Title  The ClusterSignificance package provides tools to assess if class clusters in dimensionality reduced data representations have a separation different from permuted data

Version  1.32.0

Author  Jason T. Serviss [aut, cre], Jesper R. Gadin [aut]

Maintainer  Jason T Serviss <jason.serviss@ki.se>

Description  The ClusterSignificance package provides tools to assess if class clusters in dimensionality reduced data representations have a separation different from permuted data. The term class clusters here refers to, clusters of points representing known classes in the data. This is particularly useful to determine if a subset of the variables, e.g. genes in a specific pathway, alone can separate samples into these established classes. ClusterSignificance accomplishes this by, projecting all points onto a one dimensional line. Cluster separations are then scored and the probability of the seen separation being due to chance is evaluated using a permutation method.

Depends  R (>= 3.3.0)

URL  https://github.com/jasonserviss/ClusterSignificance/

BugReports  https://github.com/jasonserviss/ClusterSignificance/issues

Imports  methods, pracma, princurve (>= 2.0.5), scatterplot3d, RColorBrewer, grDevices, graphics, utils, stats

License  GPL-3

LazyData  true

Suggests  knitr, rmarkdown, testthat, BiocStyle, ggplot2, plsgenomics, covr

VignetteBuilder  knitr

biocViews  Clustering, Classification, PrincipalComponent, StatisticalMethod
The ClusterSignificance package provides tools to assess if clusters have a separation different from random or permuted data.

The ClusterSignificance package provides tools to assess if clusters have a separation different from random or permuted data. ClusterSignificance investigates clusters of two or more groups by first, projecting all points onto a one dimensional line. Cluster separations are then scored and the probability of the seen separation being due to chance is evaluated using a permutation method.

Details

<table>
<thead>
<tr>
<th>Package:</th>
<th>ClusterSignificance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type:</td>
<td>Package</td>
</tr>
<tr>
<td>Version:</td>
<td>1.0</td>
</tr>
<tr>
<td>Date:</td>
<td>2016-02-28</td>
</tr>
<tr>
<td>License:</td>
<td>GPL-3</td>
</tr>
</tbody>
</table>
Author(s)

Author: Jason T. Serviss, Jesper R. Gadin

References

Reference to published application note (work in progress)

---

**ClassifiedPoints-class**

*Classification of the one dimensional points in a Pcp or Mlp object.*

**Description**

Classification based on ROC params (TN TP FP FN).

**Usage**

```r
## S4 method for signature 'ClassifiedPoints'
getData(x, n = NULL)

classify(x, ...)

## S4 method for signature 'Pcp'
classify(x, ...)

## S4 method for signature 'Mlp'
classify(x, ...)

## S4 method for signature 'ClassifiedPoints'
initialize(.Object, ..., scores,
  scores.points = scores.points, scores.index = scores.index, ROC, AUC,
  class.color)

## S4 method for signature 'ClassifiedPoints,missing'
plot(x, y, comparison = "all",
  class.color = NULL, ...)

## S4 method for signature 'ClassifiedPoints'
show(object)
```
Arguments

x: Pcp or Mlp Object for the function classify otherwise it is a ClassifiedPoints object
n: data to extract from ClassifiedPoints (NULL gives all)
...: additional arguments to pass on
.Object: internal object
scores: final scores
scores.points: sorted points
scores.index: index of sorted points
ROC: parameters (TN, TP, FN and FP)
AUC: area under the curve
class.color: user assigned group coloring scheme
y: default plot param, which should be set to NULL
comparison: Specify a comparison i.e. ("grp1 vs grp2") and plot only that comparison.
object: ClassifiedPoints Object

Details
Tests all possible discrimination lines and picks the one with highest score based on a score which is simply calculated by the formula (TP - FP) + (TN - FN).
The plot shows the distribution of scores for different discrimination lines. Each line is a separator that has a score for the separation of the two groups, and the height of the line marks the score for this separation.

Value
The classify function returns an object of class ClassifiedPoints

Author(s)
Jesper R. Gadin and Jason T. Serviss

Examples

#use demo data
data(pcpMatrix)
classes <- rownames(pcpMatrix)

#run function
prj <- pcp(pcpMatrix, classes)
c1 <- classify(prj)

#getData accessor
data(c1)

#getData accessor specific
Mlp-class

```r
get_data(cl, "scores")

# plot result
plot(cl)
```

---

**Mlp-class**

*Projection of points into one dimension.*

---

**Description**

Project points onto the mean based line.

**Usage**

```r
## S4 method for signature 'Mlp'
get_data(x, n = NULL)

## S4 method for signature 'Mlp'
initialize(.Object, ..., classes, points.orig, line,
          points.onedim, class.color)

## S4 method for signature 'Mlp,missing'
plot(x, y, steps = "all", ...)  
mlp(mat, ...)

## S4 method for signature 'matrix'
mlp(mat, classes, class.color = NULL, ...)

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

**Arguments**

- `x` matrix object for the function mlp otherwise it is a Mlp object
- `n` data to extract from Mlp (NULL gives all)
- `.Object` internal object
- `...` additional arguments to pass on
- `classes` vector in same order as rows in matrix
- `points.orig` multidimensional points describing the original data
- `line` multidimensional points describing a line
- `points.onedim` a vector of points
- `class.color` user assigned group coloring scheme
- `y` default plot param, which should be set to NULL (default: NULL)
steps 1,2,3,4,5,6 or "all"
mat matrix with samples on rows, PCs in columns. Ordered PCs, with PC1 to the left.
object Mlp object

Details

Projection of the points onto a line between the mean of two groups. Mlp is the abbreviation for 'mean line projection'. The function accepts, at the moment, only two groups and two PCs at a time.

An object containing results from a mean line projection reduction to one dimension.

The group and the one dimensional points are the most important information to carry out a classification using the classify() function. As a help to illustrate the details of the dimension reduction, the information from some critical steps are stored in the object. To visually explore these there is a dedicated plot method for Mlp objects, use plot().

Value

The mlp function returns an object of class Mlp

Author(s)

Jesper R. Gadin and Jason T. Serviss

Examples

# use demo data
data(mlpMatrix)
groups <- rownames(mlpMatrix)

# run function
prj <- mlp(mlpMatrix, groups)

# getData accessor
getInfo(prj)

# getData accessor specific
getInfo(prj, "line")

# plot result
plot(prj)
mlpMatrix

Simulated data used to demonstrate the Mlp method.

Description
MLp demonstration matrix.

Usage
mlpMatrix

Format
Matrix
  rownames Groups
  colnames dimension number

Value
simulated matrix

Examples
mlpMatrix

Pcp-class
Projection of points into one dimension.

Description
Project points onto a principal curve.

Usage
getData(x, ...)

## S4 method for signature 'Pcp'
getData(x, n = NULL)

## S4 method for signature 'Pcp'
initialize(.Object, ..., classes, points.orig, line,
  points.onedim, index, class.color)

## S4 method for signature 'Pcp,missing'
plot(x, y, steps = "all", class.color = NULL, ...)

pcp(mat, ...)

## S4 method for signature 'matrix'
pcp(mat, classes, df = NULL, warn = TRUE,
     class.color = NULL, ...)

## S4 method for signature 'Pcp'
show(object)

Arguments

x matrix object for the function pcp otherwise it is a Pcp object
...
additional arguments to pass on
n data to extract from Pcp (NULL gives all)
.Object internal object
classes vector in same order as rows in matrix
points.orig multidimensional points describing the original data
line multidimensional points describing a line
points.onedim a vector of points
index internal index from the projection
class.color user assigned group coloring scheme
y default plot param, which should be set to NULL
steps 1,2,3,4,5,6 or "all"
mat matrix with samples on rows, PCs in columns. Ordered PCs, with PC1 to the left.
df degrees of freedom, passed to smooth.spline
warn logical indicating if a change in the default df argument should generate a warning. mostly for internal use.
object Pcp object

Details

The resulting Pcp object containing results from a principal curve reduction to one dimension. The group and the one dimensional points will be the information needed to carry out a classification using the classify() function. As a help to illustrate the details of the dimension reduction, the information from some critical steps is stored in the object. To visually explore these there is a dedicated plot method for Pcp objects, use plot().

Value

The pcp function returns an object of class Pcp
Author(s)
Jesper R. Gadin and Jason T. Serviss

Examples

#use demo data
data(pcpMatrix)
classes <- rownames(pcpMatrix)

#run function
prj <- pcp(pcpMatrix, classes)

#getData accessor
data(prj)

#getData accessor specific
data(prj, "line")

#plot the result (if dim >2, then plot in 3d)
plot(prj)

#plot the result (if dim=2, then plot in 2d)
prj2 <- pcp(pcpMatrix[,1:2], classes)
plot(prj2)

---

**pcpMatrix**  
*Simulated data used to demonstrate the Pcp method.*

Description
Pcp demonstration matrix.

Usage

```R
pcpMatrix
```

Format

Matrix

- **rownames** Groups
- **colnames** dimension number

Value

simulated matrix
Examples

```r
pcpMatrix
```

---

**PermutationResults-class**

*Permutation test*

---

**Description**

Test how the classification performs compared to random (eg. permuted) data.

**Usage**

```r
## S4 method for signature 'PermutationResults'
getData(x, n = NULL)

## S4 method for signature 'PermutationResults'
c(x, ..., recursive = FALSE)

pvalue(x, ...)

## S4 method for signature 'PermutationResults'
conf.int(x, conf.level = 0.99, ...)

## S4 method for signature 'PermutationResults'
initialize(.Object, ..., scores.real, scores.vec)

permute(mat, ...)

## S4 method for signature 'matrix'
permute(mat, classes, projmethod = "pcp", iter = 100,
         user.permutations = NULL, seed = 3, df = NULL, verbose = TRUE, ...)

## S4 method for signature 'PermutationResults,missing'
plot(x, y, comparison = "all", ...)

## S4 method for signature 'PermutationResults'
show(object)
```
**Arguments**

- `x` matrix for the function `permute`, otherwise it is a PermutationResults object
- `n` data to extract from `ClassifiedPoints` (NULL gives all)
- `...` arguments to pass on
- `recursive` don't use (belongs to default generic of combine 'c()')
- `conf.level` confidence level for the returned confidence interval
- `.Object` internal object
- `scores.real` the real score
- `scores.vec` all permuted scores
- `mat` matrix with samples on rows, PCs in columns. Ordered PCs, with PC1 to the left.
- `classes` vector in same order as rows in matrix
- `projmethod` 'pcp' or 'mlp'
- `iter` integer number of iterations to be performed.
- `user.permutations` user defined permutation matrix
- `seed` random seed to be used by the internal permutation
- `df` degrees of freedom, passed to `smooth.spline`
- `verbose` makes function more talkative
- `y` default plot param, which should be set to NULL
- `comparison` Specify a comparison i.e. ("grp1 vs grp2") and plot only that comparison.
- `object` `ClassifiedPoints Object`

**Details**

This is a test suite and will return a summarized object. The default of the parameter 'iter' is set quite low, and in principle the more iterations the better, or until the pvalue converges to a specific value. If no pre-permuted data has been supplied by the user, then the internal permutation method will perform a sampling without replacement within each dimension.

**Value**

The `permute` function returns an object of class `PermutationResults`

**Author(s)**

Jesper R. Gadin and Jason T. Serviss
Examples

# use pcp method
data(pcpMatrix)
classes <- rownames(pcpMatrix)

# run function
iterations <- 10
pe <- permute(
    mat=pcpMatrix,
    classes=classes,
    iter=iterations,
    projmethod="pcp"
)

# use mlp method
data(mlpMatrix)
classes <- rownames(mlpMatrix)
pe <- permute(
    mat=mlpMatrix,
    classes=classes,
    iter=iterations,
    projmethod="mlp"
)

# getData accessor
data(pe)

# getData accessor specific
data(pe, "scores.vec")

# get pvalue
pvalue(pe)

# plot result
plot(pe)

# combine three (parallel) jobs on the same matrix
pe2 <- c(pe, pe, pe)
Index

* classification
  ClassifiedPoints-class, 3
* package
  ClusterSignificance-package, 2
* permutation
  PermutationResults-class, 10
* projection
  Mlp-class, 5
  Pcp-class, 7
  .ClassifiedPoints
    (ClassifiedPoints-class, 3
  .Mlp (Mlp-class), 5
  .Pcp (Pcp-class), 7
  .PermutationResults
    (PermutationResults-class, 10
  c,PermutationResults-method
    (PermutationResults-class, 10
  ClassifiedPoints
    (ClassifiedPoints-class, 3
  ClassifiedPoints-class, 3
  classify (ClassifiedPoints-class), 3
  classify,Mlp-method
    (ClassifiedPoints-class, 3
  classify,Pcp-method
    (ClassifiedPoints-class, 3
  ClusterSignificance
    (ClusterSignificance-package), 2
  ClusterSignificance-package, 2
  conf.int (PermutationResults-class), 10
  conf.int,PermutationResults-method
    (PermutationResults-class, 10
  getData (Pcp-class), 7
  getData,ClassifiedPoints-method
    (ClassifiedPoints-class, 3
  getData,Mlp-method (Mlp-class), 5
  getData,Pcp-method (Pcp-class), 7
  getData,PermutationResults-method
    (PermutationResults-class, 10
  initialize,ClassifiedPoints-method
    (ClassifiedPoints-class), 3
  initialize,Mlp-method (Mlp-class), 5
  initialize,Pcp-method (Pcp-class), 7
  initialize,PermutationResults-method
    (PermutationResults-class, 10
  Mlp (Mlp-class), 5
  mlp (Mlp-class), 5
  mlp,matrix-method (Mlp-class), 5
  Mlp-class, 5
  mlpMatrix, 7
  Pcp (Pcp-class), 7
  pcp (Pcp-class), 7
  pcp,matrix-method (Pcp-class), 7
  Pcp-class, 7
  pcpMatrix, 9
  PermutationResults-class, 10
  permute (PermutationResults-class), 10
  permute,matrix-method
    (PermutationResults-class, 10
  plot,ClassifiedPoints,missing-method
    (ClassifiedPoints-class), 3
  plot,Mlp,missing-method (Mlp-class), 5
  plot,Pcp,missing-method (Pcp-class), 7
  plot,PermutationResults,missing-method
    (PermutationResults-class, 10
  pvalue (PermutationResults-class), 10
  pvalue,PermutationResults-method
    (PermutationResults-class, 10
  show,ClassifiedPoints-method
    (ClassifiedPoints-class, 3
  show,Mlp-method (Mlp-class), 5
  show,Pcp-method (Pcp-class), 7
  show,PermutationResults-method
    (PermutationResults-class, 10