Package: metaCCA (via r-universe)

November 25, 2026

Type Package

Title Summary Statistics-Based Multivariate Meta-Analysis of Genome-Wide Association Studies Using Canonical Correlation Analysis

Version 1.42.0

Date 2016-01-26

Author Anna Cichonska <anna.cichonska@gmail.com>

Maintainer Anna Cichonska <anna.cichonska@gmail.com>

Suggests knitr

VignetteBuilder knitr

Description metaCCA performs multivariate analysis of a single or multiple GWAS based on univariate regression coefficients. It allows multivariate representation of both phenotype and genotype. metaCCA extends the statistical technique of canonical correlation analysis to the setting where original individual-level records are not available, and employs a covariance shrinkage algorithm to achieve robustness.

License MIT + file LICENSE

URL https://doi.org/10.1093/bioinformatics/btw052

LazyData TRUE

biocViews GenomeWideAssociation, SNP, Genetics, Regression, StatisticalMethod, Software

NeedsCompilation no

Repository Bioconductor 3.24

Date/Publication 2026-11-23 22:36:42 UTC

RemoteUrl https://github.com/bioc/metaCCA

RemoteRef RELEASE_3_24

RemoteSha 1ee8f140aefaf460229fc8cac963fa4fe8f14b7a

Contents

estimateSyy 2
metaCcaGp 3
metaCcaPlusGp 7
N1 10
N2 11
S_XX_study1 11
S_XX_study2 12
S_XY_full_study1 12
S_XY_full_study2 13
S_XY_study1 13
S_XY_study2 14

Index **15**

`estimateSyy` *Function to estimate correlations between phenotypic variables from summary statistics*

Description

This function computes phenotypic correlation matrix `S_YY` based on univariate summary statistics `S_XY`.

Usage

```
estimateSyy( S_XY )
```

Arguments

`S_XY`	Univariate summary statistics. Data frame with row names corresponding to SNP IDs (e.g., position or rs_id) and the following columns: - `allele_0` - string composed of "A", "C", "G" or "T", - `allele_1` - string composed of "A", "C", "G" or "T", - then, two columns for each trait (phenotypic variable) to be included in the analysis; in turn: 1) `traitID_b` with linear regression coefficients, 2) `traitID_se` with corresponding standard errors ("traitID" in the column name must be an ID of a trait specified by a user; do not use underscores "_" in trait IDs outside "_b"/"_se" in order for the IDs to be processed correctly).

Value

`S_YY`	Matrix containing correlations between traits given as input. Row and column names correspond to trait IDs.

Note

In practice, summary statistics of at least one chromosome should be used in order to ensure good quality of the estimate of phenotypic correlation structure.

Author(s)

Anna Cichonska

References

Cichonska et al. (2016) metaCCA: Summary statistics-based multivariate meta-analysis of genome-wide association studies using canonical correlation analysis. Bioinformatics, 32(13):1981-1989.

Examples

```
# Estimating correlations between 10 traits given their
# univariate summary statistics across 1000 SNPs
S_YY = estimateSyy( S_XY = S_XY_full_study1 )

# Viewing the resulting phenotypic correlation matrix
print( S_YY, digit = 3 )
```

`metaCcaGp` *Function to perform genotype-phenotype association analysis according to metaCCA algorithm.*

Description

This function performs genotype-phenotype association analysis according to metaCCA algorithm (univariate summary statistics-based analysis of a single or multiple genome-wide association studies (GWAS) that allows multivariate representation of both genotype and phenotype).

The function accepts a varying number of arguments, depending on the type of the analysis. By default, single-SNP–multi-trait association analysis is performed, where each given SNP is tested against all given phenotypic variables. Other options are to perform single-SNP–multi-trait analysis of one selected SNP, as well as multi-SNP–multi-trait analysis.

Usage

```
metaCcaGp( nr_studies, S_XY, std_info, S_YY, N, analysis_type, SNP_id, S_XX )
```

Arguments

`nr_studies`	Number of studies to be analysed.

`S_XY` Univariate summary statistics of the variables to be analysed.

A list of data frames (one for each study) with row names corresponding to SNP IDs (e.g., position or rs_id) and the following columns:

- `allele_0` - string composed of "A", "C", "G" or "T",

- `allele_1` - string composed of "A", "C", "G" or "T",

- then, two columns for each trait (phenotypic variable) to be included in the analysis; in turn:

1) `traitID_b` with linear regression coefficients,

2) `traitID_se` with corresponding standard errors

("traitID" in the column name must be an ID of a trait specified by a user; do not use underscores "_" in trait IDs outside "_b"/"_se" in order for the IDs to be processed correctly).

`std_info` A vector with numerical values `0/1` (one value for each study) indicating if the univariate analysis has been performed on standardised (`1`) or non-standardised (`0`) data;

(most likely the data were not standardised - the genotypes were not standardised before univariate regression coefficients and standard errors were computed - option `0` should be used).

`S_YY` A list of phenotypic correlation matrices (one for each study) estimated using `estimateSyy` function.

`N` A vector with numbers of individuals in each study.

Arguments below are OPTIONAL and depend on the type of the analysis.

`analysis_type` Indicator of the analysis type.

1) Single-SNP–multi-trait analysis of one selected SNP: `1`.

2) Multi-SNP–multi-trait analysis: `2`.

`SNP_id` 1) Single-SNP–multi-trait analysis of one selected SNP:

An ID of the SNP of interest.

2) Multi-SNP–multi-trait analysis:

A vector with IDs of SNPs to be analysed jointly.

`S_XX` A list of data frames (one for each study) containing correlations between SNPs. Row names (and, optionally, column names) must correspond to SNP IDs. This argument needs to be given only in case of multi-SNP–multi-trait analysis.

Value

`result` Data frame with row names corresponding to SNP IDs.

Columns contain:

1) `r_1` - leading canonical correlation value,

2) `-log10(p-val)` - p-value in the -log10 scale,

3) `trait_weights` - trait-wise canonical weights,

4) `snp_weights` - SNP-wise canonical weights (only for multi-SNP-–multi-trait analysis).

Author(s)

Anna Cichonska

References

Cichonska et al. (2016) metaCCA: Summary statistics-based multivariate meta-analysis of genome-wide association studies using canonical correlation analysis. Bioinformatics, 32(13):1981-1989.

Examples

```
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
#           Analysis of one study according to metaCCA algorithm.           #
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #


# Default single-SNP--multi-trait analysis.
# Here, we will test each of 10 SNPs for an association with a set of 10 traits.
result1 = metaCcaGp( nr_studies = 1,
                     S_XY = list( S_XY_study1 ),
                     std_info = 0,
                     S_YY = list( estimateSyy(S_XY_full_study1 ) ),
                     N = N1 )

# Viewing association results
print( result1, digits = 3 )


# Single-SNP--multi-trait analysis of one selected SNP.
# Here, we will test one of 10 SNPs for an association with a set of 10 traits.
result2 = metaCcaGp( nr_studies = 1,
                     S_XY = list( S_XY_study1 ),
                     std_info = 0,
                     S_YY = list( estimateSyy(S_XY_full_study1) ),
                     N = N1,
                     analysis_type = 1,
                     SNP_id = 'rs80' )

# Viewing association results
print( result2, digits = 3 )


# Multi-SNP--multi-trait analysis.
# Here, we will test a set of 5 SNPs for an association with a set of 10 traits.
result3 = metaCcaGp( nr_studies = 1,
                     S_XY = list( S_XY_study1 ),
                     std_info = 0,
                     S_YY = list( estimateSyy(S_XY_full_study1) ),
                     N = N1,
                     analysis_type = 2,
                     SNP_id = c( 'rs10', 'rs80', 'rs140', 'rs170', 'rs172' ),
                     S_XX = list( S_XX_study1 ) )
```

```
# Viewing association results
print( result3, digits = 3 )



# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
#       Meta-analysis of two studies according to metaCCA algorithm.        #
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #


# Default single-SNP--multi-trait analysis.
# Here, we will test each of 10 SNPs for an association with a set of 10 traits.
meta_result1 = metaCcaGp( nr_studies = 2,
                          S_XY = list( S_XY_study1, S_XY_study2 ),
                          std_info = c( 0, 0 ),
                          S_YY = list( estimateSyy(S_XY_full_study1),
                                       estimateSyy(S_XY_full_study2) ),
                          N = c( N1, N2 ) )

# Viewing association results
print( meta_result1, digits = 3 )


# Single-SNP--multi-trait analysis of one selected SNP.
# Here, we will test one of 10 SNPs for an association with a set of 10 traits.
meta_result2 = metaCcaGp( nr_studies = 2,
                          S_XY = list( S_XY_study1, S_XY_study2 ),
                          std_info = c( 0, 0 ),
                          S_YY = list( estimateSyy(S_XY_full_study1),
                                       estimateSyy(S_XY_full_study2) ),
                          N = c( N1, N2 ),
                          analysis_type = 1,
                          SNP_id = 'rs80' )

# Viewing association results
print( meta_result2, digits = 3 )


# Multi-SNP--multi-trait analysis.
# Here, we will test a set of 5 SNPs for an association with a set of 10 traits.
meta_result3 = metaCcaGp( nr_studies = 2,
                          S_XY = list( S_XY_study1, S_XY_study2 ),
                          std_info = c( 0, 0 ),
                          S_YY = list( estimateSyy(S_XY_full_study1),
                                       estimateSyy(S_XY_full_study2) ),
                          N = c( N1, N2 ),
                          analysis_type = 2,
                          SNP_id = c( 'rs10', 'rs80', 'rs140', 'rs170', 'rs172' ),
                          S_XX = list( S_XX_study1, S_XX_study2 ) )

# Viewing association results
print( meta_result3, digits = 3 )
```

`metaCcaPlusGp` *Function to perform genotype-phenotype association analysis according to metaCCA+ algorithm.*

Description

This function performs genotype-phenotype association analysis according to metaCCA+ algorithm (the variant of metaCCA, where the full covariance matrix is shrunk beyond the level guaranteeing its positive semidefinite property).

`metaCcaPlusGp` requires exactly the same inputs as `metaCcaGp` function, and it has the same output format.

Usage

```
metaCcaPlusGp( nr_studies, S_XY, std_info, S_YY, N, analysis_type, SNP_id, S_XX )
```

Arguments

`nr_studies`	Number of studies to be analysed.
`S_XY`	Univariate summary statistics of the variables to be analysed. A list of data frames (one for each study) with row names corresponding to SNP IDs (e.g., position or rs_id) and the following columns: - `allele_0` - string composed of "A", "C", "G" or "T", - `allele_1` - string composed of "A", "C", "G" or "T", - then, two columns for each trait (phenotypic variable) to be included in the analysis; in turn: 1) `traitID_b` with linear regression coefficients, 2) `traitID_se` with corresponding standard errors ("traitID" in the column name must be an ID of a trait specified by a user; do not use underscores "_" in trait IDs outside "_b"/"_se" in order for the IDs to be processed correctly).
`std_info`	A vector with numerical values `0/1` (one value for each study) indicating if the univariate analysis has been performed on standardised (`1`) or non-standardised (`0`) data; (most likely the data were not standardised - the genotypes were not standardised before univariate regression coefficients and standard errors were computed - option `0` should be used).
`S_YY`	A list of phenotypic correlation matrices (one for each study) estimated using `estimateSyy` function.
`N`	A vector with numbers of individuals in each study.

Arguments below are OPTIONAL and depend on the type of the analysis.

analysis_type	Indicator of the analysis type. 1) Single-SNP–multi-trait analysis of one selected SNP: 1. 2) Multi-SNP–multi-trait analysis: 2.
SNP_id	1) Single-SNP–multi-trait analysis of one selected SNP: An ID of the SNP of interest. 2) Multi-SNP–multi-trait analysis: A vector with IDs of SNPs to be analysed jointly.
S_XX	A list of data frames (one for each study) containing correlations between SNPs. Row names (and, optionally, column names) must correspond to SNP IDs. This argument needs to be given only in case of multi-SNP–multi-trait analysis.

Value

result	Data frame with row names corresponding to SNP IDs. Columns contain: 1) r_1 - leading canonical correlation value, 2) -log10(p-val) - p-value in the -log10 scale, 3) trait_weights - trait-wise canonical weights, 4) snp_weights - SNP-wise canonical weights (only for multi-SNP-–multi-trait analysis).

Author(s)

Anna Cichonska

References

Cichonska et al. (2016) metaCCA: Summary statistics-based multivariate meta-analysis of genome-wide association studies using canonical correlation analysis. Bioinformatics, 32(13):1981-1989.

Examples

```
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
#          Analysis of one study according to metaCCA+ algorithm.           #
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #


# Default single-SNP--multi-trait analysis.
# Here, we will test each of 10 SNPs for an association with a set of 10 traits.
result1 = metaCcaPlusGp( nr_studies = 1,
                         S_XY = list( S_XY_study1 ),
                         std_info = 0,
                         S_YY = list( estimateSyy(S_XY_full_study1 ) ),
                         N = N1 )

# Viewing association results
print( result1, digits = 3 )
```

```
# Single-SNP--multi-trait analysis of one selected SNP.
# Here, we will test one of 10 SNPs for an association with a set of 10 traits.
result2 = metaCcaPlusGp( nr_studies = 1,
                         S_XY = list( S_XY_study1 ),
                         std_info = 0,
                         S_YY = list( estimateSyy(S_XY_full_study1) ),
                         N = N1,
                         analysis_type = 1,
                         SNP_id = 'rs80' )

# Viewing association results
print( result2, digits = 3 )


# Multi-SNP--multi-trait analysis.
# Here, we will test a set of 5 SNPs for an association with a set of 10 traits.
result3 = metaCcaPlusGp( nr_studies = 1,
                         S_XY = list( S_XY_study1 ),
                         std_info = 0,
                         S_YY = list( estimateSyy(S_XY_full_study1) ),
                         N = N1,
                         analysis_type = 2,
                         SNP_id = c( 'rs10', 'rs80', 'rs140', 'rs170', 'rs172' ),
                         S_XX = list( S_XX_study1 ) )

# Viewing association results
print( result3, digits = 3 )




# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
#      Meta-analysis of two studies according to metaCCA+ algorithm.        #
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #


# Default single-SNP--multi-trait analysis.
# Here, we will test each of 10 SNPs for an association with a set of 10 traits.
meta_result1 = metaCcaPlusGp( nr_studies = 2,
                              S_XY = list( S_XY_study1, S_XY_study2 ),
                              std_info = c( 0, 0 ),
                              S_YY = list( estimateSyy(S_XY_full_study1),
                                           estimateSyy(S_XY_full_study2) ),
                              N = c( N1, N2 ) )

# Viewing association results
print( meta_result1, digits = 3 )


# Single-SNP--multi-trait analysis of one selected SNP.
# Here, we will test one of 10 SNPs for an association with a set of 10 traits.
meta_result2 = metaCcaPlusGp( nr_studies = 2,
                              S_XY = list( S_XY_study1, S_XY_study2 ),
```

```
                              std_info = c( 0, 0 ),
                              S_YY = list( estimateSyy(S_XY_full_study1),
                                           estimateSyy(S_XY_full_study2) ),
                              N = c( N1, N2 ),
                              analysis_type = 1,
                              SNP_id = 'rs80' )

# Viewing association results
print( meta_result2, digits = 3 )


# Multi-SNP--multi-trait analysis.
# Here, we will test a set of 5 SNPs for an association with a set of 10 traits.
meta_result3 = metaCcaPlusGp( nr_studies = 2,
                              S_XY = list( S_XY_study1, S_XY_study2 ),
                              std_info = c( 0, 0 ),
                              S_YY = list( estimateSyy(S_XY_full_study1),
                                           estimateSyy(S_XY_full_study2) ),
                              N = c( N1, N2 ),
                              analysis_type = 2,
                              SNP_id = c( 'rs10', 'rs80', 'rs140', 'rs170', 'rs172' ),
                              S_XX = list( S_XX_study1, S_XX_study2 ) )

# Viewing association results
print( meta_result3, digits = 3 )
```

N1 *Number of individuals in study 1.*

Description

Number of individuals in study 1.

Format

Numeric value

Value

Test data

Source

Part of the simulated toy data set.

N2 *Number of individuals in study 2.*

Description

Number of individuals in study 2.

Format

Numeric value

Value

Test data

Source

Part of the simulated toy data set.

S_XX_study1 *Correlations between 10 SNPs corresponding to the population underlying study 1.*

Description

Data frame containing correlations between SNPs estimated from a reference database matching the study 1 population, e.g., the 1000Genomes. Here, [10 SNPs x 10 SNPs].

Format

Data frame

Value

Test data

Source

Part of the simulated toy data set.

S_XX_study2 *Correlations between 10 SNPs corresponding to the population underlying study 2.*

Description

Data frame containing correlations between SNPs estimated from a reference database matching the study 2 population, e.g., the 1000Genomes. Here, [10 SNPs x 10 SNPs].

Format

Data frame

Value

Test data

Source

Part of the simulated toy data set.

S_XY_full_study1 *Univariate summary statistics of 10 traits across 1000 SNPs (study 1).*

Description

Data frame containing univariate summary statistics (regression coefficients and standard errors) of study 1 for 1000 SNPs and 10 traits. It will be used for estimating phenotypic correlation structure S_YY of study 1.

Format

Data frame

Value

Test data

Source

Part of the simulated toy data set.

S_XY_full_study2 *Univariate summary statistics of 10 traits across 1000 SNPs (study 2).*

Description

Data frame containing univariate summary statistics (regression coefficients and standard errors) of study 2 for 1000 SNPs and 10 traits. It will be used for estimating phenotypic correlation structure S_YY of study 2.

Format

Data frame

Value

Test data

Source

Part of the simulated toy data set.

S_XY_study1 *Univariate summary statistics of 10 traits across 10 SNPs (study 1).*

Description

Data frame containing univariate summary statistics (regression coefficients and standard errors) of study 1 corresponding to the variables to be included in the association analysis: 10 SNPs and 10 traits.

Format

Data frame

Value

Test data

Source

Part of the simulated toy data set.

S_XY_study2 *Univariate summary statistics of 10 traits across 10 SNPs (study 2).*

Description

Data frame containing univariate summary statistics (regression coefficients and standard errors) of study 2 corresponding to the variables to be included in the association analysis: 10 SNPs and 10 traits.

Format

Data frame

Value

Test data

Source

Part of the simulated toy data set.

Index

∗ **Genetics**
    metaCcaGp, 3
    metaCcaPlusGp, 7
∗ **GenomeWideAssociation**
    estimateSyy, 2
    metaCcaGp, 3
    metaCcaPlusGp, 7
∗ **SNP**
    metaCcaGp, 3
    metaCcaPlusGp, 7
∗ **datasets**
    N1, 10
    N2, 11
    S_XX_study1, 11
    S_XX_study2, 12
    S_XY_full_study1, 12
    S_XY_full_study2, 13
    S_XY_study1, 13
    S_XY_study2, 14

estimateSyy, 2

metaCcaGp, 3
metaCcaPlusGp, 7

N1, 10
N2, 11

S_XX_study1, 11
S_XX_study2, 12
S_XY_full_study1, 12
S_XY_full_study2, 13
S_XY_study1, 13
S_XY_study2, 14