepiGen4c11mRNASeq,10)
intersectoR(k.RNAseq613c3t, k.ESepiGen4c11, pval=.05)

h.RNAseq613c3t<-hclust(as.dist(1-(cor(t(p.RNAseq613c3t)))))
h.ESepiGen4c11<-hclust(as.dist(1-(cor(t(ESepiGen4c11mRNASeq)))))
intersectoR(pSet1=h.ESepiGen4c11, pSet2=h.RNAseq613c3t, pval=.05, k=c(3,4))
```

map.ESepiGen4c11  RNAseqing and ChIPSeq of matched genes in differentiated human iPS cells

Description

map.ESepiGen4c11 contains gene annotations

Usage

map.ESepiGen4c11

Format

A data frames with 93 rows and 9 variables:
References


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
**p.RNAseq6l3c3t**

**Description**

p.RNAseq6l3c3t contains log2(RPKM + 1) values for polyA bulk sequencing of 6 cell lines in 3 experimental conditions at 3 time points.

**Usage**

p.RNAseq6l3c3t

**Format**

A data frames with 108 rows and 54 variables:

---

**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 line 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 93 rows and between 4 and 9 variables:

**References**

**pd.ESepiGen4c1l**

**Description**

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

**Usage**

`pd.ESepiGen4c1l`

**Format**

A data frames with 9 rows and 2 variables:

**References**


---

**pd.RNAseq6l3c3t**

**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:
Description

Generate point and line confidence intervals from provided estimates.

Usage

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

Arguments

- `confidence_intervals`: A dataframe of features x estimates.
- `interval_name`: names of columns that contain the low and high estimates, respectively. Default: c("low", "high")
- `pattern_name`: string to use as the title for plots.
- `sort`: Boolean. Whether or not to sort genes by their estimates (default = T)
- `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 processed version of weights to visualize as a heatmap. Options are "none" (which uses provided weights) or "quantiles". Default: none

Value

A list with pointrange estimates and, if requested, a heatmap of pattern weights.
Description

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

Usage

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

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

loadingsNames  a vector with names of loading rows. Defaults to rownames.

pattern_name  column of loadings for which drivers will be calculated.

pvalue       confidence level for the bonferroni confidence intervals. Default 1e-5

display      boolean. Whether or not to plot and display confidence intervals

normalize_pattern  Boolean. Whether or not to normalize pattern weights.

Value

A list with weighted mean differences, mean differences, and differential genes that meet the provided significance 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, ...)
```

**R Code Snippet**

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

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

```r
## 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'

```r
projectR(
  data,
  loadings,
  dataNames = NULL,
  loadingsNames = NULL,
  NP = NA,
  full = FALSE
)
```

## S4 method for signature 'matrix,rotatoR'

```r
projectR(
  data,
  loadings,
  dataNames = NULL,
  loadingsNames = NULL,
  NP = NA,
  full = FALSE
)
```

## S4 method for signature 'matrix,correlateR'

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

## S4 method for signature 'matrix,hclust'

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

## S4 method for signature 'matrix,kmeans'

```r
projectR(
  data,
  loadings,
  dataNames = NULL,
  loadingsNames = NULL,
  NP = NA,
  full = FALSE
)
```
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 of loadings 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.
- **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 arguments 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

```r
projectR(data=p.ESepiGen4c11$mRNA.Seq, loadings=AP.RNAseq6l3c3t$Amean,
dataNames = map.ESepiGen4c11[["GeneSymbols"]])

library("CoGAPS")
# CR.RNAseq6l3c3t <- CoGAPS(p.RNAseq6l3c3t, params = new("CogapsParams", nPatterns=5))
projectR(data=p.ESepiGen4c11$mRNA.Seq, loadings=CR.RNAseq6l3c3t,
dataNames = map.ESepiGen4c11[["GeneSymbols"]])

c.RNAseq6l3c3t<-prcomp(t(p.RNAseq6l3c3t))
pca.ESepiGen4c11<-projectR(data=p.ESepiGen4c11$mRNA.Seq,
loadings=c.RNAseq6l3c3t, dataNames = map.ESepiGen4c11[["GeneSymbols"]])

r.RNAseq6l3c3t<-'rotatoR(1,1,-1,-1,pca.RNAseq6l3c3t$rotation[,1:2])
pca.ESepiGen4c11<-projectR(data=p.ESepiGen4c11$mRNA.Seq,
loadings=r.RNAseq6l3c3t, dataNames = map.ESepiGen4c11[["GeneSymbols"]])

c.RNAseq6l3c3t<-correlateR(genes="T", dat=p.RNAseq6l3c3t, threshtype="N",
threshold=10, absR=TRUE)
cor.ESepiGen4c11<-projectR(data=p.ESepiGen4c11$mRNA.Seq, loadings=c.RNAseq6l3c3t,
NP="PositiveCOR", dataNames = map.ESepiGen4c11[["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


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

Value

An object of class `rotatoR`.

Examples

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

class of rotatoR output.

Slots

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

* datasets
  - AP.RNAseq6l3c3t, 3
  - CR.RNAseq6l3c3t, 9
  - glial_counts, 11
  - map.ESepiGen4c1l, 14
  - map.RNAseq6l3c3t, 15
  - microglial_counts, 15
  - p.ESepiGen4c1l, 16
  - p.RNAseq6l3c3t, 16
  - pd.ESepiGen4c1l, 17
  - pd.RNAseq6l3c3t, 17
  - retinal_patterns, 23

- alluvialMat, 2
- AP.RNAseq6l3c3t, 3
- aucMat, 4

- bonferroniCorrectedDifferences, 4

- cluster2pattern, 5
  - cluster2pattern, character-method (cluster2pattern), 5
  - cluster2pattern, hclust-method (cluster2pattern), 5
  - cluster2pattern, kmeans-method (cluster2pattern), 5
  - cluster2pattern, numeric-method (cluster2pattern), 5
  - cluster2pattern-class, 6
  - clusterPlotR, 6
  - clusterPlotR, ANY, hclust-method (clusterPlotR), 6
  - clusterPlotR, ANY, kmeans-method (clusterPlotR), 6

- correlateR, 7
- correlateR-class, 8
- CR.RNAseq6l3c3t, 9

- geneMatchR, 9
- getTSNE, 10
- getUMAP, 10
- glial_counts, 11

- initialize, cluster2pattern-method, 12
- initialize, correlateR-method, 12
- initialize, rotatoR-method, 13
- intersectoR, 13
  - intersectoR, hclust, hclust-method (intersectoR), 13
  - intersectoR, kmeans, kmeans-method (intersectoR), 13

- map.ESepiGen4c1l, 14
- map.RNAseq6l3c3t, 15
- microglial_counts, 15

- p.ESepiGen4c1l, 16
- p.RNAseq6l3c3t, 16
- pd.ESepiGen4c1l, 17
- pd.RNAseq6l3c3t, 17

- plotConfidenceIntervals, 18
- projectionDriveR, 19
- projectR, 20
  - projectR, dgCMatrix, matrix-method (projectR), 20
  - projectR, matrix, cluster2pattern-method (projectR), 20
  - projectR, matrix, correlateR-method (projectR), 20
  - projectR, matrix, hclust-method (projectR), 20
  - projectR, matrix, kmeans-method (projectR), 20
  - projectR, matrix, LinearEmbeddingMatrix-method (projectR), 20
  - projectR, matrix, matrix-method (projectR), 20
  - projectR, matrix, prcomp-method (projectR), 20
INDEX

projectR, matrix, rotatoR-method (projectR), 20

retinal_patterns, 23
rotatoR, 24
rotatoR-class, 25