Package: PCGSE (via r-universe)

October 13, 2024

Type Package

Title Principal Component Gene Set Enrichment

Version 0.5.0

Date 2023-08-20

Author H. Robert Frost

Maintainer H. Robert Frost <rob.frost@dartmouth.edu>

Description Contains logic for computing the statistical association of variable groups, i.e., gene sets, with respect to the principal components of genomic data.

Depends R (>= 2.15.0), RMTstat, MASS, methods

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2023-08-20 15:42:38 UTC

Contents

`PCGSE-package` *Implementation of the Principal Component Gene Set Enrichment (PCGSE) and Spectral Gene Set Enrichment (SGSE) algorithms*

Description

Contains logic to compute the statistical association between gene sets and the principal components of experimental data.

Details

Package:	PCGSE
Type:	Package
Version:	0.4.1
Date:	2023-08
License:	GPL-3

Principal component gene set enrichment is performed using the function `pcgse`. Spectral gene set enrichment is performed using the function `sgse`.

Note

This work was supported by the National Institutes of Health R01 grants LM010098, LM011360, EY022300, GM103506 and GM103534.

Author(s)

H. Robert Frost

References

- Frost, H. R., Li, Z., and Moore, J. H. (2014). Principal component gene set enrichment (PCGSE). ArXiv e-prints. arXiv:1403.5148.
- Frost, H. R., Li, Z., and Moore, J. H. (2014). Spectral gene set enrichment (SGSE). ArXiv e-prints.

`pcgse` *Principal component gene set enrichment (PCGSE) algorithm*

Description

Implementation of the PCGSE algorithm. Computes the statistical association between gene sets and the principal components of experimental data using a two-stage competitive test. Supported gene-level test statistics include the PC loadings for each genomic variable, the Pearson correlation coefficients between each genomic variable and each PC and the Fisher-transformed correlation coefficients. The input data is centered and scaled so that eigendecomposition is computed on the sample correlation matrix rather than the sample covariance matrix. Because the PC loadings for PCA on a correlation matrix are proportional to the Pearson correlation coefficients between each PC and each variable, all supported gene-level statistics provide a measure of correlation between genomic variables and PCs. Each gene set is quantified using either a standardized mean difference statistic or a standardized rank sum statistic. The statistical significance of each gene set test statistic is computed according to a competitive null hypothesis using either a parametric test, a correlation-adjusted parametric test or a permutation test.

Usage

```
pcgse(data, prcomp.output, pc.indexes=1, gene.sets, gene.statistic="z",
    transformation="none", gene.set.statistic="mean.diff",
    gene.set.test="cor.adj.parametric", nperm=9999)
```

Arguments

`data`	Empirical data matrix, observations-by-variables. Must be specified. Cannot contain missing values.
`prcomp.output`	Output of prcomp(data,center=T,scale=T). If not specified, it will be computed.
`pc.indexes`	Indices of the PCs for which enrichment should be computed. Defaults to 1.
`gene.sets`	Data structure that holds gene set membership information. Must be either a binary membership matrix or a list of gene set member indexes. For the member matrix, rows are gene sets, columns are genes, elements are binary membership values. For the membership index list, each element of the list represents a gene set and holds a vector of indexes of genes that are members. Must be a matrix if gene.set.test is set to "permutation".
`gene.statistic`	The gene-level statistic used to quantify the association between each genomic variable and each PC. Must be one of the following (default is "z"): • "loading": PC loading associated with the genomic variable. • "cor": Pearson correlation coefficient between the PC and the genomic variable. • "z": Fisher-transformed Pearson correlation coefficient.
`transformation`	Optional transformation to apply to the gene-level statistics. Must be one of the following (default is "none"): • "none": No transformations are applied to the gene-level statistics. • "abs.value": The absolute value of the gene-level statistics is used.
`gene.set.statistic`	The gene set statisic computed from the gene-level statistics. Must be one of the following (default is "mean.diff"): • "mean.diff": The standardized difference between the mean of the gene-level statistics for members of the gene set and the mean of the gene-level statistics for genomic variables not in the gene set. Equivalent to the U_D statistic from Barry et al. • "rank.sum": The standardized Wilcoxon rank sum statistic computed from the gene-level statistics for members of the gene set. Equivalent to the U_W statistic from Barry et al.
`gene.set.test`	The statistical test used to compute the significance of the gene set statistics under a competitive null hypothesis. The "parametric" test is in the "class 1" test category according to Barry et al., the "cor.adj.parametric" and "permutation" tests are in the "class 2" test category according to Barry et al. Must be one of the following (default is "cor.adj.parametric"): • "parametric": If the mean difference is being used as the gene set statistic, corresponds to a two-sided, two-sample t-test with equal variances. If

the rank sum is being used as the gene set statistic, this corresponds to a two-sided, two-sample z-test based on the standardized rank sum statistic. NOTE: both of these tests incorrectly assume the gene-level statistics are i.i.d. and should therefore only be used for comparative purposes.

- "cor.adj.parametric": Tests statistical significance of the standardized and correlation-adjusted gene set statistic using a two-sided t-test or z-test. Similar to the CAMERA method by Wu et al., standardization of either the mean different statistic or rank sum statistic is performed using a variation inflation factor based on the average pair-wise correlation between the gene-level statistics for members of the gene set. Per Barry et al., this is approximated by the average correlation between the genomic variables. Per Wu et al., the significance of the correlation-adjusted t-statistic is tested using a two-sided t-test with n-2 df and the significance of the correlation-adjusted rank sum z-statistic is tested using a two-sided z-test.
- "permutation": Tests gene set enrichment via the permutation distribution of the gene set statistic. The permutation distribution is computed via permutation of the sample labels, which, in this case, is equivalent to permutation of the elements of the target PC. This test is realized using the safe() function from the R safe package. The number of permutations is controlled by the "nperm" parameter. The gene.statistic cannot be set to "loadings" with this option. Per Barry et al., this correlation is approximated by the average correlation between the genomic variables. This option can be extremely computationally expensive so should not be used for most applications.

`nperm` Number of permutations to perform. Only relevant if gene.set.test is set to "permutation".

Value

List with the following elements:

- p.values: Matrix with one row per gene set and one column for each tested PC. Elements are the two-sided competitive enrichment p-values. Multiple hypothesis correction is NOT applied to these p-values.
- statistics: Matrix with one row per gene set and one column for each tested PC. Elements are the gene set test statistics for each gene set.

Examples

```
library(MASS)

p=200 ## number of genomic variables
n=50 ## number of observations
f=20 ## number of gene sets

## Create annotation matrix with disjoint gene sets
gene.sets = matrix(0, nrow=f, ncol=p)
for (i in 1:f) {
   gene.sets[i, ((i-1)*p/f + 1):(i*p/f)] = 1
```

```
}

## Simulate MVN data with two population PCs whose loadings are
## associated with the first and second gene sets, respectively.
var1=2 ## variance of first population PC
var2=1 ## variance of second population PC
default.var=.1 ## background variance of population PCs
load = sqrt(.1) ## value of population loading vector for gene set 1 on PC 1 and set 2 on PC 2

## Creates a first PC with loadings for just the first 20 genes and a
## second PC with loadings for just the second 20 genes
loadings1 = c(rep(load,p/f), rep(0,p-p/f))
loadings2 = c(rep(0,p/f), rep(load, p/f), rep(0, p-2*p/f))

## Create the population covariance matrix
sigma = var1 * loadings1 %*% t(loadings1) + var2 * loadings2 %*% t(loadings2) +
 diag(rep(default.var, p))

## Simulate MVN  data
data = mvrnorm(n=n, mu=rep(0, p), Sigma=sigma)

## Perform PCA on the standardized data
prcomp.output = prcomp(data, center=TRUE, scale=TRUE)

## Execute PCGSE using Fisher-transformed correlation coefficients as the gene-level statistics,
## the standardized mean difference as the gene set statistic and an unadjusted two-sided,
## two-sample t-test for the determination of statistical significance.
pcgse.results = pcgse(data=data,
                      prcomp.output=prcomp.output,
                      pc.indexes=1:2,
                      gene.sets=gene.sets,
                      gene.statistic="z",
                      transformation="none",
                      gene.set.statistic="mean.diff",
                      gene.set.test="parametric")

## Apply Bonferroni correction to p-values
for (i in 1:2) {
pcgse.results$p.values[,i] = p.adjust(pcgse.results$p.values[,i], method="bonferroni")
}

## Display the p-values for the first 5 gene sets for PCs 1 and 2
pcgse.results$p.values[1:5,]

## Execute PCGSE again but using a correlation-adjusted t-test
pcgse.results = pcgse(data=data,
                      prcomp.output=prcomp.output,
                      pc.indexes=1:2,
                      gene.sets=gene.sets,
                      gene.statistic="z",
                      transformation="none",
                      gene.set.statistic="mean.diff",
                      gene.set.test="cor.adj.parametric")
```

```
## Apply Bonferroni correction to p-values
for (i in 1:2) {
pcgse.results$p.values[,i] = p.adjust(pcgse.results$p.values[,i], method="bonferroni")
}

## Display the p-values for the first 5 gene sets for PCs 1 and 2
pcgse.results$p.values[1:5,]
```

---

sgse *Spectral gene set enrichment (SGSE) algorithm*

Description

Implementation of the SGSE algorithm. Computes the statistical association between gene sets and the spectra of the specified data set. The association between each gene set and each PC is first computed using the pcgse function. The PC-specific p-values are then combined using the weighted Z-method with weights set to either the PC variance or the PC variance scaled by the lower-tailed p-value calculated for the variance according to the Tracey-Widom distribution.

Usage

```
sgse(data, prcomp.output, gene.sets,
     gene.statistic="z", transformation="none",
     gene.set.statistic="mean.diff", gene.set.test="cor.adj.parametric",
     nperm=999, pc.selection.method="all", pc.indexes=NA, rmt.alpha=.05,
     pcgse.weight="rmt.scaled.var")
```

Arguments

data	Empirical data matrix, observations-by-variables. Must be specified. Cannot contain missing values.
prcomp.output	Output of prcomp(data,center=T,scale=T). If not specified, it will be computed.
gene.sets	See documentation for gene.sets argument for pcgse function.
gene.statistic	See documentation for gene.statistic argument for pcgse function.
transformation	See documentation for transformation argument for pcgse function.
gene.set.statistic	See documentation for gene.set.statistic argument for pcgse function.
gene.set.test	See documentation for gene.set.test argument for pcgse function.
nperm	See documentation for nperm argument for pcgse function.
pc.selection.method	Method used to determine the PCs for which enrichment will be computed. Must be one of the following: • "all": All PCs with non-zero variance will be used. • "specific": The set of PCs specified by pc.indexes will be used.

- "rmt": The set of PCs with significant eigenvalues according to the Tracy-Widom distribution for a white Wishart at the alpha specified by the "rmt.alpha" parameter.

`pc.indexes` Indices of the PCs for which enrichment should be computed. Must be specified if pc.selection.method is "specific".

`rmt.alpha` Significance level for selection of PCs according to the Tracy-Widom distribution. Must be specified if pc.selection.method is "rmt".

`pcgse.weight` Type of weight to use with the weighted Z-method to combine the p-values from the PCGSE tests on all PCs selected according to the pc.selection.method parameter value. Must be one of the following:

- "variance": The PC variance is used as the weight. NOTE: this should only be used for evaluation and testing.
- "rmt.scaled.var": The product of the PC variance and the Tracey-Widom lower-tailed p-value for the eigenvalue associated with the PC is used as the weight.

Value

List with the following elements:

- "pc.indexes": Indices of the PCs on which enrichment was performed.
- "pcgse": Output from `pcgse` function on the PCs identified by pc.indexes.
- "sgse": Vector of combined p-values for all PCs identified by pc.indexes.
- "weights": Vector of PC-specific weights for the PCs identified by pc.indexes.

See Also

`pcgse`

Examples

```
library(MASS)

p=200 ## number of genomic variables
n=50 ## number of observations
f=20 ## number of gene sets

## Create annotation matrix with disjoint gene sets
gene.sets = matrix(0, nrow=f, ncol=p)
for (i in 1:f) {
   gene.sets[i, ((i-1)*p/f + 1):(i*p/f)] = 1
}

## Simulate MVN data where the
## first population PC loadings are
## associated with the first gene set.
var=2 ## variance of first population PC
default.var=.1 ## background variance of population PCs
load = sqrt(.1) ## value of population loading vector for gene set 1 on PC 1
```

```
## Creates a first PC with loadings for just the first 20 genes and a
loadings = c(rep(load,p/f), rep(0,p-p/f))

## Create the population covariance matrix
sigma = var * loadings %*% t(loadings) + diag(rep(default.var, p))

## Simulate MVN  data
data = mvrnorm(n=n, mu=rep(0, p), Sigma=sigma)

## Perform PCA on the standardized data
prcomp.output = prcomp(data, center=TRUE, scale=TRUE)

## Execute SGSE using Fisher-transformed correlation coefficients as
## the gene-level statistics, the standardized mean difference as the
## gene set statistic and a correlation adjusted two-sided,
## two-sample t-test for the determination of statistical significance,
## all PCs with non-zero eigenvalues for spectral enrichment and
## variance weights
sgse.results = sgse(data=data,
                    prcomp.output=prcomp.output,
                    gene.sets=gene.sets,
                    gene.statistic="z",
                    transformation="none",
                    gene.set.statistic="mean.diff",
                    gene.set.test="cor.adj.parametric",
                    pc.selection.method="all",
                    pcgse.weight="variance")

## Display the PCGSE p-values for the first 5 gene sets for PC 1
sgse.results$pcgse$p.values[1:5,1]

## Display the SGSE weights for the first 5 PCs
sgse.results$weights[1:5]

## Display the SGSE p-values for the first 5 gene sets
sgse.results$sgse[1:5]

## Execute SGSE again but using RMT scaled variance weights
sgse.results = sgse(data=data,
                    prcomp.output=prcomp.output,
                    gene.sets=gene.sets,
                    gene.statistic="z",
                    transformation="none",
                    gene.set.statistic="mean.diff",
                    gene.set.test="cor.adj.parametric",
                    pc.selection.method="all",
                    pcgse.weight="rmt.scaled.var")

## Display the SGSE weights for the first 5 PCs
sgse.results$weights[1:5]

## Display the SGSE p-values for the first 5 gene sets
```

```
sgse.results$sgse[1:5]

## Execute SGSE again using RMT scaled variance weights and
## all RMT-significant PCs at alpha=.05
sgse.results = sgse(data=data,
                    prcomp.output=prcomp.output,
                    gene.sets=gene.sets,
                    gene.statistic="z",
                    transformation="none",
                    gene.set.statistic="mean.diff",
                    gene.set.test="cor.adj.parametric",
                    pc.selection.method="rmt",
                    rmt.alpha=.05,
                    pcgse.weight="rmt.scaled.var")

## Display the indexes of the RMT-significant PCs
sgse.results$pc.indexes

## Display the SGSE p-values for the first 5 gene sets
sgse.results$sgse[1:5]
```

Index