

Package: IntegratedJM (via r-universe)

August 30, 2024

Type Package

Title Joint Modeling of the Gene-Expression and Bioassay Data, Taking Care of the Effect Due to a Fingerprint Feature

Version 1.6

Date 2017-08-03

Author Rudradev Sengupta, Nolen Joy Perualila

Maintainer Rudradev Sengupta <rudradev.sengupta@uhassel.t.be>

Description Offers modeling the association between gene-expression and bioassay data, taking care of the effect due to a fingerprint feature and helps with several plots to better understand the analysis.

Depends R (>= 3.0.0), grid

Imports ggplot2, nlme, Biobase

License GPL-3

LazyLoad yes

Repository CRAN

Repository/R-Forge/Project integratedjm

Repository/R-Forge/Revision 14

Repository/R-Forge/DateTimeStamp 2017-08-03 19:03:01

Date/Publication 2017-08-03 22:37:38 UTC

NeedsCompilation no

Contents

fitJM	2
getCorrUnad	3
IntegratedJM	4
multiplot	4
non_nullcov	5
nullcov	6

plot1gene	7
plotAsso	8
plotEff	8
sampleData	9
topkGenes	10
volcano	11

Index	13
--------------	-----------

fitJM	<i>fitJM</i>
-------	--------------

Description

The fitJM function fits the model for all the genes for a specific bio-activity vector and a particular fingerprint feature.

Usage

```
fitJM(dat, responseVector, covariate = NULL, methodMultTest)
```

Arguments

dat	Contains the gene expression data matrix for all the genes - can be a matrix or an expression set.
responseVector	Vector containing the bio-activity data.
covariate	Vector of 0's and 1's, containing data about the fingerprint feature.
methodMultTest	Character string to specify the multiple testing method. Default is the BH-FDR method.

Details

The default for the covariate parameter is NULL and if no covariate is specified it returns a data frame containing 5 variables, named as "Pearson", "Spearman", "p", "adj-p", "logratio" and the data frame is ordered based on the column "p" which is the p-value obtained from the Log-Ratio Test. If there is a covariate, then the output is a dataframe containing 13 variables for all the genes, named as "adjPearson", "adjSpearman", "pPearson", "Pearson", "Spearman", "pAdjR", "CovEffect1", "adjPeffect1", "CovEffect2", "adjPeffect2", "rawP1", "rawP2", "logratio" and sorted based on "rawP1" and "pPearson" which are p-value corresponding to the effect of the fingerprint feature on the gene expression data as obtained from the t-table after fitting the model using gls and the p-value obtained from the Log-Ratio Test, respectively. In the first case without any covariate it calls the nullcov function inside it, otherwise the non_nullcov function is called to do the analysis.

Value

A data frame, containing the results of the model, to be used later for plots or to identify the top genes.

Examples

```
## Not run:
jmRes <- fitJM(dat=gene_eset,responseVector=activity,methodMultTest='fdr')
jmRes <- fitJM(dat=gene_eset,responseVector=activity,covariate = fp,methodMultTest='fdr')

## End(Not run)
```

getCorrUnad	<i>getCorrUnad</i>
-------------	--------------------

Description

The getCorrUnad function is a support function for the function plot1gene.

Usage

```
getCorrUnad(geneName, fp, fpName, responseVector, dat, resPlot)
```

Arguments

geneName	Character string, specifying the name of the gene.
fp	Vector containing 0's and 1's - the data about the fingerprint feature.
fpName	Character string, used to make the title of the plots. If not specified, the plot title will be blank.
responseVector	Vector containing the bio-activity data.
dat	Contains the gene expression data matrix for all the genes - can be a matrix or an expression set.
resPlot	Logical. If TRUE, creates the plot data for the residual plot

Details

Works as a support function for plot1gene.

Value

A list containing the data to create the respective plots and the unadjusted association between the gene expression and bio-activity data.

Examples

```
## Not run:
getCorrUnad(geneName="Gene21", fp=fp, fpName="Fingerprint",
responseVector=activity, dat=gene_eset, resPlot=TRUE)

## End(Not run)
```

IntegratedJM	<i>Joint Modelling of Gene Expression and Bio-activity data, taking care of the effect due to Fingerprint Feature.</i>
--------------	--

Description

The IntegratedJM package contains the functions to fit the Joint Model, to classify the genes based on different criteria and necessary plot functions.

multiplot	<i>multiplot</i>
-----------	------------------

Description

The multiplot function plots multiple ggplots in the same window.

Usage

```
multiplot(..., cols = 1)
```

Arguments

...	ggplot2 objects, separated by comma.
cols	Integer, specifying the number of plots in one row in the layout.

Details

Plots multiple ggplots in the same window - `multiplot(p1,p2,p3,p4, cols=2)` is similar to the standard R notation `par(mfrow=c(2,2))`.

Value

Creates multiple ggplots in same window

Examples

```
## Not run:  
multiplot(p1,p2,p3,cols=3)  
  
## End(Not run)
```

non_nullcov	<i>non_nullcov</i>
-------------	--------------------

Description

The `non_nullcov` function is called while fitting the model when the covariate is specified in the `fitJM` function. It returns a `data.frame` containing the results after fitting the model. The output of this function is also the output of the `fitJM` function.

Usage

```
non_nullcov(dat, responseVector, covariate, methodMultTest, data_type)
```

Arguments

<code>dat</code>	Contains the gene expression data matrix for all the genes - can be a matrix or an expression set.
<code>responseVector</code>	Vector containing the bio-activity data.
<code>covariate</code>	Vector of 0's and 1's, containing data about the fingerprint feature.
<code>methodMultTest</code>	Character string to specify the multiple testing method.
<code>data_type</code>	Binary, specifying the type of the parameter <code>dat</code> : 0 - <code>expressionSet</code> , 1 - matrix.

Details

Fits the model, adjusting for the covariate effect, using `gls`, calculates the correlation, p-values, adjusted p-values (based on the multiple testing method) and `logratio` from LRT and returns the required results.

Value

A data frame, containing the results of the model - same as the output of the `fitJM` function.

Examples

```
## Not run:  
non_nullcov(dat=gene_eset, responseVector=activity, covariate=fp, methodMultTest='fdr', data_type=0)  
  
## End(Not run)
```

nullcov	<i>nullcov</i>
---------	----------------

Description

The nullcov function is called while fitting the model when there no covariate is specified in the fitJM function. It returns a data.frame containing the results after fitting the model. The output of this function is also the output of the fitJM function.

Usage

```
nullcov(dat, responseVector, methodMultTest, data_type)
```

Arguments

dat	Contains the gene expression data matrix for all the genes - can be a matrix or an expression set.
responseVector	Vector containing the bio-activity data.
methodMultTest	Character string to specify the multiple testing method. Default is the BH-FDR method.
data_type	Binary, specifying the type of the parameter dat: 0 - expressionSet, 1 - matrix.

Details

Fits the model using gls, calculates the correlation, p-values, adjusted p-values (based on the multiple testing method) and logratio from LRT and returns the required results.

Value

A data frame, containing the results of the model - same as the output of the fitJM function.

Examples

```
## Not run:  
nullcov(dat=gene_eset,responseVector=activity,methodMultTest='fdr',data_type=0)  
  
## End(Not run)
```

`plot1gene`*plot1gene*

Description

The `plot1gene` function plots the data for a single gene.

Usage

```
plot1gene(geneName, fp, fpName = "", responseVector, dat, resPlot = TRUE,  
          colP = "blue", colA = "white")
```

Arguments

<code>geneName</code>	Character string, specifying the name of the gene.
<code>fp</code>	Vector containing 0's and 1's - the data about the fingerprint feature.
<code>fpName</code>	Character string, used to make the title of the plots. If not specified, the plot title will be blank.
<code>responseVector</code>	Vector containing the bio-activity data.
<code>dat</code>	Contains the gene expression data matrix for all the genes - can be a matrix or an expression set.
<code>resPlot</code>	Logical. If TRUE, also plots the residual from the gls fit. Default is TRUE.
<code>colP</code>	Character string, specifying the colour for the 1's in the <code>fp</code> parameter. Default is blue.
<code>colA</code>	Character string, specifying the colour for the 0's in the <code>fp</code> parameter. Default is white.

Details

Calls the `getCorrUnad` function and creates the plot(s) accordingly.

Value

Creates a plot

Examples

```
## Not run:  
plot1gene(geneName="Gene21", fp=fp, fpName="Fingerprint", responseVector=activity, dat=gene_eset)  
## End(Not run)
```

plotAsso

plotAsso

Description

The plotAsso function is used to plot the unadjusted association vs the adjusted association for all the genes.

Usage

```
plotAsso(jointModelResult, type)
```

Arguments

jointModelResult
Data frame, containing the results from the fitJM function.

type
Character string, specifying the type of association - Pearson or Spearman.

Details

Plots the unadjusted association vs the adjusted association for all the genes.

Value

Creates a plot

Examples

```
## Not run:  
plotAsso(jointModelResult=jmRes, type="Pearson")  
  
## End(Not run)
```

plotEff

plotEff

Description

The plotEff function is used to plot the fingerprint effect on gene expression vs the adjusted association for all the genes.

Usage

```
plotEff(jointModelResult, type)
```


Arguments

`jointModelResult`
Data frame, containing the results from the fitJM function.

`type`
Character string, specifying the type of association - Pearson or Spearman.

Details

Plots the fingerprint effect on gene expression vs the specified type of adjusted association for all the genes.

Value

Creates a plot

Examples

```
## Not run:  
plotEff(jointModelResult=jmRes,type="Pearson")  
  
## End(Not run)
```

sampleData

Sample Data Example

Description

sample_data; Sample data included in the package. Gene expression data for 500 genes and 20 compounds and data on bio-activity and fingerprint feature.

Usage

```
data(sampleData)
```

Format

The format is: List containing one gene expression data matrix, one vector each on bio-activity and fingerprint feature data, respectively.

Examples

```
data(sampleData)  
gene_mx <- sample_data[[1]]  
activity <- sample_data[[2]]  
fp <- sample_data[[3]]
```

topkGenes

topkGenes

Description

The topkGenes function is to identify the top genes based on different criteria.

Usage

```
topkGenes(jointModelResult, subset_type, ranking, k = 10, sigLevel = 0.01)
```

Arguments

jointModelResult	Data frame, containing the results from the fitJM function.
subset_type	Character string to specify the set of genes. It can have four values: "Effect" for only differentially expressed genes, "Correlation" for only correlated genes, "Effect and Correlation" for genes which are both differentially expressed & correlated and "Other" for the genes which are neither differentially expressed nor correlated.
ranking	Character string, specifying one of the columns of the jointModelResult data frame, based on the genes will be ranked within the selected subset.
k	Integer, specifying the number of genes, to be returned from the list of top genes. Default is 10.
sigLevel	Numeric between 0 and 1, specifying the level of significance, used to select the subset of genes.

Details

Returned data frame contains 6 columns, named as "Genes", "FP-Effect", "p-adj(Effect)", "Un-adj.Asso.", "Adj.Asso.", "p-adj(Adj.Asso.)".

Value

A data frame containing top k genes according to the specified criteria from the specified set of genes.

Examples

```
## Not run:
topkGenes(jointModelResult=jmRes, subset_type="Effect", ranking="Pearson", k=10, sigLevel = 0.05)

## End(Not run)
```

<code>volcano</code>	<i>volcano</i>
----------------------	----------------

Description

The volcano function produces the volcano plot for logratio / fp-effect vs corresponding p-values.

Usage

```
volcano(x, pValue, pointLabels, topPValues = 10, topXvalues = 10,
        smoothScatter = TRUE, xlab = NULL, ylab = NULL, main = NULL,
        newpage = TRUE, additionalPointsToLabel = NULL,
        additionalLabelColor = "red", dir = TRUE)
```

Arguments

<code>x</code>	Numeric vector of logratios or covariate effect values to be plotted.
<code>pValue</code>	Numeric vector of corresponding p-values obtained from some statistical test.
<code>pointLabels</code>	Character vector providing the texts for the points to be labelled in the plot.
<code>topPValues</code>	Number of top p-values to be labelled. Default value is 10.
<code>topXvalues</code>	Number of top logratios or covariate effect values to be labelled. Default value is 10.
<code>smoothScatter</code>	Logical parameter to decide if a smooth plot is expected or not. Default is TRUE.
<code>xlab</code>	Text for the x-axis of the plot. Default is NULL.
<code>ylab</code>	Text for the y-axis of the plot. Default is NULL.
<code>main</code>	Text for the main title of the plot. Default is NULL.
<code>newpage</code>	Logical parameter
<code>additionalPointsToLabel</code>	Set of points other than the top values to be labelled in the plot. Default is NULL.
<code>additionalLabelColor</code>	Colour of the additionally labelled points. Default colour is red.
<code>dir</code>	Logical parameter deciding if the top values should be in decreasing (= TRUE) or increasing (= FALSE) order. Default is TRUE.

Details

Creates a plot which looks like a volcano with the interesting points labelled within the plot.

Value

A plot which looks like a volcano.

Examples

```
## Not run:  
volcano(x=jmRes$CovEffect1,pValue=jmRes$rawP1,pointLabels=rownames(jmRes),  
topPValues = 10, topXvalues = 10,xlab="FP Effect (alpha)",ylab="-log(p-values)")  
  
## End(Not run)
```

Index

* datasets

sampleData, 9

fitJM, 2

getCorrUnad, 3

IntegratedJM, 4

IntegratedJM-package (IntegratedJM), 4

multiplot, 4

non_nullcov, 5

nullcov, 6

plot1gene, 7

plotAsso, 8

plotEff, 8

sample_data(sampleData), 9

sampleData, 9

topkGenes, 10

volcano, 11