Package 'CoxMK'

August 16, 2025

Type Package

Title Cox Regression with Model-X Knockoffs for Survival Analysis

Version 0.1.0

Author yangchen

Description Implements Cox regression analysis with Model-X knockoffs for variable selection in survival analysis. The package provides functions for generating knockoff variables, fitting null models using SPACox (<https://github.com/WenjianBI/SPACox>), performing association analysis, and calculating W statistics for variable selection with false discovery rate control. SPACox is recommended for fitting null models with large-scale genetic data.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Depends R (>= 3.5.0)

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

Suggests bigmemory,
SPACox,
SPAtest,
testthat (>= 3.0.0),
knitr,
rmarkdown

VignetteBuilder knitr

Config/testthat/edition 3

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

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

Contents

`CoxMK-package` *CoxMK: Cox Regression with Model-X Knockoffs*

Description

This package provides a complete workflow for Cox regression analysis with Model-X knockoffs for variable selection in survival analysis.

For detailed workflow information, see `CoxMK-workflow`.

The workflow follows four main steps:

1. **Generate Knockoffs:** `step1_create_knockoffs`
2. **Fit Null Model:** `step2_fit_null_cox_model`
3. **Association Testing:** `step3_perform_association_testing`
4. **Variable Selection:** `step4_knockoff_filter`

Quick Start

See `CoxMK-workflow` for complete workflow documentation and examples.

For one-step analysis, use `cox_knockoff_analysis`.

Four-Step Functions

- `step1_create_knockoffs` - Step 1: Generate knockoff variables
- `step2_fit_null_cox_model` - Step 2: Fit null Cox model with SPACox
- `step3_perform_association_testing` - Step 3: Association testing
- `step4_knockoff_filter` - Step 4: Variable selection with FDR control

Author(s)

CoxMK Development Team

`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(Z_orig, Z_ko, method = "median")
```

Arguments

`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})
`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.

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)
```

CoxMK *CoxMK: Cox Regression with Model-X Knockoffs*

Description

Main interface functions for Cox regression analysis with Model-X knockoffs. This package provides a complete workflow for survival analysis with variable selection using the knockoff methodology.

The workflow follows four main steps: 1. **Generate Knockoffs**: Create knockoff variables using `create_knockoffs` 2. **Fit Null Model**: Fit null Cox model using `fit_null_cox_model` 3. **Perform Testing**: Conduct association testing using `perform_association_testing` 4. **Apply Filter**: Select variables using `knockoff_filter`

Main Functions

- `cox_knockoff_analysis` - Complete knockoff analysis workflow
- `create_knockoffs` - Step 1: Generate knockoff variables
- `fit_null_cox_model` - Step 2: Fit null Cox model for testing
- `perform_association_testing` - Step 3: Perform association testing
- `knockoff_filter` - Step 4: Apply knockoff filter for variable selection

CoxMK-workflow *CoxMK: Four-Step Cox Regression with Model-X Knockoffs Workflow*

Description

Complete workflow for Cox regression analysis with Model-X knockoffs for survival analysis. This package implements a four-step methodology for variable selection with false discovery rate control.

Overview

The CoxMK workflow consists of four main steps:

Step 1: Generate Knockoff Variables: Main Function: `create_knockoffs`
Supporting Functions:

- `load_plink_data` - Load genotype data from PLINK format

Purpose: Generate knockoff variables that preserve the correlation structure of the original variables while being conditionally independent.
Input: Genotype matrix or PLINK files **Output:** Knockoff matrices stored in GDS format

Step 2: Fit Null Cox Model: Main Function: `fit_null_cox_model`
Supporting Functions:

- `prepare_phenotype` - Prepare survival time and status data
- `load_covariates` - Load covariate data

Purpose: Fit null Cox proportional hazards model using SPACox for efficient large-scale analysis.
Input: Phenotype data (time, status) and covariates **Output:** Fitted null model object

Step 3: Perform Association Testing: Main Function: `perform_association_testing`
Purpose: Conduct association testing using both original and knockoff variables with the fitted null model.
Input: Original and knockoff genotype data, fitted null model **Output:** Test statistics for original and knockoff variables

Step 4: Apply Knockoff Filter: Main Function: `knockoff_filter`
Supporting Functions:

- `calculate_w_statistics` - Calculate W statistics

Purpose: Apply the knockoff filter to select significant variables with false discovery rate control.
Input: Test statistics from original and knockoff variables **Output:** Selected variables with FDR control

Complete Workflow Function

One-Step Analysis: `cox_knockoff_analysis`

For users who want to run the complete analysis in one step, the `cox_knockoff_analysis` function provides a wrapper that executes all four steps automatically.

Example Workflow

```
# Step-by-step analysis
library(CoxMK)

# Step 1: Generate knockoffs
knockoffs <- create_knockoffs(X, M = 5)

# Step 2: Fit null model
null_model <- fit_null_cox_model(time, status, covariates)

# Step 3: Perform association testing
results <- perform_association_testing(X_orig, X_knockoffs, null_model)

# Step 4: Apply knockoff filter
selected <- knockoff_filter(results$W_stats, fdr = 0.05)

# Alternative: Complete analysis in one step
results <- cox_knockoff_analysis(
  plink_prefix = "data/genotypes",
  time = survival_time,
  status = event_status,
  covariates = covar_data,
  fdr = 0.05
)
```

See Also

Step 1: `step1_create_knockoffs`, `create_knockoffs`

Step 2: `step2_fit_null_cox_model`, `fit_null_cox_model`

Step 3: step3_perform_association_testing, perform_association_testing

Step 4: step4_knockoff_filter, knockoff_filter

Complete Workflow: cox_knockoff_analysis

---

cox_knockoff_analysis *Complete Cox Knockoff Analysis Workflow*

Description

Performs a complete Model-X knockoff analysis following the four-step workflow: 1. Generate knockoff variables from PLINK data and save to GDS format 2. Fit null Cox model using SPACox for efficient large-scale analysis 3. Perform SPA testing using original and knockoff variables 4. Apply knockoff filter for variable selection with FDR control

Usage

```
cox_knockoff_analysis(
  plink_prefix,
  time,
  status,
  covariates = NULL,
  sample_ids = NULL,
  null_model = NULL,
  gds_file = NULL,
  M = 5,
  fdr = 0.05,
  method = "median",
  output_dir = NULL
)
```

Arguments

plink_prefix	Path prefix for PLINK files (without extension)
time	Survival times
status	Event indicators (1=event, 0=censored)
covariates	Optional covariate matrix/data.frame
sample_ids	Sample IDs (optional, will be generated from .fam file)
null_model	Pre-fitted null Cox model or path to RDS file with fitted model (optional)
gds_file	Path to pre-generated GDS file with knockoff data (optional, if provided, knockoffs will be loaded instead of generated)
M	Number of knockoff copies to generate (default: 5)
fdr	Target false discovery rate (default: 0.05)
method	Statistical method for W statistics ("median", "difference")
output_dir	Directory to save GDS files (default: extdata folder)

Value

List containing:

selected_vars	Indices of selected variables
W_stats	W statistics for all variables
threshold	Knockoff threshold used
gds_file	Path to GDS file used
null_model	Fitted null Cox model
test_results	SPA test results

Examples

```
## Not run:
# Standard workflow with PLINK data
extdata_path <- system.file('extdata', package = 'CoxMK')
plink_prefix <- file.path(extdata_path, 'sample')
pheno_data <- prepare_phenotype(file.path(extdata_path, 'tte_phenotype.txt'))
covar_data <- load_covariates(file.path(extdata_path, 'covariates.txt'))

# Option 1: Complete analysis in one step
result <- cox_knockoff_analysis(
  plink_prefix = plink_prefix,
  time = pheno_data$time,
  status = pheno_data$status,
  covariates = covar_data,
  M = 3,
  fdr = 0.1
)

# Option 2: Step-by-step workflow with GDS file reuse
# Step 2a: Generate knockoffs and save to GDS file
knockoffs <- create_knockoffs(
  X = load_plink_data(plink_prefix)$genotypes,
  pos = load_plink_data(plink_prefix)$positions,
  M = 3
)
gds_file <- knockoffs$gds_file  # GDS file path

# Step 2b: Fit null model separately
null_model <- fit_null_cox_model(
  time = pheno_data$time,
  status = pheno_data$status,
  covariates = covar_data
)

# Step 2c: Load knockoffs from GDS file and run analysis
result <- cox_knockoff_analysis(
  plink_prefix = plink_prefix,
  time = pheno_data$time,
  status = pheno_data$status,
  covariates = covar_data,
  null_model = null_model,
  gds_file = gds_file,  # Use pre-generated GDS file
  M = 3,
  fdr = 0.05
```

```
)
# View selected variables
print(result$selected_vars)
print(result$summary)

## End(Not run)
```

`create_knockoffs` *Create Model-X Knockoffs for Genetic Data*

Description

Generate knockoff variables for genotype data using the Model-X 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 data frame with chromosome information from BIM file. If provided, will extract chromosome number automatically.
`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.

---

data-processing *Load PLINK Data for Analysisata Processing Utilities*

Description

Utility functions for data loading, processing, and validation specifically designed for genetic association studies.

---

fit_null_cox_model *Fit Null Cox Model*

Description

Fits a null Cox model with only covariates (no genetic variants). This function uses SPACox when available for large-scale analysis, and falls back to standard Cox regression otherwise.

Usage

```
fit_null_cox_model(time, status, covariates = NULL)
```

Arguments

time	Survival times
status	Event indicators
covariates	Covariate data frame (optional)

Value

Fitted Cox model object (SPACox or coxph)

Examples

```
## Not run:
# Example with covariates
data(example_phenotype)
data(example_covariates)

null_model <- fit_null_cox_model(
  time = example_phenotype$time,
  status = example_phenotype$status,
  covariates = example_covariates
)

# Example without covariates
null_model <- fit_null_cox_model(
  time = example_phenotype$time,
  status = example_phenotype$status
)

## End(Not run)
```

---

knockoff-generation *Knockoff Variable Generation*

Description

Functions for generating Model-X knockoff variables using leveraging scores, clustering, and adaptive lasso methods specifically designed for genetic data.

---

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)
```

---

load_covariates *Load Covariate Data*

Description

Loads and processes covariate data for inclusion in survival models.

Usage

```
load_covariates(covariate_file, exclude_cols = c("FID", "IID"))
```

Arguments

covariate_file	Path to covariate file
exclude_cols	Column names to exclude from covariate matrix, typically the analysis (default: c("FID", "IID")).

Value

Data frame of covariates

---

perform_association_testing *Perform Association Testing*

Description

Performs association testing with genotype data using a fitted null model. This function handles both SPACox and standard Cox regression methods for genetic variant association analysis.

Usage

```
perform_association_testing(X, null_model)
```

Arguments

X	Genotype matrix (samples × SNPs)
null_model	Fitted null Cox model from fit_null_cox_model()

Value

List with test statistics and p-values

Examples

```
## Not run:
# Fit null model first
null_model <- fit_null_cox_model(time, status, covariates)

# Perform association testing
results <- perform_association_testing(genotype_matrix, null_model)

# Extract results
test_stats <- results$test_stats
p_values <- results$p_values

## End(Not run)
```

---

step1_create_knockoffs

Step 1: Generate Knockoff Variables

Description

Generate knockoff variables using the Model-X approach for variable selection in Cox regression analysis.

Usage

```
create_knockoffs(X, pos = NULL, M = 5, method = "sdp")
```

Arguments

X	Genotype matrix (samples × SNPs)
pos	Genomic positions (optional)
M	Number of knockoff copies (default: 5)
method	Method for knockoff generation ("sdp", "equi", "asdp")

Details

This is Step 1 of the 4-step CoxMK workflow. It generates knockoff variables that preserve the correlation structure of the original variables while being conditionally independent given the original variables.

Value

List containing:

knockoffs	List of M knockoff matrices
method	Method used
Q	Correlation matrix (if applicable)

See Also

step2_fit_null_cox_model, step3_perform_association_testing, step4_knockoff_filter

Examples

```
## Not run:
# Generate knockoffs for genotype data
knockoff_result <- create_knockoffs(X = genotype_matrix, M = 3)

## End(Not run)
```

---

step2_fit_null_cox_model

Step 2: Fit Null Cox Model

Description

Fit a null Cox model using only covariates (no genetic variants). This function uses SPACox when available for large-scale analysis.

Usage

```
fit_null_cox_model(time, status, covariates = NULL)
```

Arguments

time	Survival times
status	Event indicators (1=event, 0=censored)
covariates	Covariate data frame (optional)

Details

This is Step 2 of the 4-step CoxMK workflow. The fitted null model is used for efficient association testing in Step 3.

Value

Fitted Cox model object (SPACox or coxph) with additional metadata:

model_type	Type of model fitted ("SPACox" or "Standard Cox")
formula	Formula used for fitting
n_samples	Number of samples
n_events	Number of events

See Also

step1_create_knockoffs, step3_perform_association_testing, step4_knockoff_filter

Examples

```
## Not run:
# Fit null model with covariates
null_model <- fit_null_cox_model(
  time = survival_time,
  status = event_status,
  covariates = covariate_data
)

## End(Not run)
```

---

step3_perform_association_testing

Step 3: Perform Association Testing

Description

Perform association testing with genotype data using a fitted null model. Handles both SPACox and standard Cox regression methods.

Usage

```
perform_association_testing(X, null_model)
```

Arguments

X	Genotype matrix (samples × SNPs)
null_model	Fitted null Cox model from step2_fit_null_cox_model

Details

This is Step 3 of the 4-step CoxMK workflow. It performs association testing for both original and knockoff variables using the null model from Step 2.

Value

List containing:

test_stats	Test statistics for each SNP
p_values	P-values for each SNP
method	Method used for testing
n_snps	Number of SNPs tested

See Also

step1_create_knockoffs, step2_fit_null_cox_model, step4_knockoff_filter

Examples

```
## Not run:
# Perform association testing
test_results <- perform_association_testing(
  X = genotype_matrix,
  null_model = fitted_null_model
)

## End(Not run)
```

`step4_knockoff_filter` *Step 4: Apply Knockoff Filter*

Description

Apply 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 computed from original and knockoff test statistics
`fdr`	Target false discovery rate (default: 0.1)
`offset`	Offset parameter for knockoff filter (default: 1)

Details

This is Step 4 of the 4-step CoxMK workflow. It applies the knockoff filter to control FDR while selecting truly associated variables.

The W statistics should be computed by comparing test statistics from Step 3 for original variables with those from their knockoff counterparts.

Value

Vector of indices of selected variables. The `threshold` attribute contains the threshold value used for selection.

See Also

`step1_create_knockoffs`, `step2_fit_null_cox_model`, `step3_perform_association_testing`

Examples

```
## Not run:
# Apply knockoff filter
selected_vars <- knockoff_filter(W_statistics, fdr = 0.1)

# Check threshold used
threshold <- attr(selected_vars, "threshold")

## End(Not run)
```

Index