Package: seam (via r-universe)

November 16, 2026

Title Sparse Extended Analysis of Modules

Version 0.1.0

Maintainer Marketa Barac Makarova <marketa.makarova@mail.muni.cz>

Description Provides tools for comparing groups using sparse principal component analysis (SPCA). Includes functions for identifying and visualizing influential variables, assessing variable selection stability through bootstrapping, and exploring associations with clinical traits. SPCA is implemented using the method described in <doi:10.1198/106186006X113430>.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1.0)

Imports dplyr, elasticnet, ggplot2, graphics, rlang, stats, tidyr

Suggests spelling, testthat (>= 3.0.0)

Config/testthat/edition 3

Roxygen list(markdown = TRUE)

NeedsCompilation no

Author Marketa Barac Makarova [aut, cre], Jan Kolacek [ctb]

URL https://github.com/marketa-makarova/seam

BugReports https://github.com/marketa-makarova/seam/issues

Config/roxygen2/version 8.1.0

Language en-US

Config/pak/sysreqs libicu-dev

Repository https://marketa-makarova.r-universe.dev

Date/Publication 2026-11-10 19:03:01 UTC

RemoteUrl https://github.com/marketa-makarova/seam

RemoteRef HEAD

RemoteSha 0651509522a6a8ce33c869299b0fa3de9fe4bfb3

Contents

compute_activity . . . 2
compute_module_trait_correlations . . . 3
compute_variable_trait_correlations . . . 4
insight_correlations_lollipop . . . 5
insight_correlations_reconstruct . . . 7
insight_correlations_sign_changes . . . 9
insight_weights_lollipop . . . 10
plot_activity_bar . . . 12
plot_activity_bubble . . . 13
plot_explained_variance . . . 14
plot_lambdas . . . 16
plot_module_heatmaps . . . 17
plot_module_trait_correlations . . . 18
plot_sparsity . . . 20
seam . . . 21
seam_demo . . . 21
spca_all_modules . . . 22
spca_all_modules_grid . . . 23

Index **25**

`compute_activity` *Compute activity statuses of variables using SPCA*

Description

The activity column describes the original SPCA fit. When bootstrap fits are available, pct_* columns give the percentage of bootstrap replicates assigned to each status. Percentages sum to 100 per variable (up to rounding).

Usage

```
compute_activity(spca_data, return_information = TRUE, threshold = 0.8, labels)
```

Arguments

`spca_data`	Result of spca_all_modules() or spca_all_modules_grid().
`return_information`	`TRUE` for refined undead classifications, `FALSE` for raw selection, or `"both"` to print both plots.
`threshold`	Absolute within-group correlation threshold for reclassification to undead status.
`labels`	Group labels, if present, in the same order as the expression data.

Details

A variable is active in a group when its SPCA loading is nonzero. A variable excluded in all groups is classified as undead in a given group only when it has absolute correlation above `threshold` with at least one variable selected by SPCA in that same group. Variables selected in other groups do not contribute to this classification.

Bootstrap resampling is used to assess the stability of variable selection under repeated sampling. Selection frequencies describe the reproducibility of nonzero SPCA loadings and should not be interpreted as statistical significance or evidence of differential population support.

With the presence of groups, activity labels are `dead`, `active`, `active_<label>`, `undead`, and `undead_<label>`. The labels `active` and `undead` indicate all groups. Bootstrap percentages use the selected variables from each corresponding group and bootstrap replicate, with within-group correlations calculated from the original expression matrices.

Value

Data frame with variable_id, module, `w_<label>` columns, activity, and optional `pct_<status>` bootstrap percentages.

Examples

```
data("seam_demo", package = "seam")
fit = spca_all_modules(
 seam_demo$expression_data,
 seam_demo$module_assignment,
 lambda = 0.05,
 bootstrap = FALSE
)
activity = compute_activity(
 fit,
 labels = seam_demo$labels
)
table(
 activity$module,
 activity$activity
)
```

---

compute_module_trait_correlations

Compute Pearson's correlations between clinical traits and sparse or classic eigengenes

---

Description

The original correlations are always returned. If spca_data contains bootstrap replicates, bootstrap correlations are returned separately. Each bootstrap replicate uses the same sampled rows for eigengenes and traits.

Usage

```
compute_module_trait_correlations(
  trait_data,
  spca_data,
  module_assignment,
  mode = "spca"
)
```

Arguments

trait_data	List of numeric trait data frames, one per group, with rows in the same order as the corresponding expression matrix.
spca_data	Result of spca_all_modules() or spca_all_modules_grid().
module_assignment	Vector of modules assigned to variables. One module per variable.
mode	One of "spca", "classic", or "both".

Value

A list containing mtc_spca and/or mtc_classic (lists of original module-by-trait correlation matrices), trait_data, module_sizes, reduced_module_sizes, and bootstrap. The bootstrap element is NULL if no resampling was performed.

Examples

```
data("seam_demo", package = "seam")
fit = spca_all_modules(
 seam_demo$expression_data,
 seam_demo$module_assignment,
 lambda = 0.05,
 bootstrap = FALSE
)
mtc = compute_module_trait_correlations(
 seam_demo$trait_data,
 fit, seam_demo$module_assignment,
 mode = "both"
)
mtc$mtc_spca[[1]]
```

---

compute_variable_trait_correlations

Compute variable–trait correlations

Description

Computes Pearson correlations between expression variables and clinical traits, alogside SPCA-weighted correlations (correlation multiplied by corresponding SPCA weight). When bootstrap fits are available, adds bootstrap confidence intervals using combined resampling of traits and samples.

Usage

```
compute_variable_trait_correlations(
  trait_data,
  spca_data,
  return_information = TRUE,
  threshold = 0.8,
  labels
)
```

Arguments

trait_data	List of numeric trait data frames, one per group, with rows in the same order as the corresponding expression matrix.
spca_data	Result of spca_all_modules() or spca_all_modules_grid().
return_information	TRUE for refined undead classifications, FALSE for raw selection, or "both" to print both plots.
threshold	Absolute within-group correlation threshold for reclassification to undead status.
labels	Group labels, if present, in the same order as the expression data.

Value

List of data frames, one per group. Each contains variable_id, module, original SPCA weights, activity, and <trait>_cor and <trait>_w_cor columns.

Examples

```
data("seam_demo", package = "seam")
fit = spca_all_modules(
 seam_demo$expression_data,
 seam_demo$module_assignment,
 lambda = 0.05,
 bootstrap = FALSE
)
vtc = compute_variable_trait_correlations(
 seam_demo$trait_data,
 fit,
 labels = seam_demo$labels)
head(vtc[[1]])
```

---

insight_correlations_lollipop

Lollipop plot of variable-trait correlations and SPCA-weighted correlations

---

Description

Selects module-trait pairs whose SPCA eigengene correlation exceeds interest_threshold in at least one group. For each pair, plots variable-trait correlations and/or correlations multiplied by SPCA weights. Original estimates are used whether or not spca_data contains bootstrap fits.

Usage

```
insight_correlations_lollipop(
  module_trait_correlations,
  spca_data,
  type = "both",
  interest_threshold = 0.6,
  cex = 20,
  cex_point = 5,
  labels,
  n = 10,
  xlab = "",
  ylab = "Weighted correlations",
  col = NULL,
  col_active = 3,
  y_dodge = 2
)
```

Arguments

module_trait_correlations	Result of compute_module_trait_correlations() calculated with mode = "spca" or "both".
spca_data	Result of spca_all_modules() or spca_all_modules_grid().
type	One of "classic" (unweighted), "weighted", or "both".
interest_threshold	Miminal absolute module-trait correlation threshold.
cex	Base text size.
cex_point	Point size.
labels	Group labels, if present, in the same order as the expression data.
n	Number of top variable labels to display per plot.
xlab, ylab	Axis labels.
col	Optional colors for the groups.
col_active	Color of segments for variables active in multiple groups.
y_dodge	Number of rows used to dodge variable labels.

Value

A named list of ggplot objects is returned invisibly. Plots are printed separately.

Examples

```
data("seam_demo", package = "seam")
fit = spca_all_modules(
 seam_demo$expression_data,
 seam_demo$module_assignment,
 lambda = 0.05,
 bootstrap = FALSE
)
mtc = compute_module_trait_correlations(
 seam_demo$trait_data,
 fit,
 seam_demo$module_assignment,
 mode = "both")
insight_correlations_lollipop(
 mtc,
 fit,
 labels = seam_demo$labels,
 interest_threshold = 0.2,
 n = 5
)
```

`insight_correlations_reconstruct` *Reconstruct module-trait correlations from top contributing variables*

Description

Ranks variables within each group by the absolute product of their SPCA weight and variable-trait correlation. Reconstructs the SPCA score from the top variables and compares its trait correlation with the original full-module correlation. Optional bootstrap fits do not change the reconstruction; only original estimates are used.

Usage

```
insight_correlations_reconstruct(
  module_trait_correlations,
  spca_data,
  labels,
  col = NULL,
  cex = 20,
  top_n_upper = 50,
  top_n_lower = 10,
  interest_threshold = 0.6,
  threshold = 0.8,
  return_information = TRUE,
  ylab = "Correlation",
```

```
  xlab = "Top n variables"
)
```

Arguments

module_trait_correlations	Result of compute_module_trait_correlations() calculated with mode = "spca" or "both".
spca_data	Result of spca_all_modules() or spca_all_modules_grid().
labels	Group labels, if present, in the same order as the expression data.
col	Optional colors for the groups.
cex	Base text size.
top_n_upper	Maximum number of selected variables.
top_n_lower	Minimum number of selected variables.
interest_threshold	Miminal absolute module-trait correlation threshold.
threshold	Absolute within-group correlation threshold for reclassification to undead status.
return_information	TRUE for refined undead classifications, FALSE for raw selection, or "both" to print both plots.
ylab, xlab	Axis labels.

Value

A named list of ggplot objects is returned invisibly. Plots are printed separately.

Examples

```
data("seam_demo", package = "seam")
fit = spca_all_modules(
 seam_demo$expression_data,
 seam_demo$module_assignment,
 lambda = 0.05,
 bootstrap = FALSE
)
mtc = compute_module_trait_correlations(
 seam_demo$trait_data,
 fit,
 seam_demo$module_assignment,
 mode = "both"
)
insight_correlations_reconstruct(
 mtc,
 fit,
 labels = seam_demo$labels,
 top_n_lower = 1,
 top_n_upper = 8,
 interest_threshold = 0.2
```

```
)
```

`insight_correlations_sign_changes` *Visualize variable-trait correlations and negative SPCA weights*

Description

Selects module-trait pairs exceeding the absolute SPCA correlation threshold in at least one group. For each pair, plots the density of unweighted variable-trait correlations among variables active in SPCA. The title reports the percentage of active variables with negative SPCA weights in each group.

Usage

```
insight_correlations_sign_changes(
  module_trait_correlations,
  spca_data,
  labels,
  col = NULL,
  interest_threshold = 0.6,
  cex = 20,
  xlab = "Correlations",
  ylab = ""
)
```

Arguments

`module_trait_correlations`	Result of compute_module_trait_correlations() with mode = "spca" or "both".
`spca_data`	Result of spca_all_modules() or spca_all_modules_grid().
`labels`	Group labels, if present, in the same order as the expression data.
`col`	Optional colors for the groups.
`interest_threshold`	Miminal absolute module-trait correlation threshold.
`cex`	Base text size.
`xlab, ylab`	Axis labels.

Value

A named list of ggplot objects is returned invisibly. Plots are printed separately.

Examples

```
data("seam_demo", package = "seam")
fit = spca_all_modules(
 seam_demo$expression_data,
 seam_demo$module_assignment,
 lambda = 0.05,
 bootstrap = FALSE
)
mtc = compute_module_trait_correlations(
 seam_demo$trait_data,
 fit,
 seam_demo$module_assignment,
 mode = "both")
insight_correlations_sign_changes(
 mtc,
 fit,
 labels = seam_demo$labels,
 interest_threshold = 0.2
)
```

`insight_weights_lollipop` *Compare SPCA variable weights between two groups*

Description

Draws a lollipop plot of the original SPCA weight for each variable in two groups. Variables are ranked by between-group weight difference, weight magnitude, or bootstrap selection instability. When bootstrap results are available, optional 95% percentile intervals and selection frequencies describe the variability of each loading.

Usage

```
insight_weights_lollipop(
  spca_data,
  labels,
  modules = "all",
  n = 20,
  bootstrap = TRUE,
  show_intervals = TRUE,
  show_stability = TRUE,
  rank_by = c("difference", "magnitude", "instability"),
  align_signs = TRUE,
  cex = 14,
  col = c("#377EB8", "#E69F00"),
  xlab = "SPCA weight"
)
```

Arguments

spca_data	Result of spca_all_modules() or spca_all_modules_grid().
labels	Group labels, if present, in the same order as the expression data.
modules	Module labels to plot, or "all" (default).
n	Maximum number of variables shown per module. Default is 20.
bootstrap	Logical. Whether to use available bootstrap results. Default is TRUE. If results are absent, only original weights are shown.
show_intervals	Logical. Whether to draw 95% percentile intervals for bootstrap weights. Default is TRUE.
show_stability	Logical. Whether to encode bootstrap selection frequency as interval opacity (darker means more frequently selected). Default is TRUE.
rank_by	One of "difference" (absolute difference in aligned original weights), "magnitude" (largest absolute weight in either group), or "instability" (lowest mean bootstrap selection frequency). "instability" requires bootstrap results.
align_signs	Logical. Whether to align the second group's original and bootstrap loading vectors to the first group by the sign of their dot product. Default is TRUE.
cex	Base text size.
col	Optional colors for the groups.
xlab	Label for the weight axis.

Details

The same variables must be present in both groups for each module. Group B is sign-aligned to group A *for visualization only*; the input object is not modified. Bootstrap loading vectors are assumed to have already been aligned to their respective original solutions by the SPCA fitting function. The sign of an SPCA component is arbitrary, and the dot-product alignment can be ambiguous for nearly orthogonal solutions.

Selection frequency is the percentage of bootstrap fits with a nonzero loading. Intervals are the 2.5th and 97.5th percentiles of the full bootstrap loading distribution, including zero values. These percentile ranges are descriptive and may not have nominal confidence coverage after sparse variable selection.

Value

A named list of ggplot objects is returned invisibly. Plots are printed separately.

Examples

```
data("seam_demo", package = "seam")
fit = spca_all_modules(
 seam_demo$expression_data,
 seam_demo$module_assignment,
 lambda = 0.05,
 bootstrap = FALSE
)
```

```
insight_weights_lollipop(
 fit, labels = seam_demo$labels,
 modules = "blue",
 n = 8,
 bootstrap = FALSE
)
```

`plot_activity_bar` *Plot variable activity statuses as a stacked barplot*

Description

Displays counts of original SPCA activity classifications across modules as a stacked barplot. Statuses are differentiated by colour.

Usage

```
plot_activity_bar(
  spca_data,
  return_information = TRUE,
  modules = "all",
  threshold = 0.6,
  labels,
  cex = 20,
  col_both_active = "forestgreen",
  col_active = NULL,
  col_dead = "grey55",
  col_both_undead = "lightgrey",
  col_undead = NULL,
  xlab = "Module",
  ylab = "Count"
)
```

Arguments

`spca_data`	Result of spca_all_modules() or spca_all_modules_grid().
`return_information`	`TRUE` for refined undead classifications, `FALSE` for raw selection, or `"both"` to print both plots.
`modules`	Module labels to plot, or `"all"` (default).
`threshold`	Absolute within-group correlation threshold for reclassification to undead status.
`labels`	Group labels, if present, in the same order as the expression data.
`cex`	Base text size.
`col_both_active`	Color for variables selected in all groups.

col_active	Colors for variables selected in exactly one group; one color per group, in labels order.
col_dead	Color for variables excluded in every group.
col_both_undead	Color for variables undead in all groups.
col_undead	Colors for variables undead in exactly one group; one color per group, in labels order.
xlab, ylab	Axis labels.

Value

A named list of ggplot objects is returned invisibly. Multiple plots are printed separately.

Examples

```
data("seam_demo", package = "seam")
fit = spca_all_modules(
 seam_demo$expression_data,
 seam_demo$module_assignment,
 lambda = 0.05,
 bootstrap = FALSE
)
plot_activity_bar(
 fit,
 labels = seam_demo$labels
)
```

---

plot_activity_bubble *Plot variable activity statuses as a bubble chart*

Description

Bubble size describes counts of variables assigned a given status. When bootstrap results are available, a second number optionally displays an average status count across bootstrap replicates.

Usage

```
plot_activity_bubble(
  spca_data,
  labels,
  modules = "all",
  threshold = 0.8,
  return_information = TRUE,
  cex = 20,
  scale_range = c(5, 30),
  number_color = "white",
```

```
  bubble_colors = NA,
  show_bootstrap = TRUE
)
```

Arguments

spca_data	Result of spca_all_modules() or spca_all_modules_grid().
labels	Group labels, if present, in the same order as the expression data.
modules	Module labels to plot, or "all" (default).
threshold	Absolute within-group correlation threshold for reclassification to undead status.
return_information	TRUE for refined undead classifications, FALSE for raw selection, or "both" to print both plots.
cex	Base text size.
scale_range	Minimum and maximum bubble sizes.
number_color	Color of numbers inside bubbles.
bubble_colors	Optional colors for modules; NA uses module colors when they are valid.
show_bootstrap	Show average bootstrap counts below original counts.

Value

A ggplot object, printed and returned invisibly.

Examples

```
data("seam_demo", package = "seam")
fit = spca_all_modules(
 seam_demo$expression_data,
 seam_demo$module_assignment,
 lambda = 0.05,
 bootstrap = FALSE
)
plot_activity_bubble(
 fit,
 labels = seam_demo$labels
)
```

---

plot_explained_variance

Plot explained variance in modules as a heatmap

Description

Cell colors show percentages of variance explained by SPCA.

Usage

```
plot_explained_variance(
  spca_data,
  labels,
  modules = "all",
  y_axis_labels = NA,
  cex = 1.2,
  main = "Approximate explained variance [in %]",
  color_palette = (grDevices::colorRampPalette(c("blue", "white", "red")))(50),
  show_legend = TRUE,
  show_size = TRUE
)
```

Arguments

spca_data	Result of spca_all_modules() or spca_all_modules_grid().
labels	Group labels, if present, in the same order as the expression data.
modules	Module labels to plot, or "all" (default).
y_axis_labels	Optional custom module labels.
cex	Base text size.
main	Plot title.
color_palette	Colors for the continuous heatmap scale.
show_legend	Logical. Whether to display the color legend.
show_size	Logical. Whether to include module sizes in row labels.

Value

A ggplot object, printed and returned invisibly.

Examples

```
data("seam_demo", package = "seam")
fit = spca_all_modules(
 seam_demo$expression_data,
 seam_demo$module_assignment,
 lambda = 0.05
)
plot_explained_variance(
 fit,
 labels = seam_demo$labels
)
```

plot_lambdas *Plot SPCA sparsity and explained variance across different lambda values*

Description

A connected scatter plot show the percentage of explained variance and variables assigned zero weights.

Usage

```
plot_lambdas(
  expression_data,
  module_assignment,
  modules = "all",
  lambda_grid,
  cex = 17,
  labels,
  legend_title = "Module",
  legend_position = "right",
  linewidth = 2,
  xlab = "Lambda",
  ylab = "Percentage of dead variables",
  separate_modules = FALSE
)
```

Arguments

expression_data	List of numeric expression matrices (samples in rows).
module_assignment	Vector of modules assigned to variables. One module per variable.
modules	Module labels to plot, or "all" (default).
lambda_grid	Numeric vector of non-negative sparsity penalties.
cex	Base text size.
labels	Group labels, if present, in the same order as the expression data.
legend_title	Legend title.
legend_position	ggplot legend position.
linewidth	Line width.
xlab	X-axis label.
ylab	Y-axis label.
separate_modules	Logical. Whether to facet by module as well as group.

Value

A named list of ggplot objects is returned invisibly. Plots are printed separately.

Examples

```
data("seam_demo", package = "seam")
plot_lambdas(
 seam_demo$expression_data,
 seam_demo$module_assignment,
 lambda_grid = c(0.02, 0.05, 0.1),
 labels = seam_demo$labels
)
```

---

plot_module_heatmaps *Plot module correlation heatmaps and sparsified heatmaps*

Description

The SPCA fit determines which variables are excluded. When bootstrap results are present, masked cells are shaded according to how often either variable received a zero weight across bootstrap replicates.

Usage

```
plot_module_heatmaps(
  spca_data,
  labels,
  modules = "all",
  return_information = TRUE,
  threshold = 0.8,
  cex = 20,
  midpoint = 0.5,
  col_low = "darkred",
  col_mid = "orange",
  col_high = "white"
)
```

Arguments

spca_data	Result of spca_all_modules() or spca_all_modules_grid().
labels	Group labels, if present, in the same order as the expression data.
modules	Module labels to plot, or "all" (default).
return_information	TRUE for refined undead classifications, FALSE for raw selection, or "both" to print both plots.

threshold	Absolute within-group correlation threshold for reclassification to undead status.
cex	Base text size.
midpoint	Midpoint of the distance color scale.
col_low, col_mid, col_high	Colors of the distance scale.

Value

A named list of ggplot objects is returned invisibly. Plots are printed separately.

Examples

```
data("seam_demo", package = "seam")
fit = spca_all_modules(
 seam_demo$expression_data,
 seam_demo$module_assignment,
 lambda = 0.05,
 bootstrap = FALSE
)
plot_module_heatmaps(
 fit,
 labels = seam_demo$labels,
 modules = "blue",
 return_information = FALSE
)
```

---

plot_module_trait_correlations

Plot module-trait correlations for sparse and classic eigengenes

---

Description

Cell colors represent strength of correlations. When bootstrap results exist, 95% percentile intervals are also included.

Usage

```
plot_module_trait_correlations(
  module_trait_correlations,
  labels,
  modules = "all",
  mfrow = c(1, 2),
  main = NA,
  plot_avg_differences = FALSE,
  mode = "spca",
  color_palette = (grDevices::colorRampPalette(c("blue", "white", "red")))(50),
```

```
  cex = 1.2,
  show_legend = TRUE
)
```

Arguments

module_trait_correlations	Result of compute_module_trait_correlations().
labels	Group labels, if present, in the same order as the expression data.
modules	Module labels to plot, or "all" (default).
mfrow	Number of rows and columns per page (grid layout).
main	Optional title. If NA, titles identify method and group.
plot_avg_differences	Show per-module RMSE across traits between classic and SPCA correlations; requires mode = "both".
mode	"spca", "classic", or "both".
color_palette	Colors for the diverging correlation scale.
cex	Base text size.
show_legend	Logical. Whether to display color legends.

Value

A named list of ggplot objects is returned invisibly. Plots are printed separately.

Examples

```
data("seam_demo", package = "seam")
fit = spca_all_modules(
 seam_demo$expression_data,
 seam_demo$module_assignment,
 lambda = 0.05,
 bootstrap = FALSE
)
mtc = compute_module_trait_correlations(
 seam_demo$trait_data,
 fit,
 seam_demo$module_assignment,
 mode = "both"
)
plot_module_trait_correlations(
 mtc,
 labels = seam_demo$labels,
 mode = "spca"
)
```

plot_sparsity *Plot percentage of inactive variables (sparsity) across modules as a heatmap*

Description

Cell colors show sparsity.

Usage

```
plot_sparsity(
  spca_data,
  labels = NA,
  modules = "all",
  y_axis_labels = NA,
  cex = 1.2,
  main = "Dead variables [in %]",
  color_palette = (grDevices::colorRampPalette(c("blue", "white", "red")))(50),
  show_legend = TRUE,
  show_size = TRUE
)
```

Arguments

spca_data	Result of spca_all_modules() or spca_all_modules_grid().
labels	Group labels, if present, in the same order as the expression data.
modules	Module labels to plot, or "all" (default).
y_axis_labels	Optional custom module labels.
cex	Base text size.
main	Plot title.
color_palette	Colors for the continuous heatmap scale.
show_legend	Logical. Whether to display the color legend.
show_size	Logical. Whether to include module sizes in row labels.

Value

A ggplot object, printed and returned invisibly.

Examples

```
data("seam_demo", package = "seam")
fit = spca_all_modules(
 seam_demo$expression_data,
 seam_demo$module_assignment,
 lambda = 0.05
```

```
)
plot_sparsity(
 fit,
 labels = seam_demo$labels
)
```

`seam` *seam: Sparse Eigengene Analysis of Modules*

Description

Tools for extended analysis of weighted gene co-expression networks using sparse principal component analysis. Includes functions for group comparison, visualizing of drivers, and clinical trait correlation analysis.

`seam_demo` *Simulated expression and trait data for two groups and two modules*

Description

A reproducible example with 48 samples per group, 24 variables, two modules (`blue` and `red`), and two continuous traits. Different subsets of variables dominate the module signals in groups A and B; the first trait has an opposite relationship to the blue module across groups. Data are simulated and have no biological interpretation.

Usage

```
data(seam_demo)
```

Format

A list with four components:

expression_data Named list of two 48-by-24 numeric matrices.

trait_data Named list of two 48-by-2 numeric data frames.

module_assignment Character vector of 24 module labels.

labels Character vector `c("A", "B")`.

Source

Simulated with a fixed random seed; see `data-raw/seam_demo.R`.

`spca_all_modules` *Perform SPCA across groups and modules*

Description

Fits SPCA to the original data and optionally to bootstrap resamples. Input expression matrices must contain no missing values.

Usage

```
spca_all_modules(
  expression_data,
  module_assignment,
  lambda,
  modules = "all",
  bootstrap = TRUE,
  n = 100
)
```

Arguments

`expression_data`	List of numeric expression matrices (samples in rows).
`module_assignment`	Vector of modules assigned to variables. One module per variable.
`lambda`	Non-negative sparsity penalty passed to the `para` argument of `elasticnet::spca()`.
`modules`	Module labels to plot, or `"all"` (default).
`bootstrap`	Logical. Whether to compute bootstrap replicates.
`n`	Number of bootstrap replicates.

Details

Bootstrap resampling may be used to assess the stability of variable selection under repeated sampling.

#' @examples data("seam_demo", package = "seam") fit = spca_all_modules(seam_demo$expression_data, seam_demo$module_assignment, lambda = 0.05, bootstrap = FALSE) fit$var_exp

Value

List with original PCs, var_exp, vs, module_matrices, module_colors, module_sizes, module_assignment, module_signs, and optional bootstrap list (PCs, var_exp, vs, module_signs, samples).

`spca_all_modules_grid` *Perform SPCA across groups and modules with module-specific sparsity*

Description

Fits SPCA to the original data and optionally to bootstrap resamples. Input expression matrices must contain no missing values. Allows for different sparsity penalties across modules.

Usage

```
spca_all_modules_grid(
  expression_data,
  module_assignment,
  modules = "all",
  lambdas,
  bootstrap = TRUE,
  n = 100
)
```

Arguments

`expression_data`	List of numeric expression matrices (samples in rows).
`module_assignment`	Vector of modules assigned to variables. One module per variable.
`modules`	Module labels to plot, or `"all"` (default).
`lambdas`	Non-negative sparsity penalties passed to the `para` argument of `elasticnet::spca()`. For no module assignment, supply a single value.
`bootstrap`	Logical. Whether to compute bootstrap replicates.
`n`	Number of bootstrap replicates.

Details

Bootstrap resampling may be used to assess the stability of variable selection under repeated sampling.

Value

List with original PCs, var_exp, vs, module_matrices, module_colors, module_sizes, module_assignment, module_signs, and optional bootstrap list (PCs, var_exp, vs, module_signs, samples).

Examples

```
data("seam_demo", package = "seam")
fit = spca_all_modules_grid(
 seam_demo$expression_data,
 seam_demo$module_assignment,
 lambdas = c(blue = 0.05, red = 0.08),
 bootstrap = FALSE
)
```

Index

∗ **datasets**
seam_demo, 21

compute_activity, 2
compute_module_trait_correlations, 3
compute_variable_trait_correlations, 4

elasticnet::spca(), *22, 23*

insight_correlations_lollipop, 5
insight_correlations_reconstruct, 7
insight_correlations_sign_changes, 9
insight_weights_lollipop, 10

plot_activity_bar, 12
plot_activity_bubble, 13
plot_explained_variance, 14
plot_lambdas, 16
plot_module_heatmaps, 17
plot_module_trait_correlations, 18
plot_sparsity, 20

seam, 21
seam_demo, 21
spca_all_modules, 22
spca_all_modules_grid, 23