

Package: MetaHD (via r-universe)

October 7, 2026

Type Package

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

Version 0.1.6

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. Visualisation tools are provided for interpreting and comparing results across methods, including Venn diagrams, UpSet plots, and ROC curves, heatmaps of pooled effect sizes and correlations among outcomes. 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 (>= 4.4.0)

LazyData true

LazyDataCompression xz

Imports dplyr, tidyr, metafor, corpcor, nloptr, Matrix, matrixcalc, Rcpp (>= 1.0.0), dynamicTreeCut, cluster, future.apply, metapro, metap, UpSetR, pROC, graphics, grDevices, tools, ComplexHeatmap, circlize

LinkingTo Rcpp, RcppArmadillo
Suggests gVenn, testthat (>= 3.0.0)
Config/roxygen2/version 8.0.0
NeedsCompilation yes
Config/pak/sysreqs cmake libfftw3-dev libicu-dev libpng-dev perl
Repository <https://jayaminicl.r-universe.dev>
Date/Publication 2026-07-09 12:30:02 UTC
RemoteUrl <https://github.com/cran/MetaHD>
RemoteRef HEAD
RemoteSha f49e086e282125d3128d68b8eb0f7e99c7144ea7

Contents

estimateCorMat	2
MetaHD	3
MetaHD-outputs	8
MetaHDInput	9
MetaHDPval	10
MetaHDResult	12
plot.MetaHDResult	14
plot_correlation_heatmap	18
plot_dendrogram	20
plot_effect_heatmap	21
realdata	24
simdata.1	25
simdata.2	26
simData.complete	26
simData.missing	27
Index	29

estimateCorMat	<i>Estimate the correlation matrix using observed treatment effects</i>
----------------	---

Description

Estimates the correlation matrix among outcomes from observed treatment effect sizes. This is the correlation structure that [MetaHD](#) uses internally: for the divide-and-conquer clustering of outcomes, as an estimate of the within-study correlations when these are unavailable, and to estimate the between-study correlations. When the number of outcomes exceeds the number of studies ($N > K$), a shrinkage estimator (Schaefer and Strimmer, 2005) is used by default to obtain a well-conditioned estimate.

Usage

```
estimateCorMat(Y, shrinkCor = TRUE, impute.na = FALSE)
```

Arguments

<code>Y</code>	A $K \times N$ matrix of treatment effect sizes (K studies, N outcomes).
<code>shrinkCor</code>	Logical. Whether to use a shrinkage estimator for the correlation matrix when $N > K$. Default is TRUE.
<code>impute.na</code>	Logical. Whether to impute missing values (each missing entry is replaced by its column mean) before estimating the correlations. Default is FALSE.

Value

An $N \times N$ correlation matrix. When there are two or fewer studies ($K \leq 2$), the correlations cannot be estimated and $\emptyset$ is returned.

References

Schaefer, J. and 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.

See Also

[plot_correlation_heatmap](#) for visualising a correlation matrix.

Examples

```
Y <- simdata.1$Y
cormat <- estimateCorMat(Y)
dim(cormat)
```

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,
  DCgroups = NULL,
  parallel = FALSE,
  dendro.method = c("dynamicTreeCut", "fixedHeight", "fixedK", "optimalK"),
  dendro.height = NULL,
  dendro.k = NULL,
  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 x 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 x N matrix and set est.wscor = TRUE. For method = "REM" or method = "FEM", provide the associated variances of the treatment effects as a K x 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 multivariate 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 (covariances) 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.
<code>useDivideConquer</code>	a logical value indicating whether to use the divide-and-conquer implementation of the fastMetaHD model. This option is used only when <code>method = "multi"</code> . Default is FALSE.
<code>DCgroups</code>	A list of outcome groups for the divide-and-conquer approach. Each element should be a character vector containing the names of outcomes belonging to the same cluster. Outcome names must match column names in the input data (Y), and each outcome may appear in at most one group. If NULL, groups are determined automatically within the function using hierarchical clustering followed by a dynamic tree cut.
<code>parallel</code>	a logical value indicating whether to enable parallel computation for the divide-and-conquer approach. Default is FALSE. See also Details.
<code>dendro.method</code>	Character string specifying the method used to cut the dendrogram when <code>useDivideConquer = TRUE</code> and <code>DCgroups = NULL</code> . One of: "dynamicTreeCut" Default. Automatically determines the number of clusters using the dynamic tree cut algorithm. "fixedHeight" Cuts the dendrogram at a fixed height specified via <code>dendro.height</code> . "fixedK" Cuts the dendrogram into a fixed number of clusters specified via <code>dendro.k</code> . "optimalK" Automatically selects the optimal number of clusters by maximising the average silhouette width across all candidate partitions.
<code>dendro.height</code>	A numeric value specifying the height at which to cut the dendrogram. Only used when <code>dendro.method = "fixedHeight"</code> . Must be a positive number within the range of the dendrogram heights. Use <code>plot_dendrogram(Y)</code> to visualize the dendrogram and choose an appropriate height before running MetaHD. If NULL (default) and <code>dendro.method = "fixedHeight"</code> , an error is thrown.
<code>dendro.k</code>	A positive integer specifying the desired number of clusters. Only used when <code>dendro.method = "fixedK"</code> . Must be between 2 and <code>ncol(Y) - 1</code> . If NULL (default) and <code>dendro.method = "fixedK"</code> , an error is thrown.
<code>est.wscor</code>	a logical value indicating whether the within-study correlation matrix needs to be estimated or not. Default is FALSE.
<code>shrinkCor</code>	a logical value indicating whether a shrinkage estimator should be used to estimate within- or between-study correlation matrix. TRUE.
<code>impute.na</code>	a logical value indicating whether missing values need to be imputed or not. Default is FALSE.
<code>optim.algorithm</code>	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 <code>rigls.iter</code> iterations of the RIGLS algorithm, followed by quasi-Newton (BFGS algorithm) iterations until convergence. If <code>rigls.iter</code> 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.
<code>optim.maxiter</code>	maximum number of iterations in methods involving optimization procedures.
<code>rigls.iter</code>	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.
<code>initPsi</code>	$N \times N$ 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.
<code>impute.var</code>	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.
- `clustering_diagnostics`: A list of clustering diagnostic outputs returned when `useDivideConquer = TRUE` and `DCgroups = NULL`. Contains:
 - `avg_silhouette`: Overall average silhouette width across all outcomes.
 - `cluster_avg`: Average silhouette width per cluster.
 - `n_clusters`: Number of clusters identified.
 - `cluster_sizes`: Number of outcomes in each cluster.
 - `silhouette_object`: The full silhouette object of class "silhouette", which can be passed directly to `plot()` for visualization.
 - `silhouette_by_k`: A data frame with columns `k` and `avg_silhouette`, returned only when `dendro.method = "optimalK"`, showing the average silhouette width at each candidate partition used to select the optimal number of clusters.

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.
- Schaefer 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.
- Langfelder, P., Zhang, B. and Horvath, S. (2008). *Defining clusters from a hierarchical cluster tree: the Dynamic Tree Cut package for R*. Bioinformatics, 24(5), pp. 719–720.
- Rousseeuw, P.J. (1987). *Silhouettes: a graphical aid to the interpretation and validation of cluster analysis*. Journal of Computational and Applied Mathematics, 20, pp. 53–65.

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
```

Description

`summary()` returns the combined meta-analysis results from a [MetaHD](#) fit as a tidy data frame, one row per outcome. It is convenient for inspecting the results and for exporting them to a .csv file with [write.csv](#). `print()` shows a concise overview of the fit.

Usage

```
## S3 method for class 'MetaHD'
summary(object, outcome_names = NULL, ...)

## S3 method for class 'MetaHD'
print(x, ...)
```

Arguments

<code>object, x</code>	A "MetaHD" object returned by MetaHD .
<code>outcome_names</code>	Optional character vector of outcome labels, one per outcome. If NULL (default), the names of <code>x\$estimate</code> are used when available, otherwise <code>Var1</code> , <code>Var2</code> , ...
<code>...</code>	Additional arguments. For <code>print()</code> these are forwarded to <code>summary()</code> (e.g. <code>outcome_names</code>); <code>summary()</code> itself ignores them.

Value

`summary()` returns a data frame with one row per outcome and the columns `outcome`, `estimate`, `std.err`, `pVal` and `I2.stat`. `print()` displays a concise overview and invisibly returns `x`.

See Also

[MetaHD](#)

Examples

```
Y <- simdata.1$Y
Slist <- simdata.1$Slist
model <- MetaHD(Y, Slist)

model # concise overview
results <- summary(model, outcome_names = colnames(Y))
head(results)

## Not run:
# Export the results table to CSV
write.csv(summary(model, outcome_names = colnames(Y)),
           "metahd_results.csv", row.names = FALSE)
```

```
## End(Not run)
```

MetaHDIInput	<i>Creating Input Data for MetaHD When Individual-Level Data are Available</i>
--------------	--

Description

The MetaHDIInput 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

```
MetaHDIInput(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 <- MetaHDIInput(realdata)

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

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

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

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

MetaHDResult

Create a MetaHD result object

Description

Constructs an object containing combined results from different meta-analysis methods, for downstream visualisation via Venn diagrams (Tav, 2025), UpSet plots (Conway et al., 2017), and ROC curves (Robin et al., 2011). The `sets` argument accepts numeric vectors of p-values (thresholded by `alpha`) or effect sizes (ranked and selected by `top_n`). Note that ROC curves require numeric p-values and a truth vector.

Usage

```
MetaHDResult(
  sets,
  outcome_names = NULL,
  truth = NULL,
  alpha = 0.05,
  top_n = NULL
)
```

Arguments

<code>sets</code>	A named list where each element is a numeric vector of p-values or effect sizes for one method, one value per outcome. All vectors must have the same length. Selection of outcomes requires either <code>alpha</code> (for p-values; available for Venn, UpSet, and ROC plots) or <code>top_n</code> (for effect sizes; available for Venn and UpSet plots). Names of the list are used as method labels in the plots.
<code>outcome_names</code>	An optional character vector of outcome names matching the length of the numeric vectors in <code>sets</code> . If <code>NULL</code> (default), outcomes are indexed by position.
<code>truth</code>	An optional logical or numeric (0/1) vector of length <code>N</code> indicating true signals. Required for ROC curve visualisation via <code>plot(res, type = "ROC")</code> .
<code>alpha</code>	A numeric value specifying the significance threshold used when <code>sets</code> contains p-value vectors. Default is <code>0.05</code> . Ignored when <code>top_n</code> is specified.
<code>top_n</code>	Optional positive integer. If specified, the <code>top_n</code> outcomes with the largest absolute values are selected from each numeric vector in <code>sets</code> . Useful when <code>sets</code> contains effect sizes rather than p-values. Overrides <code>alpha</code> when specified. Note: ROC curves are not available when <code>top_n</code> is used.

Details

This object is designed to be used with the S3 method `plot.MetaHDResult` for visualisation of **MetaHD** results via Venn diagrams (Tav, 2025), UpSet plots (Conway et al., 2017), and ROC curves (Robin et al., 2011).

The function supports two input modes:

1. **P-value thresholding:** Supply numeric p-value vectors with alpha (and optionally outcome_names). Available for Venn, UpSet, and ROC plots.
2. **Top-N selection:** Supply numeric vectors (e.g. effect sizes) with top_n (and optionally outcome_names). Available for Venn and UpSet plots only.

Value

An object of class "MetaHdResult" containing:

sig_df A data frame of binary indicators (1 = significant/selected, 0 = otherwise).

sets The original input list.

truth The provided ground truth vector (if any).

alpha The significance threshold used (if applicable).

top_n The top-N value used (if applicable).

References

Tav, C. (2025). *gVenn: Proportional Venn and UpSet Diagrams for Gene Sets and Genomic Regions*. doi:10.18129/B9.bioc.gVenn. R package version 1.1.1, <https://bioconductor.org/packages/gVenn>

Conway, J.R., Lex, A., and Gehlenborg, N. (2017). *UpSetR: an R package for the visualization of intersecting sets and their properties*. Bioinformatics, 33(18), 2938–2940. doi:10.1093/bioinformatics/btx364

Robin, X., Turck, N., Hainard, A., Tiberti, N., Lisacek, F., Sanchez, J., and Mueller, M. (2011). *pROC: an open-source package for R and S+ to analyze and compare ROC curves*. BMC Bioinformatics, 12, 77. doi:10.1186/147121051277

See Also

`plot.MetaHdResult` for Venn diagrams, UpSet plots, and ROC curves based on this object. `plot_effect_heatmap` for comparing pooled effect size estimates across methods. `plot_correlation_heatmap` for visualising correlations among outcomes.

Examples

```
set.seed(123)
N <- 100
truth <- rbinom(N, 1, 0.2) # 20% of features are true signals

# Example 1: p-value threshold (default alpha = 0.05)
# Supports venn, upset, and ROC plots
res <- MetaHdResult(
  sets = list(
    method_A = runif(N)^ifelse(truth, 5, 1),
    method_B = runif(N)^ifelse(truth, 3, 1),
    method_C = runif(N)^ifelse(truth, 2, 1)
  ),
  truth = truth
)
```

```
# Example 2: top-N selection by effect size
# Supports venn and upset plots only
res2 <- MetaHdResult(
  sets = list(
    method_A = rnorm(N),
    method_B = rnorm(N),
    method_C = rnorm(N)
  ),
  outcome_names = paste0("Var", seq_len(N)),
  top_n = 10
)
```

plot.MetaHdResult	<i>Plot MetaHD results</i>
-------------------	----------------------------

Description

Visualizes a `MetaHdResult` object using a Venn diagram (Tav, 2025) for a compact view of overlapping outcomes across methods, an UpSet plot (Conway et al., 2017) to show overlap of significant outcomes across methods, or ROC curves (Robin et al., 2011) to assess performance when ground truth is available. Optionally saves the plot to a file in any of several formats.

Usage

```
## S3 method for class 'MetaHdResult'
plot(
  x,
  type = c("upset", "venn", "ROC"),
  roc.colors = NULL,
  venn.colors = NULL,
  queries = NULL,
  highlight = NULL,
  highlight.colors = NULL,
  show.truth = TRUE,
  file = NULL,
  width = 8,
  height = 6,
  units = c("in", "cm", "mm"),
  dpi = 300,
  ...
)
```

Arguments

`x` An object of class `"MetaHdResult"`.

type	Character string specifying the type of plot: "upset" (default) for an UpSet plot (Conway et al., 2017); "venn" for a Venn diagram (Tav, 2025); or "ROC" for ROC curves (Robin et al., 2011). See Details for more information.
roc.colors	Optional vector of colors for ROC curves. Must match the number of methods. If NULL (default), colors are generated automatically via <code>grDevices::hcl.colors()</code> . Used only when <code>type = "ROC"</code> .
venn.colors	Optional vector of colors for the Venn diagram sets. Must match the number of methods. If NULL (default), colors are generated automatically via <code>grDevices::hcl.colors()</code> . Used only when <code>type = "venn"</code> .
queries	For <code>type = "upset"</code> : a list of pre-built UpSetR query lists for highlighting, or NULL. For details and examples on creating UpSetR queries see R package UpSetR (Conway et al., 2017).
highlight	For <code>type = "upset"</code> : a convenience argument for highlighting intersections. A list of character vectors; each vector names the methods (or "Truth") whose intersection should be highlighted. Ignored if <code>queries</code> is supplied.
highlight.colors	Optional colors for highlight. If NULL (default), distinct colors are generated automatically via <code>grDevices::hcl.colors()</code> .
show.truth	Logical. If TRUE (default) and <code>x\$truth</code> is available, includes "Truth" as an additional set in the UpSet plot. Not applicable for <code>type = "ROC"</code> or <code>type = "venn"</code> .
file	Optional file path to save the plot. If NULL (default), the plot is drawn to the active graphics device. See the Saving the plot section below.
width, height	Plot dimensions. <i>Used only when file is set.</i> Default 8 x 6.
units	Units for width and height: "in" (default), "cm", or "mm". <i>Used only when file is set.</i>
dpi	Resolution for raster formats (PNG, JPEG, TIFF, BMP). Ignored for vector formats (PDF, SVG). Default 300. <i>Used only when file is set.</i>
...	Further arguments passed to <code>gVenn::plotVenn()</code> for <code>type = "venn"</code> , <code>UpSetR::upset()</code> for <code>type = "upset"</code> , or <code>graphics::plot()</code> for <code>type = "ROC"</code> .

Details

Venn diagram (Tav, 2025): Displays proportional overlap of significant or selected outcomes across methods, implemented via the **gVenn** package. Recommended for up to three methods; for more methods use `type = "upset"`. Requires R ≥ 4.5 and **gVenn** to be installed:

```
BiocManager::install("gVenn")
```

Available for both input modes supported by [MetaHdResult](#).

UpSet plot (Conway et al., 2017): Visualizes the intersection of significant or selected outcomes across meta-analysis methods, implemented via the **UpSetR** package. Available for both input modes supported by [MetaHdResult](#).

ROC curves (Robin et al., 2011): Requires a ground truth vector (provided when creating the object) and displays ROC curves along with AUC values for each method, implemented via the

pROC package. AUC values are computed as the area under each ROC curve and displayed in the plot legend. **ROC curves require numeric p-values** — they are not available when `top_n` was used to create the `MetaHdResult` object.

Value

Invisibly returns: a named list of character vectors containing the significant or top-ranked outcome names per method for `type = "venn"`; and a named numeric vector of AUCs for `type = "ROC"`.

Saving the plot

To save the plot directly, pass a file path to `file`. The graphics device is auto-selected from the file extension, the plot is written to that file, and the plot is *not* also drawn to the screen. Supported extensions:

- Vector formats: `.pdf`, `.svg`
- Raster formats: `.png`, `.jpeg/.jpg`, `.tiff/.tif`, `.bmp`

Use `width`, `height`, `units`, and `dpi` to control the saved output. `dpi` is ignored for vector formats. The path may be absolute or relative to `getwd()`; missing parent directories are created automatically.

References

- Tav, C. (2025). *gVenn: Proportional Venn and UpSet Diagrams for Gene Sets and Genomic Regions*. doi:10.18129/B9.bioc.gVenn. R package version 1.1.1, <https://bioconductor.org/packages/gVenn>
- Conway, J.R., Lex, A., and Gehlenborg, N. (2017). *UpSetR: an R package for the visualization of intersecting sets and their properties*. Bioinformatics, 33(18), 2938–2940. doi:10.1093/bioinformatics/btx364
- Robin, X., Turck, N., Hainard, A., Tiberti, N., Lisacek, F., Sanchez, J., and Mueller, M. (2011). *pROC: an open-source package for R and S+ to analyze and compare ROC curves*. BMC Bioinformatics, 12, 77. doi:10.1186/147121051277

See Also

`MetaHdResult` for creating objects to be plotted. `plot_effect_heatmap` for comparing pooled effect size estimates across methods. `plot_correlation_heatmap` for visualising correlations among outcomes.

Examples

```
set.seed(123)
N <- 100
truth <- rbinom(N, 1, 0.2) # 20% of features are true signals
res <- MetaHdResult(
  sets = list(
    method_A = runif(N)^ifelse(truth, 5, 1),
    method_B = runif(N)^ifelse(truth, 3, 1),
    method_C = runif(N)^ifelse(truth, 2, 1)
```

```

    ),
    truth = truth
  )

# Example 1: simple UpSet plot
plot(res, type = "upset")

# Example 2: highlights via the `highlight` argument
plot(res, type = "upset",
      highlight = list(c("Truth", "method_A", "method_B", "method_C")),
      highlight.colors = "darkgreen")

# Example 3: passing in-built queries
plot(res, type = "upset",
      queries = list(
        list(
          query = UpSetR::intersects,
          params = list(c("method_A", "method_B", "method_C")),
          color = "dodgerblue3",
          active = TRUE,
          query.name = "Identified by all methods"
        ),
        list(
          query = function(row) {
            row["method_A"] == 1 && sum(row) < length(row)
          },
          color = "orange",
          active = TRUE,
          query.name = "Others identified by method_A"
        )
      ),
      show.truth = FALSE)

# Example 4: ROC curves with AUC values
# Requires numeric p-values in sets
aucs <- plot(res, type = "ROC")
print(aucs)

# Example 5: Venn diagram (Tav, 2025)
# Requires R >= 4.5 and gVenn installed
if (requireNamespace("gVenn", quietly = TRUE)) {
  plot(res, type = "venn")
}

## Not run:
# Example 6: save plots to file
plot(res, type = "venn", file = "venn.pdf", width = 8, height = 7)
plot(res, type = "upset", file = "upset.pdf", width = 10, height = 6)
plot(res, type = "ROC", file = "roc.png", width = 8, height = 6, dpi = 600)

## End(Not run)

```

plot_correlation_heatmap

Plot a heatmap of correlations among outcomes

Description

Visualises correlations among outcomes from a single covariance or correlation matrix. A covariance matrix is converted to correlations internally; a correlation matrix can be supplied directly via `is.corr = TRUE` (for example, the within-study or between-study correlations among outcomes). This is useful for understanding the correlation structure of the outcomes and assessing whether the multivariate meta-analysis approach is likely to offer advantages over univariate methods. This function uses the **ComplexHeatmap** package (Gu, 2022) for rendering.

Usage

```
plot_correlation_heatmap(
  mat,
  is.corr = FALSE,
  outcome_names = NULL,
  title = NULL,
  col_low = "blue",
  col_mid = "white",
  col_high = "red",
  cluster_rows = TRUE,
  cluster_columns = TRUE,
  row_km = 1,
  show_row_names = NULL,
  show_column_names = NULL,
  names_fontsize = 7,
  legend_title = "Correlation",
  file = NULL,
  width = 8,
  height = 7,
  units = c("in", "cm", "mm"),
  dpi = 300,
  ...
)
```

Arguments

<code>mat</code>	An $N \times N$ covariance or correlation matrix, where N is the number of outcomes, such as one of the within-study covariance matrices returned by <code>MetaHDIInput()</code> or a correlation matrix computed by the user. By default it is treated as a covariance matrix and converted to correlations internally; set <code>is.corr = TRUE</code> to supply a correlation matrix directly.
<code>is.corr</code>	Logical. If <code>FALSE</code> (default), <code>mat</code> is treated as a covariance matrix and converted to correlations with <code>cov2cor</code> . If <code>TRUE</code> , it is used directly as a correlation matrix.

outcome_names	An optional character vector of outcome names for axis labels. If NULL (default), row and column names of mat are used.
title	Character string for the heatmap title. If NULL (default), a default title is used. Use title = "" to omit the title.
col_low	Character string specifying the colour for correlation of -1. Default is "blue".
col_mid	Character string specifying the colour for correlation of 0. Default is "white".
col_high	Character string specifying the colour for correlation of 1. Default is "red".
cluster_rows	Logical. Whether to cluster rows. Default is TRUE.
cluster_columns	Logical. Whether to cluster columns. Default is TRUE.
row_km	Integer ≥ 1 . Number of groups into which outcomes are partitioned by k-means clustering, drawn as separate heatmap slices, showing groups of correlated outcomes (Gu, 2022). Because the correlation matrix is symmetric, the same partition is applied to both rows and columns so the diagonal blocks stay aligned. Default 1 (no partitioning).
show_row_names, show_column_names	Logical or NULL. Whether to display row / column (outcome) labels. If NULL (default), labels are shown when there are at most 50 outcomes and hidden otherwise (with a message). Set TRUE or FALSE to override.
names_fontsize	Numeric. Font size for row and column labels. Default is 7.
legend_title	Character string for the colour legend title. Default is "Correlation".
file	Optional file path to save the plot. If NULL (default), the plot is drawn to the active graphics device. Supported extensions: .pdf, .svg, .png, .jpeg/.jpg, .tiff/.tif, .bmp.
width, height	Plot dimensions. <i>Used only when file is set.</i> Default 8 x 6.
units	Units for width and height: "in" (default), "cm", or "mm". <i>Used only when file is set.</i>
dpi	Resolution for raster formats (PNG, JPEG, TIFF, BMP). Ignored for vector formats (PDF, SVG). Default 300. <i>Used only when file is set.</i>
...	Further arguments passed to ComplexHeatmap::Heatmap().

Details

Unless `is.corr = TRUE`, `mat` is treated as a covariance matrix and converted to a correlation matrix internally using `cov2cor()` before plotting. The colour scale runs from `col_low` at -1 through `col_mid` at 0 to `col_high` at 1.

Value

Invisibly returns the Heatmap object.

Large numbers of outcomes

With many outcomes the row and column labels become unreadable and individual cells shrink to sub-pixel size. To keep the plot readable, axis labels are hidden automatically beyond 50 outcomes (override via `show_row_names` / `show_column_names`). Very large heatmaps are rasterised automatically by **ComplexHeatmap** to keep file size and rendering manageable.

References

Gu, Z. (2022). *Complex heatmap visualization*. iMeta, 1, e43. doi:10.1002/imt2.43

See Also

[plot_effect_heatmap](#) for comparing pooled effect sizes across methods. [plot.MetaHDResult](#) for Venn diagrams, UpSet plots, and ROC curves.

Examples

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

# Example 1: a within-study covariance matrix (converted to correlations)
plot_correlation_heatmap(Slist[[1]])

# Example 2: partition into k-means blocks
set.seed(123) # reproducible k-means blocks
plot_correlation_heatmap(Slist[[1]], row_km = 4)

# Example 3: supply a correlation matrix directly
cormat <- estimateCorMat(Y)
plot_correlation_heatmap(cormat, is.corr = TRUE)

## Not run:
# Example 4: save to PNG
plot_correlation_heatmap(
  Slist[[1]],
  file = "correlation_heatmap.png",
  width = 10,
  height = 8,
  dpi = 600
)

## End(Not run)
```

plot_dendrogram

Plot dendrogram

Description

Plots the dendrogram of outcomes to help users choose an appropriate cutting height when using `dendro.method = "fixedHeight"` in `MetaHD`.

Usage

```
plot_dendrogram(Y, shrinkCor = TRUE, impute.na = FALSE, h = NULL, ...)
```

Arguments

Y	A K x N matrix of treatment effect sizes of the outcomes (K studies, N outcomes).
shrinkCor	Logical. Whether to use shrinkage estimation for the correlation matrix. Default is TRUE.
impute.na	Logical. Whether to impute missing values. Default is FALSE.
h	Optional numeric value. If provided, draws a horizontal red dashed line at this height to preview the cut.
...	Further arguments passed to <code>plot()</code> .

Value

Invisibly returns the `hclust` object.

plot_effect_heatmap	<i>Plot a heatmap of pooled effect sizes across meta-analysis methods</i>
---------------------	---

Description

Produces a heatmap with outcomes on the y-axis and meta-analysis methods on the x-axis, allowing visual comparison of pooled effect sizes across methods. Rows (outcomes) are clustered by default to reveal patterns of agreement and disagreement across methods. This function uses the **ComplexHeatmap** package (Gu, 2022) for rendering.

Usage

```
plot_effect_heatmap(
  estimates,
  outcome_names = NULL,
  title = "Pooled Effect Sizes Across Methods",
  col_low = "blue",
  col_mid = "white",
  col_high = "red",
  cluster_rows = TRUE,
  cluster_columns = FALSE,
  row_km = 1,
  show_row_names = NULL,
  label_top_n = NULL,
  row_names_fontsize = 8,
  column_names_fontsize = 10,
  legend_title = "Pooled\nEffect Size",
  file = NULL,
  width = 8,
  height = 6,
  units = c("in", "cm", "mm"),
```

```

    dpi = 300,
    ...
)

```

Arguments

<code>estimates</code>	A named list of numeric vectors, where each element corresponds to a meta-analysis method and contains the pooled effect size estimates for all outcomes. All vectors must have the same length. Names of the list are used as column labels (methods).
<code>outcome_names</code>	An optional character vector of outcome names to use as row labels. If NULL (default), outcomes are labelled Var1, Var2, ...
<code>title</code>	Character string for the heatmap title. Default is "Pooled Effect Sizes Across Methods". Use <code>title = ""</code> to omit the title.
<code>col_low</code>	Character string specifying the colour for the lowest values. Default is "blue".
<code>col_mid</code>	Character string specifying the colour for the midpoint (zero). Default is "white".
<code>col_high</code>	Character string specifying the colour for the highest values. Default is "red".
<code>cluster_rows</code>	Logical. Whether to cluster rows (outcomes). Default is TRUE.
<code>cluster_columns</code>	Logical. Whether to cluster columns (methods). Default is FALSE.
<code>row_km</code>	Integer ≥ 1 . Number of groups into which outcomes (rows) are partitioned by k-means clustering, each drawn as a separate heatmap slice, showing subgroups of outcomes with similar effect patterns (Gu, 2022). Default 1 (no partitioning).
<code>show_row_names</code>	Logical or NULL. Whether to display row (outcome) labels. If NULL (default), labels are shown when there are at most 50 outcomes (and no <code>label_top_n</code> is given) and hidden otherwise (with a message). Set TRUE or FALSE to override.
<code>label_top_n</code>	Optional positive integer. If set, the <code>label_top_n</code> outcomes with the largest absolute pooled effect (across methods) are labelled with connector lines via a mark annotation (Gu, 2022). Useful for highlighting key outcomes when row labels are hidden. When labels are marked this way, the full row labels are hidden by default.
<code>row_names_fontsize</code>	Numeric. Font size for row (outcome) labels. Default is 8.
<code>column_names_fontsize</code>	Numeric. Font size for column (method) labels. Default is 10.
<code>legend_title</code>	Character string for the colour legend title. Default is "Pooled\nEffect Size".
<code>file</code>	Optional file path to save the plot. If NULL (default), the plot is drawn to the active graphics device. Supported extensions: .pdf, .svg, .png, .jpeg/.jpg, .tiff/.tif, .bmp.
<code>width, height</code>	Plot dimensions. <i>Used only when file is set.</i> Default 8 x 6.
<code>units</code>	Units for width and height: "in" (default), "cm", or "mm". <i>Used only when file is set.</i>
<code>dpi</code>	Resolution for raster formats (PNG, JPEG, TIFF, BMP). Ignored for vector formats (PDF, SVG). Default 300. <i>Used only when file is set.</i>
<code>...</code>	Further arguments passed to <code>ComplexHeatmap::Heatmap()</code> .

Details

The colour scale is centred at zero by default, with `col_low` indicating negative effect sizes and `col_high` indicating positive effect sizes. This allows easy visual identification of outcomes with consistent direction across methods as well as those where methods disagree.

Value

Invisibly returns the Heatmap object.

Large numbers of outcomes

With many outcomes per-row labels become unreadable and individual rows are squeezed to sub-pixel height. To keep the plot readable, row labels are hidden automatically beyond 50 outcomes (override via `show_row_names`). Alternatively, label only the key outcomes with `label_top_n`, which marks the outcomes with the largest absolute effect with connector lines while leaving the rest unlabelled. Very large heatmaps are rasterised automatically by **ComplexHeatmap** to keep file size and rendering manageable.

References

Gu, Z. (2022). *Complex heatmap visualization*. iMeta, 1, e43. doi:10.1002/imt2.43

See Also

`plot_correlation_heatmap` for visualising correlations among outcomes. `plot.MetaHDResult` for Venn diagrams, UpSet plots, and ROC curves.

Examples

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

K <- nrow(Y)
N <- ncol(Y)
Smat <- matrix(0, nrow = K, ncol = N,
               dimnames = list(rownames(Y), colnames(Y)))
for (i in 1:K) Smat[i, ] <- diag(Slist[[i]])

model_multi <- MetaHD(Y = Y, Slist = Slist, method = "multi")
model_rem <- MetaHD(Y = Y, Slist = Smat, method = "REM")
model_fem <- MetaHD(Y = Y, Slist = Smat, method = "FEM")

# Example 1: basic heatmap with custom colours
plot_effect_heatmap(
  estimates = list(
    fastMetaHD = model_multi$estimate,
    REM = model_rem$estimate,
    FEM = model_fem$estimate
  ),
  outcome_names = colnames(Y),
  col_low = "purple",
```

```

    col_high = "orange"
  )

# Example 2: partition rows into k-means slices
set.seed(123) # reproducible k-means slices
plot_effect_heatmap(
  estimates = list(
    fastMetaHD = model_multi$estimate,
    REM = model_rem$estimate,
    FEM = model_fem$estimate
  ),
  outcome_names = colnames(Y),
  row_km = 4
)

# Example 3: label the outcomes with the largest absolute effect with connector lines.
# useful when there are many outcomes.
plot_effect_heatmap(
  estimates = list(
    fastMetaHD = model_multi$estimate,
    REM = model_rem$estimate,
    FEM = model_fem$estimate
  ),
  outcome_names = colnames(Y),
  label_top_n = 15
)

## Not run:
# Example 4: save to PNG
plot_effect_heatmap(
  estimates = list(
    fastMetaHD = model_multi$estimate,
    REM = model_rem$estimate,
    FEM = model_fem$estimate
  ),
  outcome_names = colnames(Y),
  file = "effect_heatmap.png",
  width = 10,
  height = 8,
  dpi = 600
)

## End(Not run)

```

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]])
```

simData.complete

*Simulated Study-Level Complete Data***Description**

A simulated dataset for demonstrating meta-analysis workflows in **MetaHD**. It contains the observed treatment effects, within-study covariance matrices, the associated variances of the treatment effects, individual two-sided *p*-values, true effect sizes, and sample sizes for each study.

Usage

```
simData.complete
```

Format

A list with six elements:

Y A numeric matrix of dimension 10 x 200 containing the observed effect sizes. Each row corresponds to a study and each column corresponds to one of 200 outcomes.

Slist A list of length 10 containing the within-study covariance matrices, one per study.

wsvar A numeric matrix of dimension 10 x 200 containing the within-study variances for each study and outcome.

pvals A numeric matrix of dimension 10 x 200 containing the two-sided p-values for each study and outcome.

true.theta A numeric vector of length 200 containing the true effect sizes.

sample.size An integer vector of length 10 giving the sample size of each study.

See Also

[simData.missing](#) for the version with missing values.

simData.missing	<i>Simulated Study-Level Data with Missing Values</i>
-----------------	---

Description

A simulated dataset for demonstrating meta-analysis workflows in **MetaHD** when some outcomes are unobserved in some studies. The structure matches [simData.complete](#), with missing values generated under a missing-at-random (MAR) mechanism using a logistic regression model that determines the probability of missingness based on the values of other outcomes. Missing values were introduced on ten randomly selected outcomes; for each selected outcome, approximately 50%

Usage

```
simData.missing
```

Format

A list with six elements:

Y A numeric matrix of dimension 10 x 200 containing the observed effect sizes, with NA values where outcomes are missing. Each row corresponds to a study and each column to one of 200 outcomes.

Slist A list of length 10 containing the within-study covariance matrices, one per study, with NA values where outcomes are missing.

wsvar A numeric matrix of dimension 10 x 200 containing the within-study variances for each study and outcome, with NA values where outcomes are missing.

pvals A numeric matrix of dimension 10 x 200 containing the two-sided p-values for each study and outcome, with NA values where outcomes are missing.

true.theta A numeric vector of length 200 containing the true effect sizes.

sample.size An integer vector of length 10 giving the sample size of each study.

See Also

[simData.complete](#) for the version without missing values.

Index

* datasets

- realdata, [24](#)
- simdata.1, [25](#)
- simdata.2, [26](#)
- simData.complete, [26](#)
- simData.missing, [27](#)

cov2cor, [18](#)

estimateCorMat, [2](#)

MetaHD, [2](#), [3](#), [8](#)

MetaHD-outputs, [8](#)

MetaHDInput, [9](#)

MetaHdpval, [10](#)

MetaHDResult, [12](#), [15](#), [16](#)

plot.MetaHDResult, [12](#), [13](#), [14](#), [20](#), [23](#)

plot_correlation_heatmap, [3](#), [13](#), [16](#), [18](#),
[23](#)

plot_dendrogram, [20](#)

plot_effect_heatmap, [13](#), [16](#), [20](#), [21](#)

print.MetaHD (MetaHD-outputs), [8](#)

realdata, [24](#)

simdata.1, [25](#)

simdata.2, [26](#)

simData.complete, [26](#), [27](#), [28](#)

simData.missing, [27](#), [27](#)

summary.MetaHD (MetaHD-outputs), [8](#)

write.csv, [8](#)