Package 'MetaHD'

February 5, 2026

Type Package

Title A Multivariate Meta-Analysis Model for High-Dimensional Data

Version 0.1.4

Maintainer Jayamini Liyanage <j.liyanage@latrobe.edu.au>

Author Jayamini Liyanage [aut, cre],
Alysha De Livera [aut],
Luke Prendergast [aut]

Description Performs multivariate meta-analysis for high-dimensional data to integrate and collectively analyse individual-level data from multiple studies, as well as to combine summary estimates. This approach accounts for correlation between outcomes, incorporates within- and between-study variability, handles missing values, and uses shrinkage estimation to accommodate high dimensionality. The 'MetaHD' R package provides access to our multivariate meta-analysis approach, along with a comprehensive suite of existing meta-analysis methods, including fixed-effects and random-effects models, Fisher's method, Stouffer's method, the weighted Z method, Lancaster's method, the weighted Fisher's method, and vote-counting approach. A detailed vignette with example datasets and code for data preparation and analysis is available at <https://alyshadelivera.github.io/MetaHD_vignette/>.

License GPL-3

Encoding UTF-8

Depends R (>= 3.5.0)

LazyData true

RoxygenNote 7.3.3

Imports dplyr, tidyr, metafor, corpcor, nloptr, Matrix, matrixcalc, Rcpp (>= 1.0.0), dynamicTreeCut, future.apply, metapro, metap

LinkingTo Rcpp, RcppArmadillo

NeedsCompilation yes

Repository CRAN

Date/Publication 2026-02-05 08:30:16 UTC

Contents

`MetaHD` *A Multivariate Meta-Analysis Model for High-Dimensional Data*

Description

The MetaHD function performs a multivariate meta-analysis for high-dimensional data, combining summary estimates obtained from multiple studies by using restricted maximum likelihood estimation. In its default settings, the function fits the fastMetaHD model, which provides a memory-efficient and computationally faster implementation of the MetaHD methodology. Assuming a meta-analysis is based on N outcomes and K studies:

Usage

```
MetaHD(
  Y,
  Slist,
  Psi = NULL,
  method = c("multi","REM","FEM"),
  bscov = c("unstructured","diag","none"),
  useDivideConquer = FALSE,
  parallel = FALSE,
  est.wscor = FALSE,
  shrinkCor = TRUE,
  impute.na = FALSE,
  optim.algorithm = c("BOBYQA","hybrid","L-BFGS-B"),
  optim.maxiter = 2000,
  rigls.iter = 1,
  initPsi = NULL,
  impute.var = 10^4
)
```

Arguments

`Y`	treatment effect sizes of the outcomes. This should be in the form of a K x N matrix.

`Slist` A K-dimensional list of N × N matrices representing within-study variances and covariances of the treatment effects. If within-study correlations are not available, provide the associated variances of the treatment effects as a K × N matrix and set est.wscor = TRUE. For method = "REM" or method = "FEM", provide the associated variances of the treatment effects as a K × N matrix.

`Psi` N x N matrix representing between-study variances and covariances of the treatment effects. (optional, if not specified this will be estimated internally by "MetaHD" using "estimateBSvar" and "estimateCorMat" functions in "MetaHD" package).

`method` estimation method: "multi" for multivarite meta-analysis model fitted through restricted maximum likelihood estimation where the between-study covariance structure can be selected via 'bscov', "REM" for univariate random-effects model fitted through restricted maximum likelihood estimation and "FEM" for univariate fixed-effects model.

`bscov` a character vector defining the structure of the random-effects covariance matrix. Among available covariance structures, the user can select "unstructured" to obtain between-study covariance matrix with diagonal elements (variances) estimated using restricted maximum likelihood and off-diagonal elements (co-variances) reflecting the correlations estimated via shrinkage, "diag" (diagonal) for between-study variances as diagonal elements and zero co-variances, and "none" for zero between-study variances and co-variances.

`useDivideConquer` a logical value indicating whether to use the divide-and-conquer implementation of the fastMetaHD model. This option is used only when method = "multi". Default is FALSE.

`parallel` a logical value indicating whether to enable parallel computation for the divide-and-conquer approach. Default is `FALSE`. See also Details.

`est.wscor` a logical value indicating whether the within-study correlation matrix needs to be estimated or not. Default is `FALSE`.

`shrinkCor` a logical value indicating whether a shrinkage estimator should be used to estimate within- or between-study correlation matrix. `TRUE`.

`impute.na` a logical value indicating whether missing values need to be imputed or not. Default is `FALSE`.

`optim.algorithm` specifies the algorithm used to maximize the restricted log-likelihood function for estimating between-study variances. The default algorithm is "BOBYQA", which offers derivative-free, bound-constrained optimization by iteratively constructing a quadratic approximation of the objective function. The "hybrid" option performs up to rigls.iter iterations of the RIGLS algorithm, followed by quasi-Newton (BFGS algorithm) iterations until convergence. If rigls.iter is set to zero, only the quasi-Newton method (BFGS algorithm) is used for estimation. The "L-BFGS-B" algorithm is a limited-memory version of the BFGS quasi-Newton method, which supports box constraints, allowing each variable to have specified lower and/or upper bounds.

`optim.maxiter` maximum number of iterations in methods involving optimization procedures.

`rigls.iter`	number of iterations of the restricted iterative generalized least square algorithm (RIGLS) when used in the initial phase of hybrid optimization procedure. Default is set to 1.
`initPsi`	N x N diagonal matrix representing the starting values of the between-study variances to be used in the optimization procedures. If not specified, the starting values in Psi default to a diagonal matrix with variances set to 1.
`impute.var`	multiplier for replacing the missing variances in Slist.(a large value, default is 10^4).

Details

If `parallel = TRUE`, the divide-and-conquer approach may be evaluated in parallel. Parallel execution is implemented using the `future` R package.

On Windows, users must set a future plan (e.g., `future::plan(future::multisession, workers = ncores)`) before calling `MetaHD()` in order to enable parallel computation.

On Linux and macOS, users may alternatively use `future::plan(future::multicore, workers = ncores)`.

If no future plan is set, or if `parallel = FALSE`, computations are performed sequentially.

Value

A list of objects containing :

- `estimate`: An N-dimensional vector of the combined estimates.
- `std.err`: An N-dimensional vector of the associated standard errors.
- `pVal`: An N-dimensional vector of the p-values.
- `I2.stat`: I^2 statistics.

References

Liyanage JC, Prendergast L, Staudte R, De Livera AM (2024). *MetaHD: a multivariate meta-analysis model for metabolomics data*. Bioinformatics, 40(7), btae470. doi:10.1093/bioinformatics/btae470

Powell MJ (2009). *The BOBYQA algorithm for bound constrained optimization without derivatives*. Cambridge NA Report NA2009/06, University of Cambridge, 26, 26–46.

Sera F, Armstrong B, Blangiardo M, et al. (2019). *An extended mixed-effects framework for meta-analysis*. Statistics in Medicine, 38, 5429–5444.

Schäfer J, Strimmer K (2005). *A shrinkage approach to large-scale covariance estimation and implications for functional genomics*. Statistical Applications in Genetics and Molecular Biology, 4, 32.

Examples

```
# CREATE INPUT DATA
input_data <- MetaHDInput(realdata)
Y <- input_data$Y
Slist <- input_data$Slist
```

```
N <- ncol(Y)
K <- nrow(Y)

Smat <- matrix(0, nrow = K, ncol = N)
for (i in 1:K) {
 Smat[i, ] <- diag(Slist[[i]])
}

# MULTIVARIATE RANDOM-EFFECTS META-ANALYSIS
model <- MetaHD(Y = Y, Slist = Slist, method = "multi")
model$estimate
model$pVal

# UNIVARIATE RANDOM-EFFECTS META-ANALYSIS
model <- MetaHD(Y = Y, Slist = Smat, method = "REM")
model$estimate
model$pVal

# UNIVARIATE FIXED-EFFECTS META-ANALYSIS
model <- MetaHD(Y = Y, Slist = Smat, method = "FEM")
model$estimate
model$pVal
```

`MetaHDInput` *Creating Input Data for MetaHD When Individual-Level Data are Available*

Description

The MetaHDInput function creates input data `Y` (treatment effects) and `Slist` (within-study covariance matrices) for `MetaHD` when individual-level data are available. Assuming that the individual-level data are in the following format, with 'study' in column 1, 'group' in column 2 and outcomes in rest of the columns, with samples in rows.

Usage

```
MetaHDInput(data)
```

Arguments

`data`	a dataframe consisting of individual-level data in the format, where 'study' in column 1, 'group' in column 2 and outcomes in rest of the columns and samples in rows.

Value

A list of objects containing :

- `Y`: A $K \times N$ matrix of treatment effect sizes, where K is the number of studies and N is the number of outcomes.
- `Slist`: A list of length K containing $N \times N$ within-study variance–covariance matrices of the treatment effects.

Examples

```
# CREATE INPUT DATA
input_data <- MetaHDInput(realdata)

## treatment effect-sizes
Y <- input_data$Y
head(Y)

## within-study variance-covariance matrices
Slist <- input_data$Slist
head(Slist[[1]])
```

MetaHDpval *P-value Combination Methods for High-Dimensional Data*

Description

Combines individual p-values across multiple studies for each outcome using p-value combination methods applied independently per outcome. Includes traditional and weighted p-value combination approaches and a vote counting method.

Usage

```
MetaHDpval(
  pmat,
 method = c("Fisher", "Stouffer", "wZ", "Lancaster", "wFisher", "Vote counting"),
  weight = NULL,
  is.onetail = TRUE,
  eff.sign = NULL,
  alpha = 0.5
)
```

Arguments

`pmat`	A $K \times N$ matrix of individual p-values, where K is the number of studies and N is the number of outcomes.

`method`	Character string specifying the p-value combination method. One of `"Fisher"`, `"Stouffer"`, `"wZ"`, `"Lancaster"`, `"wFisher"`, or `"Vote counting"`. See Details for more information.
`weight`	An optional $K \times N$ matrix of weights or sample sizes for each outcome in each study. Not relevant for `"Vote counting"`.
`is.onetail`	Logical. If `TRUE`, p-values are combined without considering effect directions. If `FALSE`, effect directions are used via `eff.sign`. Default is `TRUE`. Not relevant for `"Vote counting"`.
`eff.sign`	An optional $K \times N$ matrix indicating the signs of effect sizes (e.g., `+1` or `-1`). Only used when `is.onetail = FALSE`. Not relevant for `"Vote counting"`.
`alpha`	Numeric value defining the p-value cutoff for the `"Vote counting"` method. By default, `alpha = 0.5` splits p-values at 0.5. Values between `alpha` and `1 - alpha` are treated as neutral. If `alpha > 1`, it is interpreted as a percentage.

Details

The `MetaHDpval` function offers five traditional and more recent p-value combination methods implemented using the `metapro` R package, as well as a vote counting method implemented using the `metap` R package:

- **Fisher's method** (Fisher, 1932), which combines logarithmically transformed p-values from individual studies for each outcome using Fisher's statistic.
- **Stouffer's method** (Stouffer et al., 1949), which combines inverse normal–transformed p-values derived from individual study test statistics for each outcome.
- **Weighted Z-method (wZ)** (Mosteller and Bush, 1954), an extension of Stouffer's method that incorporates study-specific weights, resulting in a weighted inverse normal combination.
- **Lancaster's method** (Lancaster, 1961), which generalizes Fisher's method by introducing weights and exploits the additive property of the χ^2-distribution.
- **Weighted Fisher's method (wFisher)** (Yoon et al., 2021), which extends Fisher's method by allowing non-integer weights reflecting study-specific information (e.g., sample sizes). This approach replaces the χ^2-distribution with the gamma distribution to accommodate non-integer degrees of freedom.
- **Vote counting method** (Becker, 1994), that classifies a study as positive if its p-value is less than `alpha` and as negative if it exceeds `1 - alpha`, with studies falling in between treated as neutral and excluded. The number of positive studies is then counted, and a one-sided binomial test is applied to the non-neutral studies to obtain a combined p-value for each outcome.

Value

A numeric vector of length N containing the combined p-values for each outcome.

References

Yoon, S., Baik, B., Park, T., et al. (2021). *Powerful p-value combination methods to detect incomplete association*. Scientific Reports, 11, 6980. doi:10.1038/s4159802186465y

Yoon, S. (2023). *metapro: Robust P-Value Combination Methods* (R package version 1.5.11). Comprehensive R Archive Network (CRAN). doi:10.32614/CRAN.package.metapro

Becker, B.J. (1994). *Combining significance levels. In Cooper H, Hedges LV (eds.), A handbook of research synthesis*, 215–230. Russell Sage, New York.

Dewey, M. (2025). *metap: Meta-Analysis of Significance Values* (R package version 1.13). Comprehensive R Archive Network (CRAN). doi:10.32614/CRAN.package.metap

Examples

```
## Example with 5 studies and 12 outcomes
set.seed(123)
pmat <- matrix(runif(15), nrow = 5, ncol = 12)
eff.sign <- matrix(sample(c(-1, 1), 60, replace = TRUE), nrow = 5, ncol = 12)
wmat <- matrix(sample(50:200, 60, replace = TRUE), nrow = 5, ncol = 12)

## Fisher's method
MetaHDpval(pmat, method = "Fisher", is.onetail = FALSE, eff.sign = eff.sign)

## Weighted Z method
MetaHDpval(pmat, method = "wZ", weight = wmat, is.onetail = FALSE, eff.sign = eff.sign)

## Vote counting
MetaHDpval(pmat, method = "Vote counting", alpha = 0.4)
```

---

realdata *An Individual-Level Metabolomics Dataset*

Description

This is a subset of data, publicly available on MetaboAnalyst example datasets.

Usage

```
realdata
```

Format

A data frame with 172 observations on 14 metabolites.

Examples

```
head(realdata)
```

simdata.1 *Simulated Dataset 1 : With Complete Data*

Description

This dataset consists of a list of two data frames containing treatment effect-sizes and within-study covariance matrices

Usage

```
simdata.1
```

Format

A list of data frames as follows:

`Y` treatment effect sizes of the metabolites in the form of a 12 x 30 matrix, where 12 is the number of studies and 30 is the number of metabolites.

`Slist` 12-dimensional list of 30 x 30 matrices representing within-study variances and covariances of the treatment effects

Examples

```
Y <- simdata.1$Y
Slist <- simdata.1$Slist

head(Y)
head(Slist[[1]])
head(Slist[[12]])
```

simdata.2 *Simulated Dataset 2 : With Data Missing-At-Random*

Description

This dataset consists of a list of two data frames containing treatment effect-sizes and within-study covariance matrices with missing values

Usage

```
simdata.2
```

Format

A list of data frames as follows:

`Y` treatment effect sizes of the metabolites in the form of a 12 x 30 matrix, where 12 is the number of studies and 30 is the number of metabolites.

`Slist` 12-dimensional list of 30 x 30 matrices representing within-study variances and covariances of the treatment effects

Examples

```
Y <- simdata.2$Y
Slist <- simdata.2$Slist

head(Y)
head(Slist[[1]])
head(Slist[[12]])
```

Index