Package ‘adSplit’

January 15, 2026

Title Annotation-Driven Clustering

Version 1.80.0

Date 2021-02-01

Author Claudio Lottaz, Joern Toedling

Description This package implements clustering of microarray gene expression profiles according to functional annotations. For each term genes are annotated to, splits into two subclasses are computed and a significance of the supporting gene set is determined.

Maintainer Claudio Lottaz <`Claudio.Lottaz@klinik.uni-regensburg.de`>

Depends R (>= 2.1.0), methods (>= 2.1.0)

Imports AnnotationDbi, Biobase (>= 1.5.12), cluster (>= 1.9.1), GO.db (>= 1.8.1), graphics, grDevices, KEGGREST (>= 1.30.1), multtest (>= 1.6.0), stats (>= 2.1.0)

Suggests golubEsets (>= 1.0), vsn (>= 1.5.0), hu6800.db (>= 1.8.1)

LazyLoad yes

URL `http://compdiag.molgen.mpg.de/software/adSplit.shtml`

License GPL (>= 2)

biocViews Microarray, Clustering

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

git_branch RELEASE_3_22

git_last_commit 0efcd35

git_last_commit_date 2025-10-29

Repository Bioconductor 3.22

Date/Publication 2026-01-15

Contents

adSplit *Annotation-Driven Splits*

Description

This function searches for annotation-driven splits of patients in microarray data. A split is a partitioning of patients into two groups. In order to do so it refers to GO terms and KEGG pathways. In addition, a significance measure can be computed by simulating a random distribution of scores. DLD-scores are used to judge the quality of a split.

Usage

```
adSplit(mydata, annotation.ids, chip.name,
        min.probes = 20, max.probes = NULL,
        B = NULL, min.group.size = 5, ngenes = 50,
        ignore.genes = 5)
```

Arguments

`mydata` either an expression set as defined by the package `Biobase` or a matrix of expression levels (rows=genes, columns=samples).

`annotation.ids` a vector of GO or KEGG identifiers in the form "GO:..." or "KEGG:..." respectively. The prefix "KEGG:" is removed from the KEGG-identifiers before accessing the chip's "...PATH2PROBES" hash.

`chip.name` the name of the chip by which the expression set is measured. `adSplit` attempts to load a library of the same name and expects to find a hash called "<chip-name>GO2ALLPROBES" and one called "<chip-name>PATH2PROBES" there.

`min.probes` annotation identifiers with fewer than this associated genes are skipped.

`max.probes` annotation identifiers with more than this associated genes are skipped. The default is ten percent of the genes on the chip.

`B` the number of random gene set samplings to be performed to compute empirical p-values.

`min.group.size` filter criteria to avoid splits suggesting tiny groups. Splits where one of the two suggested groups are smaller than this number are removed from the split set.

`ngenes` number of genes used to compute DLD scores.

`ignore.genes` number of best scoring genes to be ignored when computing DLD scores.

Details

This function applies the same splitting procedure to all annotation identifiers provided. Firstly, the associated genes for one identifier are determined and extracted from the expression data. Then the diana2means function is applied to the restricted data and the different splits generated are collected into a single splitSet object.

As annotation identifiers vectors of identifiers of the KEGG:nnnnn and GO:nnnnnn are valid. In addition, the keywords "KEGG", "GO" and "all" are allowed, representing all terms in the corresponding ontology.

If B is set to a integer number this number of samplings are used to generate a null-distribution of DLD-scores. This distribution is used to compute empirical p-values for each split. If more than one valid split is found, multiple testing is corrected for by applying Benjamini-Hochbergs correction from the multtest package.

Value

Returns an object of class splitSet with the following list elements:

cuts	a matrix of split attributions. One row per annotation identifier (GO term or KEGG pathway for which a split has been generated. One column per object in the dataset.
score	one score per generated split.
pvalue	one empirical p-value per generated split, or NULL
qvalue	one q-value computed according Benjamini-Hochberg's correction for multiple testing per generated split, or NULL

Author(s)

Claudio Lottaz, Joern Toedling

See Also

diana2means, randomDiana2means, image.splitSet

Examples

```
# prepare data
library(golubEsets)
data(Golub_Merge)

# generate annotation-driven splits for apoptosis and signal transduction
x <- adSplit(Golub_Merge, "GO:0006915", "hu6800")
x <- adSplit(Golub_Merge, c("GO:0007165","GO:0006915"), "hu6800", max.probes=7000)

# generate a split for alanine, aspartate and glutamate metabolism including
# an empirical p-value
x <- adSplit(Golub_Merge, "KEGG:00250", "hu6800", B=100)

# generate splits for all KEGG pathways.
x <- adSplit(Golub_Merge, "KEGG", "hu6800")
image(x)
```

`adSplit-internal` *adSplit Internal Function*

Description

Function for internal use only.

Details

No details given. This function is subject to change without further notice.

Author(s)

Claudio Lottaz

`diana2means` *2-Means with Hierarchical Initialization*

Description

Split a set of data points into two coherent groups using the k-means algorithm. Instead of random initialization, divisive hierarchical clustering is used to determine initial groups and the corresponding centroids.

Usage

```
diana2means(mydata, mingroupsize = 5,
            ngenes = 50, ignore.genes = 5,
            return.cut = FALSE)
```

Arguments

`mydata`	either an expression set as defined by the package `Biobase` or a matrix of expression levels (rows=genes, columns=samples).
`mingroupsize`	report only splits where both groups are larger than this size.
`ngenes`	number of genes used to compute cluster quality DLD-score.
`ignore.genes`	number of best scoring genes to be ignored when computing DLD-scores.
`return.cut`	logical, whether to return the attributions of samples to groups.

Details

This function uses divisive hierarchical clustering (diana) to generate a first split of the data. Thereby, each column of the data matrix is considered to represent a data element. From the thus generated temptative groups, centroids are deduced and used to initialize the k-means clustering algorithm.

For the split optimized by k-means the DLD-score is determined using the `ngenes` and `ignore.genes` arguments.

Value

If the logical `return.cut` is set to `FALSE` (the default), a single number is representing the DLD-score for the generated split is returned. Otherwise an object of class `split` containing the following elements is returned:

`cut`	one number out of 0 and 1 per column in the original data, specifying the split attribution.
`score`	the DLD-score achieved by the split.

Author(s)

Joern Toedling, Claudio Lottaz

See Also

`diana`

Examples

```
# get golub data
library(vsn)
library(golubEsets)
data(Golub_Merge)

# use 10% most variable genes
e <- exprs(Golub_Merge)
vars <- apply(e, 1, var)
e <- e[vars > quantile(vars,0.9),]

# use diana2means to get splits and scores
diana2means(e)
diana2means(e, return.cut=TRUE)
```

`drawRandomPS` *Draw sets of probe-sets*

Description

This function draws a given number of probe-sets randomly, such that probe-sets referring to the same are either included or excluded as a whole.

Usage

```
drawRandomPS(nps, EID2PSenv, allEIDs)
```

Arguments

`nps`	number of probe-sets to be drawn.
`EID2PSenv`	a hash mapping EntrezGene to probe-set identifiers.
`allEIDs`	vector of all EntrezGene identifiers represented on a chip.

Value

A named vector of probe-set identifiers. The names correspond to the EntrezGene identifiers.

Author(s)

Claudio Lottaz

Examples

```
# draw ten random probe-sets from hu6800
library(hu6800.db)
EID2PSenv <- makeEID2PROBESenv(hu6800ENTREZID)
drawRandomPS(10, EID2PSenv, ls(EID2PSenv))
```

`golubKEGGSplits` *Examplar splitSet*

Description

This is a data object precomputed by `adSplit` for illustration.

Usage

```
data(golubKEGGSplits)
```

Format

Annotation-driven split set holds 70 splits on 72 elements, scores range is: 3.382672 17.31385, empirical p-values range is: 0.005 0.955, q-value range is: 0.1633333 0.955.

Details

This object is generated by the following call:

golubKEGGSplits <- adSplit(golubNorm, "KEGG", "hu6800", B=1000)

where `golubNorm` is a normalized version of `Golub_Merge` from the `golubEsets` package.

Examples

```
data(golubKEGGSplits)
```

`hist.splitSet` *Overview Histogram for splitSets*

Description

Draws a histogram of empirical p-values and shows the corresponding q-values corrected for multiple testing.

Usage

```
## S3 method for class 'splitSet'
hist(x, main = "Distribution of p-Values",
              xlab = "p-values", col = "grey", xlim = c(0, 1), ...)
```

Arguments

`x` object of type `splitSet`. Should hold a considerable number of splits.

`main` main title of the histogram.

`xlab` legend for the x-axis.

`col` color for the histogram bars.

`xlim` limits for the x-axis (p-values).

`...` further parameters passed on to the default `hist` function.

Details

This function draws a regular histogram of empirical p-values observed in the splitSet at hand. The corresponding q-values, corrected by the method suggested by Benjamini-Hochberg, are plotted into the same graph. The scale for the q-values is shown at the left hand side of the plot.

Author(s)

Claudio Lottaz

See Also

`adSplit`

Examples

```
data(golubKEGGSplits)
hist(golubKEGGSplits, col="red")
```

image.splitSet *Illustrate Split Sets*

Description

Draws an image of all splits, one per row, of a splitSet object. Each column corresponds to a patient.

Usage

```
## S3 method for class 'splitSet'
image(x, filter.fdr = 1, main = "", max.label.length = 50,
               full.names = TRUE, xlab = NULL, sample.labels = FALSE,
               col = c("yellow", "red"), invert = FALSE,
               outfile = NULL, res = 72, pointsize = 7, ...)
```

Arguments

x	the object of class splitSet to be illustrated.
filter.fdr	worst acceptable false discovery rate for the shown set of splits. All splits with q-values below this level are dropped from the image.
main	a title for the image.
max.label.length	Maximal length of the annotations shown to the right of the image. Longer annotations are truncated.
full.names	Show full names for annotations instead of their identifiers only.
xlab	additional annotation on the x-axis.
sample.labels	whether names of samples are to be shown on the x-axis.
col	two strings encoding the colors to be used to illustrate to which group a sample is attributed.
invert	whether to draw in white on black background.
outfile	the filename on which to draw the image in postscript format. The default is NULL, meaning to produce the image interactively.
res	resolution for bitmap output on postscript.
pointsize	size of font.
...	further arguments passed to image.

Details

The set of splits given is illustrated as an image. Each row corresponds to an annotation, each column to a patient. In position (x,y), the association of patient x to a group with respect to annotation y is coded as colors (yellow and red by default). The image is ordered by hierarchical clustering such that similar patients and similar splits are brought closer together.

Value

Always returns NULL.

Author(s)

Claudio Lottaz

See Also

`adSplit`

Examples

```
data(golubKEGGSplits)
image(golubKEGGSplits, filter.fdr=0.5)
```

---

makeEID2PROBESenv *Generate EID2PROBES environment*

Description

Make hash containing probe-sets per EntrezGene identifier.

Usage

```
makeEID2PROBESenv(EIDenv)
```

Arguments

`EIDenv`	an environment containing one entry per probe-set holding all corresponding EntrezGene identifiers.

Value

An environment containing one entry per EntrezGene identifier holding all corresponding probe-sets.

Author(s)

Joern Toedling, Claudio Lottaz

Examples

```
library(hu6800.db)
makeEID2PROBESenv(hu6800ENTREZID)
```

`print.split` *Print Information on Single Splits*

Description

Print information on a single split.

Usage

```
## S3 method for class 'split'
print(x, ...)
```

Arguments

`x`	the split-object to be described.
`...`	not used.

Author(s)

Claudio Lottaz

See Also

`diana2means`

`print.splitSet` *Print Summaries for splitSets*

Description

Prints information of splitSets.

Usage

```
## S3 method for class 'splitSet'
print(x, ...)
```

Arguments

`x`	the splitSet-object to be described.
`...`	not used.

Author(s)

Claudio Lottaz

See Also

`adSplit`

Examples

```
data(golubKEGGSplits)
print(golubKEGGSplits)
```

---

randomDiana2means *Generate null-distributions of DLD-scores*

Description

Draws a number of random sets of probe-sets consisting of the needed size and applies `diana2means` to compute DLD scores.

Usage

```
randomDiana2means(nprobes, data, chip, ndraws = 10000,
                    ngenes = 50, ignore.genes = 5)
```

Arguments

`nprobes`	the size of gene sets.
`data`	a matrix of expression data, rows correspond to genes, columns to samples.
`chip`	the name of the used chip.
`ndraws`	the number of DLD scores computed.
`ngenes`	the number of genes used to compute DLD scores (passed to `diana2means`).
`ignore.genes`	the number of best scoring genes to be ignored when computing DLD scores (passed to `diana2means`)

Details

This function uses `drawRandomPS` to draw `ndraws` gene sets. On these it applies `diana2means` to determine a null-distribution of DLD-scores.

Value

A vector of DLD-scores.

Author(s)

Joern Toedling, Claudio Lottaz

See Also

`drawRandomPS`, `diana2means`

Examples

```
# prepare data
library(vsn)
library(golubEsets)
data(Golub_Merge)

# generate DLD scores
scores <- randomDiana2means(20, exprs(Golub_Merge), "hu6800", ndraws = 500)
```

Index