Package 'gcpca'

April 1, 2026

Type Package

Title Generalized Contrastive Principal Component Analysis

Version 0.0.1

Description Implements dense and sparse generalized contrastive principal component analysis (gcPCA) with S3 fit objects and methods for prediction, summaries, and plotting. The gcPCA is a hyperparameter-free method for comparing high-dimensional datasets collected under different experimental conditions to reveal low-dimensional patterns enriched in one condition compared to the other. Method details are described in de Oliveira, Garg, Hjerling-Leffler, Batista-Brito, and Sjulson (2025) <doi:10.1371/journal.pcbi.1012747>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

URL https://github.com/SjulsonLab/generalized_contrastive_PCA

BugReports https://github.com/SjulsonLab/generalized_contrastive_PCA/issues

Imports graphics, stats

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Eliezyer de Oliveira [aut, cre]

Maintainer Eliezyer de Oliveira <eliezyer.deoliveira@gmail.com>

Repository CRAN

Date/Publication 2026-04-01 08:30:08 UTC

Contents

`coef.gcPCA` *Extract gcPCA loadings*

Description

Extract gcPCA loadings

Usage

```
## S3 method for class 'gcPCA'
coef(object, ...)

## S3 method for class 'sparse_gcPCA'
coef(object, ...)
```

Arguments

`object`	A fitted `"gcPCA"` or `"sparse_gcPCA"` object.
`...`	Unused.

Value

For dense models, a loadings matrix. For sparse models, a named list of loading matrices (one per lasso penalty).

`fitted.gcPCA` *Return fitted training scores*

Description

Return fitted training scores

Usage

```
## S3 method for class 'gcPCA'
fitted(object, ...)
```

Arguments

object	A fitted "gcPCA" or "sparse_gcPCA" object.
...	Passed to predict().

Value

A list with Ra_scores and Rb_scores.

---

gcPCA *Fit a generalized contrastive PCA model*

Description

gcPCA() fits dense generalized contrastive principal components from two data matrices (Ra and Rb) measured on the same feature set.

Usage

```
gcPCA(
  Ra,
  Rb,
  method = "v4",
  Ncalc = NULL,
  Nshuffle = 0,
  normalize_flag = TRUE,
  alpha = 1,
  alpha_null = 0.975,
  cond_number = 10^13
)
```

Arguments

Ra	Matrix for condition A (n_a x p).
Rb	Matrix for condition B (n_b x p).
method	Character scalar selecting optimization variant.
Ncalc	Optional number of gcPCs for orthogonal variants (*.1).
Nshuffle	Number of shuffles used to estimate a null distribution.
normalize_flag	Logical; if TRUE, z-score and L2-normalize columns.
alpha	Contrastive coefficient used by method "v1".
alpha_null	Quantile for null thresholding (reserved in current code).
cond_number	Maximum allowed denominator condition number.

Value

An object of class "gcPCA" with loadings, scores, objective values, and fit metadata.

Examples

```
set.seed(1)
Ra <- matrix(rnorm(40 * 5), ncol = 5)
Rb <- matrix(rnorm(35 * 5), ncol = 5)
fit <- gcPCA(Ra, Rb, method = "v4", Ncalc = 3)
pred <- predict(fit, Ra = Ra)
str(pred$Ra_scores)
```

---

loadings *Get gcPCA loadings*

Description

Convenience wrapper around `coef()` for `"gcPCA"` and `"sparse_gcPCA"` objects.

Usage

```
loadings(object, ...)
```

Arguments

`object`	A fitted model object.
`...`	Extra arguments passed to `coef()`.

Value

A loading matrix or a named list of loading matrices.

---

plot.gcPCA *Plot gcPCA scores*

Description

Plot gcPCA scores

Usage

```
## S3 method for class 'gcPCA'
plot(x, which = c(1, 2), ...)
```

Arguments

`x`	A fitted `"gcPCA"` or `"sparse_gcPCA"` object.
`which`	Integer vector of length 2 indicating components to plot.
`...`	Extra graphical parameters passed to `plot()`.

Value

x, invisibly.

`predict.gcPCA` *Project data with a fitted gcPCA model*

Description

Project data with a fitted gcPCA model

Usage

```
## S3 method for class 'gcPCA'
predict(object, Ra = NULL, Rb = NULL, ...)

## S3 method for class 'sparse_gcPCA'
predict(object, Ra = NULL, Rb = NULL, ...)
```

Arguments

`object`	A fitted "gcPCA" or "sparse_gcPCA" object.
`Ra`	Optional matrix for condition A.
`Rb`	Optional matrix for condition B.
`...`	Unused.

Value

For dense models, a list of score matrices. For sparse models, a list of score-matrix lists, one per penalty value.

`print.gcPCA` *Print a gcPCA model*

Description

Print a gcPCA model

Usage

```
## S3 method for class 'gcPCA'
print(x, ...)
```

Arguments

x	A fitted "gcPCA" or "sparse_gcPCA" object.
...	Unused.

Value

x, invisibly.

---

print.summary.gcPCA *Print a gcPCA summary*

Description

Print a gcPCA summary

Usage

```
## S3 method for class 'summary.gcPCA'
print(x, ...)
```

Arguments

x	A "summary.gcPCA" object from summary.gcPCA().
...	Unused.

Value

x, invisibly.

---

scores *Get gcPCA scores*

Description

Convenience wrapper around predict() for "gcPCA" and "sparse_gcPCA" objects.

Usage

```
scores(object, newdata = NULL, ...)
```

Arguments

object	A fitted model object.
newdata	Optional matrix (treated as Ra) or list with Ra/Rb.
...	Extra arguments passed to predict().

Value

A list of projected scores.

`sparse_gcPCA` *Fit a sparse generalized contrastive PCA model*

Description

`sparse_gcPCA()` fits sparse gcPCA loadings over a path of lasso penalties.

Usage

```
sparse_gcPCA(
  Ra,
  Rb,
  method = "v4",
  Ncalc = NULL,
  normalize_flag = TRUE,
  Nsparse = NULL,
  Nshuffle = 0,
  lasso_penalty = exp(seq(log(0.01), log(1), length.out = 10)),
  ridge_penalty = 0,
  alpha = 1,
  alpha_null = 0.975,
  tol = 1e-05,
  max_steps = 1000,
  cond_number = 1e+13
)
```

Arguments

`Ra`	Matrix for condition A (`n_a x p`).
`Rb`	Matrix for condition B (`n_b x p`).
`method`	Character scalar selecting optimization variant.
`Ncalc`	Optional number of gcPCs for orthogonal variants (`*.1`).
`normalize_flag`	Logical; if `TRUE`, z-score and L2-normalize columns.
`Nsparse`	Number of sparse gcPCs to estimate.
`Nshuffle`	Number of shuffles passed to the dense gcPCA stage.
`lasso_penalty`	Numeric vector of lasso penalties.
`ridge_penalty`	Numeric ridge penalty.
`alpha`	Contrastive coefficient used by method `"v1"`.
`alpha_null`	Quantile for null thresholding (reserved in current code).
`tol`	Convergence tolerance for sparse optimization.
`max_steps`	Maximum number of sparse optimization iterations.
`cond_number`	Maximum allowed denominator condition number.

Value

An object of class `c("sparse_gcPCA", "gcPCA")` containing sparse loadings and training projections across penalties.

Examples

```
set.seed(1)
Ra <- matrix(rnorm(50 * 6), ncol = 6)
Rb <- matrix(rnorm(45 * 6), ncol = 6)
fit <- sparse_gcPCA(Ra, Rb, method = "v4", Nsparse = 2,
                    lasso_penalty = c(0.05, 0.1))
pred <- predict(fit, Ra = Ra)
str(pred$Ra_scores)
```

---

summary.gcPCA *Summarize a gcPCA model*

Description

Summarize a gcPCA model

Usage

```
## S3 method for class 'gcPCA'
summary(object, ...)
```

Arguments

`object`	A fitted `"gcPCA"` or `"sparse_gcPCA"` object.
`...`	Unused.

Value

A `"summary.gcPCA"` list with eigenvalues, standard deviations, explained variance summaries, and tuning settings.

Index