Package 'scPCA'

October 17, 2020

Title Sparse Contrastive Principal Component Analysis

Version 1.2.0

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 techical noise through the use of control data. Also implements and extends cPCA.

Depends R (>= 3.6)

Imports stats, methods, assertthat, tibble, dplyr, purrr, stringr, Rdpack, matrixStats, BiocParallel, elasticnet, sparsepca, cluster, kernlab, origami

Suggests testthat (>= 2.1.0), covr, knitr, rmarkdown, BiocStyle, Matrix, 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 7.1.0

RdMacros Rdpack

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

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

git_branch RELEASE_3_11

git_last_commit 83adc07

git_last_commit_date 2020-04-27

Date/Publication 2020-10-16

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

Maintainer Philippe Boileau <philippe_boileau@berkeley.edu>

R topics documented:

`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)
```

`scPCA` *Sparse Constrastive 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 penalty terms. For more information on the contrastive PCA method, consult Abid A, Zhang MJ, Bagaria VK, Zou J (2018). "Exploring patterns enriched in a dataset with contrastive principal component analysis." *Nature communications*, **9**.. Sparse PCA is performed via the method of Zou H, Hastie T, Tibshirani R (2006). "Sparse principal component analysis." *Journal of computational and graphical statistics*, **15**, 265–286..

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,
  max_iter = 10,
  linkage_method = "complete",
  n_medoids = 8,
  parallel = 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. 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.
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, thus, the default is NULL, corresponding to no cross-validation.
alg	A character indicating the SPCA algorithm used to sparsify the contrastive loadings. Currently supports iterative for the Zou H, Hastie T, Tibshirani R (2006). "Sparse principal component analysis." *Journal of computational and graphical statistics*, **15**, 265–286. implemententation, var_proj for the non-randomized Erichson NB, Zeng P, Manohar K, Brunton SL, Kutz JN, Aravkin AY (2018). "Sparse Principal Component Analysis via Variable Projection." *ArXiv*, **abs/1804.00341**. solution, and rand_var_proj for the randomized Erichson NB, Zeng P, Manohar K, Brunton SL, Kutz JN, Aravkin AY (2018). "Sparse Principal Component Analysis via Variable Projection." *ArXiv*, **abs/1804.00341**. result. Defaults to iterative.
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 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. 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.

Value

A list containing the following components:

- rotation - the matrix of variable loadings
- x - the rotated data, centred and scaled if requested, multiplied by the rotation matrix
- contrast - the optimal contrastive parameter
- penalty - the optimal L1 penalty term
- center - whether the target dataset was centered
- scale - whether the target dataset was scaled

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
)

# 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
)
```

---

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