Package 'PANPRSnext'

July 22, 2025

Type Package

Title Building PRS Models Based on Summary Statistics of GWAs

Version 1.2.1

Description Shrinkage estimator for polygenic risk prediction (PRS) models based on summary statistics of genome-wide association (GWA) studies. Based upon the methods and original 'PAN-PRS' package as found in: Chen, Chatterjee, Landi, and Shi (2020) <doi:10.1080/01621459.2020.1764849>.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Depends gtools, R (>= 3.1.0)

LinkingTo Rcpp (>= 1.0.14), RcppArmadillo (>= 14.4.3-1)

Imports Rcpp (>= 1.0.14)

NeedsCompilation yes

Author Katherine Luo [aut, cre],
Osvaldo Espin-Garcia [aut],
Ting-Huei Chen [aut]

Maintainer Katherine Luo <`hluo224@uwo.ca`>

Repository CRAN

Date/Publication 2025-07-22 10:20:22 UTC

Contents

`funcIndex` *Inputs for the functional annotations of SNPs.*

Description

A 3614 x 3 matrix with (0,1) entry with 3614 SNPs and 3 functional annotations. For the element at i-th row, j-th column, the entry 0 means SNP i without j-th functional annotation; entry 1 means otherwise. follows:

- f1: The binary index for functional annotation 1.
- f2: The binary index for functional annotation 2.
- f3: The binary index for functional annotation 3.

Usage

```
data(summaryZ)
```

Format

A matrix with 3614 rows for the 3614 SNPs and 3 columns for functional annotations.

`gsfPEN_cpp` *Main CPP function*

Description

Main CPP function

Usage

```
gsfPEN_cpp(
  summary_betas,
  ld_J,
  index_matrix,
  index_J,
  ld_vec,
  SD_vec,
  tuning_matrix,
```

```
  lambda0_vec,
  z_matrix,
  lambda_vec_func,
  func_lambda,
  Ifunc_SNP,
  dims,
  params
)
```

Arguments

`summary_betas`	matrix of summary statistics
`ld_J`	vector of indices of SNPs in LD with the current SNP
`index_matrix`	matrix of indices of SNPs in LD with the current SNP
`index_J`	vector of indices of SNPs in LD with the current SNP
`ld_vec`	vector of LD values
`SD_vec`	matrix of SD values
`tuning_matrix`	matrix of tuning parameters
`lambda0_vec`	vector of lambda0 values
`z_matrix`	matrix of z values
`lambda_vec_func`	vector of lambda values
`func_lambda`	matrix of lambda values
`Ifunc_SNP`	vector of indices of SNPs in LD with the current SNP
`dims`	vector of dimensions
`params`	vector of parameters

`gsfPEN_R` *Run the gsfPEN algorithm for multiple traits, with functional annotations.*

Description

Run the gsfPEN algorithm for multiple traits, with functional annotations.

Usage

```
gsfPEN_R(
  summary_z,
  n_vec,
  plinkLD,
  func_index,
  n_iter = 1000,
```

```
  upper_val = NULL,
  breaking = 1,
  z_scale = 1,
  tuning_matrix = NULL,
  p_threshold = NULL,
  p_threshold_params = c(0.5, 10^-4, 4),
  tau_factor = c(1/25, 1, 3),
  sub_tuning = 4,
  lim_lambda = c(0.5, 0.9),
  len_lambda = 4,
  lambda_vec = NULL,
  lambda_vec_limit_len = c(1.5, 3),
  df_max = NULL,
  sparse_beta = FALSE,
  debug_output = FALSE,
  verbose = FALSE
)
```

Arguments

`summary_z`	A matrix of summary statistics for each SNP and trait.
`n_vec`	A vector of sample sizes for each of the Q traits corresponding to the Q columns of summary_z.
`plinkLD`	A matrix of LD values for each pair of SNPs.
`func_index`	A matrix of functional annotations for each SNP and trait. For the element at i-th row, j-th column, the entry 0 means SNP i without j-th functional annotation; entry 1 means otherwise.
`n_iter`	The number of iterations to run the algorithm.
`upper_val`	The upper bound for the tuning parameter.
`breaking`	The number of iterations to run before checking for convergence.
`z_scale`	The scaling factor for the summary statistics.
`tuning_matrix`	A matrix of tuning parameters.
`p_threshold`	A vector of p-values to use for the tuning parameters.
`p_threshold_params`	A vector of parameters to use for the p-value tuning parameters.
`tau_factor`	A vector of factors to multiply the median value by to get the tuning parameters.
`sub_tuning`	The number of tuning parameters to use for the second iteration.
`lim_lambda`	The range of tuning parameters to use for the first iteration.
`len_lambda`	The number of tuning parameters to use for the second iteration.
`lambda_vec`	A vector of tuning parameters to use for the first iteration.
`lambda_vec_limit_len`	The number of tuning parameters to use for the first iteration.
`df_max`	The maximum degrees of freedom for the model.
`sparse_beta`	Whether to use the sparse version of the algorithm.
`debug_output`	Whether to output the tuning combinations that did not converge.
`verbose`	Whether to output information through the evaluation of the algorithm.

Value

A named list containing the following elements: beta_matrix: A matrix of the estimated coefficients for each SNP and trait. num_iter_vec: A vector of the number of iterations for each tuning combination. all_tuning_matrix: A matrix of the tuning parameters used for each tuning combination.

Examples

```
# Load the library and data
library(PANPRSnext)
data("summaryZ")
data("Nvec")
data("plinkLD")
data("funcIndex")

# Take random subset of the data
subset <- sample(nrow(summaryZ), 100)
subset_summary_z <- summaryZ[subset, ]
subset_func_index <- funcIndex[subset, ]

# Run gsfPEN
output <- gsfPEN_R(
  summary_z = subset_summary_z,
  n_vec = Nvec,
  plinkLD = plinkLD,
  func_index = subset_func_index
)
```

`gsfPEN_sparse_cpp` *Main CPP function*

Description

Main CPP function

Usage

```
gsfPEN_sparse_cpp(
  summary_betas,
  ld_J,
  index_matrix,
  index_J,
  ld_vec,
  SD_vec,
  tuning_matrix,
  lambda0_vec,
  z_matrix,
  lambda_vec_func,
  func_lambda,
```

```
  Ifunc_SNP,
  dims,
  params
)
```

Arguments

`summary_betas`	matrix of summary statistics
`ld_J`	vector of indices of SNPs in LD with the current SNP
`index_matrix`	matrix of indices of SNPs in LD with the current SNP
`index_J`	vector of indices of SNPs in LD with the current SNP
`ld_vec`	vector of LD values
`SD_vec`	matrix of SD values
`tuning_matrix`	matrix of tuning parameters
`lambda0_vec`	vector of lambda0 values
`z_matrix`	matrix of z values
`lambda_vec_func`	vector of lambda values
`func_lambda`	matrix of lambda values
`Ifunc_SNP`	vector of indices of SNPs in LD with the current SNP
`dims`	vector of dimensions
`params`	vector of parameters

`gsPEN_cpp` *Main CPP function*

Description

Main CPP function

Usage

```
gsPEN_cpp(
  summary_betas,
  ld_J,
  index_matrix,
  index_J,
  ld_vec,
  SD_vec,
  tuning_matrix,
  dims,
  params
)
```

Arguments

`summary_betas`	matrix of summary statistics
`ld_J`	vector of indices of SNPs in LD with the current SNP
`index_matrix`	matrix of indices of SNPs in LD with the current SNP
`index_J`	vector of indices of SNPs in LD with the current SNP
`ld_vec`	vector of LD values
`SD_vec`	matrix of SD values
`tuning_matrix`	matrix of tuning parameters
`dims`	vector of dimensions
`params`	vector of parameters

`gsPEN_R` *Run the gsPEN algorithm for multiple traits, without functional annotations.*

Description

Run the gsPEN algorithm for multiple traits, without functional annotations.

Usage

```
gsPEN_R(
  summary_z,
  n_vec,
  plinkLD,
  n_iter = 100,
  upper_val = NULL,
  breaking = 1,
  z_scale = 1,
  tuning_matrix = NULL,
  tau_factor = c(1/25, 1, 10),
  len_lim_lambda = 10,
  sub_tuning = 50,
  lim_lambda = c(0.5, 0.9),
  len_lambda = 200,
  df_max = NULL,
  sparse_beta = FALSE,
  debug_output = FALSE,
  verbose = FALSE
)
```

Arguments

`summary_z`	A matrix of summary statistics for each SNP and trait.
`n_vec`	A vector of sample sizes for each of the Q traits corresponding to the Q columns of summary_z.
`plinkLD`	A matrix of LD values for each pair of SNPs.
`n_iter`	The number of iterations to run the algorithm.
`upper_val`	The upper bound for the tuning parameter.
`breaking`	The number of iterations to run before checking for convergence.
`z_scale`	The scaling factor for the summary statistics.
`tuning_matrix`	A matrix of tuning parameters.
`tau_factor`	A vector of factors to multiply the median value by to get the tuning parameters.
`len_lim_lambda`	The number of tuning parameters to use for the first iteration.
`sub_tuning`	The number of tuning parameters to use for the second iteration.
`lim_lambda`	The range of tuning parameters to use for the first iteration.
`len_lambda`	The number of tuning parameters to use for the second iteration.
`df_max`	The maximum degrees of freedom for the model.
`sparse_beta`	Whether to use the sparse version of the algorithm.
`debug_output`	Whether to output the tuning combinations that did not converge.
`verbose`	Whether to output information through the evaluation of the algorithm.

Value

A named list containing the following elements: beta_matrix: A matrix of the estimated coefficients for each SNP and trait. num_iter_vec: A vector of the number of iterations for each tuning combination. all_tuning_matrix: A matrix of the tuning parameters used for each tuning combination.

Examples

```
# Load the library and data
library(PANPRSnext)
data("summaryZ")
data("Nvec")
data("plinkLD")

# Take random subset of the data
subset <- sample(nrow(summaryZ), 100)
subset_summary_z <- summaryZ[subset, ]

# Run gsPEN
output <- gsPEN_R(
  summary_z = subset_summary_z,
  n_vec = Nvec,
  plinkLD = plinkLD
)
```

`gsPEN_sparse_cpp` *Main CPP function*

Description

Main CPP function

Usage

```
gsPEN_sparse_cpp(
  summary_betas,
  ld_J,
  index_matrix,
  index_J,
  ld_vec,
  SD_vec,
  tuning_matrix,
  dims,
  params
)
```

Arguments

`summary_betas`	matrix of summary statistics
`ld_J`	vector of indices of SNPs in LD with the current SNP
`index_matrix`	matrix of indices of SNPs in LD with the current SNP
`index_J`	vector of indices of SNPs in LD with the current SNP
`ld_vec`	vector of LD values
`SD_vec`	matrix of SD values
`tuning_matrix`	matrix of tuning parameters
`dims`	vector of dimensions
`params`	vector of parameters

`Nvec` *A vector of sample sizes for the q traits of the* `summaryZ`*.*

Description

A vector of q sample sizes for the q set of Z statistics corresponding to the q columns of `summaryZ`.

Usage

```
data(Nvec)
```

Format

A vector with q elements, where q is the number of columns of `summaryZ`.

`plinkLD` *The LD info from output of the software (plink)*

Description

The LD information is crucial for the analysis by SummaryLasso. The reference alleles used to obtained for the Z statsitics or the regression coefficients have to be the sames as those used for the LD calculation. This file can be obtained directly from the output of the LD calculation by the software (plink); for example the output can be like plink.ld. On the other hand, the user can calcuate the LD based on their prefered tools. The variables are as follows:

- CHR_A: The chromosome of SNP_A
- BP_A: The positions of SNP_A
- SNP_A: The names of SNP_A
- CHR_B: The chromosome of SNP_B
- BP_B: The positions of SNP_B
- SNP_B: The names of SNP_B
- R: The correlation between SNP_A and SNP_B

Usage

```
data(plinkLD)
```

Format

A data frame with 205959 rows and 7 columns

References

- Purcell S, et al. (2007) PLINK: a toolset for whole-genome association and population-based linkage analysis. *American Journal of Human Genetics*, **81**.

`summaryZ`	*The Z statistics from the univariate analysis of the association between 3614 SNPs and three traits respectively.*

Description

These Z statsitics are obtained from simulated datasets. The variables are as follows:

- Z1: The Z statistics from trait 1; the primary trait.
- Z2: The Z statistics from trait 2; the secondary trait.
- Z2: The Z statistics from trait 3; the secondary trait.

Usage

```
data(summaryZ)
```

Format

A matrix with 3614 rows for the 3614 SNPs and 3 columns for 3 traits.

`test_gsfPEN`	*Run gsfPEN on a small sample of the provided data set (Only 100 samples)*

Description

Run gsfPEN on a small sample of the provided data set (Only 100 samples)

Usage

```
test_gsfPEN(...)
```

Arguments

`...`	Additional arguments to pass to gsfPEN_R

Value

The output of gsfPEN_R

`test_gsPEN` *Run gsPEN on a small sample of the provided data set (Only 100 samples)*

Description

Run gsPEN on a small sample of the provided data set (Only 100 samples)

Usage

```
test_gsPEN(...)
```

Arguments

`...`	Additional arguments to pass to gsPEN_R

Value

The output of gsPEN_R

Index