Package 'CoxMK'

August 18, 2025

Type Package

Title A Model-X Knockoff Method for Genome-Wide Survival Association Analysis

Version 0.1.0

Author Yang Chen [aut, cre], Contributors [ctb]

Maintainer Yang Chen <yangchen5@stu.scu.edu.cn>

Description A genome-wide survival framework that integrates sequential conditional independent tuples and saddlepoint approximation method, to provide SNP-level false discovery rate control while improving power, particularly for biobank-scale survival analyses with low event rates. A shrinkage algorithmic leveraging accelerates multiple knockoffs generation in large genetic cohorts.

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 3.5.0)

Imports Matrix,
survival,
irlba,
stats,
utils,
gdsfmt,
BEDMatrix

Suggests SPACox,
testthat (>= 3.0.0)

Remotes github::WenjianBI/SPAcox

Config/testthat/edition 3

URL https://github.com/xiaoxiandadada/Cox-MK

BugReports https://github.com/xiaoxiandadada/Cox-MK/issues

Contents

calculate_w_statistics

Calculate W Statistics for Knockoff Analysis

Description

Computes W statistics by comparing test statistics from original variables with those from their knockoff counterparts. These statistics are used for variable selection with FDR control.

Usage

```
calculate_w_statistics(t_orig, t_knock, method = "median")
```

Arguments

`t_orig`	Vector of test statistics for original variables
`t_knock`	Vector or list of test statistics for knockoff variables. If a list, should contain M vectors of the same length as t_orig.
`method`	Method for computing W statistics: • "difference": W_j = T_j - max(T_{j,k}) (default) • "median": Uses Model-X knockoff median-based statistics • "ratio": W_j = T_j / max(T_{j,k})

Value

Vector of W statistics for variable selection

Examples

```
## Not run:
# Example with difference method
t_orig <- c(5.2, 3.1, 8.7, 2.4, 6.9)
t_knock <- list(
  c(2.1, 4.2, 3.3, 1.8, 2.9),
  c(1.9, 3.8, 4.1, 2.2, 3.1)
)

w_median <- calculate_w_statistics(t_orig, t_knock, method = "median")
w_diff <- calculate_w_statistics(t_orig, t_knock, method = "difference")

## End(Not run)
```

create_knockoffs *Create Multiple Knockoffs for Genetic Data*

Description

Generate knockoff variables for genotype data using the Multiple knockoff method with leveraging scores and clustering specifically optimized for genetic variant data.

Usage

```
create_knockoffs(
  X,
  pos,
  chr_info = NULL,
  sample_ids = NULL,
  M = 5,
  save_gds = TRUE,
  output_dir = NULL,
  start = NULL,
  end = NULL,
  corr_max = 0.75,
  maxN_neighbor = Inf,
  maxBP_neighbor = 1e+05,
  n_AL = floor(10 * nrow(X)^(1/3) * log(nrow(X))),
  thres_ultrarare = 25,
  R2_thres = 1,
  prob_eps = 1e-12,
  irlba_maxit = 1500
)
```

Arguments

X	A sparse matrix (n x p) of genotype data where n is the number of samples and p is the number of SNPs. Typically coded as 0, 1, 2 for genotype dosages.
pos	A numeric vector of SNP positions (in base pairs) for linkage disequilibrium-aware knockoff generation.
chr_info	Optional chromosome information. Can be either: (1) A data frame with chromosome information from BIM file containing a column named "chr" or "CHR" with chromosome numbers, or (2) A vector of chromosome numbers directly. Chromosome information will be automatically extracted.
sample_ids	A character vector of sample IDs (default: NULL, will generate)
M	Number of knockoff copies to generate (default: 5). More copies can improve statistical power but increase computational cost.
save_gds	Whether to save knockoffs to GDS format (default: TRUE)
output_dir	Directory to save GDS files (default: extdata folder)
start	Start position for file naming (default: min(pos))
end	End position for file naming (default: max(pos))
corr_max	Maximum correlation threshold for clustering variants (default: 0.75). Higher values create fewer, larger clusters.

`maxN_neighbor` Maximum number of neighboring variants to consider for each variant (default: Inf).

`maxBP_neighbor` Maximum base pair distance to consider variants as neighbors (default: 100,000 bp).

`n_AL` Number of samples to use for adaptive lasso fitting (default: automatically determined based on sample size).

`thres_ultrarare` Minimum minor allele count threshold for variant inclusion (default: 25).

`R2_thres` R-squared threshold for model fitting (default: 1).

`prob_eps` Minimum probability value to prevent numerical issues (default: 1e-12).

`irlba_maxit` Maximum iterations for truncated SVD (default: 1500).

Value

If save_gds is TRUE, returns the path to the saved GDS file. Otherwise, returns a list of M matrices, each of the same dimensions as X, containing knockoff variables.

`fit_cox_model_from_files` *Step 2: Fit Cox Model from Files*

Description

Implements Step 2 of the CoxMK workflow: fitting a null Cox proportional hazards model by reading phenotype and covariate data from files. This function is designed for batch processing and large-scale analysis where data is stored in separate files.

Usage

```
fit_cox_model_from_files(
  phenotype_file,
  covariate_file,
  output_file,
  use_spacox = TRUE
)
```

Arguments

`phenotype_file` Path to CSV file with columns: IID, time, status

`covariate_file` Path to CSV file with columns: IID, covar1, covar2, ...

`output_file` Path to RDS file to save the fitted null model

`use_spacox` Whether to try using SPACox package (default: TRUE)

Value

Invisible path to the output file

Examples

```
## Not run:
# Prepare example data files
pheno_data <- data.frame(
  IID = paste0("ID", 1:100),
  time = rexp(100, 0.1),
  status = rbinom(100, 1, 0.3)
)
covar_data <- data.frame(
  IID = paste0("ID", 1:100),
  age = rnorm(100, 50, 10),
  sex = rbinom(100, 1, 0.5)
)

write.csv(pheno_data, "phenotype.csv", row.names = FALSE)
write.csv(covar_data, "covariates.csv", row.names = FALSE)

# Step 2: Fit null Cox model from files
fit_cox_model_from_files(
  phenotype_file = "phenotype.csv",
  covariate_file = "covariates.csv",
  output_file = "null_model.rds"
)

# Load the fitted model for Step 3
model_info <- readRDS("null_model.rds")

## End(Not run)
```

`knockoff_filter` *Apply Knockoff Filter for Variable Selection*

Description

Applies the knockoff filter to select variables while controlling the false discovery rate (FDR) at a specified level.

Usage

```
knockoff_filter(W, fdr = 0.1, offset = 1)
```

Arguments

`W`	Vector of W statistics from `calculate_w_statistics`
`fdr`	Target false discovery rate (default: 0.1)
`offset`	Offset parameter for knockoff filter (default: 1)

Value

Vector of indices of selected variables

Examples

```
## Not run:
# Generate some example W statistics
W <- c(2.1, -0.5, 3.8, -1.2, 4.5, 0.3, -2.1, 1.9)

# Apply knockoff filter
selected <- knockoff_filter(W, fdr = 0.1)
print(selected)  # Indices of selected variables

## End(Not run)
```

Index