Package ‘corrselect’

September 2, 2025

Title Correlation-Based Variable Subset Selection

Version 2.0.1

Description

Provides functions to extract low-correlation variable subsets using exact graph-theoretic algorithms (e.g., Eppstein–Löffler–Strash, Bron–Kerbosch) as well as greedy and spectral heuristics. Supports both numeric and mixed-type data using generalized association measures.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

LinkingTo Rcpp

Imports Rcpp, methods, stats

Suggests GO.db, WGCNA, preprocessCore, impute, energy, minerva, knitr, rmarkdown

VignetteBuilder knitr

URL `https://gillescolling.com/corrselect/`

BugReports `https://github.com/gcol33/corrselect/issues`

NeedsCompilation yes

Author Gilles Colling [aut, cre]

Maintainer Gilles Colling <`gilles.colling051@gmail.com`>

Archs x64

Contents

as.data.frame.CorrCombo *Coerce CorrCombo to a Data Frame*

Description

Converts a CorrCombo object into a data frame of variable combinations.

Usage

```
## S3 method for class 'CorrCombo'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

x	A CorrCombo object.
row.names	Optional row names for the output data frame.
optional	Logical. Passed to data.frame().
...	Additional arguments passed to data.frame().

Value

A data frame where each row corresponds to a subset of variables. Columns are named VarName01, VarName02, ..., up to the size of the largest subset. Subsets shorter than the maximum length are padded with NA.

See Also

CorrCombo

Examples

```
set.seed(1)
mat <- matrix(rnorm(100), ncol = 10)
colnames(mat) <- paste0("V", 1:10)
res <- corrSelect(cor(mat), threshold = 0.5)
as.data.frame(res)
```

assocSelect *Select Variable Subsets with Low Association (Mixed-Type Data Frame Interface)*

Description

Identifies combinations of variables of any common data type (numeric, ordered factors, or unordered) factors—whose *pair-wise association* does not exceed a user-supplied threshold. The routine wraps MatSelect() and handles all pre-processing (type conversion, missing-row removal, constant-column checks) for typical data-frame/tibble/data-table inputs.

Usage

```
assocSelect(
  df,
  threshold = 0.7,
  method = NULL,
  force_in = NULL,
  method_num_num = c("pearson", "spearman", "kendall", "bicor", "distance", "maximal"),
  method_num_ord = c("spearman", "kendall"),
  method_ord_ord = c("spearman", "kendall"),
  ...
)
```

Arguments

`df` A data frame (or tibble / data.table). May contain any mix of:

- numeric / integer (treated as numeric)
- ordered factors
- unordered factors (character vectors are coerced to factors)

`threshold` Numeric in $(0, 1)$. Maximum allowed pair-wise *absolute* association. Default `0.7`.

`method` Character; the subset-search algorithm. One of `"els"` or `"bron-kerbosch"`. If `NULL` (default) the function selects automatically: ELS when `force_in` is supplied, otherwise Bron–Kerbosch.

`force_in` Optional character vector or column indices specifying variables that must appear in every returned subset.

`method_num_num` Association measure for numeric–numeric pairs. One of `"pearson"` (default), `"spearman"`, `"kendall"`, `"bicor"`, `"distance"`, or `"maximal"`.

`method_num_ord` Association measure for numeric–ordered pairs. One of `"spearman"` (default) or `"kendall"`.

`method_ord_ord` Association measure for ordered–ordered pairs. One of `"spearman"` (default) or `"kendall"`.

`...` Additional arguments passed unchanged to `MatSelect()` (e.g., `use_pivot = TRUE` for Bron–Kerbosch).

Details

A single call can therefore screen a data set that mixes continuous and categorical features and return every subset whose internal associations are "sufficiently low" under the metric(s) you choose.

Rows containing `NA` are dropped with a warning; constant columns are treated as having zero association with every other variable.

The default association measure for each variable-type combination is:

numeric – numeric `method_num_num` (default `"pearson"`)

numeric – ordered `method_num_ord`

numeric – unordered `"eta"` (ANOVA η^2)

ordered – ordered `method_ord_ord`

ordered – unordered `"cramersv"`

unordered – unordered `"cramersv"`

All association measures are rescaled to $[0, 1]$ before thresholding. External packages are required for "bicor" (**WGCNA**), "distance" (**energy**), and "maximal" (**minerva**); an informative error is thrown if they are missing.

Value

A CorrCombo S4 object containing:

- all valid subsets,
- their summary association statistics,
- metadata (algorithm used, rows kept, forced-in variables, etc.).

The object's show() method prints the association metrics that were *actually used* for this data set.

See Also

corrSelect(), MatSelect(), corrSubset()

Examples

```
df <- data.frame(
  height = rnorm(15, 170, 10),
  weight = rnorm(15, 70, 12),
  group  = factor(rep(LETTERS[1:3], each = 5)),
  score  = ordered(sample(c("low","med","high"), 15, TRUE))
)

## keep every subset whose internal associations <= 0.6
assocSelect(df, threshold = 0.6)

## use Kendall for all rank-based comparisons and force 'height' to appear
assocSelect(df,
            threshold       = 0.5,
            method_num_num  = "kendall",
            method_num_ord  = "kendall",
            method_ord_ord  = "kendall",
            force_in        = "height")
```

CorrCombo *CorrCombo S4 class*

Description

Holds the result of corrSelect or MatSelect: a list of valid variable combinations and their correlation statistics.

This class stores all subsets of variables that meet the specified correlation constraint, along with metadata such as the algorithm used, correlation method(s), variables forced into every subset, and summary statistics for each combination.

An S4 class that stores the result of correlation-based subset selection.

Usage

```
## S4 method for signature 'CorrCombo'
show(object)
```

Arguments

`object`	A `CorrCombo` object to be printed.

Slots

`subset_list` A list of character vectors. Each vector is a valid subset (variable names).

`avg_corr` A numeric vector. Average absolute correlation within each subset.

`min_corr` A numeric vector. Minimum pairwise absolute correlation in each subset.

`max_corr` A numeric vector. Maximum pairwise absolute correlation within each subset.

`names` Character vector of all variable names used for decoding.

`threshold` Numeric scalar. The correlation threshold used during selection.

`forced_in` Character vector. Variable names that were forced into each subset.

`search_type` Character string. One of `"els"` or `"bron-kerbosch"`.

`cor_method` Character string. Either a single method (e.g. "pearson") or "mixed" if multiple methods used.

`n_rows_used` Integer. Number of rows used for computing the correlation matrix (after removing missing values).

`subset_list` A list of character vectors, each representing a subset of variable names.

`avg_corr` Numeric vector: average correlation of each subset.

`min_corr` Numeric vector: minimum correlation of each subset.

`max_corr` Numeric vector: maximum correlation of each subset.

`names` Character vector of variable names in the original matrix.

`threshold` Numeric threshold used for correlation filtering.

`forced_in` Character vector of variables that were forced into all subsets.

`search_type` Character: the search algorithm used (e.g., "els", "bron-kerbosch").

`cor_method` Character: the correlation method used.

`n_rows_used` Integer: number of rows used to compute correlations.

See Also

`corrSelect`, `MatSelect`, `corrSubset`

Examples

```
show(new("CorrCombo",
  subset_list = list(c("A", "B"), c("A", "C")),
  avg_corr = c(0.2, 0.3),
  min_corr = c(0.1, 0.2),
  max_corr = c(0.3, 0.4),
  names = c("A", "B", "C"),
  threshold = 0.5,
  forced_in = character(),
  search_type = "els",
  cor_method = "mixed",
  n_rows_used = as.integer(5)
))
```

corrSelect *Select Variable Subsets with Low Correlation (Data Frame Interface)*

Description

Identifies combinations of numeric variables in a data frame such that all pairwise absolute correlations fall below a specified threshold. This function is a wrapper around `MatSelect()` and accepts data frames, tibbles, or data tables with automatic preprocessing.

Usage

```
corrSelect(
  df,
  threshold = 0.7,
  method = NULL,
  force_in = NULL,
 cor_method = c("pearson", "spearman", "kendall", "bicor", "distance", "maximal"),
  ...
)
```

Arguments

`df` A data frame. Only numeric columns are used.

`threshold` A numeric value in (0, 1). Maximum allowed absolute correlation. Defaults to 0.7.

`method` Character. Selection algorithm to use. One of `"els"` or `"bron-kerbosch"`. If not specified, the function chooses automatically: `"els"` when `force_in` is provided, otherwise `"bron-kerbosch"`.

`force_in` Optional character vector or numeric indices of columns to force into all subsets.

`cor_method` Character string indicating which correlation method to use. One of `"pearson"` (default), `"spearman"`, `"kendall"`, `"bicor"`, `"distance"`, or `"maximal"`.

`...` Additional arguments passed to `MatSelect()`, e.g., `use_pivot`.

Details

Only numeric columns are used for correlation analysis. Non-numeric columns (factors, characters, logicals, etc.) are ignored, and their names and types are printed to inform the user. These can be optionally reattached later using `corrSubset()` with `keepExtra = TRUE`.

Rows with missing values are removed before computing correlations. A warning is issued if any rows are dropped.

The `cor_method` controls how the correlation matrix is computed:

- `"pearson"`: Standard linear correlation.
- `"spearman"`: Rank-based monotonic correlation.

- "kendall": Kendall's tau.
- "bicor": Biweight midcorrelation (WGCNA::bicor).
- "distance": Distance correlation (energy::dcor).
- "maximal": Maximal information coefficient (minerva::mine).

For "bicor", "distance", and "maximal", the corresponding package must be installed.

Value

An object of class CorrCombo, containing selected subsets and correlation statistics.

See Also

assocSelect(), MatSelect(), corrSubset()

Examples

```
set.seed(42)
n <- 100

# Create 20 variables: 5 blocks of correlated variables + some noise
block1 <- matrix(rnorm(n * 4), ncol = 4)
block2 <- matrix(rnorm(n), ncol = 1)
block2 <- matrix(rep(block2, 4), ncol = 4) + matrix(rnorm(n * 4, sd = 0.1), ncol = 4)
block3 <- matrix(rnorm(n * 4), ncol = 4)
block4 <- matrix(rnorm(n * 4), ncol = 4)
block5 <- matrix(rnorm(n * 4), ncol = 4)

df <- as.data.frame(cbind(block1, block2, block3, block4, block5))
colnames(df) <- paste0("V", 1:20)

# Add a non-numeric column to be ignored
df$label <- factor(sample(c("A", "B"), n, replace = TRUE))

# Basic usage
corrSelect(df, threshold = 0.8)

# Try Bron-Kerbosch with pivoting
corrSelect(df, threshold = 0.6, method = "bron-kerbosch", use_pivot = TRUE)

# Force in a specific variable and use Spearman correlation
corrSelect(df, threshold = 0.6, force_in = "V10", cor_method = "spearman")
```

---

corrSubset *Extract Variable Subsets from a CorrCombo Object*

Description

Extracts one or more variable subsets from a CorrCombo object as data frames. Typically used after corrSelect or MatSelect to obtain filtered versions of the original dataset containing only low-correlation variable combinations.

Usage

```
corrSubset(res, df, which = "best", keepExtra = FALSE)
```

Arguments

res A CorrCombo object returned by corrSelect or MatSelect.

df A data frame or matrix. Must contain all variables listed in res@names. Columns not in res@names are ignored unless keepExtra = TRUE.

which Subsets to extract. One of:

- "best" (default) or 1: the top-ranked subset.
- A single integer (e.g. 2): the nth ranked subset.
- A vector of integers (e.g. 1:3): multiple subsets.
- "all": all available subsets.

Subsets are ranked by decreasing size, then increasing average correlation.

keepExtra Logical. If TRUE, columns in df not in res@names (e.g., factors, characters) are retained. Defaults to FALSE.

Value

A data frame if a single subset is extracted, or a list of data frames if multiple subsets are extracted. Each data frame contains the selected variables (and optionally extras).

Note

A warning is issued if any rows contain missing values in the selected variables.

See Also

corrSelect, MatSelect, CorrCombo

Examples

```
# Simulate input data
set.seed(123)
df <- as.data.frame(matrix(rnorm(100), nrow = 10))
colnames(df) <- paste0("V", 1:10)

# Compute correlation matrix
cmat <- cor(df)

# Select subsets using corrSelect
res <- corrSelect(cmat, threshold = 0.5)

# Extract the best subset (default)
corrSubset(res, df)

# Extract the second-best subset
corrSubset(res, df, which = 2)

# Extract the first three subsets
corrSubset(res, df, which = 1:3)

# Extract all subsets
```

```
corrSubset(res, df, which = "all")

# Extract best subset and retain additional numeric column
df$CopyV1 <- df$V1
corrSubset(res, df, which = 1, keepExtra = TRUE)
```

MatSelect *Select Variable Subsets with Low Correlation or Association (Matrix Interface)*

Description

Identifies all maximal subsets of variables from a symmetric matrix (typically a correlation matrix) such that all pairwise absolute values stay below a specified threshold. Implements exact algorithms such as Eppstein–Löffler–Strash (ELS) and Bron–Kerbosch (with or without pivoting).

Usage

```
MatSelect(mat, threshold = 0.7, method = NULL, force_in = NULL, ...)
```

Arguments

mat	A numeric, symmetric matrix with 1s on the diagonal (e.g. correlation matrix). Column names (if present) are used to label output variables.
threshold	A numeric scalar in (0, 1). Maximum allowed absolute pairwise value. Defaults to 0.7.
method	Character. Selection algorithm to use. One of "els" or "bron-kerbosch". If not specified, the function chooses automatically: "els" when force_in is provided, otherwise "bron-kerbosch".
force_in	Optional integer vector of 1-based column indices to force into every subset.
...	Additional arguments passed to the backend, e.g., use_pivot (logical) for enabling pivoting in Bron–Kerbosch (ignored by ELS).

Value

An object of class CorrCombo, containing all valid subsets and their correlation statistics.

Examples

```
set.seed(42)
mat <- matrix(rnorm(100), ncol = 10)
colnames(mat) <- paste0("V", 1:10)
cmat <- cor(mat)

# Default method (Bron-Kerbosch)
res1 <- MatSelect(cmat, threshold = 0.5)

# Bron-Kerbosch without pivot
res2 <- MatSelect(cmat, threshold = 0.5, method = "bron-kerbosch", use_pivot = FALSE)

# Bron-Kerbosch with pivoting
```

```
res3 <- MatSelect(cmat, threshold = 0.5, method = "bron-kerbosch", use_pivot = TRUE)

# Force variable 1 into every subset (with warning if too correlated)
res4 <- MatSelect(cmat, threshold = 0.5, force_in = 1)
```

Index