Package 'EGG'

December 15, 2024

Type Package

Title Estimating Genetic Gaussian Networks from GWAS Summary Data

Version 0.1.0

Author Yihe Yang

Maintainer The package maintainer <yxy1234@case.edu>

Description The EGG (Estimating Genetic Gaussian Networks) package is designed for estimating multiple variable Genetic Gaussian Networks using GWAS summary data. It utilizes advanced statistical methods to infer the genetic relationships and dependencies between different traits or variables, providing a comprehensive view of the genetic architecture. This package is particularly useful for researchers and geneticists looking to understand the complex interplay between multiple genetic factors.

License MIT

Encoding UTF-8

LazyData true

Imports data.table (>= 1.12.0),
glasso (>= 1.11),
CppMatrix,
igraph,
ggplot2,
ggraph,
RColorBrewer

RoxygenNote 7.3.1

Contents

`dtrace.mcp.pearson.sampling` *Estimate genetic network using MCP-penalized D-trace loss function and Pearson's rho correlation matrix*

Description

The `entropy.mcp.spearman.sampling` function estimates the genetic network by applying a penalized D-trace loss function using MCP (Minimax Concave Penalty) and Pearson's r correlation matrix. It's designed to work with genetic data, specifically effect size estimates or Z-scores for various exposures across multiple variants.

Usage

```
dtrace.mcp.pearson.sampling(
  BETA,
  Rnoise,
  lamvec = c(1:20)/100,
  max.eps = 0.01,
  max.iter = 25,
  rho = 0.05,
  mineig = 0.01,
  subtime = 100,
  subfrac = 0.5,
  subthres = 0.95,
  alpha = 0,
  bic.factor = 0.5,
  reliability.thres = 0.8,
  PenaMatrix = "none"
)
```

Arguments

`BETA`	A matrix of dimensions mxp, where m is the number of variants and p is the number of exposures. It represents effect size estimates or Z-scores.
`Rnoise`	The covariance matrix of estimation errors in BETA. If BETA contains Z-scores, Rnoise should be a correlation matrix.
`lamvec`	A vector of tuning parameters for MCP. Default is c(1:20)/100.
`max.eps`	The threshold for stopping the algorithm. Default is 0.01.
`max.iter`	The maximum number of iterations for the algorithm. Since ADMM converges quickly but not with high precision, this should not be too large. Default is 25.
`rho`	The penalty parameter for the ADMM algorithm. Default is 0.05.
`mineig`	The minimum matrixEigenvalue for the precision matrix estimate. Default is 0.01.
`subtime`	The number of times to resample. Default is 100.
`subfrac`	The fraction of the data to resample each time. Default is 0.5.
`subthres`	The threshold for stability selection. Default is 0.95.
`alpha`	The additional parameter for MCP + alpha * Ridge. Default is 0.

`bic.factor` An additional penalty on cross-validation error. Default is 0.5.

`reliability.thres`
The threshold for rescaling Rnoise to ensure that (var(BETAj)-Rnoisejj)/var(BETAj) is greater than this value. Default is 0.8.

`PenaMatrix` A penalty weight matrix to rescale the tuning parameter in MCP. Default is a matrix of ones.

Details

The function performs subsampling and uses the MCP method to estimate a sparse precision matrix (inverse covariance matrix) representing the genetic network. It uses Spearman's rank correlation to handle non-linear relationships and applies stability selection to enhance the reliability of the selected network.

Value

A list containing: - Theta: The estimated precision matrix. - Pvalue: P-values for the entries in Thetabased on subsampling. - K: Stability selection matrix indicating the proportion of subsamples where each entry in Theta was non-zero. - cv.error: The cross-validation error for each lambda in lamvec. - R: The Spearman rank correlation matrix of BETA. - ThetaSE: The standard error of Theta based on subsampling.

Examples

```
# Assuming BETA and Rnoise are already defined:
result <- dtrace.mcp.pearson.sampling(BETA, Rnoise)
```

---

dtrace.mcp.spearman.sampling

Estimate genetic network using MCP-penalized D-trace loss function and Spearman's rho correlation matrix

---

Description

The `entropy.mcp.spearman.sampling` function estimates the genetic network by applying a penalized D-trace loss function using MCP (Minimax Concave Penalty) and Spearman's rho correlation matrix. It's designed to work with genetic data, specifically effect size estimates or Z-scores for various exposures across multiple variants.

Usage

```
dtrace.mcp.spearman.sampling(
  BETA,
  Rnoise,
  lamvec = c(1:20)/100,
  max.eps = 0.01,
  max.iter = 25,
  rho = 0.05,
  mineig = 0.01,
  subtime = 100,
```

```
  subfrac = 0.5,
  subthres = 0.95,
  alpha = 0,
  bic.factor = 0.5,
  reliability.thres = 0.8,
  PenaMatrix = "none"
)
```

Arguments

BETA	A matrix of dimensions mxp, where m is the number of variants and p is the number of exposures. It represents effect size estimates or Z-scores.
Rnoise	The covariance matrix of estimation errors in BETA. If BETA contains Z-scores, Rnoise should be a correlation matrix.
lamvec	A vector of tuning parameters for MCP. Default is c(1:20)/100.
max.eps	The threshold for stopping the algorithm. Default is 0.01.
max.iter	The maximum number of iterations for the algorithm. Since ADMM converges quickly but not with high precision, this should not be too large. Default is 25.
rho	The penalty parameter for the ADMM algorithm. Default is 0.05.
mineig	The minimum matrixEigenvalue for the precision matrix estimate. Default is 0.01.
subtime	The number of times to resample. Default is 100.
subfrac	The fraction of the data to resample each time. Default is 0.5.
subthres	The threshold for stability selection. Default is 0.95.
alpha	The additional parameter for MCP + alpha * Ridge. Default is 0.
bic.factor	An additional penalty on cross-validation error. Default is 0.5.
reliability.thres	The threshold for rescaling Rnoise to ensure that (var(BETAj)-Rnoisejj)/var(BETAj) is greater than this value. Default is 0.8.
PenaMatrix	A penalty weight matrix to rescale the tuning parameter in MCP. Default is a matrix of ones.

Details

The function performs subsampling and uses the MCP method to estimate a sparse precision matrix (inverse covariance matrix) representing the genetic network. It uses Spearman's rank correlation to handle non-linear relationships and applies stability selection to enhance the reliability of the selected network.

Value

A list containing: - Theta: The estimated precision matrix. - Pvalue: P-values for the entries in Thetabased on subsampling. - K: Stability selection matrix indicating the proportion of subsamples where each entry in Theta was non-zero. - cv.error: The cross-validation error for each lambda in lamvec. - R: The Spearman rank correlation matrix of BETA. - ThetaSE: The standard error of Theta based on subsampling.

Examples

```
# Assuming BETA and Rnoise are already defined:
result <- entropy.mcp.spearman.sampling(BETA, Rnoise)
```

`entropy.mcp.pearson.sampling`

Estimate genetic network using MCP-penalized entropy loss function and Pearson's r correlaiton matrix

---

Description

The `entropy.mcp.spearman.sampling` function estimates the genetic network by applying a penalized entropy loss function using MCP (Minimax Concave Penalty) and Pearson's r correlation matrix. It's designed to work with genetic data, specifically effect size estimates or Z-scores for various exposures across multiple variants.

Usage

```
entropy.mcp.pearson.sampling(
  BETA,
  Rnoise,
  lamvec = c(1:20)/100,
  max.eps = 0.01,
  max.iter = 25,
  rho = 0.05,
  mineig = 0.01,
  subtime = 100,
  subfrac = 0.5,
  subthres = 0.95,
  alpha = 0,
  bic.factor = 0.5,
  reliability.thres = 0.8,
  PenaMatrix = "none"
)
```

Arguments

`BETA`	A matrix of dimensions mxp, where m is the number of variants and p is the number of exposures. It represents effect size estimates or Z-scores.
`Rnoise`	The covariance matrix of estimation errors in BETA. If BETA contains Z-scores, Rnoise should be a correlation matrix.
`lamvec`	A vector of tuning parameters for MCP. Default is c(1:20)/100.
`max.eps`	The threshold for stopping the algorithm. Default is 0.01.
`max.iter`	The maximum number of iterations for the algorithm. Since ADMM converges quickly but not with high precision, this should not be too large. Default is 25.
`rho`	The penalty parameter for the ADMM algorithm. Default is 0.05.
`mineig`	The minimum matrixEigenvalue for the precision matrix estimate. Default is 0.01.

`subtime`	The number of times to resample. Default is 100.
`subfrac`	The fraction of the data to resample each time. Default is 0.5.
`subthres`	The threshold for stability selection. Default is 0.95.
`alpha`	The additional parameter for MCP + alpha * Ridge. Default is 0.
`bic.factor`	An additional penalty on the cross-validation error. Default is 0.5.
`reliability.thres`	The threshold for rescaling Rnoise to ensure that (var(BETAj)-Rnoisejj)/var(BETAj) is greater than this value. Default is 0.8.
`PenaMatrix`	A penalty weight matrix to rescale the tuning parameter in MCP. Default is a matrix of ones.

Details

The function performs subsampling and uses the MCP method to estimate a sparse precision matrix (inverse covariance matrix) representing the genetic network. It uses Spearman's rank correlation to handle non-linear relationships and applies stability selection to enhance the reliability of the selected network.

Value

A list containing: - Theta: The estimated precision matrix. - Pvalue: P-values for the entries in Theta. - K: Stability selection matrix indicating the proportion of subsamples where each entry in Theta was non-zero. - cv.error: The cross-validation error for each lambda in lamvec. - R: The Spearman rank correlation matrix of BETA. - ThetaSE: The standard error of Theta based on subsampling.

Examples

```
# Assuming BETA and Rnoise are already defined:
result <- entropy.mcp.pearson.sampling(BETA, Rnoise)
```

`entropy.mcp.spearman.sampling` *Estimate genetic network using MCP-penalized entropy loss function and Spearman's rho correlation matrix*

Description

The `entropy.mcp.spearman.sampling` function estimates the genetic network by applying a penalized entropy loss function using MCP (Minimax Concave Penalty) and Spearman's rho correlation matrix. It's designed to work with genetic data, specifically effect size estimates or Z-scores for various exposures across multiple variants.

Usage

```
entropy.mcp.spearman.sampling(
  BETA,
  Rnoise,
  lamvec = c(1:20)/100,
  max.eps = 0.01,
  max.iter = 25,
  rho = 0.05,
  mineig = 0.01,
  subtime = 100,
  subfrac = 0.5,
  subthres = 0.95,
  alpha = 0,
  bic.factor = 0.5,
  reliability.thres = 0.8,
  PenaMatrix = "none"
)
```

Arguments

BETA	A matrix of dimensions mxp, where m is the number of variants and p is the number of exposures. It represents effect size estimates or Z-scores.
Rnoise	The covariance matrix of estimation errors in BETA. If BETA contains Z-scores, Rnoise should be a correlation matrix.
lamvec	A vector of tuning parameters for MCP. Default is c(1:20)/100.
max.eps	The threshold for stopping the algorithm. Default is 0.01.
max.iter	The maximum number of iterations for the algorithm. Since ADMM converges quickly but not with high precision, this should not be too large. Default is 25.
rho	The penalty parameter for the ADMM algorithm. Default is 0.05.
mineig	The minimum matrixEigenvalue for the precision matrix estimate. Default is 0.01.
subtime	The number of times to resample. Default is 100.
subfrac	The fraction of the data to resample each time. Default is 0.5.
subthres	The threshold for stability selection. Default is 0.95.
alpha	The additional parameter for MCP + alpha * Ridge. Default is 0.
bic.factor	An additional penalty on the cross-validation error. Default is 0.5.
reliability.thres	The threshold for rescaling Rnoise to ensure that (var(BETAj)-Rnoisejj)/var(BETAj) is greater than this value. Default is 0.8.
PenaMatrix	A penalty weight matrix to rescale the tuning parameter in MCP. Default is a matrix of ones.

Details

The function performs subsampling and uses the MCP method to estimate a sparse precision matrix (inverse covariance matrix) representing the genetic network. It uses Spearman's rank correlation to handle non-linear relationships and applies stability selection to enhance the reliability of the selected network.

Value

A list containing: - Theta: The estimated precision matrix. - Pvalue: P-values for the entries in Theta. - K: Stability selection matrix indicating the proportion of subsamples where each entry in Theta was non-zero. - cv.error: The cross-validation error for each lambda in lamvec. - R: The Spearman rank correlation matrix of BETA. - ThetaSE: The standard error of Theta based on subsampling.

Examples

```
# Assuming BETA and Rnoise are already defined:
result <- entropy.mcp.spearman.sampling(BETA, Rnoise)
```

---

hapmap3 *HapMap3 and UKBB Genotype SNP Data*

Description

A data frame containing combined SNP data from 1000 Genomes Project Phase 3 and UK Biobank (UKBB) genotypes. It includes a total of 1,664,852 SNPs.

Usage

```
data(hapmap3)
```

Format

A data frame with 1,664,852 rows and 3 variables:

SNP SNP identifier.

A1 Effect allele.

A2 Reference allele.

Source

1000 Genomes Project Phase 3 and UK Biobank genotype data.

---

merge_gwas *Filter and Align GWAS Data to a Reference Panel*

Description

The `merge_gwas` function processes a list of GWAS summary statistics data frames, harmonizes alleles according to a reference panel, removes duplicates, and aligns data to common SNPs. It's used to prepare data for further analysis such as LDSC.

Usage

```
merge_gwas(gwas_data_list, ref_panel)
```

Arguments

gwas_data_list	A list of data.frames where each data.frame contains GWAS summary statistics for a trait. Each data.frame should include columns for SNP identifiers, Z-scores of effect size estimates, sample sizes (N), effect allele (A1), and reference allele (A2).
ref_panel	A data.frame containing the reference panel data. It must include columns for SNP, A1, and A2.

Details

The function performs several key steps: adjusting alleles according to a reference panel, removing duplicate SNPs, and aligning all GWAS data frames to a set of common SNPs. This is often a necessary preprocessing step before performing genetic correlation and heritability analyses.

Value

A list of data.frames, each corresponding to an input GWAS summary statistics data frame, but filtered, harmonized, and aligned to the common SNPs found across all data frames.

Examples

```
# Assuming GWAS_List and ref_panel are already defined:
GWAS_List <- merge_gwas(GWAS_List, ref_panel)
```

Index