Package ‘GSReg’

May 15, 2024

Version 1.39.0

Date 2016-11-28

Title Gene Set Regulation (GS-Reg)

Author Bahman Afsari <bahman@jhu.edu>, Elana J. Fertig
<ejfertig@jhmi.edu>

Maintainer Bahman Afsari <bahman@jhu.edu>, Elana J. Fertig
<ejfertig@jhmi.edu>

Depends R (>= 2.13.1), Homo.sapiens, org.Hs.eg.db, GenomicFeatures, AnnotationDbi

Suggests GenomicRanges, GSBenchMark

Description A package for gene set analysis based on the variability of expressions as well as a method to detect Alternative Splicing Events . It implements DIfferential RAnk Conservation (DIRAC) and gene set Expression Variation Analysis (EVA) methods. For detecting Differentially Spliced genes, it provides an implementation of the Spliced-EVA (SEVA).

License GPL-2

biocViews GeneRegulation, Pathways, GeneExpression, GeneticVariability, GeneSetEnrichment, AlternativeSplicing

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

git_branch devel

git_last_commit 5f15cec

git_last_commit_date 2024-04-30

Repository Bioconductor 3.20

Date/Publication 2024-05-15

Contents

GSReg-package 2
GSReg.GeneSets.DIRAC 3

GSReg.GeneSets.EVA 4
GSReg.kendall.tau.distance 6
GSReg.overlapJunction 8
GSReg.SEVA 9

Index **12**

`GSReg-package` *A package for Gene Set Analysis based on the variability of gene expression in different phenotypes.*

Description

The GSReg package applies the analysis of variety among phenotypes for each gene set from gene expression as well as alternative splicing from junction expression. The user can also use Differential Rank Conservation (DIRAC) (Eddy et al. 2010) and a modified version which allows for efficient and easy p value calculation. Both DIRAC and its modified version are rank-based methods, i.e. they only consider the ordering of the expressions within the pathway.

GSReg package features

The package contains several utilities enabling to:

A) Prune Gene Sets based on the available genes in the expression data;

B) Calculate the DIRAC measure and p-value for it based on permutation test;

C) Calculate for a modified DIRAC method and a fast-efficient p-valuebased on U-Statistic theory;

D) Alternative Splicing Events in genes from a phenotype to other phenotype using their junction expression;

Author(s)

Bahman Afsari <bahman.afsari@gmail.com>, Elana J. Fertig <ejfertig@jhmi.edu>

Source

http://www.ncbi.nlm.nih.gov/pubmed/20523739

References

Eddy et al., "Identifying tightly regulated and variably expressed networks by Differential Rank Conservation (DIRAC).", *PLoS Comp. Bio.*, 2010, **6**(5)

`GSReg.GeneSets.DIRAC` *Performs DIRAC for gene set analysis from the paper Eddy et al (2010).*

Description

GSReg.GeneSets.DIRAC performs DIRAC for gene set analysis from the paper Eddy et al (2010). In fact, the Null hypothesis is that the conservation index is not significantly different under two phenotypes. The function calculates the p-value using permutation test; hence, extremely low p-value cannot be reached.

Usage

```
GSReg.GeneSets.DIRAC(geneexpres, pathways, phenotypes, Nperm = 0, alpha = 0.05, minGeneNum=5)
```

Arguments

`geneexpres`	the matrix of gene expressions. The rownames must represent gene names and the columns represent samples. There must not be any missing values. Please use imputation or remove the genes with missing values.
`pathways`	a list containing pathway information. Each element represents a pathway as a character vector. The genes shown in the pathway must be present in geneexpres. Please prune the genes in pathways using GSReg.Prune before applying this function.
`phenotypes`	a binary factor containing the phenotypes for samples in geneexpres; hence, the column number of geneexpres and the length of phenotypes must be equal.
`Nperm`	The number of permutation tests required for p-value calculation. If Nperm==0 then the function reports a normal approximation of the p-value.
`alpha`	A parameter smoothes the template estimate. The template corresponding to a comparison is 1, if the probability of the comparison of the two genes is bigger than 0.5 +alpha/n (where n is the number of samples in the phenotype), and otherwise zero.
`minGeneNum`	the minimum number of genes required in a pathway.

Value

IThe output is a list with three elements. Each element of the output list is a vector are named according to the pathway.

`$mu1`	a vector containing the variability in DIRAC sense (1- conservation indices in Eddy et al (2010) paper) for all pathways in phenotype == levels(phenotypes)[1].
`$mu2`	a vector containing the variability in DIRAC sense (1- conservation indices in Eddy et al (2010) paper) for all pathways in phenotype == levels(phenotypes)[2].
`$pvalues`	a vector containing p-values for each pathway. Low p-values means that the gene expressions have different orderings under different phenotypes.

Author(s)

Bahman Afsari

References

Eddy, James A., et al. "Identifying tightly regulated and variably expressed networks by Differential Rank Conservation (DIRAC)." PLoS computational biology 6.5 (2010): e1000792.

See Also

GSReg.GeneSet.VReg

Examples

```
library(GSBenchMark)
### loading and pruning the pathways
data(diracpathways)
### loading the data
data(leukemia_GSEA)


### extracting gene names
genenames = rownames(exprsdata);


### DIRAC analysis
DIRAna = GSReg.GeneSets.DIRAC(pathways=diracpathways,geneexpres=exprsdata,Nperm=0,phenotypes=phenotypes)
dysregulatedpathways = rbind(DIRAna$mu1[which(DIRAna$pvalues<0.05)],
DIRAna$mu2[which(DIRAna$pvalues<0.05)],DIRAna$pvalues[which(DIRAna$pvalues<0.05)]);
rownames(dysregulatedpathways)<-c("mu1","mu2","pvalues");
print(dysregulatedpathways[,1:5])
plot(x=dysregulatedpathways["mu1",],y=dysregulatedpathways["mu2",],
xlim=range(dysregulatedpathways[1:2,]),ylim=range(dysregulatedpathways[1:2,]))
lines(x=c(min(dysregulatedpathways[1:2,]),max(dysregulatedpathways[1:2,])),
y=c(min(dysregulatedpathways[1:2,]),max(dysregulatedpathways[1:2,])),type="l")
```

GSReg.GeneSets.EVA — *Performs Gene Set Analysis using Expression Variation Analysis (EVA).*

Description

GSReg.GeneSets.EVA performs modified version DIRAC papers. Using a theoretical analysis, we can calculate p-value which makes extreme low p-values available.

Usage

```
GSReg.GeneSets.EVA(geneexpres, pathways, phenotypes,minGeneNum=5)
```

Arguments

`geneexpres`	the matrix of gene expressions. The rownames must represent gene names and the columns represent samples. There must not be any missing values. Please use imputation or remove the genes with missing values.
`pathways`	a list containing pathway information. Each element represents a pathway as a character vector. The genes shown in the pathway must be present in geneexpres. geneexpres must have numeric and finite numbers.
`phenotypes`	a binary factor containing the phenotypes for samples in geneexpres; hence, the column number of geneexpres and the length of phenotypes must be equal.
`minGeneNum`	the minimum number of genes required in a pathway.

Value

a list of analysis for all pathways.

`$E1`	the modified variance on the pathway within the samples from levels(phenotypes)[1].
`$E2`	the modified variance on the pathway within the samples from levels(phenotypes)[2].
`$E12`	the modified variance on the pathway across the samples from levels(phenotypes)[1] tolevels(phenotypes)[2].
`$VarEta1`	the estimation of the modified variance on the pathway within the samples from levels(phenotypes)[1].
`$VarEta2`	the estimation of the modified variance on the pathway within the samples from levels(phenotypes)[2].
`$zscore`	zscore for the modified variance.
`$pvalue`	theoretical p-value for null E1 = E2. (Standard EVA).
`$pvalueD12D1`	theoretical p-value for null E1 = E12.
`$pvalueD12D2`	theoretical p-value for null E2 = E12.
`$pvalueTotal`	Bonferonni corrected p-value of the three p-values.

Author(s)

Bahman Afsari

See Also

GSReg.GeneSets.DIRAC,cor

Examples

```
### loading and pruning the pathways
library(GSBenchMark)
data(diracpathways)
### loading the data
data(leukemia_GSEA)
```

```
### removing genes which contain not a number.
if(sum(apply(is.nan(exprsdata),1,sum)>0))
  exprsdata = exprsdata[-which(apply(is.nan(exprsdata),1,sum)>0),];

### extracting gene names
genenames = rownames(exprsdata);

### DIRAC analysis
VarAnKendallV = GSReg.GeneSets.EVA(geneexpres=exprsdata,
 pathways=diracpathways, phenotypes=as.factor(phenotypes))
E1 = sapply(VarAnKendallV,function(x) x$E1);
E2 = sapply(VarAnKendallV,function(x) x$E2);
Kpvalues = sapply(VarAnKendallV,function(x) x$pvalue);

dysregulatedpathways = rbind(E1[which(Kpvalues<0.05)],
E2[which(Kpvalues<0.05)],Kpvalues[which(Kpvalues<0.05)]);
rownames(dysregulatedpathways)<-c("E1","E2","pvalues");
print(dysregulatedpathways)
plot(x=dysregulatedpathways["E1",],y=dysregulatedpathways["E2",],
xlim=range(dysregulatedpathways[1:2,]),ylim=range(dysregulatedpathways[1:2,]))
lines(x=c(min(dysregulatedpathways[1:2,]),max(dysregulatedpathways[1:2,])),
y=c(min(dysregulatedpathways[1:2,]),max(dysregulatedpathways[1:2,])),type="l")
```

`GSReg.kendall.tau.distance`

Calculate Kendall-tau distance in different forms

Description

Different types of calculations of kendall-tau distance.

Usage

```
GSReg.kendall.tau.distance(V)
GSReg.kendall.tau.distance.template(V, Temp)
GSReg.kendall.tau.distance.restricted(V, RestMat)
```

Arguments

`V`	A matrix on which the distances will be calculated. The distance will be calculated between any pair of columns
`Temp`	A square binary template matrix, like DIRAC template, whose columns and rows correspond to the rows of V. In general, Temp provides a template for comparisons in V. Temp[i,j]==1 means that the template expects that V[i,.]<V[j,.] and 0 means the reverse.
`RestMat`	Restricted matrix. It must be a square matrix and symmetric with binary 0 or 1 whose columns and rows correspond to the rows of V. Only comparisons with one will be considered.

Details

GSReg.kendall.tau.distance returns kendall-tau calculates distance matrix between any pair of columns of V.

GSReg.kendall.tau.distance.template returns kendall-tau calculates distance matrix between any column V and a template. Temp[i,j] represent a comaprison between the i-th and j-th element of a vector. Hence, the out come's k-th element is (V[i,k]<V[j,k] & RestMat[i,j]==1)/((nrow(RestMat)*(nrow(RestMat)-1))/2).

GSReg.kendall.tau.distance.restricted calculates the Kendall-tau distance and the only considered comparisons are those RestMat[i,j]==1. It is a modified Kendall-tau distance used by SEVA.

Value

Kendall-tau distance.

Author(s)

Bahman Afsari

See Also

GSReg.GeneSets.DIRAC,GSReg.GeneSets.EVA

Examples

```
library(GSReg)
V <- cbind(c(1,5,3),c(3,2,1))
rownames(V) <- c("F1","F2","F3")
colnames(V) <- c("S1","S2")


myRest1 <- cbind(c(0,1,1),c(1,0,1),c(1,1,0))
rownames(myRest1) <- rownames(V)
colnames(myRest1) <- rownames(V)

GSReg.kendall.tau.distance.restricted(V,myRest1)

GSReg.kendall.tau.distance(V)

myRest2 <- cbind(c(0,0,1),c(0,0,1),c(1,1,0))
rownames(myRest2) <- rownames(V)
colnames(myRest2) <- rownames(V)
GSReg.kendall.tau.distance.restricted(V,myRest2)


Temp1 <- cbind(c(0,1,1),c(0,0,0),c(0,1,0))
rownames(Temp1) <- rownames(V)
colnames(Temp1) <- rownames(V)

GSReg.kendall.tau.distance.template(V,Temp = Temp1)
```

`GSReg.overlapJunction` *Generates junction overlap matrices required for SEVA*

Description

GSReg.overlapJunction generates junction overlap matrices required for SEVA. It may also perform the filtering the junctions based on the expression of the gene.

Usage

```
GSReg.overlapJunction <- function(juncExprs,
                                  GenestoStudy=NULL,
                                  geneexpr=NULL,
                                  minmeanloggeneexp= 3,
                                  alpha =0,
                                  sparse = F,
                        genesCoordinatesTxDB = TxDb.Hsapiens.UCSC.hg19.knownGene,
                                  geneIDInTxDB = 'ENTREZID',
                                  geneIDOut = 'SYMBOL',
                                  org=org.Hs.eg.db, ...)
```

Arguments

`juncExprs` A matrix containing junction expression whose columns represent samples. Rows correspond to junctions and whose names are formed in the following format, chrN:D-A, N represents chromosome number, D represents the start coordination coordinate and A the acceptor end coordinationcoordinate. If geneexprs is not specified, overlaps and consequently SEVA statistics become independent of the quantification method for expression (assuming they do not affect the order of the junction expression) because SEVA is a rank-based method. Otherwise, junction expression and total gene expression in geneexprs (expected to have the same quantification method as junction expression) would be log transformed for applying optimal filtering.

`geneexpr` gene expression matrix whose values have been calculated using the same quantification as juncExprs. Columns must contain the same samples and same sample order as juncExprs. It is used for two types of filtering: see minmeanloggeneexp, alpha.If this parameter is missing, both filters will not be applied.

`minmeanloggeneexp` The parameter for filtering a genes based on the gene expression. If the geneexpr after transformed in log2 does not pass minmeanloggeneexp, all its corresponding junctions would be filtered.

`alpha` The parameter for the filter junction expression . By default this filter is off by alpha=0 but we recommend alpha= 0.1 for a default analysis. In fact, if a junction maximum expression in log2 does not pass the average of its corresponding gene expression in log2 times alpha, the junction will be filtered.

`sparse` Use sparse matrices for junctions overlap. Not recommended unless you run out of memory.

`genesCoordinatesTxDB` The annotation database used for alignment. The default is hg19. It must be the annotation dataset used for gene and junction expression alignment.

`geneIDInTxDB` Gene IDs in the database.

`geneIDOut` Gene IDs used for geneexpr.

`...` Other parameters to be sent to mapIds function.

Value

`$Rest` a list for all genestoStudy. Each of them contains a square matrix whose rows and columns corresponds to the genes junction and the value is one if they overlap otherwise zero i.e. $Rest[["genes"]]["junci","juncj" = I(junc i and junc j overlap)

`$genesJunction` a list for all geensToStudy. Each of them contains a vector of the names of the junctions.

Author(s)

Bahman Afsari

See Also

GSReg.GeneSets.DIRAC,GSReg.GeneSets.EVA

Examples

```
library(GSReg)
require('Homo.sapiens')
require('org.Hs.eg.db')
require('GenomicRanges')

data(juncExprsSimulated)


overlapMat <- GSReg.overlapJunction(juncExprs =  junc.RPM.Simulated,
                                    geneexpr = geneExrsGSReg)
```

`GSReg.SEVA` *Applies Splice-EVA (SEVA) algorithm*

Description

GSReg.SEVA identifies Differential Spliced genes by assigning p-value by SEVA.

Usage

```
function(juncExprs,
 phenoVect,
         verbose=T,
         sparse =F, ...)
```

Arguments

juncExprs	A matrix containing junction expression whose columns represent samples. Rows correspond to junctions and whose names are formed in the following format, chrN:D-A, N represents chromosome number, D represents the start coordination coordinate and A the acceptor end coordinationcoordinate. If geneexprs is not specified, SEVA statistics are independent of the coordinates for junction expression because they are rank-based . Otherwise, junction expression would be log transformed and apply the same quantification method as used for total gene expression in geneexprs for optimal filtering.
phenoVect	a factor containing the labels for columns of juncExprs.
verbose	If True, reports that if the progress of the analysis after every 100 genes are analyzed.
sparse	Use sparse matrices for junctions overlap. Not recommended unless you run out of memory.
...	Parameters would be passed to GSReg.OverlapJunction function.

Value

a list of analysis for all genes.

$E1	the modified variance on the pathway within the samples from levels(phenotypes)[1].
$E2	the modified variance on the pathway within the samples from levels(phenotypes)[2].
$E12	the modified variance on the pathway across the samples from levels(phenotypes)[1] tolevels(phenotypes)[2].
$VarEta1	the estimation of the modified variance on the pathway within the samples from levels(phenotypes)[1].
$VarEta2	the estimation of the modified variance on the pathway within the samples from levels(phenotypes)[2].
$zscore	zscore for the modified variance.
$zscoreD12D1	zscoreD12D1 for the modified variance.
$zscoreD12D2	zscoreD12D2 for the modified variance.
$pvalue	theoretical p-value for null E1 = E2. (Standard EVA).
$pvalueD12D1	theoretical p-value for null E1 = E12.
$pvalueD12D2	theoretical p-value for null E2 = E12.
$pvalueTotal	Bonferonni corrected p-value of the three p-values.

Author(s)

Bahman Afsari

See Also

GSReg.OverlapJunction,GSReg.GeneSets.DIRAC,GSReg.GeneSets.EVA

Examples

```
library(GSReg)
require('Homo.sapiens')
require('org.Hs.eg.db')
require('GenomicRanges')



data(juncExprsSimulated)
SEVAjunc <- GSReg.SEVA(juncExprs = junc.RPM.Simulated,
                       phenoVect = phenotypes,
                       geneexpr = geneExrsGSReg)

print(sapply(SEVAjunc,function(x) x$pvalue))
```

Index

∗ **DIRAC Analysis**
GSReg.GeneSets.DIRAC, 3
∗ **Expression Variation Analysis**
GSReg.GeneSets.EVA, 4
∗ **Junction overlap**
GSReg.kendall.tau.distance, 6
GSReg.overlapJunction, 8
GSReg.SEVA, 9
∗ **package**
GSReg-package, 2

GSReg *(GSReg-package)*, 2
GSReg-package, 2
GSReg.GeneSets.DIRAC, 3
GSReg.GeneSets.EVA, 4
GSReg.kendall.tau.distance, 6
GSReg.kendall.tau.template
(GSReg.kendall.tau.distance), 6
GSReg.overlapJunction, 8
GSReg.SEVA, 9