Package 'GISPA'

April 11, 2018

Type Package

Title GISPA: Method for Gene Integrated Set Profile Analysis

Version 1.2.0

Date 2017-02-24

Author Bhakti Dwivedi and Jeanne Kowalski

Maintainer Bhakti Dwivedi <bhakti.dwivedi@emory.edu>

Description GISPA is a method intended for the researchers who are interested in defining gene sets with similar, a priori specified molecular profile. GISPA method has been previously published in Nucleic Acid Research (Kowalski et al., 2016; PMID: 26826710).

Depends R (>= 3.3.2)

Imports Biobase, changepoint, data.table, genefilter, graphics, GSEABase, HH, lattice, latticeExtra, plyr, scatterplot3d, stats

License GPL-2

LazyData true

Collate 'cptPlot.R' 'cptModel.R' 'computePS.R' 'GISPA.R' 'cptSlopeplot.R' 'data.R' 'propBarplot.R' 'stackedBarplot.R'

biocViews StatisticalMethod,GeneSetEnrichment,GenomeWideAssociation

Suggests knitr

VignetteBuilder knitr

NeedsCompilation no

RoxygenNote 6.0.1

R topics documented:

`cnvset` *Copy Number Variation (CNV) data*

Description

A dataset containing the genome-wide gene copy change identified in the 3 multiple myeloma cell line samples. The variables are as follows

Usage

```
cnvset
```

Format

A data matrix with 534 genes copy change and 3 samples

Details

- gene
- copy number variation segment ID
- sample 1 copy change segment mean value
- sample 2 copy change segment mean value
- sample 3 copy change segment mean value

@source https://research.themmrf.org/

`computePS` *Computes the profile statistics*

Description

Computes the increased or decreased profile statistics for each row (or gene) across the three samples within a feature or data type (expression, methylation, or copy-number variation)

Usage

```
computePS(rd1, cd1, cd2, profile)
```

Arguments

`rd1`	: A numeric value of the reference sample (R) on which to estimate the profile statistics for a given gene, gene probe or gene copy segment
`cd1`	: A numeric value of the comparison sample 1 (S1) for a given gene, gene probe or gene copy segment
`cd2`	: A numeric value of the comparison sample 2 (S2) for a given gene, gene probe or gene copy segment
`profile`	: The desired direction of genomic change. The values are "up" (default) or "down" to select for increased or decreased gene set profile, respectively

Details

This function requires three data values corrosponding to three samples for a given gene (or row), respectively

Value

The returned value is profile statistics computed considering the specified change in the reference sample when compared to the remaining two relative samples.

Author(s)

Bhakti Dwivedi & Jeanne Kowalski

Examples

```
rd1 = 40
cd1 = 20
cd2 = 20
computePS(rd1, cd1, cd2, profile="up")
```

`cptModel` *Computes within and between gene feature profile statistics by feature and amongst features*

Description

Computes the percentiles on the estimated profile statistics within a gene and across genes for one or more combination of feature or data types (expression, methylation, copy-number variation, or variant change)

Usage

```
cptModel(psm, genelist, cpt.data, cpt.method, cpt.max)
```

Arguments

`psm`	: A data matrix of estimated gene profile statistics for each feature
`genelist`	: A vector of gene names or gene symbols corrosponding to the profile statistics
`cpt.data`	: Identify changepoints in the data using variance (cpt.var), mean (cpt.mean) or both (meanvar). Default is cpt.var.
`cpt.method`	: Choice of single or multiple changepoint model. Default is "BinSeg".
`cpt.max`	: The maximum number of changepoints to search for using "BinSeg" method. Default is 60. This number is dependent on the number of input data points

Details

This function estimates within and between feature profile statistics by gens in addition to the summed percentiles and successive differences

Value

Estimated change points in the input data set

Author(s)

Bhakti Dwivedi & Jeanne Kowalski

Examples

```
id <- 1000 ## number of probes
s <- 3 ## number of sample groups
dm <- matrix(runif(id*s,0,200), nrow=id, ncol=s, dimnames=list(paste("gene", 1:id, sep="") , paste("fs", 1:s,
genelist <- rownames(dm)
cptModel(dm, genelist, cpt.data="var", cpt.method="BinSeg", cpt.max=60)
```

---

cptPlot *Scatterplot representation of gene sets by change points*

Description

Scatterplot representation of identified change points on the estimated profile statistics within the data

Usage

```
cptPlot(psv, cut.pts)
```

Arguments

psv	: A data vector of estimated profile statistics on which changepoints are identified
cut.pts	: The estimated profile statistics cutoffs corrosponding to the locations in psv

Details

This function expects 'gispa.output' profile statistics output from GISPA.R main function

Value

Plot representing all the identified gene sets by change points in the data

Author(s)

Bhakti Dwivedi & Jeanne Kowalski

Examples

```
x <- runif(100, 0.0, 1.0)
y <- c(0.2, 0.6, 0.8)
cpt.plot <- cptPlot(psv=x,cut.pts=y)
```

cptSlopeplot *Scatterplot representation of identified change points gene set slopes*

Description

This function will plot the average slopes estimated over all gene sets within each change point by data types

Usage

```
cptSlopeplot(gispa.output,feature,type)
```

Arguments

gispa.output	: A data matrix of between gene feature profile statistics for each feature with corrosponding identified changepoints. The row names should corrospond to genes or names to be displayed on y-axis
feature	: Analysis type i.e., one ('1'), two ('2') or three ('3') dimensional feature analysis.
type	: Type of data, e.g., EXP (default) for expression, VAR of variants, CNV for copy number change.

Details

This function expects the output from GISPA function of GISPA package, and highlights the gene set slope profile in the selected changepoints

Value

Scatterplot illustrating the average slopes by change point to access the best gene set profile

Author(s)

Bhakti Dwivedi & Jeanne Kowalski

Examples

```
id <- 200 ## number of rows
s <- 3 ## number of columns
dm <- matrix(runif(id*s,0,10), nrow=id, ncol=s,
                dimnames=list(paste("gene", 1:id, sep=""),
                paste("sample", 1:s, sep="")))
changepoints <- sort(sample(1:2, id, replace=TRUE))
dm <- cbind(dm,changepoints)
cptSlopeplot(gispa.output=dm,feature=1,type="EXP")
```

exprset *Gene Variant (EXP) data*

Description

A dataset containing the genome-wide gene expression values from 3 multiple myeloma cell line samples. The variables are as follows

Usage

```
exprset
```

Format

A data matrix with 1500 genes and 3 samples

Details

- gene
- gene names
- sample 1 log2 transformed normalized expression count values
- sample 2 log2 transformed normalized expression count values
- sample 3 log2 transformed normalized expression count values

@source https://research.themmrf.org/

GISPA *Gene Integrates Set Profile Analysis*

Description

Identifies gene sets with a similar a prior defined profile using any combination of three feature or data types

Usage

```
GISPA(feature,f.sets,g.set,ref.samp.idx,comp.samp.idx,f.profiles,cpt.data,cpt.method,cpt.max)
```

Arguments

feature	: Analysis type i.e., one ('1'), two ('2') or three ('3') dimensional feature analysis.
f.sets	: A list of ExpressionSet data objects corrosponding to a data type
g.set	: A GeneSet from an ExpressionSet to subset the f.sets for analysis purposes. 'geneIds' should corrospond to the gene names. Default is null, i.e., genome-wide analysis
ref.samp.idx	: Reference sample column index on which to determine the gene set profile. The default is 3

`comp.samp.idx` : The other two relative sample column index against which the profile is being determined. The default is 4 and 5

`f.profiles` : A vector of the desired direction of genomic change (or profile) corrosponding to each data type. The values are "up" or "down" for increased and decreased gene set profile, respectively

`cpt.data` : Identify changepoints for data using variance (cpt.var), mean (cpt.mean) or both (meanvar). Default is cpt.var.

`cpt.method` : Choice of single or multiple changepoint model. Default is "BinSeg".

`cpt.max` : The maximum number of changepoints to search for using "BinSeg" method. Default is 60. This number is dependent on the number of input data points

Value

The returned value is a data matrix including the original data along with between gene profile statistics and identified changepoints.

`propBarplot` *A plotting function for proportion by sample*

Description

Given a gene, this function will plot the proportion of each sample over the three samples within each data type

Usage

```
propBarplot(gispa.output,feature,cpt,input.cex,input.cex.lab,ft.col,strip.col)
```

Arguments

`gispa.output` : A data matrix of between gene feature profile statistics for each feature with corrosponding identified changepoints. The row names should corrospond to genes or names to be displayed on y-axis

`feature` : Analysis type i.e., one ('1'), two ('2') or three ('3') dimensional feature analysis.

`cpt` : Changepoints to be plotted.

`input.cex` : character (or symbol expansion) for the x- and y-axis labels

`input.cex.lab` : character (or symbol expansion) for the horizontal and vertial strip labels

`ft.col` : a vector of colors of the bar for the features or data types

`strip.col` : color to be used for the vertial strip

Details

This function expects the output from the main function of GISPA package, and highlights the gene set in the selected changepoints and their proportion in each of the three sample groups by data type.

Value

Barplot illustrating each sample proportion for each gene in the selected change point

Author(s)

Bhakti Dwivedi & Jeanne Kowalski

Examples

```
id <- 20 ## number of rows
s <- 3 ## number of columns
dm <- matrix(runif(id*s,0,10), nrow=id, ncol=s,
                 dimnames=list(paste("gene", 1:id, sep=""),
                 paste("fs", 1:s, sep="")))
changepoints <- sort(sample(1:2, id, replace=TRUE))
dm <- cbind(dm,changepoints)
propBarplot(gispa.output=dm,feature=2,cpt=1,
           input.cex=0.5,input.cex.lab=0.5,
           ft.col=c("grey0", "grey60"),strip.col="yellow")
```

---

stackedBarplot *A plotting function for each sample proportion*

Description

Given a gene, this function will plot the proportion of each sample over all the samples for each gene or row within each data type

Usage

```
stackedBarplot(gispa.output,feature,cpt,type,input.cex,input.cex.lab,input.gap,samp.col,strip.co
```

Arguments

gispa.output	: A data frame containing genes as rows followed by between gene feature profile statistics for each sample.
feature	: Analysis type i.e., one ('1d'), two ('2d') or three ('3d') dimensional feature analysis.
cpt	: Changepoint gene set to be plotted.
type	: Type of data, e.g., EXP (default) for expression, VAR of variants, CNV for copy number change.
input.cex	: character (or symbol expansion): x- and y-axis labels
input.cex.lab	: character (or symbol expansion) for the horizontal and vertial strip labels
input.gap	: gap or distance between each data type plot
samp.col	: a vector of colors of the bar for the three sample groups, reference followed by other two samples
strip.col	: color to be used for the vertial strip

Details

This function expects the output from GISPA function of GISPA package, and highlights the gene set in the selected changepoints and their proportion in each of the three sample groups.

Value

Barplot illustrating each sample proportion for each gene in the selected change point

Author(s)

Bhakti Dwivedi & Jeanne Kowalski

Examples

```
id <- 20 ## number of rows
s <- 4 ## number of columns
dm <- matrix(runif(id*s,min=0,max=100), nrow=id, ncol=s,
                 dimnames=list(paste("gene", 1:id, sep=""),
                 paste("sample", 1:s, sep="")))
changepoints <- sort(sample(1:2, id, replace=TRUE))
dm <- cbind(dm,changepoints)
stackedBarplot(gispa.output=dm, feature=1, cpt=1, type="EXP",
              input.cex=1.5, input.cex.lab=1.5, input.gap=0.5,
              samp.col=c("red", "green", "blue"), strip.col="yellow")
```

---

varset *Variant (VAR) data*

Description

A dataset containing the genome-wide gene variant proportion from 3 multiple myeloma cell line samples. The variables are as follows

Usage

```
varset
```

Format

A data matrix with 1101 genes variants and 3 samples

Details

- gene
- gene names
- sample 1 transformed variant proportion data
- sample 2 transformed variant proportion data
- sample 3 transformed variant proportion data

@source https://research.themmrf.org/

Index