

Package ‘jackstraw’

May 8, 2026

Type Package

Title Statistical Inference for Unsupervised Learning

Version 1.3.21

Description Test for association between the observed data and their estimated latent variables. 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 (PCA), factor analysis (FA), K-means clustering, and related unsupervised learning 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 used again to test against that estimated latent variables. When latent variables are estimated by PCA, the jackstraw enables statistical testing for association between observed variables and latent variables, as estimated by low-dimensional principal components (PCs). This essentially leads to identifying variables that are significantly associated with PCs. Similarly, unsupervised clustering, such as K-means clustering, partition around medoids (PAM), and others, finds coherent groups in high-dimensional data. The jackstraw estimates statistical significance of cluster membership, by testing association between data and cluster centers. Clustering membership can be improved by using the resulting jackstraw p-values and posterior inclusion probabilities (PIPs), with an application to unsupervised evaluation of cell identities in single cell RNA-seq (scRNA-seq).

LazyData true

Depends R (>= 3.0.0)

Imports methods, stats, stats4, corpcor, irlba, rsvd, ClusterR,
cluster, BEDMatrix, genio (>= 1.0.15.9000)

Suggests qvalue, lfa (>= 2.0.6.9000), gctest (>= 2.0.4.9000),
testthat (>= 3.0.0)

License GPL-2

Encoding UTF-8

RoxygenNote 7.3.3

Config/testthat/edition 3

NeedsCompilation no

Author Neo Christopher Chung [aut, cre] (ORCID: <https://orcid.org/0000-0001-6798-8867>),
 John D. Storey [aut] (ORCID: <https://orcid.org/0000-0001-5992-402X>),
 Wei Hao [aut],
 Alejandro Ochoa [aut] (ORCID: <https://orcid.org/0000-0003-4928-3403>)

Maintainer Neo Christopher Chung <nchchung@gmail.com>

Repository CRAN

Date/Publication 2026-04-22 07:20:02 UTC

Contents

find_k	2
jackstraw	3
jackstraw_alstructure	4
jackstraw_cluster	6
jackstraw_irlba	8
jackstraw_kmeans	10
jackstraw_kmeanspp	11
jackstraw_lfa	13
jackstraw_MiniBatchKmeans	16
jackstraw_pam	17
jackstraw_pca	19
jackstraw_rpca	21
jackstraw_subspace	23
Jurkat293T	25
nep_est	26
permutationPA	27
pip	28
pvals_nc_chisq	29
Index	31

find_k	<i>Find a number of clusters or principal components</i>
--------	--

Description

There are a wide range of algorithms and visual techniques to identify a number of clusters or principal components embedded 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 estimated latent variables. 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 (PCA), factor analysis (FA), K-means clustering, and related unsupervised learning 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 used again to test against that estimated latent variables. When latent variables are estimated by PCA, the `jackstraw` enables statistical testing for association between observed variables and latent variables, as estimated by low-dimensional principal components (PCs). This essentially leads to identifying variables that are significantly associated with PCs. Similarly, unsupervised clustering, such as K-means clustering, partition around medoids (PAM), and others, finds coherent groups in high-dimensional data. The `jackstraw` estimates statistical significance of cluster membership, by testing association between data and cluster centers. Clustering membership can be improved by using the resulting `jackstraw` p-values and posterior inclusion probabilities (PIPs), with an application to unsupervised evaluation of cell identities in single cell RNA-seq (scRNA-seq).

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 (PCA), 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 used again to test against those 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 [doi:10.1093/bioinformatics/btu674](https://doi.org/10.1093/bioinformatics/btu674)

Chung (2020) Statistical significance of cluster membership for unsupervised evaluation of cell identities. *Bioinformatics*, 36(10): 3107–3114 [doi:10.1093/bioinformatics/btaa087](https://doi.org/10.1093/bioinformatics/btaa087)

See Also

[jackstraw_pca](#) [jackstraw_subspace](#) [jackstraw_kmeans](#) [jackstraw_cluster](#)

jackstraw_alstructure *Non-Parametric Jackstraw for ALStructure*

Description

Test association between the observed variables and population structure estimated by ALStructure.

Usage

```
jackstraw_alstructure(  
  dat,  
  r,  
  FUN,  
  r1 = NULL,  
  s = NULL,  
  B = NULL,  
  covariate = NULL,  
  verbose = TRUE  
)
```

Arguments

dat	a genotype matrix with m rows as variables and n columns as observations.
r	a number of significant LFs.
FUN	a function to ALStructure
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.

Details

This function uses ALStructure from Cabrerros and Storey (2019). A deviation dev in logistic regression (the full model with r LFs vs. the intercept-only model) is used to assess association. This function also requires the Bioconductor gctest package to be installed.

Value

jackstraw_alstructure returns a list consisting of

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

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 [doi:10.1093/bioinformatics/btu674](https://doi.org/10.1093/bioinformatics/btu674)

See Also

[jackstraw_pca](#) [jackstraw](#)

Examples

```
## Not run:  
# load genotype data to analyze (not shown) into this variable  
X  
# choose the number of ancestries  
r <- 3
```

```

# load alstructure package (install from https://github.com/StoreyLab/alstructure)
library(alstructure)
# define the function this way, a function of the genotype matrix only
FUN <- function(x) t( alstructure(x, d_hat = r)$Q_hat )

# calculate p-values (and other statistics) for each SNP
out <- jackstraw_alstructure( X, r, FUN )

## End(Not run)

```

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,
  centers,
  algorithm = function(x, centers, ...) stats::kmeans(x, centers, ...),
  s = 1,
  B = 1000,
  center = TRUE,
  noise = NULL,
  covariate = NULL,
  pool = TRUE,
  verbose = FALSE,
  ...
)

```

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.
pool	a logical specifying to pool the null statistics across all clusters. By default, TRUE.
verbose	a logical specifying to print the computational progress. By default, FALSE.
...	additional, optional arguments to <code>algorithm</code> .

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

<code>F.obs</code>	m observed F statistics between variables and cluster centers.
<code>F.null</code>	F null statistics between null variables and cluster centers, from the jackstraw method.
<code>p.F</code>	m p-values of membership.

Author(s)

Neo Christopher Chung <nchchung@gmail.com>

References

Chung (2020) Statistical significance of cluster membership for unsupervised evaluation of cell identities. *Bioinformatics*, 36(10): 3107–3114 [doi:10.1093/bioinformatics/btaa087](https://doi.org/10.1093/bioinformatics/btaa087)

jackstraw_irlba	<i>Non-Parametric Jackstraw for Principal Component Analysis (PCA) using the augmented implicitly restarted Lanczos bidiagonalization algorithm (IRLBA)</i>
-----------------	---

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,
  r = NULL,
  r1 = NULL,
  s = NULL,
  B = NULL,
  covariate = NULL,
  verbose = TRUE,
  ...
)
```

Arguments

dat	a data matrix with m rows as variables and n columns as observations.
r	a number (a positive integer) of significant principal components. See permutationPA and other methods.
r1	a numeric vector of principal components of interest. Choose a subset of r significant PCs to be used.
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.
...	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 variables and their r PCs.

You could specify a subset of significant PCs that you are interested in ($r1$). If $r1$ is given, then this function computes statistical significance of association between m variables and $r1$, while adjusting for other PCs (i.e., significant PCs that are not your interest). For example, if you want to identify variables associated with first and second PCs, when your data contains three significant PCs, set $r=3$ and $r1=c(1,2)$.

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 $10\sqrt{m}$. If B is not supplied, B is set to $m\sqrt{10}/s$.

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 [doi:10.1093/bioinformatics/btu674](https://doi.org/10.1093/bioinformatics/btu674)

See Also

[jackstraw](#) [jackstraw_subspace](#) [permutationPA](#)

Examples

```
## simulate data from a latent variable model: Y = BL + E
B = c(rep(1,10),rep(-1,10), rep(0,180))
L = rnorm(20)
E = matrix(rnorm(200*20), nrow=200)
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=200)
## 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 = FALSE,
  covariate = NULL,
  match = TRUE,
  pool = TRUE,
  verbose = FALSE,
  ...
)
```

Arguments

<code>dat</code>	a matrix with m rows as variables and n columns as observations.
<code>kmeans.dat</code>	an output from applying <code>kmeans()</code> onto <code>dat</code> .
<code>s</code>	a number of “synthetic” null variables. Out of m variables, s variables are independently permuted.
<code>B</code>	a number of resampling iterations.
<code>center</code>	a logical specifying to center the rows of the null samples. By default, TRUE.
<code>covariate</code>	a model matrix of covariates with n observations. Must include an intercept in the first column.
<code>match</code>	a logical specifying to match the observed clusters and jackstraw clusters using minimum Euclidean distances.
<code>pool</code>	a logical specifying to pool the null statistics across all clusters. By default, TRUE.
<code>verbose</code>	a logical specifying to print the computational progress. By default, FALSE.
<code>...</code>	optional arguments to control the k-means clustering algorithm (refers to <code>kmeans</code>).

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

<code>F.obs</code>	m observed F statistics between variables and cluster centers.
<code>F.null</code>	F null statistics between null variables and cluster centers, from the jackstraw method.
<code>p.F</code>	m p-values of membership.

Author(s)

Neo Christopher Chung <nchchung@gmail.com>

References

Chung (2020) Statistical significance of cluster membership for unsupervised evaluation of cell identities. *Bioinformatics*, 36(10): 3107–3114 [doi:10.1093/bioinformatics/btaa087](https://doi.org/10.1093/bioinformatics/btaa087)

Examples

```
## Not run:
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)
```

<code>jackstraw_kmeanspp</code>	<i>Non-Parametric Jackstraw for K-means Clustering using RcppArmadillo</i>
---------------------------------	--

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,
  ...
)
```

Arguments

<code>dat</code>	a matrix with m rows as variables and n columns as observations.
<code>kmeans.dat</code>	an output from applying <code>ClusterR::KMeans_rcpp</code> onto <code>dat</code> .
<code>s</code>	a number of “synthetic” null variables. Out of m variables, s variables are independently permuted.
<code>B</code>	a number of resampling iterations.
<code>center</code>	a logical specifying to center the rows. By default, <code>TRUE</code> .
<code>covariate</code>	a model matrix of covariates with n observations. Must include an intercept in the first column.
<code>verbose</code>	a logical specifying to print the computational progress. By default, <code>FALSE</code> .
<code>pool</code>	a logical specifying to pool the null statistics across all clusters. By default, <code>TRUE</code> .
<code>...</code>	optional arguments to control the k-means clustering algorithm (refers to <code>ClusterR::KMeans_rcpp</code>).

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

<code>F.obs</code>	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 (2020) Statistical significance of cluster membership for unsupervised evaluation of cell identities. *Bioinformatics*, 36(10): 3107–3114 doi:[10.1093/bioinformatics/btaa087](https://doi.org/10.1093/bioinformatics/btaa087)

Examples

```
## Not run:
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,
  r,
  FUN,
  r1 = NULL,
  s = NULL,
  B = NULL,
  covariate = NULL,
  permute_alleles = TRUE,
  verbose = TRUE
)
```

Arguments

<code>dat</code>	either a genotype matrix with m rows as variables and n columns as observations, or a <code>BEDMatrix</code> object (see package <code>BEDMatrix</code> , these objects are transposed compared to the above but this works fine as-is, see example, no need to modify a <code>BEDMatrix</code> input). A <code>BEDMatrix</code> input triggers a low-memory mode where permuted data is also written and processed from disk, whereas a regular matrix input stores permutations in memory. The tradeoff is <code>BEDMatrix</code> version typically runs considerably slower, but enables analysis of very large data that is otherwise impossible.
<code>r</code>	a number of significant LFs.
<code>FUN</code>	a function to use for LFA.
<code>r1</code>	a numeric vector of LFs of interest (implying you are not interested in all r LFs).
<code>s</code>	a number of “synthetic” null variables. Out of m variables, s variables are independently permuted.
<code>B</code>	a number of resampling iterations. There will be a total of $s*B$ null statistics.
<code>covariate</code>	a data matrix of covariates with corresponding n observations (do not include an intercept term).
<code>permute_alleles</code>	If <code>TRUE</code> (default), alleles (rather than genotypes) are permuted, which results in a more Binomial synthetic null when data is highly structured. Changing to <code>FALSE</code> is not recommended, except for research purposes to confirm that it performs worse than the default.
<code>verbose</code>	a logical specifying to print the computational progress.

Details

This function uses logistic factor analysis (LFA) from Hao et al. (2016). Particularly, the deviance in logistic regression (the full model with r LFs vs. the intercept-only model) is used to assess significance. This function requires the `gctest` package, and in practice also the `lfa` package, to be installed from Bioconductor.

The random outputs of the regular matrix versus the `BEDMatrix` versions are equal in distribution. However, fixing a seed and providing the same data to both versions does not result in the same exact outputs. This is because the `BEDMatrix` version permutes loci in a different order by necessity.

Value

`jackstraw_lfa` returns a list consisting of

<code>p.value</code>	m p-values of association tests between variables and their LFs
<code>obs.stat</code>	m observed deviances
<code>null.stat</code>	$s*B$ null deviances

Author(s)

Neo Christopher Chung <nchchung@gmail.com>

Alejandro Ochoa <alejandro.ochoa@duke.edu>

References

Chung and Storey (2015) Statistical significance of variables driving systematic variation in high-dimensional data. *Bioinformatics*, 31(4): 545-554 [doi:10.1093/bioinformatics/btu674](https://doi.org/10.1093/bioinformatics/btu674)

See Also

[jackstraw_pca\(\)](#) [jackstraw](#) [jackstraw_subspace\(\)](#)
[pvals_nc_chisq\(\)](#) for calculating high-precision p-values from jackstraw_lfa outputs if desired.

Examples

```
## 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=n)
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)

# load lfa package (install from Bioconductor)
library(lfa)
# choose the number of logistic factors, including the intercept
r <- 2
# define the function this way, a function of the genotype matrix only
FUN <- function(x) lfa( x, r )

## apply the jackstraw_lfa
out <- jackstraw_lfa( dat, r, FUN )

# if you had very large genotype data in plink BED/BIM/FAM files,
# use BEDMatrix and save memory by reading from disk (at the expense of speed)
library(BEDMatrix)
dat_BM <- BEDMatrix( 'filepath' ) # assumes filepath.bed, .bim and .fam exist
# run jackstraw!
out <- jackstraw_lfa( dat_BM, r, FUN )

# if desired, this calculates new p-values with much higher precision for highly significant cases
out2 <- pvals_nc_chisq( out, r-1 )
# these are the desired p-values
out2$p.value
# and non-centrality estimates from two methods
out2$ncp

## 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,
  batch_size = floor(nrow(dat)/100),
  initializer = "kmeans++",
  pool = TRUE,
  ...
)
```

Arguments

<code>dat</code>	a data matrix with m rows as variables and n columns as observations.
<code>MiniBatchKmeans.output</code>	an output from applying <code>ClusterR::MiniBatchKmeans()</code> onto <code>dat</code> . This provides more controls over the algorithm and subsequently the initial centroids used.
<code>s</code>	a number of “synthetic” null variables. Out of m variables, s variables are independently permuted.
<code>B</code>	a number of resampling iterations.
<code>center</code>	a logical specifying to center the rows. By default, <code>TRUE</code> .
<code>covariate</code>	a model matrix of covariates with n observations. Must include an intercept in the first column.
<code>verbose</code>	a logical specifying to print the computational progress. By default, <code>FALSE</code> .
<code>batch_size</code>	the size of the mini batches.
<code>initializer</code>	the method of initialization. By default, <code>kmeans++</code> .
<code>pool</code>	a logical specifying to pool the null statistics across all clusters. By default, <code>TRUE</code> .
<code>...</code>	optional arguments to control the Mini Batch K-means clustering algorithm (refers to <code>ClusterR::MiniBatchKmeans</code>).

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 (2020) Statistical significance of cluster membership for unsupervised evaluation of cell identities. *Bioinformatics*, 36(10): 3107–3114 [doi:10.1093/bioinformatics/btaa087](https://doi.org/10.1093/bioinformatics/btaa087)

Examples

```
## Not run:
library(ClusterR)
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,
  ...
)
```

Arguments

<code>dat</code>	a matrix with m rows as variables and n columns as observations.
<code>pam.dat</code>	an output from applying <code>cluster::pam()</code> on <code>dat</code> .
<code>s</code>	a number of “synthetic” null variables. Out of m variables, s variables are independently permuted.
<code>B</code>	a number of resampling iterations.
<code>center</code>	a logical specifying to center the rows. By default, <code>TRUE</code> .
<code>covariate</code>	a model matrix of covariates with n observations. Must include an intercept in the first column.
<code>verbose</code>	a logical specifying to print the computational progress. By default, <code>FALSE</code> .
<code>pool</code>	a logical specifying to pool the null statistics across all clusters. By default, <code>TRUE</code> .
<code>...</code>	optional arguments to control the k-means clustering algorithm (refers to <code>kmeans</code>).

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

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

Author(s)

Neo Christopher Chung <nchchung@gmail.com>

References

Chung (2020) Statistical significance of cluster membership for unsupervised evaluation of cell identities. *Bioinformatics*, 36(10): 3107–3114 doi:[10.1093/bioinformatics/btaa087](https://doi.org/10.1093/bioinformatics/btaa087)

Examples

```
## Not run:
library(cluster)
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,
  r = NULL,
  r1 = NULL,
  s = NULL,
  B = NULL,
  covariate = NULL,
  verbose = TRUE
)
```

Arguments

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

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

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

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.

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 variables and their r PCs.

You could specify a subset of significant PCs that you are interested in (r1). If r1 is given, then this function computes statistical significance of association between m variables and r1, while adjusting for other PCs (i.e., significant PCs that are not your interest). For example, if you want to identify variables associated with first and second PCs, when your data contains three significant PCs, set r=3 and r1=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. 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 [doi:10.1093/bioinformatics/btu674](https://doi.org/10.1093/bioinformatics/btu674)

See Also

[jackstraw](#) [jackstraw_subspace](#) [permutationPA](#)

Examples

```

## Not run:
## 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)

## End(Not run)

```

jackstraw_rpca	<i>Non-Parametric Jackstraw for Principal Component Analysis (PCA) using Randomized Singular Value Decomposition</i>
----------------	--

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,
  r = NULL,
  r1 = NULL,
  s = NULL,
  B = NULL,
  covariate = NULL,
  verbose = TRUE,
  ...
)

```

Arguments

dat	a data matrix with m rows as variables and n columns as observations.
r	a number (a positive integer) of significant principal components. See permutationPA and other methods.
r1	a numeric vector of principal components of interest. Choose a subset of r significant PCs to be used.

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.
...	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 variables and their r PCs.

You could specify a subset of significant PCs that you are interested in (r1). If r1 is given, then this function computes statistical significance of association between m variables and r1, while adjusting for other PCs (i.e., significant PCs that are not your interest). For example, if you want to identify variables associated with first and second PCs, when your data contains three significant PCs, set r=3 and r1=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\ 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 [doi:10.1093/bioinformatics/btu674](https://doi.org/10.1093/bioinformatics/btu674)

See Also

[jackstraw](#) [jackstraw_subspace](#) [permutationPA](#)

Examples

```

## simulate data from a latent variable model: Y = BL + E
B = c(rep(1,10),rep(-1,10), rep(0,180))
L = rnorm(20)
E = matrix(rnorm(200*20), nrow=200)
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=200)

## 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,
  r,
  FUN,
  r1 = NULL,
  s = NULL,
  B = NULL,
  covariate = NULL,
  noise = NULL,
  verbose = TRUE
)

```

Arguments

dat	a data matrix with m rows as variables and n columns as observations.
r	a number of significant latent variables.
FUN	Provide a specific function to estimate LVs. Must output r estimated LVs in a n*r matrix.

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.

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 [doi:10.1093/bioinformatics/btu674](https://doi.org/10.1093/bioinformatics/btu674)

Chung (2020) Statistical significance of cluster membership for unsupervised evaluation of cell identities. *Bioinformatics*, 36(10): 3107–3114 [doi:10.1093/bioinformatics/btaa087](https://doi.org/10.1093/bioinformatics/btaa087)

See Also

[jackstraw_pca](#) [jackstraw](#)

Examples

```
## 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%:50% Jurkat:293T

Usage

Jurkat293T

Format

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

Source

Supplementary Data 1 from Zheng et al. (2017) https://static-content.springer.com/esm/art%3A10.1038%2Fncomms14049/MediaObjects/41467_2017_BFncomms14049_M0ESM829_ESM.xlsx

References

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

`ncp_est`*Estimate parameter of non-central Chi squared distributed data*

Description

This function estimates the non-centrality parameter (NCP) of data assuming a non-central chi squared distribution with known degrees of freedom. The function first uses the method of moments (MOM) to get a rough estimate of the NCP, then uses that as a starting point for calculating the maximum likelihood estimate (MLE).

Usage

```
ncp_est(x, df)
```

Arguments

- | | |
|-----------------|---|
| <code>x</code> | The vector of data whose parameter we are interested in. The data is assumed to follow a non-central chi squared distribution, and appropriate inputs are deviance or likelihood ratio test statistics, or other tests that ordinarily follow a central chi-squared distribution but become non-central due to "double dipping" (where fixed variables being tested were in fact estimated from the same data). Any values that are NA, infinities, or negatives are ignored (removed before all calculations). |
| <code>df</code> | The degrees of freedom of the test used to calculate the statistics. |

Details

There is one case where this function does not calculate MLEs directly, due to numerical issues, namely when the MOM estimate is negative, in which case estimates of zero for both are returned. Negative MOM NCP estimates arise in two common cases. First, the data has a central chi squared distribution, in which case the NCP estimate may take on small negative values, and overall these estimates have a mean of zero and variance dependent on sample size; in this case the return values of zero are appropriate. Second, the data may not have a central or non-central chi squared distribution at all, in which case the estimation model is completely misspecified; we opted to return zero estimates quietly, without issuing errors or warnings, as this case overlaps with central chi-squared data and it's not trivial to tell the difference. This function does not aim to detect misspecified data, it is up to the researcher to compare the model against the observed statistics for goodness of fit, for example using q-q plots.

Value

A numeric vector with two values, namely the MLE and MOM estimates.

See Also

[pvals_nc_chisq\(\)](#), a wrapper around this function that facilitates p-value calculation for Jackstraw objects.

Examples

```
# true parameters of toy data
df <- 1
ncp_true <- 1
x <- rchisq( 100, df, ncp_true )

# get estimates!
ncp_ests <- ncp_est( x, df )
# this is MLE
ncp_ests[1]
# this is MOM estimate
ncp_ests[2]
```

permutationPA	<i>Permutation Parallel Analysis</i>
---------------	--------------------------------------

Description

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

Usage

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

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.

Details

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

Value

permutationPA returns

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

References

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

pip *Compute posterior inclusion probabilities (PIPs)*

Description

From a set of p-values, computes posterior probabilities that a feature should be truly included. For example, membership inclusion in a given cluster can be improved by filtering low quality members. In using PCA and related methods, it helps select variables that are truly associated with given latent variables.

Usage

```
pip(pvalue, group = NULL, pi0 = NULL, verbose = TRUE, ...)
```

Arguments

pvalue	a vector of p-values.
group	a vector of group indicators (optional). If provided, PIP analysis is stratified. Assumes groups are in 1:k where k is the number of unique groups.
pi0	a vector of pi0 values (optional). Its length has to be either 1 or equal the number of groups.
verbose	If TRUE, reports information.
...	optional arguments for lfd to control a local FDR estimation.

Details

This function requires the Bioconductor `qvalue` package to be installed.

Value

pip returns a vector of posterior inclusion probabilities

Author(s)

Neo Christopher Chung <nchchung@gmail.com> John R. Yamamoto-Wilson

References

Chung (2020) Statistical significance of cluster membership for unsupervised evaluation of cell identities. *Bioinformatics*, 36(10): 3107–3114 [doi:10.1093/bioinformatics/btaa087](https://doi.org/10.1093/bioinformatics/btaa087)

Chung (2014) "Jackstraw Weighted Shrinkage for Principal Component Analysis and Covariance Matrix" in *Statistical Inference of Variables Driving Systematic Variation in High-Dimensional Biological Data*. PhD thesis, Princeton University. <https://www.proquest.com/openview/e90b562d689cf3a021c35a93c6f31?pq-origsite=scholar&cbl=18750>

pvals_nc_chisq	<i>Calculate high precision p-values from Jackstraw data under non-central chi squared distribution for null statistics</i>
----------------	---

Description

This function takes a Jackstraw object for convenience, estimates the non-centrality parameter (NCP) from the null statistics (using `ncp_est()`), then uses this non-central chi squared model to calculate p-values for the observed statistics. The goal is to be able to calculate small p-values with arbitrary precision, which is normally not possible with the ordinary Jackstraw functions since the minimum empirical p-value is determined by the inverse of the number of null samples ($1 / (s * B)$) where s and B are Jackstraw parameters (see `jackstraw_lfa()`). In contrast, large p-values are typically similar between empirical and NCP model versions, and in fact this must hold if the null model is correctly specified. This model works well empirically for `jackstraw_lfa()` and `jackstraw_alstructure()`, whose statistics are deviances that would ordinarily be central chi-squared distributed were the test factors not estimated from the same data. This model is not appropriate for data from `jackstraw_pca()`, whose statistics have a roughly F distribution, and possibly many other functions from this package.

Usage

```
pvals_nc_chisq(out, df, null.stat = out$null.stat, obs.stat = out$obs.stat)
```

Arguments

out	The object returned by <code>jackstraw_lfa()</code> and <code>jackstraw_alstructure()</code> . In particular, a list with at least the following two named elements: <code>null.stat</code> containing the null statistics on which the NCP null model is estimated, and <code>obs.stat</code> containing the observed or actual statistics for which p-values are calculated.
df	The degrees of freedom of the test used to calculate the statistics. Note that for ordinary jackstraw runs this is $r-1$, since the null model/intercept is counted among the r latent variables, but alternate values can be provided for more complex tests.
null.stat	Alternate way to provide these values if the input is not a Jackstraw object, so out is not available.
obs.stat	Alternate way to provide these values if the input is not a Jackstraw object, so out is not available.

Value

A list with two named values:

- `p.value`: the vector of p-values calculated for the input vector `obs.stat`
- `ncp`: the numeric vector of two values of NCP estimates calculated from `null.stat`, namely the maximum likelihood (MLE) and method-of-moments estimates (see `ncp_est()`). Only MLE is used to calculate p-values.

See Also

[ncp_est\(\)](#) for more notes on estimating non-centrality parameters only.
[jackstraw_lfa\(\)](#), [jackstraw_alstructure\(\)](#)

Examples

```
# instead of running jackstraw here, we simulate toy data
df <- 1
ncp_true <- 1
# null data are truly non-central chi squared distributed
# observed data are the same but with huge power reflected in a larger NCP
out <- list(
  null.stat = rchisq( 100, df, ncp_true ),
  obs.stat = rchisq( 11, df, ncp_true + 10 )
)

# this calculates new p-values with much higher precision for highly significant cases
out2 <- pvals_nc_chisq( out, df )
# these are the desired p-values
out2$p.value
# and NCP estimates from two methods
out2$ncp

## Not run:
# This is the more typical usage in practice
# first run Jackstraw-LFA, which returns empirical p-values with limited precision and raw stats
out <- jackstraw_lfa( dat, r, FUN = function(x) lfa( x, r ) )
# then fit NCP model and use it to calculate high-precision p-values!
out2 <- pvals_nc_chisq( out, r-1 )

# compare p-values! In linear scale they should agree very well
plot( out$p.value, out2$p.value, pch = '.' )
# but in log scale it's clear the NCP values can take on much smaller values
plot( out$p.value, out2$p.value, pch = '.', log = 'xy' )

## End(Not run)
```

Index

* datasets

Jurkat293T, 25

find_k, 2

irlba, 8

jackstraw, 3, 5, 9, 15, 20, 22, 24

jackstraw-package (jackstraw), 3

jackstraw_alstructure, 4

jackstraw_alstructure(), 29, 30

jackstraw_cluster, 4, 6

jackstraw_irlba, 8

jackstraw_kmeans, 3, 4, 10

jackstraw_kmeanspp, 11

jackstraw_lfa, 13

jackstraw_lfa(), 29, 30

jackstraw_MinibatchKmeans, 16

jackstraw_pam, 17

jackstraw_pca, 3–5, 19, 24

jackstraw_pca(), 15, 29

jackstraw_rpca, 21

jackstraw_subspace, 4, 9, 20, 22, 23

jackstraw_subspace(), 15

Jurkat293T, 25

kmeans, 6

lfdr, 28

nep_est, 26

nep_est(), 29, 30

permutationPA, 8, 9, 19–22, 27

pip, 4, 28

pvals_nc_chisq, 29

pvals_nc_chisq(), 15, 26

rsvd, 21