Package ‘jackstraw’

December 3, 2018

Type Package

Title Statistical Inference for Unsupervised Learning

Version 1.3

Date 2018-12-03

Author Neo Christopher Chung <nchchung@gmail.com>, John D. Storey <jstorey@princeton.edu>, Wei Hao <whao@princeton.edu>

Maintainer Neo Christopher Chung <nchchung@gmail.com>

Description Test for association between the observed data and their systematic patterns of variations. Systematic patterns may be captured by latent variables using principal component analysis (PCA), factor analysis (FA), and related methods. The jackstraw enables statistical testing for association between observed variables and latent variables, as captured by PCs or other estimates. Similarly, unsupervised clustering, such as K-means clustering, partition around medoids (PAM), and others, finds subpopulations among the observed variables. The jackstraw estimates statistical significance of cluster membership, including unsupervised evaluation of cell identities in single cell RNA-seq. P-values and posterior probabilities allows one to rigorously evaluate the strength of cluster membership assignments.

LazyData true

Depends R (>= 3.0.0)

biocViews

Imports corpcor, cluster, methods, ClusterR, qvalue, lfa, stats, irlba, rsvd

Suggests parallel, knitr, rmarkdown

License GPL-2

Encoding UTF-8

RoxygenNote 6.1.0

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2018-12-03 16:10:03 UTC

R topics documented:

dev.R *Compute Deviance for Logistic Factors*

Description

This function computes deviance between the full model and the null (intercept-only) model. It uses built-in R functions, namely `glm`; slow but no C++ dependencies. Make sure that `LFr1` and `LFr0` do not have intercept terms.

Usage

```
dev.R(dat, LFr1, LFr0 = NULL, p = FALSE)
```

Arguments

`dat`	a matrix with `m` rows and `n` columns.
`LFr1`	alternative logistic factors (an output from lfa or lfa.corpcor)
`LFr0`	null logistic factors (an output from lfa or lfa.corpcor)
`p`	estimate p-values (by default, 'FALSE')

Value

When p=FALSE (by default), `dev.R` returns a vector of `m` deviances.

When p=TRUE, a list consisting of

`dev`	the `m` deviances
`p.value`	the `m` p-values based on a chisq distribution

Author(s)

Neo Christopher Chung <nchchung@gmail.com>

find_k *Find a number of clusters or principal components*

Description

There are a wide range of algorithms and visual techniques to identify a number of clusters or principal components embeddd in the observed data.

Usage

```
find_k()
```

Details

It is critical to explore the eigenvalues, cluster stability, and visualization. See R packages bootcluster, EMCluster, and nFactors.

Please see the R package SC3, which provides estkTW() function to find the number of significant eigenvalues according to the Tracy-Widom test.

ADPclust package includes adpclust() function that runs the algorithm on a range of K values. It helps you to identify the most suitable number of clusters.

This package also provides an alternative methods in permutationPA. Through a resampling-based Parallel Analysis, it finds a number of significant components.

jackstraw *jackstraw: Statistical Inference for Unsupervised Learning*

Description

Test for association between the observed data and their systematic patterns of variations. Systematic patterns may be captured by latent variables using principal component analysis (PCA), factor analysis (FA), and related methods. The jackstraw enables statistical testing for association between observed variables and latent variables, as captured by PCs or other estimates. Similarly, unsupervised clustering, such as K-means clustering, partition around medoids (PAM), and others, finds subpopulations among the observed variables. The jackstraw estimates statistical significance of cluster membership, including unsupervised evaluation of cell identities in single cell RNA-seq. P-values and posterior probabilities allows one to rigorously evaluate the strength of cluster membership assignments.

Details

The jackstraw package provides a resampling strategy and testing scheme to estimate statistical significance of association between the observed data and their latent variables. Depending on the data type and the analysis aim, the latent variables may be estimated by principal component analysis, K-means clustering, and related algorithms. The jackstraw methods learn over-fitting characteristics inherent in this circular analysis, where the observed data are used to estimate the latent variables and to again test against the estimated latent variables.

The jackstraw tests enable us to identify the data features (i.e., variables or observations) that are driving systematic variation, in a completely unsupervised manner. Using jackstraw_pca, we can find statistically significant features with regard to the top `r` principal components. Alternatively, jackstraw_kmeans can identify the data features that are statistically significant members of the data-dependent clusters. Furthermore, this package includes more general algorithms such as jackstraw_subspace for the dimension reduction techniques and jackstraw_cluster for the clustering algorithms.

Overall, it computes `m` p-values of association between the `m` data features and their corresponding latent variables. From `m` p-values, pip computes posterior inclusion probabilities, that are useful for feature selection and visualization.

Author(s)

Neo Christopher Chung <`nchchung@gmail.com`>

References

Chung and Storey (2015) Statistical significance of variables driving systematic variation in high-dimensional data. Bioinformatics, 31(4): 545-554 `http://bioinformatics.oxfordjournals.org/content/31/4/545`

Chung (2018) Statistical significance for cluster membership. biorxiv, doi:10.1101/248633 `https://www.biorxiv.org/content/early/2018/01/16/248633`

See Also

jackstraw_pca jackstraw_subspace jackstraw_kmeans jackstraw_cluster

---

`jackstraw_cluster` *Jackstraw for the User-Defined Clustering Algorithm*

Description

Test the cluster membership using a user-defined clustering algorithm

Usage

```
jackstraw_cluster(dat, k, cluster = NULL, centers = NULL,
  algorithm = function(x, centers) kmeans(x, centers, ...), s = 1,
  B = 1000, center = TRUE, noise = NULL, covariate = NULL,
  verbose = FALSE, seed = NULL, ...)
```

Arguments

`dat` a data matrix with `m` rows as variables and `n` columns as observations.

`k` a number of clusters.

`cluster` a vector of cluster assignments.

`centers` a matrix of all cluster centers.

`algorithm` a clustering algorithm to use, where an output must include ‘cluster‘ and ‘centers‘. For exact specification, see `kmeans`.

`s` a number of “synthetic” null variables. Out of `m` variables, `s` variables are independently permuted.

`B` a number of resampling iterations.

`center` a logical specifying to center the rows. By default, `TRUE`.

`noise` specify a parametric distribution to generate a noise term. If `NULL`, a non-parametric jackstraw test is performed.

`covariate` a model matrix of covariates with `n` observations. Must include an intercept in the first column.

`verbose` a logical specifying to print the computational progress. By default, `FALSE`.

`seed` a seed for the random number generator.

`...` optional arguments to control the clustering algorithm.

Details

The clustering algorithms assign `m` rows into `K` clusters. This function enable statistical evaluation if the cluster membership is correctly assigned. Each of `m` p-values refers to the statistical test of that row with regard to its assigned cluster. Its resampling strategy accounts for the over-fitting characteristics due to direct computation of clusters from the observed data and protects against an anti-conservative bias.

The user is expected to explore the data with a given clustering algorithm and determine the number of clusters `k`. Furthermore, provide `cluster` and `centers` as given by applying `algorithm` onto `dat`. The rows of `centers` correspond to `k` clusters, as well as available levels in `cluster`. This function allows you to specify a parametric distribution of a noise term. It is an experimental feature.

Value

`jackstraw_cluster` returns a list consisting of

`F.obs` `m` observed F statistics between variables and cluster centers.

`F.null` F null statistics between null variables and cluster centers, from the jackstraw method.

`p.F` `m` p-values of membership.

Author(s)

Neo Christopher Chung `<nchchung@gmail.com>`

References

Chung (2018) Statistical significance for cluster membership. biorxiv, doi:10.1101/248633 `https://www.biorxiv.org/content/early/2018/01/16/248633`

`jackstraw_irlba` *Non-Parametric Jackstraw for Principal Component Analysis (PCA) using the augmented implicitly restarted Lanczos bidiagonalization algorithm (IRLBA)*

Description

Test association between the observed variables and their latent variables captured by principal components (PCs). PCs are computed using the augmented implicitly restarted Lanczos bidiagonalization algorithm (IRLBA; see `irlba`).

Usage

```
jackstraw_irlba(dat, r1 = NULL, r = NULL, s = NULL, B = NULL,
  covariate = NULL, verbose = TRUE, seed = NULL, ...)
```

Arguments

`dat`	a data matrix with `m` rows as variables and `n` columns as observations.
`r1`	a numeric vector of principal components of interest. Choose a subset of `r` significant PCs to be used.
`r`	a number (a positive integer) of significant principal components. See permutationPA and other methods.
`s`	a number (a positive integer) of "synthetic" null variables. Out of `m` variables, `s` variables are independently permuted.
`B`	a number (a positive integer) of resampling iterations. There will be a total of `s*B` null statistics.
`covariate`	a data matrix of covariates with corresponding `n` observations (do not include an intercept term).
`verbose`	a logical specifying to print the computational progress.
`seed`	a numeric seed for the random number generator.
`...`	additional arguments to `irlba`.

Details

This function computes `m` p-values of linear association between `m` variables and their PCs. Its resampling strategy accounts for the over-fitting characteristics due to direct computation of PCs from the observed data and protects against an anti-conservative bias.

Provide the data matrix, with `m` variables as rows and `n` observations as columns. Given that there are `r` significant PCs, this function tests for linear association between `m` varibles and their `r` PCs.

You could specify a subset of significant PCs that you are interested in (PC). If PC is given, then this function computes statistical significance of association between m variables and PC, while adjusting for other PCs (i.e., significant PCs that are not your interest). For example, if you want to identify variables associated with 1st and 2nd PCs, when your data contains three significant PCs, set r=3 and PC=c(1,2).

Please take a careful look at your data and use appropriate graphical and statistical criteria to determine a number of significant PCs, r. The number of significant PCs depends on the data structure and the context. In a case when you fail to specify r, it will be estimated from a permutation test (Buja and Eyuboglu, 1992) using a function permutationPA.

If s is not supplied, s is set to about 10% of m variables. If B is not supplied, B is set to m*10/s.

Value

jackstraw_irlba returns a list consisting of

p.value	m p-values of association tests between variables and their principal components
obs.stat	m observed F-test statistics
null.stat	s*B null F-test statistics

Author(s)

Neo Christopher Chung <nchchung@gmail.com>

References

Chung and Storey (2015) Statistical significance of variables driving systematic variation in high-dimensional data. Bioinformatics, 31(4): 545-554 http://bioinformatics.oxfordjournals.org/content/31/4/545

See Also

jackstraw jackstraw_subspace permutationPA

Examples

```
set.seed(1234)
## simulate data from a latent variable model: Y = BL + E
B = c(rep(1,50),rep(-1,50), rep(0,900))
L = rnorm(20)
E = matrix(rnorm(1000*20), nrow=1000)
dat = B %*% t(L) + E
dat = t(scale(t(dat), center=TRUE, scale=TRUE))

## apply the jackstraw
out = jackstraw_irlba(dat, r=1)

## Use optional arguments
## For example, set s and B for a balance between speed of the algorithm and accuracy of p-values
## Not run:
## out = jackstraw_irlba(dat, r=1, s=10, B=1000, seed=5678)
## End(Not run)
```

jackstraw_kmeans *Non-Parametric Jackstraw for K-means Clustering*

Description

Test the cluster membership for K-means clustering

Usage

```
jackstraw_kmeans(dat, kmeans.dat, s = NULL, B = NULL, center = TRUE,
  covariate = NULL, verbose = FALSE, pool = TRUE, seed = NULL, ...)
```

Arguments

`dat` a matrix with `m` rows as variables and `n` columns as observations.

`kmeans.dat` an output from applying `kmeans()` onto `dat`.

`s` a number of “synthetic” null variables. Out of `m` variables, `s` variables are independently permuted.

`B` a number of resampling iterations.

`center` a logical specifying to center the rows. By default, `TRUE`.

`covariate` a model matrix of covariates with `n` observations. Must include an intercept in the first column.

`verbose` a logical specifying to print the computational progress. By default, `FALSE`.

`pool` a logical specifying to pool the null statistics across all clusters. By default, `TRUE`.

`seed` a seed for the random number generator.

`...` optional arguments to control the k-means clustering algorithm (refers to `kmeans`).

Details

K-means clustering assign `m` rows into `K` clusters. This function enable statistical evaluation if the cluster membership is correctly assigned. Each of `m` p-values refers to the statistical test of that row with regard to its assigned cluster. Its resampling strategy accounts for the over-fitting characteristics due to direct computation of clusters from the observed data and protects against an anti-conservative bias.

The input data (`dat`) must be of a class ‘matrix‘.

Value

`jackstraw_kmeans` returns a list consisting of

`F.obs` `m` observed F statistics between variables and cluster centers.

`F.null` F null statistics between null variables and cluster centers, from the jackstraw method.

`p.F` `m` p-values of membership.

Author(s)

Neo Christopher Chung <`nchchung@gmail.com`>

References

Chung (2018) Statistical significance for cluster membership. biorxiv, doi:10.1101/248633 `https://www.biorxiv.org/content/early/2018/01/16/248633`

Examples

```
## Not run:
set.seed(1234)
dat = t(scale(t(Jurkat293T), center=TRUE, scale=FALSE))
kmeans.dat <- kmeans(dat, centers=2, nstart = 10, iter.max = 100)
jackstraw.out <- jackstraw_kmeans(dat, kmeans.dat)

## End(Not run)
```

`jackstraw_kmeanspp` *Non-Parametric Jackstraw for K-means Clustering using RcppArmadillo*

Description

Test the cluster membership for K-means clustering, using K-means++ initialization

Usage

```
jackstraw_kmeanspp(dat, kmeans.dat, s = NULL, B = NULL,
  center = TRUE, covariate = NULL, verbose = FALSE, pool = TRUE,
  seed = NULL, ...)
```

Arguments

`dat` a matrix with `m` rows as variables and `n` columns as observations.

`kmeans.dat` an output from applying `ClusterR::KMeans_rcpp` onto `dat`.

`s` a number of “synthetic” null variables. Out of `m` variables, `s` variables are independently permuted.

`B` a number of resampling iterations.

`center` a logical specifying to center the rows. By default, `TRUE`.

`covariate` a model matrix of covariates with `n` observations. Must include an intercept in the first column.

`verbose` a logical specifying to print the computational progress. By default, `FALSE`.

`pool` a logical specifying to pool the null statistics across all clusters. By default, `TRUE`.

`seed` a seed for the random number generator.

`...` optional arguments to control the k-means clustering algorithm (refers to `ClusterR::KMeans_rcpp`).

Details

K-means clustering assign `m` rows into `K` clusters. This function enable statistical evaluation if the cluster membership is correctly assigned. Each of `m` p-values refers to the statistical test of that row with regard to its assigned cluster. Its resampling strategy accounts for the over-fitting characteristics due to direct computation of clusters from the observed data and protects against an anti-conservative bias.

Generally, it functions identical to `jackstraw_kmeans`, but this uses `ClusterR::KMeans_rcpp` instead of `stats::kmeans`. A speed improvement is gained by K-means++ initialization and `RcppArmadillo`. If the input data is still too large, consider using `jackstraw_MiniBatchKmeans`.

The input data (`dat`) must be of a class ‘matrix‘.

Value

`jackstraw_kmeanspp` returns a list consisting of

`F.obs` `m` observed F statistics between variables and cluster centers.

`F.null` F null statistics between null variables and cluster centers, from the jackstraw method.

`p.F` `m` p-values of membership.

Author(s)

Neo Christopher Chung <`nchchung@gmail.com`>

References

Chung (2018) Statistical significance for cluster membership. biorxiv, doi:10.1101/248633 `https://www.biorxiv.org/content/early/2018/01/16/248633`

Examples

```
## Not run:
set.seed(1234)
library(ClusterR)
dat = t(scale(t(Jurkat293T), center=TRUE, scale=FALSE))
kmeans.dat <- KMeans_rcpp(dat,  clusters = 10, num_init = 1,
max_iters = 100, initializer = 'kmeans++')
jackstraw.out <- jackstraw_kmeanspp(dat, kmeans.dat)

## End(Not run)
```

---

jackstraw_lfa *Non-Parametric Jackstraw for Logistic Factor Analysis*

Description

Test association between the observed variables and their latent variables captured by logistic factors (LFs).

Usage

```
jackstraw_lfa(dat, FUN = function(x) lfa(x, r)[, , drop = FALSE],
  devR = FALSE, r = NULL, r1 = NULL, s = NULL, B = NULL,
  covariate = NULL, verbose = TRUE, seed = NULL)
```

Arguments

dat	a genotype matrix with m rows as variables and n columns as observations.
FUN	a function to use for LFA (by default, it uses the lfagen package)
devR	use a R function to compute deviance. By default, FALSE (uses C++).
r	a number of significant LFs.
r1	a numeric vector of LFs of interest (implying you are not interested in all r LFs).
s	a number of “synthetic” null variables. Out of m variables, s variables are independently permuted.
B	a number of resampling iterations. There will be a total of s*B null statistics.
covariate	a data matrix of covariates with corresponding n observations (do not include an intercept term).
verbose	a logical specifying to print the computational progress.
seed	a seed for the random number generator.

Details

This function uses logistic factor analysis (LFA) from Wei et al. (2014). Particularly, a deviation dev in logistic regression (the full model with r LFs vs. the intercept-only model) is used to assess association.

Value

jackstraw_lfa returns a list consisting of

p.value	m p-values of association tests between variables and their LFs
obs.stat	m observed devs
null.stat	s*B null devs

Author(s)

Neo Christopher Chung <nchchung@gmail.com>

See Also

jackstraw_pca jackstraw jackstraw_subspace

Examples

```
set.seed(1234)
## Not run:
## simulate genotype data from a logistic factor model: drawing rbinom from logit(BL)
m=5000; n=100; pi0=.9
m0 = round(m*pi0)
m1 = m-round(m*pi0)
B = matrix(0, nrow=m, ncol=1)
B[1:m1,] = matrix(runif(m1*n, min=-.5, max=.5), nrow=m1, ncol=1)
L = matrix(rnorm(n), nrow=1, ncol=n)
BL = B %*% L
prob = exp(BL)/(1+exp(BL))

dat = matrix(rbinom(m*n, 2, as.numeric(prob)), m, n)

## apply the jackstraw_lfa
out = jackstraw_lfa(dat, 2)

## apply the jackstraw_lfa using self-contained R functions
out = jackstraw_lfa(dat, FUN = function(x) lfa.corpcor(x, 2)[, , drop = FALSE], r = 2, devR = TRUE)

## End(Not run)
```

jackstraw_MiniBatchKmeans

Non-Parametric Jackstraw for Mini Batch K-means Clustering

Description

Test the cluster membership for K-means clustering

Usage

```
jackstraw_MiniBatchKmeans(dat, MiniBatchKmeans.output = NULL, s = NULL,
  B = NULL, center = TRUE, covariate = NULL, verbose = FALSE,
  seed = NULL, batch_size = floor(nrow(dat)/100),
  initializer = "kmeans++", pool = TRUE, ...)
```

Arguments

`dat` a data matrix with `m` rows as variables and `n` columns as observations.

`MiniBatchKmeans.output` an output from applying `ClusterR::MiniBatchKmeans()` onto `dat`. This provides more controls over the algorithm and subsequently the initial centroids used.

`s` a number of “synthetic” null variables. Out of `m` variables, `s` variables are independently permuted.

`B` a number of resampling iterations.

`center` a logical specifying to center the rows. By default, `TRUE`.

`covariate` a model matrix of covariates with `n` observations. Must include an intercept in the first column.

`verbose` a logical specifying to print the computational progress. By default, `FALSE`.

`seed` a seed for the random number generator.

`batch_size` the size of the mini batches.

`initializer` the method of initialization. By default, `kmeans++`.

`pool` a logical specifying to pool the null statistics across all clusters. By default, `TRUE`.

`...` optional arguments to control the Mini Batch K-means clustering algorithm (refers to `ClusterR::MiniBatchKmeans`).

Details

K-means clustering assign `m` rows into `K` clusters. This function enable statistical evaluation if the cluster membership is correctly assigned. Each of `m` p-values refers to the statistical test of that row with regard to its assigned cluster. Its resampling strategy accounts for the over-fitting characteristics due to direct computation of clusters from the observed data and protects against an anti-conservative bias.

Value

`jackstraw_MiniBatchKmeans` returns a list consisting of

`F.obs` `m` observed F statistics between variables and cluster centers.

`F.null` F null statistics between null variables and cluster centers, from the jackstraw method.

`p.F` `m` p-values of membership.

Author(s)

Neo Christopher Chung <nchchung@gmail.com>

References

Chung (2018) Statistical significance for cluster membership. biorxiv, doi:10.1101/248633 https://www.biorxiv.org/content/early/2018/01/16/248633

Examples

```
## Not run:
library(ClusterR)
set.seed(1234)
dat = t(scale(t(Jurkat293T), center=TRUE, scale=FALSE))
MiniBatchKmeans.output <- MiniBatchKmeans(data=dat, clusters = 2, batch_size = 300,
initializer = "kmeans++")
jackstraw.output <- jackstraw_MiniBatchKmeans(dat,
MiniBatchKmeans.output = MiniBatchKmeans.output)

## End(Not run)
```

jackstraw_pam *Non-Parametric Jackstraw for Partitioning Around Medoids (PAM)*

Description

Test the cluster membership for Partitioning Around Medoids (PAM)

Usage

```
jackstraw_pam(dat, pam.dat, s = NULL, B = NULL, center = TRUE,
  covariate = NULL, verbose = FALSE, pool = TRUE, seed = NULL, ...)
```

Arguments

`dat` a matrix with `m` rows as variables and `n` columns as observations.

`pam.dat` an output from applying `cluster::pam()` on `dat`.

`s` a number of “synthetic” null variables. Out of `m` variables, `s` variables are independently permuted.

`B` a number of resampling iterations.

`center` a logical specifying to center the rows. By default, `TRUE`.

`covariate` a model matrix of covariates with `n` observations. Must include an intercept in the first column.

`verbose` a logical specifying to print the computational progress. By default, `FALSE`.

`pool` a logical specifying to pool the null statistics across all clusters. By default, `TRUE`.

`seed` a seed for the random number generator.

`...` optional arguments to control the k-means clustering algorithm (refers to `kmeans`).

Details

PAM assigns m rows into K clusters. This function enable statistical evaluation if the cluster membership is correctly assigned. Each of m p-values refers to the statistical test of that row with regard to its assigned cluster. Its resampling strategy accounts for the over-fitting characteristics due to direct computation of clusters from the observed data and protects against an anti-conservative bias.

For a large dataset, PAM could be too slow. Consider using `cluster::clara` and `jackstraw::jackstraw_clara`.

The input data (`dat`) must be of a class ‘matrix‘.

Value

`jackstraw_pam` returns a list consisting of

`F.obs`	`m` observed F statistics between variables and cluster medoids.
`F.null`	F null statistics between null variables and cluster medoids, from the jackstraw method.
`p.F`	`m` p-values of membership.

Author(s)

Neo Christopher Chung <nchchung@gmail.com>

References

Chung (2018) Statistical significance for cluster membership. biorxiv, doi:10.1101/248633 https://www.biorxiv.org/content/early/2018/01/16/248633

Examples

```
## Not run:
library(cluster)
set.seed(1234)
dat = t(scale(t(Jurkat293T), center=TRUE, scale=FALSE))
pam.dat <- pam(dat, k=2)
jackstraw.out <- jackstraw_pam(dat, pam.dat = pam.dat)

## End(Not run)
```

`jackstraw_pca` *Non-Parametric Jackstraw for Principal Component Analysis (PCA)*

Description

Test association between the observed variables and their latent variables captured by principal components (PCs).

Usage

```
jackstraw_pca(dat, r1 = NULL, r = NULL, s = NULL, B = NULL,
  covariate = NULL, verbose = TRUE, seed = NULL)
```

Arguments

dat a data matrix with m rows as variables and n columns as observations.

r1 a numeric vector of principal components of interest. Choose a subset of r significant PCs to be used.

r a number (a positive integer) of significant principal components. See permutationPA and other methods.

s a number (a positive integer) of "synthetic" null variables. Out of m variables, s variables are independently permuted.

B a number (a positive integer) of resampling iterations. There will be a total of s*B null statistics.

covariate a data matrix of covariates with corresponding n observations (do not include an intercept term).

verbose a logical specifying to print the computational progress.

seed a numeric seed for the random number generator.

Details

This function computes m p-values of linear association between m variables and their PCs. Its resampling strategy accounts for the over-fitting characteristics due to direct computation of PCs from the observed data and protects against an anti-conservative bias.

Provide the data matrix, with m variables as rows and n observations as columns. Given that there are r significant PCs, this function tests for linear association between m varibles and their r PCs.

You could specify a subset of significant PCs that you are interested in (PC). If PC is given, then this function computes statistical significance of association between m variables and PC, while adjusting for other PCs (i.e., significant PCs that are not your interest). For example, if you want to identify variables associated with 1st and 2nd PCs, when your data contains three significant PCs, set r=3 and PC=c(1,2).

Please take a careful look at your data and use appropriate graphical and statistical criteria to determine a number of significant PCs, r. The number of significant PCs depends on the data structure and the context. In a case when you fail to specify r, it will be estimated from a permutation test (Buja and Eyuboglu, 1992) using a function permutationPA.

If s is not supplied, s is set to about 10% of m variables. If B is not supplied, B is set to m*10/s.

Value

jackstraw_pca returns a list consisting of

p.value m p-values of association tests between variables and their principal components

obs.stat m observed F-test statistics

null.stat s*B null F-test statistics

Author(s)

Neo Christopher Chung <nchchung@gmail.com>

References

Chung and Storey (2015) Statistical significance of variables driving systematic variation in high-dimensional data. Bioinformatics, 31(4): 545-554 http://bioinformatics.oxfordjournals.org/content/31/4/545

See Also

jackstraw jackstraw_subspace permutationPA

Examples

```
## Not run:
set.seed(1234)
## simulate data from a latent variable model: Y = BL + E
B = c(rep(1,50),rep(-1,50), rep(0,900))
L = rnorm(20)
E = matrix(rnorm(1000*20), nrow=1000)
dat = B %*% t(L) + E
dat = t(scale(t(dat), center=TRUE, scale=TRUE))

## apply the jackstraw
out = jackstraw_pca(dat, r=1)

## Use optional arguments
## For example, set s and B for a balance between speed of the algorithm and accuracy of p-values
## out = jackstraw_pca(dat, r=1, s=10, B=1000, seed=5678)

## End(Not run)
```

jackstraw_rpca *Non-Parametric Jackstraw for Principal Component Analysis (PCA) using Randomized Singular Value Decomposition*

Description

Test association between the observed variables and their latent variables captured by principal components (PCs). PCs are computed by randomized Singular Value Decomposition (see rsvd).

Usage

```
jackstraw_rpca(dat, r1 = NULL, r = NULL, s = NULL, B = NULL,
  covariate = NULL, verbose = TRUE, seed = NULL, ...)
```

Arguments

`dat` a data matrix with `m` rows as variables and `n` columns as observations.

`r1` a numeric vector of principal components of interest. Choose a subset of `r` significant PCs to be used.

`r` a number (a positive integer) of significant principal components. See permutationPA and other methods.

`s` a number (a positive integer) of “synthetic” null variables. Out of `m` variables, `s` variables are independently permuted.

`B` a number (a positive integer) of resampling iterations. There will be a total of `s*B` null statistics.

`covariate` a data matrix of covariates with corresponding `n` observations (do not include an intercept term).

`verbose` a logical specifying to print the computational progress.

`seed` a numeric seed for the random number generator.

`...` additional arguments to `rpca`.

Details

This function computes `m` p-values of linear association between `m` variables and their PCs. Its resampling strategy accounts for the over-fitting characteristics due to direct computation of PCs from the observed data and protects against an anti-conservative bias.

Provide the data matrix, with `m` variables as rows and `n` observations as columns. Given that there are `r` significant PCs, this function tests for linear association between `m` varibles and their `r` PCs.

You could specify a subset of significant PCs that you are interested in (`PC`). If `PC` is given, then this function computes statistical significance of association between `m` variables and `PC`, while adjusting for other PCs (i.e., significant PCs that are not your interest). For example, if you want to identify variables associated with 1st and 2nd PCs, when your data contains three significant PCs, set `r=3` and `PC=c(1,2)`.

Please take a careful look at your data and use appropriate graphical and statistical criteria to determine a number of significant PCs, `r`. The number of significant PCs depends on the data structure and the context. In a case when you fail to specify `r`, it will be estimated from a permutation test (Buja and Eyuboglu, 1992) using a function permutationPA.

If `s` is not supplied, `s` is set to about 10% of `m` variables. If `B` is not supplied, `B` is set to `m*10/s`.

Value

`jackstraw_rpca` returns a list consisting of

`p.value` `m` p-values of association tests between variables and their principal components

`obs.stat` `m` observed F-test statistics

`null.stat` `s*B` null F-test statistics

Author(s)

Neo Christopher Chung <`nchchung@gmail.com`>

References

Chung and Storey (2015) Statistical significance of variables driving systematic variation in high-dimensional data. Bioinformatics, 31(4): 545-554 http://bioinformatics.oxfordjournals.org/content/31/4/545

See Also

jackstraw jackstraw_subspace permutationPA

Examples

```
set.seed(1234)
## simulate data from a latent variable model: Y = BL + E
B = c(rep(1,50),rep(-1,50), rep(0,900))
L = rnorm(20)
E = matrix(rnorm(1000*20), nrow=1000)
dat = B %*% t(L) + E
dat = t(scale(t(dat), center=TRUE, scale=TRUE))

## apply the jackstraw
out = jackstraw_rpca(dat, r=1)

## Use optional arguments
## For example, set s and B for a balance between speed of the algorithm and accuracy of p-values
## Not run:
## out = jackstraw_rpca(dat, r=1, s=10, B=1000, seed=5678)

## End(Not run)
```

---

jackstraw_subspace *Jackstraw for the User-Defined Dimension Reduction Methods*

Description

Test association between the observed variables and their latent variables, captured by a user-defined dimension reduction method.

Usage

```
jackstraw_subspace(dat, FUN, r = NULL, r1 = NULL, s = NULL,
  B = NULL, covariate = NULL, noise = NULL, verbose = TRUE,
  seed = NULL)
```

Arguments

dat	a data matrix with m rows as variables and n columns as observations.
FUN	optionally, provide a specfic function to estimate LVs. Must output r estimated LVs in a n*r matrix.

`r` a number of significant latent variables.

`r1` a numeric vector of latent variables of interest.

`s` a number of “synthetic” null variables. Out of `m` variables, `s` variables are independently permuted.

`B` a number of resampling iterations.

`covariate` a model matrix of covariates with `n` observations. Must include an intercept in the first column.

`noise` specify a parametric distribution to generate a noise term. If `NULL`, a non-parametric jackstraw test is performed.

`verbose` a logical specifying to print the computational progress.

`seed` a seed for the random number generator.

Details

This function computes `m` p-values of linear association between `m` variables and their latent variables, captured by a user-defined dimension reduction method. Its resampling strategy accounts for the over-fitting characteristics due to direct computation of PCs from the observed data and protects against an anti-conservative bias.

This function allows you to specify a parametric distribution of a noise term. It is an experimental feature. Then, a small number `s` of observed variables are replaced by synthetic null variables generated from a specified distribution.

Value

`jackstraw_subspace` returns a list consisting of

`p.value` `m` p-values of association tests between variables and their principal components

`obs.stat` `m` observed statistics

`null.stat` `s*B` null statistics

Author(s)

Neo Christopher Chung <`nchchung@gmail.com`>

References

Chung and Storey (2015) Statistical significance of variables driving systematic variation in high-dimensional data. Bioinformatics, 31(4): 545-554 `http://bioinformatics.oxfordjournals.org/content/31/4/545`

Chung (2018) Statistical significance for cluster membership. biorxiv, doi:10.1101/248633 `https://www.biorxiv.org/content/early/2018/01/16/248633`

See Also

jackstraw_pca jackstraw

Examples

```
set.seed(1234)
## simulate data from a latent variable model: Y = BL + E
B = c(rep(1,50),rep(-1,50), rep(0,900))
L = rnorm(20)
E = matrix(rnorm(1000*20), nrow=1000)
dat = B %*% t(L) + E
dat = t(scale(t(dat), center=TRUE, scale=TRUE))

## apply the jackstraw with the svd as a function
out = jackstraw_subspace(dat, FUN = function(x) svd(x)$v[,1,drop=FALSE], r=1, s=100, B=50)
```

`Jurkat293T` *A Jurkat:293T equal mixture dataset from Zheng et. al. (2017)*

Description

50

Usage

```
Jurkat293T
```

Format

A data frame with 3381 rows correspoding to single cells and 10 columns corresponding to the top 10 principal components

Source

Supplementary Data 1 from Zheng et al. (2017) https://media.nature.com/original/nature-assets/ncomms/2017/170116/ncomms14049/extref/ncomms14049-s2.xlsx

References

Zheng et al. (2017) Massively parallel digital transcriptional profiling of single cells. Nature Communications. 8:14049. DOI: 10.1038/ncomms14049

lfa.corpcor *Logistic Factor Analysis without C++ Dependency*

Description

Estimate populatoin structure in genome-wide genotype matrices.

Usage

```
lfa.corpcor(x, d, ltrace = FALSE)
```

Arguments

x	a matrix with m loci (rows) and n observations (columns).
d	a number of logistic factors.
ltrace	a logical indicator as to whether to print the progress.

Details

It performs the logistic factor analysis, similar to lfa function in the lfa package. This function works without C++ dependencies. However, it would be much slower, does not include any other LFA-related functions, checks, and warnings.

Value

lfa.corpcor returns a n*d matrix of d logistic factors. The last column is always an intercept term.

See Also

jackstraw_lfa

Examples

```
set.seed(1234)
## simulate genotype data from a logistic factor model
m=5000; n=100; pi0=.9
m0 = round(m*pi0)
m1 = m-round(m*pi0)
B = matrix(0, nrow=m, ncol=1)
B[1:m1,] = matrix(runif(m1*n, min=-.5, max=.5), nrow=m1, ncol=1)
L = matrix(rnorm(n), nrow=1, ncol=n)
BL = B %*% L
prob = exp(BL)/(1+exp(BL))

dat = matrix(rbinom(m*n, 2, as.numeric(prob)), m, n)
out = lfa.corpcor(x=dat, d=2)
```

permutationPA *Permutation Parallel Analysis*

Description

Estimate a number of significant principal components from a permutation test.

Usage

```
permutationPA(dat, B = 100, threshold = 0.05, verbose = TRUE,
  seed = NULL)
```

Arguments

`dat`	a data matrix with `m` rows as variables and `n` columns as observations.
`B`	a number (a positive integer) of resampling iterations.
`threshold`	a numeric value between 0 and 1 to threshold p-values.
`verbose`	a logical indicator as to whether to print the progress.
`seed`	a seed for the random number generator.

Details

Adopted from sva::num.sv, and based on Buja and Eyuboglu (1992)

Value

`permutationPA` returns

`p`	a list of p-values for significance of principal components
`r`	an estimated number of significant principal components based on thresholding p-values at `threshold`

References

Buja A and Eyuboglu N. (1992) Remarks on parrallel analysis. Multivariate Behavioral Research, 27(4), 509-540

pip *Compute posterior inclusion probabilities (PIPs)*

Description

Computes posterior probabilities that a feature is a true member of an assigned cluster

Usage

```
pip(pvalue, group = NULL, ...)
```

Arguments

`pvalue`	a vector of p-values.
`group`	a vector of group indicators (optional).
`...`	optional arguments to control a local FDR estimation.

Value

`pip` returns a vector of posterior inclusion probabilities

Author(s)

Neo Christopher Chung <`nchchung@gmail.com`>

Index