Package 'scPCA'

October 24, 2025

Title Sparse Contrastive Principal Component Analysis

Version 1.23.0

Description A toolbox for sparse contrastive principal component analysis (scPCA) of high-dimensional biological data. scPCA combines the stability and interpretability of sparse PCA with contrastive PCA's ability to disentangle biological signal from unwanted variation through the use of control data. Also implements and extends cPCA.

Depends R (>= 4.0.0)

Imports stats, methods, assertthat, tibble, dplyr, purrr, stringr, Rdpack, matrixStats, BiocParallel, elasticnet, sparsepca, cluster, kernlab, origami, RSpectra, coop, Matrix, DelayedArray, ScaledMatrix, MatrixGenerics

Suggests DelayedMatrixStats, sparseMatrixStats, testthat (>= 2.1.0), covr, knitr, rmarkdown, BiocStyle, ggplot2, ggpubr, splatter, SingleCellExperiment, microbenchmark

License MIT + file LICENSE

URL https://github.com/PhilBoileau/scPCA

BugReports https://github.com/PhilBoileau/scPCA/issues

Encoding UTF-8

LazyData true

VignetteBuilder knitr

RoxygenNote 7.3.1

RdMacros Rdpack

biocViews PrincipalComponent, GeneExpression, DifferentialExpression, Sequencing, Microarray, RNASeq

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

git_branch devel

git_last_commit 18cc13d

git_last_commit_date 2025-04-15

Repository Bioconductor 3.22

Date/Publication 2025-10-23

Author Philippe Boileau [aut, cre, cph] (ORCID:
<https://orcid.org/0000-0002-4850-2507>),
Nima Hejazi [aut] (ORCID: <https://orcid.org/0000-0002-7127-2789>),
Sandrine Dudoit [ctb, ths] (ORCID:
<https://orcid.org/0000-0002-6069-8629>)

Maintainer Philippe Boileau <philippe_boileau@berkeley.edu>

Contents

`background_df` *Simulated Background Data for cPCA and scPCA*

Description

The background data consisting of 400 observations and 30 variables was simulated as follows:

- Each of the first 10 variables was drawn from $N(0, 10)$
- Variables 11 through 20 were drawn from $N(0, 3)$
- Variables 21 through 30 were drawn from $N(0, 1)$

Usage

```
data(background_df)
```

Format

A simple `data.frame`.

Examples

```
data(background_df)
```

bpContrastiveCov *Parallelized Contrastive Covariance Matrices*

Description

Compute the list of contrastive covariance matrices in parallel using bplapply.

Usage

```
bpContrastiveCov(
  target,
  background,
  contrasts,
  center,
  scale,
  scaled_matrix = FALSE
)
```

Arguments

target	The target (experimental) data set, in a standard format such as a data.frame or matrix.
background	The background data set, in a standard format such as a data.frame or matrix.
contrasts	A numeric vector of the contrastive parameters.
center	A logical indicating whether the target and background data sets should be centered to mean zero.
scale	A logical indicating whether the target and background data sets should be scaled to unit variance.
scaled_matrix	A logical indicating whether to output a ScaledMatrix object. The centering and scaling procedure is delayed until later, permitting more efficient matrix multiplication and row or column sums downstream. However, this comes at the at the cost of numerical precision. Defaults to FALSE.

Value

A list of contrastive covariance matrices. Each element has an associated contrastive parameter in the contrasts vector.

bpFitCPCA *Contrastive Principal Component Analysis in Parallel*

Description

Given target and background dataframes or matrices, cPCA will perform contrastive principal component analysis (cPCA) of the target data for a given number of eigenvectors and a vector of real valued contrast parameters. This is identical to the implementation of cPCA method by Abid et al. Abid et al. (2018). Analogous to fitCPCA, but replaces all lapply calls by bplapply.

Usage

```
bpFitCPCA(
  target,
  center,
  scale,
  c_contrasts,
  contrasts,
  n_eigen,
  n_medoids,
  eigdecomp_tol,
  eigdecomp_iter
)
```

Arguments

target	The target (experimental) data set, in a standard format such as a data.frame or matrix.
center	A logical indicating whether the target and background data sets should be centered to mean zero.
scale	A logical indicating whether the target and background data sets should be scaled to unit variance.
c_contrasts	A list of contrastive covariances.
contrasts	A numeric vector of the contrastive parameters used to compute the contrastive covariances.
n_eigen	A numeric indicating the number of eigenvectors to be computed.
n_medoids	A numeric indicating the number of medoids to consider.
eigdecomp_tol	A numeric providing the level of precision used by eigendecompositon calculations. Defaults to 1e-10.
eigdecomp_iter	A numeric indicating the maximum number of interations performed by eigendecompositon calculations. Defaults to 1000.

Value

A list of lists containing the cPCA results for each contrastive parameter deemed to be a medoid.

- rotation - the list of matrices of variable loadings
- x - the list of rotated data, centred and scaled if requested, multiplied by the rotation matrix
- contrast - the list of contrastive parameters
- penalty - set to zero, since loadings are not penalized in cPCA

References

Abid A, Zhang MJ, Bagaria VK, Zou J (2018). "Exploring patterns enriched in a dataset with contrastive principal component analysis." *Nature communications*, **9**(1), 2134.

bpFitGrid *Identify the Optimal Contrastive and Penalty Parameters in Parallel*

Description

This function is used to automatically select the optimal contrastive parameter and L1 penalty term for scPCA based on a clustering algorithm and average silhouette width. Analogous to `fitGrid`, but replaces all `lapply` calls by `bplapply`.

Usage

```
bpFitGrid(
  target,
  target_valid = NULL,
  center,
  scale,
  c_contrasts,
  contrasts,
  penalties,
  n_eigen,
  alg,
  clust_method = c("kmeans", "pam", "hclust"),
  n_centers,
  max_iter = 10,
  linkage_method = "complete",
  clusters = NULL,
  eigdecomp_tol = 1e-10,
  eigdecomp_iter = 1000
)
```

Arguments

`target`	The target (experimental) data set, in a standard format such as a `data.frame` or `matrix`.
`target_valid`	A holdout set of the target (experimental) data set, in a standard format such as a `data.frame` or `matrix`. `NULL` by default but used by `cvSelectParams` for cross-validated selection of the contrastive and penalization parameters.
`center`	A `logical` indicating whether the target and background data sets should be centered to mean zero.
`scale`	A `logical` indicating whether the target and background data sets should be scaled to unit variance.
`c_contrasts`	A `list` of contrastive covariances.
`contrasts`	A `numeric` vector of the contrastive parameters used to compute the contrastive covariances.
`penalties`	A `numeric` vector of the penalty terms.
`n_eigen`	A `numeric` indicating the number of eigenvectors to be computed.
`alg`	A `character` indicating the SPCA algorithm used to sparsify the contrastive loadings. Currently supports `iterative` for the Zou et al. (2006) implemententation, `var_proj` for the non-randomized Erichson et al. (2018) solution, and `rand_var_proj` fir the randomized Erichson et al. (2018) result.

clust_method	A character specifying the clustering method to use for choosing the optimal constrastive parameter. Currently, this is limited to either k-means, partitioning around medoids (PAM), and hierarchical clustering. The default is k-means clustering.
n_centers	A numeric giving the number of centers to use in the clustering algorithm.
max_iter	A numeric giving the maximum number of iterations to be used in k-means clustering, defaulting to 10.
linkage_method	A character specifying the agglomerative linkage method to be used if clust_method = "hclust". The options are ward.D2, single, complete, average, mcquitty, median, and centroid. The default is complete.
clusters	A numeric vector of cluster labels for observations in the target data. Defaults to NULL, but is otherwise used to identify the optimal set of hyperparameters when fitting the scPCA and the automated version of cPCA.
eigdecomp_tol	A numeric providing the level of precision used by eigendecompositon calculations. Defaults to 1e-10.
eigdecomp_iter	A numeric indicating the maximum number of interations performed by eigendecompositon calculations. Defaults to 1000.

Value

A list similar to that output by prcomp:

- rotation - the matrix of variable loadings
- x - the rotated data, centred and scaled, if requested, data multiplied by the rotation matrix
- contrast - the optimal contrastive parameter
- penalty - the optimal L1 penalty term

References

Erichson NB, Zeng P, Manohar K, Brunton SL, Kutz JN, Aravkin AY (2018). "Sparse Principal Component Analysis via Variable Projection." *ArXiv*, **abs/1804.00341**.

Zou H, Hastie T, Tibshirani R (2006). "Sparse principal component analysis." *Journal of computational and graphical statistics*, **15**(2), 265–286.

---

checkArgs *Check Arguments passed to the scPCA Function*

Description

Checks whether or not the all arguments in the scPCA functions are input properly.

Usage

```
checkArgs(
  target,
  background,
  center,
  scale,
  n_eigen,
  contrasts,
  penalties,
  clust_method,
  linkage_method,
  clusters,
  eigdecomp_tol,
  eigdecomp_iter,
  n_centers,
  scaled_matrix
)
```

Arguments

target	The target (experimental) data set, in a standard format such as a data.frame or matrix.
background	The background data set, in a standard format such as a data.frame or matrix.
center	A logical indicating whether the target and background data sets should be centered to mean zero.
scale	A logical indicating whether the target and background data sets should be scaled to unit variance.
n_eigen	A numeric indicating the number of eigenvectors to be computed.
contrasts	A numeric vector of the contrastive parameters.
penalties	A numeric vector of the penalty terms.
clust_method	A character specifying the clustering method to use for choosing the optimal constrastive parameter. Currently, this is limited to either k-means, partitioning around medoids (PAM), and hierarchical clustering. The default is k-means clustering.
linkage_method	A character specifying the agglomerative linkage method to be used if clust_method = "hclust". The options are ward.D2, single, complete, average, mcquitty, median, and centroid. The default is complete.
clusters	A numeric vector of cluster labels for observations in the target data. Defaults to NULL, but is otherwise used to identify the optimal set of hyperparameters when fitting the scPCA and the automated version of cPCA.
eigdecomp_tol	A numeric providing the level of precision used by eigendecompositon calculations.
eigdecomp_iter	A numeric indicating the maximum number of interations performed by eigendecompositon calculations.
n_centers	A numeric giving the number of centers to use in the clustering algorithm. If set to 1, cPCA, as first proposed by Erichson et al. (2018), is performed, regardless of what the penalties argument is set to.

scaled_matrix	A logical indicating whether to output a ScaledMatrix object. The centering and scaling procedure is delayed until later, permitting more efficient matrix multiplication and row or column sums downstream. However, this comes at the at the cost of numerical precision.

Value

Whether all argument conditions are satisfied

References

Erichson NB, Zeng P, Manohar K, Brunton SL, Kutz JN, Aravkin AY (2018). "Sparse Principal Component Analysis via Variable Projection." *ArXiv*, **abs/1804.00341**.

---

contrastiveCov *Contrastive Covariance Matrices*

Description

Compute the list of contrastive covariance matrices.

Usage

```
contrastiveCov(
  target,
  background,
  contrasts,
  center,
  scale,
  scaled_matrix = FALSE
)
```

Arguments

target	The target (experimental) data set, in a standard format such as a data.frame or matrix.
background	The background data set, in a standard format such as a data.frame or matrix.
contrasts	A numeric vector of the contrastive parameters.
center	A logical indicating whether the target and background data sets should be centered to mean zero.
scale	A logical indicating whether the target and background data sets should be scaled to unit variance.
scaled_matrix	A logical indicating whether to output a ScaledMatrix object. The centering and scaling procedure is delayed until later, permitting more efficient matrix multiplication and row or column sums downstream. However, this comes at the at the cost of numerical precision. Defaults to FALSE.

Value

A list of contrastive covariance matrices. Each element has an associated contrastive parameter in the contrasts vector.

covMat *Compute Sample Covariance Matrix*

Description

covMat computes the sample covariance matrix of a data set. If a variable in the dataset has zero variance, then its corresponding row and column in the covariance matrix are zero vectors.

Usage

```
covMat(data, center = TRUE, scale = TRUE, scaled_matrix = FALSE)
```

Arguments

data	The data for which to compute the sample covariance matrix.
center	A logical indicating whether the target and background data sets should be centered to mean zero.
scale	A logical indicating whether the target and background data sets should be scaled to unit variance.
scaled_matrix	A logical indicating whether to output a ScaledMatrix object. The centering and scaling procedure is delayed until later, permitting more efficient matrix multiplication and row or column sums downstream. However, this comes at the at the cost of numerical precision. Defaults to FALSE.

Value

the covariance matrix of the data.

cvSelectParams *Fold-Specific Selection of Contrastive and Penalization Parameters*

Description

A wrapper function for fitting various internal functions to select the optimal setting of the contrastive and penalization parameters via cross-validation. For internal use only.

Usage

```
cvSelectParams(
  fold,
  target,
  background,
  center,
  scale,
  n_eigen,
  alg = alg,
  contrasts,
  penalties,
  clust_method,
```

```
  n_centers,
  max_iter,
  linkage_method,
  n_medoids,
  parallel,
  clusters,
  eigdecomp_tol,
  eigdecomp_iter,
  scaled_matrix
)
```

Arguments

fold	Object specifying cross-validation folds as generated by a call to make_folds.
target	The target (experimental) data set, in a standard format such as a data.frame or matrix.
background	The background data set, in a standard format such as a data.frame or matrix. Note that the number of features must match the number of features in the target data.
center	A logical indicating whether the target and background data sets should be centered to mean zero.
scale	A logical indicating whether the target and background data sets should be scaled to unit variance.
n_eigen	A numeric indicating the number of eigenvectors (or sparse contrastive components) to be computed. The default is to compute two such eigenvectors.
alg	A character indicating the SPCA algorithm used to sparsify the contrastive loadings. Currently supports iterative for the Zou et al. (2006) implementation, var_proj for the non-randomized Erichson et al. (2018) solution, and rand_var_proj for the randomized Erichson et al. (2018) result.
contrasts	A numeric vector of the contrastive parameters. Each element must be a unique non-negative real number. The default is to use 40 logarithmically spaced values between 0.1 and 1000.
penalties	A numeric vector of the L1 penalty terms on the loadings. The default is to use 20 equidistant values between 0.05 and 1.
clust_method	A character specifying the clustering method to use for choosing the optimal contrastive parameter. Currently, this is limited to either k-means, partitioning around medoids (PAM), and hierarchical clustering. The default is k-means clustering.
n_centers	A numeric giving the number of centers to use in the clustering algorithm. If set to 1, cPCA, as first proposed by Abid et al., is performed, regardless of what the penalties argument is set to.
max_iter	A numeric giving the maximum number of iterations to be used in k-means clustering, defaulting to 10.
linkage_method	A character specifying the agglomerative linkage method to be used if clust_method = "hclust". The options are ward.D2, single, complete, average, mcquitty, median, and centroid. The default is complete.
n_medoids	A numeric indicating the number of medoids to consider if n_centers is set to 1. The default is 8 such medoids.

parallel	A logical indicating whether to invoke parallel processing via the **BiocParallel** infrastructure. The default is FALSE for sequential evaluation.
clusters	A numeric vector of cluster labels for observations in the target data. Defaults to NULL, but is otherwise used to identify the optimal set of hyperparameters when fitting the scPCA and the automated version of cPCA.
eigdecomp_tol	A numeric providing the level of precision used by eigendecompositon calculations. Defaults to 1e-10.
eigdecomp_iter	A numeric indicating the maximum number of interations performed by eigendecompositon calculations. Defaults to 1000.
scaled_matrix	A logical indicating whether to output a ScaledMatrix object. The centering and scaling procedure is delayed until later, permitting more efficient matrix multiplication and row or column sums downstream. However, this comes at the at the cost of numerical precision.

Value

Output structure matching either that of fitCPCA or fitGrid (or their parallelized variants, namely either bpFitCPCA and link{bpFitGrid}, respectively).

References

Erichson NB, Zeng P, Manohar K, Brunton SL, Kutz JN, Aravkin AY (2018). "Sparse Principal Component Analysis via Variable Projection." *ArXiv*, **abs/1804.00341**.

Zou H, Hastie T, Tibshirani R (2006). "Sparse principal component analysis." *Journal of computational and graphical statistics*, **15**(2), 265–286.

---

fitCPCA *Contrastive Principal Component Analysis*

Description

Given target and background dataframes or matrices, cPCA will perform contrastive principal component analysis (cPCA) of the target data for a given number of eigenvectors and a vector of real valued contrast parameters. This is identical to the implementation of cPCA method of Abid et al. (2018).

Usage

```
fitCPCA(
  target,
  center,
  scale,
  c_contrasts,
  contrasts,
  n_eigen,
  n_medoids,
  eigdecomp_tol,
  eigdecomp_iter
)
```

Arguments

target	The target (experimental) data set, in a standard format such as a data.frame or matrix.
center	A logical indicating whether the target and background data sets should be centered to mean zero.
scale	A logical indicating whether the target and background data sets should be scaled to unit variance.
c_contrasts	A list of contrastive covariances.
contrasts	A numeric vector of the contrastive parameters used to compute the contrastive covariances.
n_eigen	A numeric indicating the number of eigenvectors to be computed.
n_medoids	A numeric indicating the number of medoids to consider. Not used if contrasts is a single value.
eigdecomp_tol	A numeric providing the level of precision used by eigendecompositon calculations. Defaults to 1e-10.
eigdecomp_iter	A numeric indicating the maximum number of interations performed by eigendecompositon calculations. Defaults to 1000.

Value

A list of lists containing the cPCA results for each contrastive parameter deemed to be a medoid.

- rotation - the list of matrices of variable loadings
- x - the list of rotated data, centred and scaled if requested, multiplied by the rotation matrix
- contrast - the list of contrastive parameters
- penalty - set to zero, since loadings are not penalized in cPCA

References

Abid A, Zhang MJ, Bagaria VK, Zou J (2018). "Exploring patterns enriched in a dataset with contrastive principal component analysis." *Nature communications*, **9**(1), 2134.

---

fitGrid *Identify the Optimal Contrastive and Penalty Parameters*

Description

This function is used to automatically select the optimal contrastive parameter and L1 penalty term for scPCA based on a clustering algorithm and average silhouette width.

Usage

```
fitGrid(
  target,
  target_valid = NULL,
  center,
  scale,
  c_contrasts,
  contrasts,
  alg,
  penalties,
  n_eigen,
  clust_method = c("kmeans", "pam", "hclust"),
  n_centers,
  max_iter = 10,
  linkage_method = "complete",
  clusters = NULL,
  eigdecomp_tol = 1e-10,
  eigdecomp_iter = 1000
)
```

Arguments

target	The target (experimental) data set, in a standard format such as a `data.frame` or `matrix`.
target_valid	A holdout set of the target (experimental) data set, in a standard format such as a `data.frame` or `matrix`. `NULL` by default but used by `cvSelectParams` for cross-validated selection of the contrastive and penalization parameters.
center	A `logical` indicating whether the target and background data sets should be centered to mean zero.
scale	A `logical` indicating whether the target and background data sets should be scaled to unit variance.
c_contrasts	A `list` of contrastive covariances.
contrasts	A `numeric` vector of the contrastive parameters used to compute the contrastive covariances.
alg	A `character` indicating the SPCA algorithm used to sparsify the contrastive loadings. Currently supports `iterative` for the Zou et al. (2006) implemententation, `var_proj` for the non-randomized Erichson et al. (2018) solution, and `rand_var_proj` for the randomized Erichson et al. (2018) result.
penalties	A `numeric` vector of the penalty terms.
n_eigen	A `numeric` indicating the number of eigenvectors to be computed.
clust_method	A `character` specifying the clustering method to use for choosing the optimal constrastive parameter. Currently, this is limited to either k-means, partitioning around medoids (PAM), and hierarchical clustering. The default is k-means clustering.
n_centers	A `numeric` giving the number of centers to use in the clustering algorithm.
max_iter	A `numeric` giving the maximum number of iterations to be used in k-means clustering, defaulting to 10.

linkage_method	A character specifying the agglomerative linkage method to be used if clust_method = "hclust". The options are ward.D2, single, complete, average, mcquitty, median, and centroid. The default is complete.
clusters	A numeric vector of cluster labels for observations in the target data. Defaults to NULL, but is otherwise used to identify the optimal set of hyperparameters when fitting the scPCA and the automated version of cPCA.
eigdecomp_tol	A numeric providing the level of precision used by eigendecompositon calculations. Defaults to 1e-10.
eigdecomp_iter	A numeric indicating the maximum number of interations performed by eigendecompositon calculations. Defaults to 1000.

Value

A list similar to that output by prcomp:

- rotation - the matrix of variable loadings
- x - the rotated data, centred and scaled, if requested, data multiplied by the rotation matrix
- contrast - the optimal contrastive parameter
- penalty - the optimal L1 penalty term

References

Erichson NB, Zeng P, Manohar K, Brunton SL, Kutz JN, Aravkin AY (2018). "Sparse Principal Component Analysis via Variable Projection." *ArXiv*, **abs/1804.00341**.

Zou H, Hastie T, Tibshirani R (2006). "Sparse principal component analysis." *Journal of computational and graphical statistics*, **15**(2), 265–286.

---

safeColScale *Safe Centering and Scaling of Columns*

Description

safeColScale is a safe utility for centering and scaling an input matrix X. It is intended to avoid the drawback of using scale on data with constant variance by inducing adding a small perturbation to truncate the values in such columns. It also takes the opportunity to be faster than scale through relying on **matrixStats** or **DelayedMatrixStats**, depending on the type of matrix being processed, for a key internal computation.

Usage

```
safeColScale(
  X,
  center = TRUE,
  scale = TRUE,
  tol = .Machine$double.eps,
  eps = 0.01,
  scaled_matrix = FALSE
)
```

Arguments

X	An input matrix to be centered and/or scaled. If X is not of class matrix or DelayedMatrix, then it must be coercible to a matrix.
center	A logical indicating whether to re-center the columns of the input X.
scale	A logical indicating whether to re-scale the columns of the input X.
tol	A tolerance level for the lowest column variance (or standard deviation) value to be tolerated when scaling is desired. The default is set to double.eps of machine precision .Machine.
eps	The desired lower bound of the estimated variance for a given column. When the lowest estimate falls below tol, it is truncated to the value specified in this argument. The default is 0.01.
scaled_matrix	A logical indicating whether to output a ScaledMatrix object. The centering and scaling procedure is delayed until later, permitting more efficient matrix multiplication and row or column sums downstream. However, this comes at the at the cost of numerical precision. Defaults to FALSE.

Value

A centered and/or scaled version of the input data.

---

scPCA *Sparse Contrastive Principal Component Analysis*

Description

Given target and background data frames or matrices, scPCA will perform the sparse contrastive principal component analysis (scPCA) of the target data for a given number of eigenvectors, a vector of real-valued contrast parameters, and a vector of sparsity inducing penalty terms.

If instead you wish to perform contrastive principal component analysis (cPCA), set the penalties argument to 0. So long as the n_centers parameter is larger than one, the automated hyperparameter tuning heuristic described in Boileau et al. (2020) is used. Otherwise, the semi-automated approach of Abid et al. (2018) is used to select the appropriate hyperparameter.

Usage

```
scPCA(
  target,
  background,
  center = TRUE,
  scale = FALSE,
  n_eigen = 2,
  cv = NULL,
  alg = c("iterative", "var_proj", "rand_var_proj"),
  contrasts = exp(seq(log(0.1), log(1000), length.out = 40)),
  penalties = seq(0.05, 1, length.out = 20),
  clust_method = c("kmeans", "pam", "hclust"),
  n_centers = NULL,
  max_iter = 10,
  linkage_method = "complete",
```

```
  n_medoids = 8,
  parallel = FALSE,
  clusters = NULL,
  eigdecomp_tol = 1e-10,
  eigdecomp_iter = 1000,
  scaled_matrix = FALSE
)
```

Arguments

target	The target (experimental) data set, in a standard format such as a `data.frame` or `matrix`. `dgCMatrix` and `DelayedMatrix` objects are also supported.
background	The background data set, in a standard format such as a `data.frame` or `matrix`. The features must match the features of the target data set. `dgCMatrix` and `DelayedMatrix` objects are also supported.
center	A `logical` indicating whether the target and background data sets' features should be centered to mean zero.
scale	A `logical` indicating whether the target and background data sets' features should be scaled to unit variance.
n_eigen	A `numeric` indicating the number of eigenvectors (or (sparse) contrastive components) to be computed. Two eigenvectors are computed by default.
cv	A `numeric` indicating the number of cross-validation folds to use in choosing the optimal contrastive and penalization parameters from over the grids of `contrasts` and `penalties`. Cross-validation is expected to improve the robustness and generalization of the choice of these parameters. However, it increases the time the procedure costs. The default is therefore `NULL`, corresponding to no cross-validation.
alg	A `character` indicating the sparse PCA algorithm used to sparsify the contrastive loadings. Currently supports `iterative` for the Zou et al. (2006) implementation, `var_proj` for the non-randomized Erichson et al. (2018) solution, and `rand_var_proj` for the randomized Erichson et al. (2018) implementation. Defaults to `iterative`.
contrasts	A `numeric` vector of the contrastive parameters. Each element must be a unique, non-negative real number. By default, 40 logarithmically spaced values between 0.1 and 1000 are used. If a single value is provided and `penalties` is set to 0, then `n_centers`, `clust_method`, `max_iter`, `linkage_method`, `n_medoids`, and `parallel` can be safely ignored.
penalties	A `numeric` vector of the L1 penalty terms on the loadings. The default is to use 20 equidistant values between 0.05 and 1. If `penalties` is set to 0, then cPCA is performed in place of scPCA. See `contrasts` and `n_centers` arguments for more infotmation.
clust_method	A `character` specifying the clustering method to use for choosing the optimal contrastive parameter. Currently, this is limited to either k-means, partitioning around medoids (PAM), and hierarchical clustering. The default is k-means clustering.
n_centers	A `numeric` giving the number of centers to use in the clustering algorithm. If set to 1, cPCA, as first proposed by Abid et al. (2018), is performed, regardless of what the `penalties` argument is set to.
max_iter	A `numeric` giving the maximum number of iterations to be used in k-means clustering. Defaults to 10.

linkage_method	A character specifying the agglomerative linkage method to be used if clust_method = "hclust". The options are ward.D2, single, complete, average, mcquitty, median, and centroid. The default is complete.
n_medoids	A numeric indicating the number of medoids to consider if n_centers is set to 1 and contrasts is a vector of length 2 or more. The default is 8 medoids.
parallel	A logical indicating whether to invoke parallel processing via the **BiocParallel** infrastructure. The default is FALSE for sequential evaluation.
clusters	A numeric vector of cluster labels for observations in the target data. Defaults to NULL, but is otherwise used to identify the optimal set of hyperparameters when fitting the scPCA and the automated version of cPCA. If a vector is provided, the n_centers, clust_method, max_iter, linkage_method, and n_medoids arguments can be safely ignored.
eigdecomp_tol	A numeric providing the level of precision used by eigendecompositon calculations. Defaults to 1e-10.
eigdecomp_iter	A numeric indicating the maximum number of interations performed by eigendecompositon calculations. Defaults to 1000.
scaled_matrix	A logical indicating whether to output a ScaledMatrix object. The centering and scaling procedure is delayed until later, permitting more efficient matrix multiplication and row or column sums downstream. However, this comes at the at the cost of numerical precision. Defaults to FALSE.

Value

A list containing the following components:

- rotation: The matrix of variable loadings if n_centers is larger than one. Otherwise, a list of rotation matrices is returned, one for each medoid. The number of medoids is specified by n_medoids.
- x: The rotated data, centred and scaled if requested, multiplied by the rotation matrix if n_centers is larger than one. Otherwise, a list of rotated data matrices is returned, one for each medoid. The number of medoids is specified by n_medoids.
- contrast: The optimal contrastive parameter.
- penalty: The optimal L1 penalty term.
- center: A logical indicating whether the target dataset was centered.
- scale: A logical indicating whether the target dataset was scaled.

References

Abid A, Zhang MJ, Bagaria VK, Zou J (2018). "Exploring patterns enriched in a dataset with contrastive principal component analysis." *Nature communications*, **9**(1), 2134.

Boileau P, Hejazi NS, Dudoit S (2020). "Exploring High-Dimensional Biological Data with Sparse Contrastive Principal Component Analysis." *Bioinformatics*. ISSN 1367-4803, doi:10.1093/bioinformatics/btaa176, btaa176, https://academic.oup.com/bioinformatics/article-pdf/doi/10.1093/bioinformatics/btaa176/32914142/bt

Erichson NB, Zeng P, Manohar K, Brunton SL, Kutz JN, Aravkin AY (2018). "Sparse Principal Component Analysis via Variable Projection." *ArXiv*, **abs/1804.00341**.

Zou H, Hastie T, Tibshirani R (2006). "Sparse principal component analysis." *Journal of computational and graphical statistics*, **15**(2), 265–286.

Examples

```
# perform cPCA on the simulated data set
scPCA(
  target = toy_df[, 1:30],
  background = background_df,
  contrasts = exp(seq(log(0.1), log(100), length.out = 5)),
  penalties = 0,
  n_centers = 4
)

# perform scPCA on the simulated data set
scPCA(
  target = toy_df[, 1:30],
  background = background_df,
  contrasts = exp(seq(log(0.1), log(100), length.out = 5)),
  penalties = seq(0.1, 1, length.out = 3),
  n_centers = 4
)

# perform cPCA on the simulated data set with known clusters
scPCA(
  target = toy_df[, 1:30],
  background = background_df,
  contrasts = exp(seq(log(0.1), log(100), length.out = 5)),
  penalties = 0,
  clusters = toy_df[, 31]
)

# cPCA as implemented in Abid et al.
scPCA(
  target = toy_df[, 1:30],
  background = background_df,
  contrasts = exp(seq(log(0.1), log(100), length.out = 10)),
  penalties = 0,
  n_centers = 1
)
```

`selectParams` *Selection of Contrastive and Penalization Parameters*

Description

A wrapper function for fitting various internal functions to select the optimal setting of the contrastive and penalization parameters. For internal use only.

Usage

```
selectParams(
  target,
  background,
  center,
  scale,
  n_eigen,
```

```
  alg,
  contrasts,
  penalties,
  clust_method,
  n_centers,
  max_iter,
  linkage_method,
  n_medoids,
  parallel,
  clusters,
  eigdecomp_tol,
  eigdecomp_iter,
  scaled_matrix
)
```

Arguments

target	The target (experimental) data set, in a standard format such as a `data.frame` or `matrix`.
background	The background data set, in a standard format such as a `data.frame` or `matrix`. Note that the number of features must match the number of features in the target data.
center	A `logical` indicating whether the target and background data sets should be centered to mean zero.
scale	A `logical` indicating whether the target and background data sets should be scaled to unit variance.
n_eigen	A `numeric` indicating the number of eigenvectors (or sparse contrastive components) to be computed. The default is to compute two such eigenvectors.
alg	A `character` indicating the SPCA algorithm used to sparsify the contrastive loadings. Currently supports `iterative` for the Zou et al. (2006) implementation, `var_proj` for the non-randomized Erichson et al. (2018) solution, and `rand_var_proj` for the randomized Erichson et al. (2018) result.
contrasts	A `numeric` vector of the contrastive parameters. Each element must be a unique non-negative real number. The default is to use 40 logarithmically spaced values between 0.1 and 1000.
penalties	A `numeric` vector of the L1 penalty terms on the loadings. The default is to use 20 equidistant values between 0.05 and 1.
clust_method	A `character` specifying the clustering method to use for choosing the optimal contrastive parameter. Currently, this is limited to either k-means, partitioning around medoids (PAM), and hierarchical clustering. The default is k-means clustering.
n_centers	A `numeric` giving the number of centers to use in the clustering algorithm. If set to 1, cPCA, as first proposed by Abid et al., is performed, regardless of what the `penalties` argument is set to.
max_iter	A `numeric` giving the maximum number of iterations to be used in k-means clustering, defaulting to 10.
linkage_method	A `character` specifying the agglomerative linkage method to be used if `clust_method = "hclust"`. The options are `ward.D2`, `single`, `complete`, `average`, `mcquitty`, `median`, and `centroid`. The default is `complete`.

n_medoids	A numeric indicating the number of medoids to consider if n_centers is set to 1. The default is 8 such medoids.
parallel	A logical indicating whether to invoke parallel processing via the **BiocParallel** infrastructure. The default is FALSE for sequential evaluation.
clusters	A numeric vector of cluster labels for observations in the target data. Defaults to NULL, but is otherwise used to identify the optimal set of hyperparameters when fitting the scPCA and the automated version of cPCA.
eigdecomp_tol	A numeric providing the level of precision used by eigendecompositon calculations. Defaults to 1e-10.
eigdecomp_iter	A numeric indicating the maximum number of interations performed by eigendecompositon calculations. Defaults to 1000.
scaled_matrix	A logical indicating whether to output a ScaledMatrix object. The centering and scaling procedure is delayed until later, permitting more efficient matrix multiplication and row or column sums downstream. However, this comes at the at the cost of numerical precision.

Value

Output structure matching either that of fitCPCA or fitGrid (or their parallelized variants, namely either bpFitCPCA and link{bpFitGrid}, respectively).

References

Erichson NB, Zeng P, Manohar K, Brunton SL, Kutz JN, Aravkin AY (2018). "Sparse Principal Component Analysis via Variable Projection." *ArXiv*, **abs/1804.00341**.

Zou H, Hastie T, Tibshirani R (2006). "Sparse principal component analysis." *Journal of computational and graphical statistics*, **15**(2), 265–286.

---

spcaWrapper *Sparse PCA Wrapper*

Description

This wrapper function specifies which implementation of sparse pricincipal component analysis (SPCA) is used to sparsify the loadings of the contrastive covariance matrix. Currently, the scPCA package supports the iterative algorithm detailed by Zou et al. (2006), and Erichson et al. (2018)'s randomized and non-randomized versions of SPCA solved via variable projection. These methods are implemented in the **elasticnet** and **sparsepca** packages.

Usage

```
spcaWrapper(
  alg,
  contrast_cov,
  contrast,
  k,
  penalty,
  eigdecomp_tol,
  eigdecomp_iter
)
```

Arguments

alg	A character indicating the SPCA algorithm used to sparsify the contrastive loadings. Currently supports iterative for the Zou et al. (2006) implemententation, var_proj for the non-randomized Erichson et al. (2018) solution, and rand_var_proj for the randomized Erichson et al. (2018) result.
contrast_cov	A contrastive covariance matrix.
contrast	A numeric contrastive parameter used to compute the contrastive covariance matrix.
k	A numeric indicating the number of eigenvectors (or sparse contrastive components) to be computed.
penalty	A numeric indicating the L1 penalty parameter applied to the loadings.
eigdecomp_tol	A numeric providing the level of precision used by eigendecompositon calculations.
eigdecomp_iter	A numeric indicating the maximum number of interations performed by eigendecompositon calculations.

Value

A p x k sparse loadings matrix, where p is the number of features, and k is the number of sparse contrastive components.

References

Erichson NB, Zeng P, Manohar K, Brunton SL, Kutz JN, Aravkin AY (2018). "Sparse Principal Component Analysis via Variable Projection." *ArXiv*, **abs/1804.00341**.

Zou H, Hastie T, Tibshirani R (2006). "Sparse principal component analysis." *Journal of computational and graphical statistics*, **15**(2), 265–286.

---

toy_df *Simulated Target Data for cPCA and scPCA*

Description

The toy data consisting of 400 observations and 31 variables was simulated as follows:

- Each of the first 10 variables was drawn from $N(0, 10)$
- For group 1 and 2, variables 11 through 20 were drawn from $N(0, 1)$
- For group 3 and 4, variables 11 through 20 were drawn from $N(3, 1)$
- For group 1 and 3, variables 21 though 30 were drawn from $N(-3, 1)$
- For group 2 and 4, variables 21 though 30 were drawn from $N(0, 1)$
- The last column provides each observations group number

Usage

```
data(toy_df)
```

Format

A simple `data.frame`.

Examples

```
data(toy_df)
```

Index