Package ‘sdef’

July 6, 2015

Type Package
Title Synthesizing List of Differentially Expressed Features
Version 1.6
Date 2015-07-03
Author Alberto Cassese, Marta Blangiardo
Maintainer Alberto Cassese <alberto.cassese@maastrichtuniversity.nl>
Description Performs two tests to evaluate if the experiments are associated and returns a list of interesting features common to all the experiments.
License GPL-2
Imports graphics,grDevices,stats,utils
NeedsCompilation no
Repository CRAN
Date/Publication 2015-07-06 18:29:16

R topics documented:

sdef-package
baymod
createTable
designCount
designMatrix
Example3Lists
extractFeatures.R
extractFeatures.T
Liver.Muscle
ratio
simulation
simulation.indep
Tmc

Index 17
sdef-package

Synthesizing List of Differentially Expressed Features

Description

The package performs two tests to evaluate if the experiments are associated and returns a list of interesting features common to all the experiments.

Details

Package: sdef
Type: Package
Version: 1.4
Date: 2010-03-10
License: GPL-2
Lazyload: yes

Author(s)

Alberto Cassese, Marta Blangiardo
Maintainer: Alberto Cassese <fratucster@gmail.com>

baymod

Bayesian model for the ratio of observed to expected probability of features to be in common

Description

The function specifies a Bayesian model for the ratio of observed to expected probability of features to be in common. A multinomial distribution is specified on the probabilities of being significant in any combination of the experiments (e.g. if two experiments are considered, the probability of being significant in none, one and two experiments is specified) and a prior distribution is put on their parameters. The quantity of interest is the ratio of the probability that a feature is in common, to the probability that a feature is in common by chance, called R(h).

Usage

baymod(output.ratio, iter = 1000, dir = getwd(), conf = 95)
**Arguments**

- `output.ratio`: The output object from the ratio function.
- `iter`: Number of iterations to be performed.
- `dir`: Directory for storing the plots.
- `conf`: Size of Confidence Interval.

**Details**

It returns an object of class list with the ratio $R(h)$ for each threshold and its quantiles specified by `conf`. $R(h)$ is significant if its CI does not include 1. We consider two rules for selecting the list of genes of interest: 1) $h_{max}$ is the maximum of Median($R(h)$) only for the subset of credibility intervals which do not include 1; 2) $h_2$ is the largest threshold where the number of features called in common at least doubles the number of features in common under independence (where $R(h)$ larger than 2).

The function returns also a plot of the credibility interval for each threshold. The same plot is also saved in the directory specified by the user.

**Value**

A matrix and a plot with the quantiles of $R(h)$ identified by `conf` for each p-value threshold.

**Author(s)**

Alberto Cassese, Marta Blangiardo

**References**

M. Blangiardo and S. Richardson (2007) Statistical tools for synthesizing lists of differentially expressed features in related experiments, Genome Biology, 8, R54.

**Examples**

```r
data = simulation(n=500,GammaA=1,GammaB=1,r1=0.5,r2=0.8,DEfirst=300, DEsecond=200,DEcommon=100)
Th<- ratio(data=data$Pval)
Rh<- baymod(iter=100,output.ratio=Th)
```

---

**createTable**

*Function to create an output table*

**Description**

This function reports the results from the Frequentist and Bayesian model for $h_{max}$ and for $h_2$. It also creates an output table with the results for all the thresholds in a csv format, so the user can select additional thresholds of interest.
createTable

Usage
createTable(output.ratio, output.bay, dir = getwd(), h=NULL)

Arguments
- output.ratio: The output object from the Frequentist model (ratio function)
- output.bay: The output object from the Bayesian model (baymod function)
- dir: Directory for storing the table
- h: Additional thresholds in the form of a vector

Details
To select a list of interesting features from the Bayesian model we suggest two decision rules in the paper: 1) the maximum of Median(R(h)) only for the subset of credibility intervals which do not include 1; 2) the largest threshold h for which the ratio R(h) is bigger than 2.

The first one is pointing out the strongest deviation from independence, whilst the second is the largest threshold where the number of features called in common at least doubles the number of features in common under independence.

Value
- max: The results of the R(hmax) statistic
- rule2: The results using the rule R(h) larger than 2 (see details)
- ruleh: The results using additional thresholds

Author(s)
Alberto Cassese, Marta Blangiardo

References
1. M.Blangiardo and S.Richardson (2007) Statistical tools for synthesizing lists of differentially expressed features in related experiments , Genome Biology, 8, R54

Examples
```r
data <- simulation(n=500,GammaA=1,GammaB=1,r1=0.5,r2=0.8, DEfirst=300,DEsecond=200,DEcommon=100)
Th <- ratio(data=data$Pval)
Rh <- baymod(iter=100,output.ratio=Th)
output.table <- createTable(output.ratio=Th, output.bay=Rh)
```
designCount  

**Internal function**

**Description**

This is an internal function. The user should not use the function directly.

**Usage**

```r
designCount(array, design)
```

**Arguments**

- `array`  
  Data matrix
- `design`  
  Design matrix

**Author(s)**

Alberto Cassese, Marta Blangiardo

**Examples**

```r
## The function is currently defined as
function(array, design) {
  sum1intersects <- function(c1,c2) return(all(c1 == c2))
  res <- vector(mode="numeric",length=(nrow(design)-1))
  for(i in 2:nrow(design)){
    res[i-1] <- sum(apply(array,1,sum1intersects,design[i,]))
  }
  return(res)
}
```

designMatrix  

**Internal function**

**Description**

This is an internal function. The user should not use the function directly.

**Usage**

```r
designMatrix(lists)
```
Arguments

lists The number of lists to be compared

Author(s)

Alberto Cassese, Marta Blangiardo

Examples

```r
## The function is currently defined as
function(lists){
 rows = 2^(lists)
ncycles = rows
 x = matrix(0,rows,lists)
 for (k in 1:lists){
   settings = c(0,1)
nycles = ncycles/2
 nreps = rows/(2*ncycles)
 settings = matrix(rep(settings,nreps),nreps,
       length(settings),byrow=TRUE)
 settings = as.vector(settings)
   #impila in un vettore settings, una colonna sotto l'altra
 settings = matrix(rep(settings,ncycles),
       length(settings),ncycles)
 x[,lists-k+1] = as.vector(settings)
 }
 return(x)
}
```

Example3Lists  Molecular Differences between Mammalian Sexes.

Description

This dataset contains three lists of p-values obtained from a publicly available experiment to evaluate differential expression between mammalian sexes in three tissues (hypothalamus, kidney and liver).

Usage

```r
data(Example3Lists)
```

Format

The format is: a matrix with 6477 rows and 3 columns. For each gene (row) it reports the p-values of being differentially expressed between male and female mice for the three tissues.
extractFeatures.R

**Source**


**References**


**Examples**

```r
#data(Example3Lists)
#Th<- ratio(data=Example3Lists)

#Rh<- baymod(iter=100,output.ratio=Th)

#MC<- Tmc(iter=100,output.ratio=Th)

#The gene names can be obtained using the command dimnames:
#feat.names = dimnames(Example3Lists)[[1]]
#feat.lists <- extractFeatures.R(output.ratio=Th,output.bay=Rh,feat.names=feat.names,h=NULL)
#feat.lists.T <- extractFeatures.T(output.ratio=Th,feat.names=feat.names)

#output.table <- createTable(output.ratio=Th,output.bay=Rh)
```

---

**extractFeatures.R Extracting the lists of features of interest**

**Description**

The function returns the list of features in common using the two suggested rules hmax and h2 (Bayesian model) and additional ones defined by the user.

**Usage**

```r
extractFeatures.R(output.ratio, output.bay, feat.names, h = NULL)
```

**Arguments**

- `output.ratio` The output object from the Frequentist model (ratio function)
- `output.bay` The output object from the Bayesian model (baymod function)
- `feat.names` Names of the features (e.g Affy ID for genes)
- `h` Additional thresholds in the form of a vector to select a list of features in common. I
Details

To select a list of interesting features from the Bayesian model we suggest two decision rules in the paper: 1) the maximum of Median(R(h)) only for the subset of credibility intervals which do not include 1; 2) the largest threshold h for which the ratio R(h) is bigger than 2.

The first one is pointing out the strongest deviation from independence, whilst the second is the largest threshold where the number of features called in common at least doubles the number of features in common under independence. The user can define additional thresholds of interest and obtain the list of associated features.

Value

The function returns an object of the class list. Each element is a matrix where the first column contains the name of the features while the other columns contain the p-values* from the experiments. It also saves a .csv file with the same information.

* instead of the p-values any other measure used to rank the features in the experiments can be used

\[ \text{max} \] The list of features of interest selected on the basis of the threshold associated to R(h_{\text{max}})

\[ \text{rule2} \] The list of features of interest selected on the basis of the threshold associated to R(h_{2})

\[ \text{User} \] The list of features of interest selected on the basis of the additional thresholds selected by the user

Author(s)

Alberto Cassese, Marta Blangiardo

References

1. M.Blangiardo and S.Richardson (2007) Statistical tools for synthesizing lists of differentially expressed features in related experiments, Genome Biology, 8, R54

Examples

data = simulation(n=500,GammaA=1,GammaB=1,r1=0.5,r2=0.8, DEFfirst=300,DEsecond=200,DEcommon=100)
Th<- ratio(data=data$Pval)
Rh<- baymod(iter=100,output.ratio=Th)
feat.names = data$names
feat.lists <- extractFeatures.R(output.ratio=Th,output.bay=Rh, feat.names=feat.names,h=NULL)
extractFeatures.T  

### Extracting the lists of features of interest

**Description**

The function returns the list of features in common using the hmax rule (Frequentist model).

**Usage**

```r
extractFeatures.T(output.ratio, feat.names)
```

**Arguments**

- `output.ratio`: The output object from the Frequentist model (ratio function)
- `feat.names`: names of the features (e.g. Affy ID for genes)

**Details**

To select a list of interesting features from the frequentist model we suggest a decision rules in the paper: the maximum of $T(h) = \text{nb genes in common}/\text{nb genes in common}$ under the hypothesis of independence. It is pointing out the strongest deviation from independence.

**Value**

The function returns an object of the class list. Each element is a matrix where the first column contains the name of the features while the other columns contain the p-values* from the experiments. It also saves a .csv file with the same information.

*instead of the p-values any other measure used to rank the features in the experiments can be used.

- `max`: The list of features in common selected on the basis of the threshold associated to $T(h_{\text{max}})$

**Author(s)**

Alberto Cassese, Marta Blangiardo

**References**

1. M.Blangiardo and S.Richardson (2007) Statistical tools for synthesizing lists of differentially expressed features in related experiments, Genome Biology, 8, R54
Examples

data = simulation(n=500, GammaA=1, GammaB=1, r1=0.5, r2=0.8, DEfirst=300, DEsecond=200, DEcommon=100)
Th<- ratio(data=data$Pval)
feat.names = data$names
feat.lists.T <- extractFeatures.T(output.ratio=Th,
    feat.names=feat.names)

Liver.Muscle  

Diabetes susceptibility in liver and skeletal muscle of mice.

Description

This dataset contains two lists of p-values obtained from a publicly available experiment to evaluate differential expression between diabetes susceptibility in liver and skeletal muscle of obese and normal mice.

Usage

data(Liver.Muscle)

Format

The format is: a matrix with 2912 rows and two columns. For each gene (row) it reports the p-values of being differentially expressed between obese and normal mice for the two tissues (columns).

Source


References


Examples

#data(Liver.Muscle)
#Th<- ratio(data=Liver.Muscle)

#Rh<- baymod(iter=100, output.ratio=Th)

#MC<- Tmc(iter=100, output.ratio=Th)

#The gene names can be obtained using the command dimnames:
#feat.names = dimnames(Liver.Muscle)[[1]]
#feat.lists <- extractFeatures.R(output.ratio=Th, output.bay=Rh, feat.names=feat.names, h=NULL)
ratio

#feat.lists.T <- extractFeatures.T(output.ratio=Th,feat.names=feat.names)

#output.table <- createTable(output.ratio=Th,output.bay=Rh)

---

**ratio**

Ratio $T(h)$ between the observed features in common and the expected ones

**Description**

The function for each experiment calculates the ratio $T(h)$ for each threshold $h$, using the list of p-values or the other measure used in the experiment to rank the features, (e.g. posterior probability, correlation).

**Usage**

```r
ratio(data, pvalue = TRUE, interval = 0.01,
name = "Distribution of T(h)", dir = getwd(),
datename = "dataratio")
```

**Arguments**

- `data` Lists of pvalues to be compared
- `pvalue` Indicate if the data are pvalues (TRUE) or posterior probability (FALSE). If they are posterior probability they are transformed in pvalues
- `interval` The interval between two threshold
- `name` The name to be used in the plots
- `dir` Directory for storing the plots
- `dataname` The name of the file containing the data (Pvalue)

**Details**

This function calculates the ratio $T(h)$ of observed number of features in common between the lists vs the expected number under the hypothesis of independence for each threshold $h$. The expected numbers are calculated as the product among the marginals divided by $(\text{number of features})^{\text{number of lists} - 1}$. $T(h_{max})$ identifies the maximum of the statistic $T(h)$ and it is shown on the plot.

**Value**

Returns a plot with the distribution of $T(h)$ showing where $T(h_{max})$ and $h_{max}$ are located. The same plot is also saved in the directory specified by the user. It returns also an object of class list with the ratio, the thresholds and other attributes. In particular:

- `h` Threshold corresponding to $T(h)$ values
- `DE` Differentially expressed features in each experiment
- `ratios` Vector or $T(h)$ values for each threshold
**Common** Features in common corresponding to the T(h) values
**interval** Interval on the p-value scale defined by the user (default is 0.01)
**name** Names to be used in the plots (defined by the user)
**pvalue** Logical: TRUE if the measures used for the analysis are p-value, FALSE if they are posterior probabilities
**dataname** The name of the file where the data has been saved

**Author(s)**
Alberto Cassese, Marta Blangiardo

**References**
M.Blangiardo and S.Richardson (2007) Statistical tools for synthesizing lists of differentially expressed features in related experiments, Genome Biology, 8, R54.

**Examples**
```r
data = simulation(n=500, GammaA=1, GammaB=1, r1=0.5, r2=0.8, DEfirst=300, DEsecond=200, DEcommon=100)
Th<- ratio(data=data$Pval)
```

---

**simulation**

*Simulate p-values for two related experiments*

**Description**
The function simulates two vectors of p-values using the procedure described in Hwang et al.

**Usage**
simulation(n, GammaA, GammaB, epsilonM = 0, epsilonSD = 1, r1, r2, DEfirst, DEsecond, DEcommon)

**Arguments**
- `n` Number of features to simulate
- `GammaA` Parameter of the Gamma distribution
- `GammaB` Parameter of the Gamma distribution
- `epsilonM` Parameter of the Gaussian noise specific to the genes and experiment
- `epsilonSD` Parameter of the Gaussian noise specific to the genes and experiment
- `r1` Additional experiment-specific noise
Details
Considering two experiments ($k=1,2$), each of them with two classes, and $n$ genes, for each gene we simulate a true difference between the classes $\delta(g)$, drawn from a Gamma distribution with random sign. The true difference $\delta(g)$ is 0 if the gene is not differentially expressed. We then add two normal random noise components, $r(k)$ that act as experiment specific components and $\epsilon(gk)$, that are the gene-experiment components. The former is assigned deterministically, whilst the latter is drawn from a standard Gaussian distribution. The log fold change ($\text{FC}(gk)$) is the sum of all these components for each gene and experiment. We assign the $n$ genes to four groups: genes differentially expressed (DE) in both experiments, genes differentially expressed only in the first experiment, genes differentially expressed only in the second experiment and genes differentially expressed in neither experiment. When the genes are differentially expressed in both experiments, they share the same $\delta(g)$ and the only difference between them is given by the random components: $\text{FC}(g1) = \delta(g) + r(1) \times \epsilon(g1)$ $\text{FC}(g2) = \delta(g) + r(2) \times \epsilon(g2)$ This group represents the true positive genes (i.e. truly DE in both experiments) that we are interested in finding using our method. The two groups of genes differentially expressed only in one of the two experiments act like additional noise and make the simulation more realistic.

Then, as described in Hwang et al., a two tails T-test is performed for each $\text{FC}(gk)$ and a p-value is generated as: $P(gk) = 2 \text{Normal cdf}(-\text{absolute value}(\text{FC}(gk)/r(k)))$.

Value

<table>
<thead>
<tr>
<th>names</th>
<th>Which group each simulated gene expression value belongs to</th>
</tr>
</thead>
<tbody>
<tr>
<td>FC1</td>
<td>T statistic for the first experiment</td>
</tr>
<tr>
<td>FC2</td>
<td>T statistic for the second experiment</td>
</tr>
<tr>
<td>Pval</td>
<td>p-value for the experiments to be compared</td>
</tr>
</tbody>
</table>

Author(s)
Alberto Cassese, Marta Blangiardo

References

M.Blangiardo and S.Richardson (2007) Statistical tools for synthesizing lists of differentially expressed features in related experiments, Genome Biology, 8, R54.
Examples

```r
data = simulation(n=500, GammaA=1, GammaB=1, r1=0.5, r2=0.8, DEfirst=300, DEsecond=200, DEcommon=100)
```

Description

The function simulates two vectors of p-values using the procedure described in Hwang et al. for independent experiments.

Usage

```r
simulation.indep(n, GammaA = 2, GammaB = 2, epsilonM = 0, epsilonSD = 1, r1, r2, DEfirst, DEsecond)
```

Arguments

- `n`: Number of features to be simulated.
- `GammaA`: Parameter of the Gamma distribution.
- `GammaB`: Parameter of the Gamma distribution.
- `epsilonM`: Parameter of the Gaussian noise.
- `epsilonSD`: Parameter of the Gaussian noise.
- `r1`: Additional experiment-specific noise.
- `r2`: Additional experiment-specific noise.
- `DEfirst`: Number of DE features in the first experiment.
- `DEsecond`: Number of DE features in the second experiment.

Details

Considering two experiments (k=1,2), each of them with two classes, and n genes, for each gene we simulate a true difference between the classes `delta(g)`, drawn from a Gamma distribution with random sign. The true difference `delta(g)` is 0 if the gene is not differentially expressed. We then add two normal random noise components, `r[k]` that act as experiment specific components and `epsilon(gk)`, that is the gene-experiment components. The former is assigned deterministically, whilst the latter is drawn from a standard Gaussian distribution. So the log fold change (`FC(gk)`) is the sum of all these components for each gene and experiment. We divide the n genes in three groups: genes differentially expressed only in the first experiment, genes differentially expressed only in the second experiment and genes differentially expressed in neither experiment. There are not true positive genes (i.e. truly DE in both experiments), so we should find no genes in common using our method.

Then, as described in Hwang et al., a two tails T-test is performed for each `FC(gk)` and a p-value is generated as: \( P(gk) = 2 \cdot \text{Normal cdf}(-\text{absolute value (FC(gk)/r(k)}) \) where FC(gk) is the t statistic that evaluates the differential expression between the two classes for the g gene and k experiment.
**Value**

- names: Which group each simulated gene expression value belongs to
- FC1: T statistic for the first experiment
- FC2: T statistic for the second experiment
- pval: p-values for the experiment to be compared

**Author(s)**

Alberto Cassese, Marta Blangiardo

**References**


M.Blangiardo and S.Richardson (2007) Statistical tools for synthesizing lists of differentially expressed features in related experiments, Genome Biology, 8, R54.

**Examples**

```r
data.indep = simulation.indep(n=500,GammaA=1,
GammaB=1,r1=0.5,r2=0.8,DEfirst=300,DEsecond=200)
```

---

**Tmc**

*Empirical null distribution of max T(h)*

**Description**

The function uses Monte Carlo permutations to calculate the empirical distribution of max T(h)=T(hmax) under the null hypothesis of independence among the experiments. An empirical p-value is calculated to evaluate where T(hmax) is located under the null distribution.

**Usage**

```r
Tmc(iter = 1000, output.ratio)
```

**Arguments**

- **iter**: Number of iteration to be performed
- **output.ratio**: The output object from the ratio function
Details

This function uses Monte Carlo permutations to calculate the empirical distribution of the maximum of $T(h)$ (i.e. $T(h_{\text{max}})$) under the null hypothesis of independence among the experiments. While the p-values* for the first list are fixed, the ones for the other lists are independently permuted $B$ times. In this way, any relationship among the lists is destroyed. At each permutation $b$ (b varies from 1 to $B$) a $T_b(h)$ is calculated for each $h$ and a maximum statistic $T_b(h_{\text{max}})$ is returned; its distribution represents the null distribution of $T(h_{\text{max}})$ under the condition of independence. The relative frequency of $T_b(h_{\text{max}})$ larger than $T(h_{\text{max}})$ identifies the p-value: it returns the proportion of $T_b(h_{\text{max}})$ from permuted dataset greater than the observed one (so indicates where the observed $T(h_{\text{max}})$ is located under the null distribution).

* instead of the p-values any other measure used to rank the features in the experiments can be used

Value

Returns the empirical p-value from testing $T(h_{\text{max}})$ and a plot of the $T_b(h_{\text{max}})$ distribution. The same plot is also saved in the directory specified by the user.

Author(s)

Alberto Cassese, Marta Blangiardo

References

Stone et al. (1988), Investigations of excess environmental risks around putative sources: statistical problems and a proposed test, Statistics in Medicine, 7, 649-660.

M. Blangiardo and S. Richardson (2007) Statistical tools for synthesizing lists of differentially expressed features in related experiments, Genome Biology, 8, R54.

Examples

data = simulation(n=500, GammaA=1, GammaB=1, r1=0.5, r2=0.8, DEfirst=300, DEsecond=200, DEmcommon=100)
Th <- ratio(data=data$Pval)
MC <- Tmc(iter=50, output.ratio=Th)
Index

baymod, 2
createTable, 3
designCount, 5
designMatrix, 5
Example3Lists, 6
extractFeatures.R, 7
extractFeatures.T, 9
Liver.Muscle, 10
ratio, 11
sdef (sdef-package), 2
sdef-package, 2
simulation, 12
simulation.indep, 14
Tmc, 15