Introduction to CoxMK

Yang Chen

2025-08-15

Contents

1 Introduction

The `CoxMK` package provides a comprehensive toolkit for performing survival analysis using the Cox proportional hazards model combined with Model-X knockoffs. This methodology allows for robust variable selection in high-dimensional settings, such as genetic association studies, while controlling the false discovery rate (FDR).

The key features of the package include:

- **Knockoff Generation**: Efficiently create knockoff variables for large-scale genetic data.
- **Association Analysis**: Fit Cox models for both original and knockoff variables.
- **Variable Selection**: Calculate W statistics and apply the knockoff filter to select significant variables.
- **Data Handling**: Utilities for loading PLINK data and managing knockoffs with the GDS file format.

2 Installation

You can install the development version of `CoxMK` from GitHub with:

```r
# install.packages("devtools")
devtools::install_github("xiaoxiandadada/Cox-MK")
```

3 The CoxMK Workflow

This section walks through a complete analysis workflow using the example data included in the package.

3.1 Step 1: Load Data

First, we load the required packages and the sample data. The package includes example PLINK files, phenotype data, and covariate data.

```r
library(CoxMK)

# Define path to external data
extdata_path <- system.file("extdata", package = "CoxMK")

# Load PLINK data (genotypes and positions)
plink_data <- load_plink_data(file.path(extdata_path, "sample"))

# Load phenotype and covariate data
phenotype_data <- read.table(file.path(extdata_path, "tte_phenotype.txt"), header = TRUE)
covariate_data <- read.table(file.path(extdata_path, "covariates.txt"), header = TRUE)
```

3.2 Step 2: Create Knockoffs

Next, we generate knockoff variables from the original genotypes. The `create_knockoffs` function handles this, and by default, saves the results to a `.gds` file for efficient storage and reuse.

```r
knockoffs_result <- create_knockoffs(
  X = plink_data$genotypes,
  pos = plink_data$positions,
  M = 5 # Number of knockoff copies
)
```

3.3 Step 3: Perform Association Analysis

We then perform Cox regression to test for associations between each variable (both original and knockoff) and the survival outcome.

```r
# Prepare merged phenotype and covariate data
pheno_data <- merge(phenotype_data, covariate_data, by = c("FID", "IID"))
covariates <- pheno_data[, c("age", "sex", "bmi")]

# Fit Cox model for original variables
original_results <- fit_cox_spa(
  X = plink_data$genotypes,
  time = pheno_data$time,
  status = pheno_data$status,
  covariates = covariates
)

# Fit Cox model for each knockoff copy
knockoff_results <- lapply(knockoffs_result$knockoffs, function(X_k) {
  fit_cox_spa(
    X = X_k,
    time = pheno_data$time,
    status = pheno_data$status,
    covariates = covariates
  )
})
```

3.4 Step 4: Calculate W Statistics

The W statistic contrasts the evidence of association for the original variable with that of its knockoff copies.

```
# Extract test statistics (e.g., z-scores) from results
original_stats <- original_results$test_stats
knockoff_stats <- lapply(knockoff_results, function(res) res$test_stats)

# Calculate W statistics
w_stats <- calculate_w_statistics(
  t_orig = original_stats,
  t_knock = knockoff_stats,
  method = "difference"
)
```

3.5 Step 5: Apply Knockoff Filter

Finally, we apply the knockoff filter to the W statistics to select variables while controlling the FDR at a specified level (e.g., 0.1).

```
selected_snps <- knockoff_filter(w_stats, fdr = 0.1)

# View selected SNP indices
print(selected_snps)
```

4 Session Information

```
sessionInfo()
#> R version 4.4.2 (2024-10-31)
#> Platform: aarch64-apple-darwin20
#> Running under: macOS Sonoma 14.5
#> 
#> Matrix products: default
#> BLAS:   /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRblas.0.dylib 
#> LAPACK: /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRlapack.dylib;  LAPACK v
#> 
#> locale:
#> [1] zh_CN.UTF-8/zh_CN.UTF-8/zh_CN.UTF-8/C/zh_CN.UTF-8/zh_CN.UTF-8
#> 
#> time zone: Asia/Shanghai
#> tzcode source: internal
#> 
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods   base     
#> 
#> loaded via a namespace (and not attached):
#>  [1] compiler_4.4.2    bookdown_0.42     fastmap_1.2.0     cli_3.6.3        
#>  [5] tools_4.4.2       htmltools_0.5.8.1 yaml_2.3.10       rmarkdown_2.29   
#>  [9] knitr_1.49        xfun_0.50         digest_0.6.37     rlang_1.1.4      
#> [13] evaluate_1.0.3
```