Package ‘QDRS’

August 20, 2020

Type Package

Title Quantitative Disease Risk Score

Version 0.1.0

Author Danqing Xu

Maintainer Danqing Xu <xud@pstat.ucsb.edu>

Description Functions for calculating quantitative disease risk scores:
(1) Phenotype Risk Score (PheRS), (2) a spectral approach (Eigen), (3) principal component approaches (PCA) that include individual PCs and a linear combination of multiple PCs (LPC), (4), Non-negative Matrix Factorization with Rank 1 (NMF1), and (5) Latent Variable Score based on an unsupervised multivariate mixed model framework (LVS).

License GPL-3

Encoding UTF-8

LazyData true

Imports MASS,
NMF,
AssocTests,
gllvm,
PRROC

RoxygenNote 7.1.1

R topics documented:

EHR *An example data set*

Description

It contains `sample.set` a set of 95 features for 5,000 subjects, `sample.group` a vector of group with two levels "Case" and "Control", and `training` a logical vector that indicates the usage for training.

Usage

```
data(EHR)
```

Format

An object of class `list` of length 3.

Examples

```
data(EHR)
## Not run:
# check the number of cases and controls
table(EHR$sample.group)
# check the prevalence of features
colMeans(EHR$sample.set)

## End(Not run)
```

eigen.score *Eigen Score*

Description

Compute Eigen weights and scores. This method is unsupervised, and assigns weights that are proportional to the balanced accuracies (the average between the sensitivity and the specificity) of the input feature.

Usage

```
eigen.score(X, training)
```

Arguments

`X`	The original data set that include training and test sets. It should be a matrix of numbers.
`training`	A logical or index vector to indicate whether the subject belongs to the training set.

Value

weights The Eigen weights for input features.

scores The resulting Eigen scores for the whole set.

References

Iuliana Ionita-Laza, Kenneth McCallum, Bin Xu, and Joseph D Buxbaum. A spectral approach integrating functional genomic annotations for coding and noncoding variants. Nature genetics, 48(2):214, 2016.

Examples

```
## Not run:
data(EHR)
res1 <- eigen.score(EHR$sample.set, EHR$training)

## End(Not run)
```

`LPC` *Linear Combination of Principal Components*

Description

Computes LPC weights and scores based on an almost unsupervised method that combines multiple PCs and only requires weak labels to help select the signs of individual PCs.

Usage

```
LPC(X, group, training)
```

Arguments

`X`	The original data set that include training and test sets. It should be a matrix of numbers.
`group`	A vector that indicate cases ("Case") and controls ("Control"). `NA` is allowed, and means that the observation is not used in individual PC sign determination.
`training`	A logical or index vector to indicate whether the subject belongs to the training set.

Value

lpc.n The number of significant PCs (eigenvalues) suggested by Tracy-Widom test, a vector of the sign of individual PC.

pc.sign A vector of the sign of individual PC.

weights The LPC weights for input features.

scores The resulting LPC scores for the whole set.

Examples

```
## Not run:
data(EHR)
res1 <- LPC(X = EHR$sample.set, group = EHR$sample.group, training = EHR$training)

## End(Not run)
```

`LVS.fit` *Fit a Latent Variable Model*

Description

`LVS.fit` computes a generalized lineatr latent variable model for multivariate bernoulli data, which assumes one latent variable.

Usage

```
LVS.fit(X, starting.choice = "random", p.seed = 3080)
```

Arguments

`X` The training data set. It should be a matrix of binary numbers.

`starting.choice` Starting values can be generated by fitting model without latent variables, and applying factorial analysis to residuals to get starting values for latent variables and their coefficients (starting.choice = "res"). Another options are to use zeros as a starting values (starting.choice = "zero") or initialize starting values for latent variables with (n x 1) matrix. Defaults to "random".

`p.seed` A random seed for model fitting.

Value

An object of class "gllvm" includes the following components.

`LVS.score` *Latent Variable Score (LVS)*

Description

Computes Latent Variable Scores based on an unsupervised multivariate mixed model framework for multivariate bernoulli data that assumes one latent variable.

Usage

```
LVS.score(X, Y = NULL, starting.choice = "random", p.seed = 3080)
```

Arguments

`X`	A training data set. It should be a matrix of binary numbers.
`Y`	A new data set. It should be a matrix of binary numbers. Default is NULL, then X will be used in prediction.
`starting.choice`	Starting values can be generated by fitting model without latent variables, and applying factorial analysis to residuals to get starting values for latent variables and their coefficients (starting.choice = "res"). Another options are to use zeros as a starting values (starting.choice = "zero") or initialize starting values for latent variables with (n x 1) matrix. Defaults to "random".
`p.seed`	A random seed for model fitting.

Value

lvs The resulting LVSs for the new set Y.

References

Jenni Niku, Wesley Brooks, Riki Herliansyah, Francis KC Hui, Sara Taskinen, and David I Warton. Efficient estimation of generalized linear latent variable models. PloS one, 14(5), 2019.

Examples

```
## Not run:
data(EHR)
sample.set = EHR$sample.set
res1 <- LVS.score(X = sample.set[seq(1,50),])

## End(Not run)
```

`NMF1` *Non-negative Matrix Factorization with Rank 1 Score*

Description

Computes NMF1 scores with factor loadings constrained to be non-negative.

Usage

```
NMF1(X, p.seed = 123)
```

Arguments

`X`	The original data set. It should be a matrix of numbers.
`p.seed`	A random seed for model fitting.

Value

lvs The resulting LVSs for the new set Y.

Examples

```
## Not run:
data(EHR)
sample.set = EHR$sample.set
res1 <- NMF1(X = sample.set[seq(1,50),])

## End(Not run)
```

---

pairwise.auc *Pairwise Area Under the ROC Curve*

Description

Calculates AUROC for pairwise levels of group contrast .

Usage

```
pairwise.auc(x, g, format = "l")
```

Arguments

x	A response vector.
g	A grouping vector or factor.
format	Format of result. The default is "l", long format. The other option is square format.

Value

A data frame with group 1, group 2, and the AUROC with comparison between them.

Examples

```
## Not run:
data(EHR)
x = c(rnorm(2500),runif(2500))
res = pairwise.auc(x, EHR$sample.group)

## End(Not run)
```

---

pairwise.upr *Pairwise Area Under the PR Curve*

Description

Calculates AUPRC for pairwise group levels contrast.

Usage

```
pairwise.upr(x, g, format = "l")
```

Arguments

x	A response vector.
g	A grouping vector or factor.
format	Format of result. The default is "l", long format. The other option is square format.

Value

A data frame with group 1, group 2, and the AUPRC with comparison between them.

Examples

```
## Not run:
data(EHR)
x = c(rnorm(2500),runif(2500))
res = pairwise.upr(x, EHR$sample.group)

## End(Not run)
```

---

pairwise.wilcox *Pairwise Wilcoxon Rank Sum Tests*

Description

Calculates pairwise comparisons between group levels with corrections for multiple testing.

Usage

```
pairwise.wilcox(x, g, p.adjust.method = "bonferroni")
```

Arguments

x	A response vector.
g	A grouping vector or factor.
p.adjust.method	Method for adjusting p values (can be abbreviated): "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none". The default is "bonferroni".

Value

A data frame with group 1, group 2, and the Wilcoxon Rank Sum test p value of comparison between them.

Examples

```
## Not run:
data(EHR)
x = c(rnorm(2500),runif(2500))
res = pairwise.wilcox(x, EHR$sample.group)

## End(Not run)
```

PC *Individual Principal Components with Selected Signs*

Description

Computes individual PC weights and scores. The approach only requires weak labels to help select the signs of individual PCs.

Usage

```
PC(X, group, training, pc.num)
```

Arguments

`X`	The original data set that include training and test sets. It should be a matrix of numbers.
`group`	A grouping vector or factor that indicate cases ("Case") and controls ("Control"). `NA` is allowed, and means that the observation is not used in individual PC sign determination.
`training`	A logical or index vector to indicate whether the subject belongs to the training set.
`pc.num`	The vector of desired individual PCs.

Value

weights The selected PC weights for input features.

scores The resulting PC scores for the whole set.

Examples

```
## Not run:
data(EHR)
res1 <- PC(X = EHR$sample.set, group = EHR$sample.group, training = EHR$training, pc.num = 1:2)

## End(Not run)
```

PheRS *Phenotype Risk Score*

Description

Computes Phenotype Risk Scores based on a approach that was proposed in the context of rare Mendelian phenotypes.

Usage

```
PheRS(X, feature.prevalence = NULL, group = NULL)
```

Arguments

`X`	The original data set that include training and test sets. It should be a matrix of binary numbers.
`feature.prevalence`	A vector of feature prevalence.
`group`	A vector that indicate cases ("Case") and controls ("Control"). `NA` is allowed. The default is NULL, meaning all observations will be used in prevalence computation if the feature prevalence vector is not provided by the user. Otherwise, only the controls will be used.

Value

weights The PheRS weights for input features.

scores The resulting PheRSs for the whole set.

References

Lisa Bastarache, Jacob J Hughey, Scott Hebbring, Joy Marlo, Wanke Zhao, Wanting T Ho, Sara L Van Driest, Tracy L McGregor, Jonathan D Mosley, Quinn S Wells, et al. Phenotype risk scores identify patients with unrecognized mendelian disease patterns. Science, 359(6381):1233– 1239, 2018.

Examples

```
## Not run:
data(EHR)
res1 <- PheRS(X = EHR$sample.set, group = EHR$sample.group)
res2 <- PheRS(X = EHR$sample.set, group = NULL)

## End(Not run)
```

`rankOne.R` *R Matrix with Rank One for Eigen Approach*

Description

Computes an estimated rank-one matrix R with unit-norm eigenvector v. Up to a sign ambiguity, the entries of v are proportional to the balanced accuracies (the average between the sensitivity and the specificity) of the input features.

Usage

```
rankOne.R(Qmat)
```

Arguments

`Qmat`	The correlation matrix Q.

Value

Rmat An estimate of the rank-one R matrix

Index