Package ‘easyEWAS’

March 16, 2026

Title Perform and Visualize EWAS Analysis

Version 1.0.1

Description Tools for conducting epigenome-wide association studies (EWAS) and visualizing results. Users provide sample metadata and methylation matrices to run EWAS with linear models, linear mixed-effects models, or Cox models. The package supports downstream visualization, bootstrap validation, enrichment analysis, batch effect correction, and differentially methylated region (DMR) analysis with optional parallel computing. Methods are described in Wang et al. (2025) <doi:10.1093/bioadv/vbaf026>, Johnson et al. (2007) <doi:10.1093/biostatistics/kxj037>, and Peters et al. (2015) <doi:10.1186/1756-8935-8-6>.

License GPL (>= 3)

URL https://github.com/ytwangZero/easyEWAS, https://easyewas-tutorial.github.io/

BugReports https://github.com/ytwangZero/easyEWAS/issues

Encoding UTF-8

RoxygenNote 7.3.3

Imports R6, boot, CMplot, ddpcr, doParallel, dplyr, foreach, lmerTest, magrittr, parallel, readxl, survival, tictoc, vroom, withr, R.utils, lubridate

Suggests AnnotationHub, ExperimentHub, DMRcate, sva, BiocParallel, clusterProfiler, enrichplot, org.Hs.eg.db, knitr, rmarkdown

Depends R (>= 4.4.0)

LazyData true

NeedsCompilation no

Author Yuting Wang [aut, cre], Xu Gao [aut]

Maintainer Yuting Wang <ytwang@pku.edu.cn>

Repository CRAN

Date/Publication 2026-03-16 19:10:14 UTC

Contents

batchEWAS 2
bootEWAS 3
dmrEWAS 4
downloadAnnotEWAS 6
enrichEWAS 7
ewasfun_lm 9
initEWAS 10
loadEWAS 10
methydata 11
plotEWAS 12
sampledata 17
startEWAS 17
transEWAS 19

Index **20**

`batchEWAS` *Perform batch effect correction*

Description

Perform batch effect correction based on the function `ComBat` form R package **sva**. It requires that the "batches" in the data set are known. It uses either parametric or non-parametric empirical Bayes frameworks for adjusting data for batch effects.

Usage

```
batchEWAS(
  input,
  adjustVar = NULL,
  batch = NULL,
  plot = FALSE,
  par.prior = TRUE,
  mean.only = FALSE,
  ref.batch = NULL,
  parallel = FALSE,
  core = NULL
)
```

Arguments

`input`	An R6 class integrated with all the information.
`adjustVar`	(Optional) Names of the variate of interest and other covariates besides batch, with each name separated by a comma. Ensure that when correcting for batch effects, the effects of other factors are appropriately considered and adjusted for.Ensure there are no space. e.g. "cov1,cov2".

batch	Name of the batch variable.
plot	Logical. TRUE give prior plots with black as a kernel estimate of the empirical batch effect density and red as the parametric. Thr default is FALSE.
par.prior	Logical. TRUE indicates parametric adjustments will be used, FALSE indicates non-parametric adjustments will be used.
mean.only	Logical. Default to FALSE. If TRUE, ComBat only corrects the mean of the batch effect (no scale adjustment).
ref.batch	(Optional) NULL If given, will use the selected batch as a reference for batch adjustment.
parallel	Logical. Whether to enable parallel computing during batch effect correction. Default is FALSE.
core	Integer. Number of CPU cores to use if parallel = TRUE. Default is NULL.

Value

input, An R6 class object integrating all information.

Examples

```
res <- initEWAS(export = FALSE)
res <- loadEWAS(input = res, ExpoData = "default", MethyData = "default")
res <- transEWAS(input = res, Vars = "cov1", TypeTo = "factor")
if (requireNamespace("sva", quietly = TRUE)) {
  res <- batchEWAS(input = res, batch = "batch", par.prior = TRUE, ref.batch = NULL)
}
```

---

bootEWAS *Perform Bootstrap-based Internal Validation*

Description

Users can perform internal validation of the identified differently methylated sites based on the bootstrap method.

Usage

```
bootEWAS(input, filterP = "PVAL", cutoff = 0.05, CpGs = NULL, times = 500,
bootCI = "perc", filename = "default", seed = NULL)
```

Arguments

input	An R6 class integrated with all the information obtained from the startEWAS or plotEWAS function.
filterP	The name of the p value columns such as "PVAL", "FDR", and "Bonfferoni." Users use this P-value to screen for significance sites and further conduct internal validation.

cutoff	The cutoff value of the P-value used to filter for further internal validation. The default is 0.05.
CpGs	The name of the methylation site specified by the user for bootstrap analysis, separated by commas. Be careful not to have spaces, such as "cpg1,cpg2".
times	Number of bootstrap times specified by the user. The default value is 100 times.
bootCI	A vector of character strings representing the type of interval to base the test on. The value should be one of "norm", "basic", "stud", "perc" (the default), and "bca".
filename	User-customized .csv file name for storing bootstrap results. If "default", it will be named as "bootresult".
seed	Optional integer seed used for reproducible bootstrap resampling. If NULL (default), the current random-number-generator state is used.

Value

input, An R6 class object integrating all information.

Examples

```
res <- initEWAS(export = FALSE)
res <- loadEWAS(input = res, ExpoData = "default", MethyData = "default")
res <- transEWAS(input = res, Vars = "cov1", TypeTo = "factor")
res$Data$Methy <- res$Data$Methy[seq_len(50), , drop = FALSE]
res <- startEWAS(
  input = res, chipType = NULL, model = "lm",
  expo = "default", adjustP = TRUE, core = 1
)
res <- bootEWAS(input = res, CpGs = res$result$probe[1], times = 5, seed = 1)
```

---

dmrEWAS *Perform Differentially Methylated Region analysis*

Description

Perform differential methylation analysis based on the R package **DMRcate**. Computes a kernel estimate against a null comparison to identify significantly DMRs.

Usage

```
dmrEWAS(
  input,
  chipType = "EPICV2",
  what = "Beta",
  epicv2Filter = "mean",
  expo = NULL,
  cov = NULL,
  genome = "hg38",
```

```
  fdrCPG = 0.05,
  pcutoff = "fdr",
  lambda = 1000,
  C = 2,
  min.cpgs = 2,
  filename = "default"
)
```

Arguments

`input`	An R6 class integrated with all the information.
`chipType`	The Illumina chip versions for user measurement of methylation data, including "450K","EPICV1", and "EPICV2". The default is "EPICV2".
`what`	Types of methylation values, including "Beta" and "M". Default to "Beta".
`epicv2Filter`	Strategy for filtering probe replicates that map to the same CpG site. "mean" takes the mean of the available probes; "sensitivity" takes the available probe most sensitive to methylation change; "precision" either selects the available probe with the lowest variation from the consensus value (most precise), or takes the mean if that confers the lowest variation instead, "random" takes a single probe at random from each replicate group.
`expo`	Name of the exposure variable used in the DMR analysis.
`cov`	Name(s) of covariate(s) used in the DMR analysis, with each name separated by a comma. Ensure there are no space. e.g. "cov1,cov2,cov3".
`genome`	Reference genome for annotating DMRs. Must be consistent with the array platform used: • Use `"hg38"` for EPICV2 arrays. • Use `"hg19"` for 450K and EPICV1 arrays. Note: the `genome` argument does not currently affect the internal behavior of `extractRanges()` (i.e., no liftover is performed).
`fdrCPG`	Used to individually assess the significance of each CpG site. If the FDR-adjusted p-value of a CpG site is below the specified fdrCPG threshold, the site will be marked as significant. The default value is 0.05.
`pcutoff`	Used to determine the threshold for DMRs. It is strongly recommended to use the default (fdr), unless you are confident about the risk of Type I errors (false positives).
`lambda`	If the distance between two significant CpG sites is greater than or equal to lambda, they will be considered as belonging to different DMRs. The default value is 1000 nucleotides, meaning that if the distance between two significant CpG sites exceeds 1000 nucleotides, they will be separated into different DMRs.
`C`	Scaling factor for bandwidth. Gaussian kernel is calculated where lambda/C = sigma. Empirical testing shows for both Illumina and bisulfite sequencing data that, when lambda=1000, near-optimal prediction of sequencing-derived DMRs is obtained when C is approximately 2, i.e. 1 standard deviation of Gaussian kernel = 500 base pairs. Cannot be < 0.2.
`min.cpgs`	Minimum number of consecutive CpGs constituting a DMR. Default to 2.

`filename` User-customized .csv file name for storing DMR results. If "default", it will be named as "DMRresult".

Value

input, An R6 class object integrating all information.

Examples

```
res <- initEWAS(export = FALSE)
res <- loadEWAS(input = res, ExpoData = "default", MethyData = "default")
res <- transEWAS(input = res, Vars = "cov1", TypeTo = "factor")
if (interactive() &&
    requireNamespace("DMRcate", quietly = TRUE) &&
    requireNamespace("ExperimentHub", quietly = TRUE) &&
    requireNamespace("AnnotationHub", quietly = TRUE)) {
  hub_cache <- file.path(tempdir(), "bioc-hub-cache")
  dir.create(hub_cache, recursive = TRUE, showWarnings = FALSE)
  ExperimentHub::setExperimentHubOption("CACHE", hub_cache)
  AnnotationHub::setAnnotationHubOption("CACHE", hub_cache)
  res <- dmrEWAS(
    input = res,
    filename = "default",
    chipType = "EPICV2",
    what = "Beta",
    expo = "var",
    cov = "cov1,cov2",
    genome = "hg38"
  )
}
```

`downloadAnnotEWAS` *Download and cache chip annotation tables*

Description

Download CpG annotation tables for supported Illumina chip types and store them in a local cache directory. The downloaded files are saved as `.rds` and reused in later analyses.

Usage

```
downloadAnnotEWAS(chipType = c("EPICV2", "EPICV1", "450K", "27K", "MSA"),
cache_dir = NULL, force = FALSE, base_url = getOption("easyEWAS.annotation_base_url",
"https://github.com/ytwangZero/easyEWAS_materials/raw/main/annotation"), quiet = FALSE)
```

Arguments

chipType	One or more chip types. Supported values: "EPICV2", "EPICV1", "450K", "27K", and "MSA".
cache_dir	Directory to store downloaded annotation files. If NULL, the default cache path tools::R_user_dir("easyEWAS", "cache")/annotation is used.
force	Logical. If TRUE, re-download and overwrite existing cached files.
base_url	Base URL where annotation .rds files are hosted.
quiet	Logical. Passed to utils::download.file().

Value

A named character vector of local file paths (invisibly).

---

enrichEWAS *Enrichment analyses*

Description

Perform GO or KEGG enrichment analysis based on the **clusterProfiler** package.

Usage

```
enrichEWAS(
  input,
  filename = "default",
  method = "GO",
  filterP = "PVAL",
  cutoff = 0.05,
  ont = "BP",
  pool = FALSE,
  plot = TRUE,
  plotType = "dot",
  plotcolor = "p.adjust",
  x = "GeneRatio",
  showCategory = 10,
  width = 11,
  height = 7,
  pvalueCutoff = 0.05,
  pAdjustMethod = "BH",
  qvalueCutoff = 0.2
)
```

Arguments

`input`	An R6 class integrated with all the information obtained from the startEWAS or plotEWAS or bootEWAS function.
`filename`	User-customized .xlsx file name for storing EWAS results. If "default" is chosen, it will be named as "enrichresult".
`method`	Methods of enrichment analysis, including "GO" and "KEGG".
`filterP`	The name of the p value columns such as "PVAL", "FDR", and "Bonfferoni." Users use this P-value to screen for significance sites and further conduct enrichment analysis.
`cutoff`	The cutoff value of the P-value used to filter for further enrichment analysis. The default is 0.05.
`ont`	When choosing GO enrichment analysis, select the GO sub-ontology for which the enrichment analysis will be performed. One of "BP", "MF", and "CC" sub-ontologies, or "ALL" for all three. Default to "BP".
`pool`	If ont='ALL', whether pool three GO sub-ontologies.
`plot`	Whether the results of enrichment analysis need to be visualized, the default is TRUE
`plotType`	Whether to draw a bar plot ("bar") or a dot plot ("dot"), the default is "dot".
`plotcolor`	It is the vertical axis of the picture of the enrichment analysis results. Users can choose "pvalue" or "p.adjust" or "qvalue". The default is "p.adjust".
`x`	Character string specifying the variable to be used on the x-axis of the plot. Common options are "GeneRatio" or "Count". • "GeneRatio": ratio of input genes annotated to a given term. • "Count": the number of input genes annotated to the term.
`showCategory`	The number of categories which will be displayed in the plots. Default to 10.
`width`	Width of the PDF output in inches. Default is 11.
`height`	Height of the PDF output in inches. Default is 7.
`pvalueCutoff`	The p-value threshold used to filter enrichment results. Only results that pass the p-value test (i.e., those smaller than this value) will be reported. This value refers to the p-value before adjustment. The p-value represents the probability of observing the current level of enrichment under the assumption of no enrichment. The smaller the p-value, the more significant the enrichment result.
`pAdjustMethod`	The p-value adjustment method used for multiple hypothesis testing, aimed at reducing false positives caused by multiple comparisons. One of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none".
`qvalueCutoff`	qvalue cutoff on enrichment tests to report as significant. The q-value is the result of controlling the false discovery rate (FDR) and represents the proportion of false positives that may occur when conducting multiple tests.Tests must pass i) pvalueCutoff on unadjusted pvalues, ii) pvalueCutoff on adjusted pvalues and iii) qvalueCutoff on qvalues to be reported. The default is 0.2.

Value

input, An R6 class object integrating all information.

Examples

```
res <- initEWAS(export = FALSE)
res <- loadEWAS(input = res, ExpoData = "default", MethyData = "default")
res <- transEWAS(input = res, Vars = "cov1", TypeTo = "factor")
if (requireNamespace("clusterProfiler", quietly = TRUE) &&
    requireNamespace("org.Hs.eg.db", quietly = TRUE)) {
  res <- startEWAS(
    input = res, chipType = NULL, model = "lm",
    expo = "default", adjustP = TRUE, core = 1
  )
  res$result$gene <- "BRCA1"
  res <- enrichEWAS(
    input = res, method = "GO", filterP = "PVAL",
    cutoff = 1, pAdjustMethod = "BH", plot = FALSE
  )
}
```

`ewasfun_lm` *EWAS Model Computation Utilities*

Description

Model fitting functions for EWAS analysis using linear, mixed, or Cox models. These are designed to be used inside parallel loops with minimal memory footprint.

Usage

```
ewasfun_lm(cg, ff, cov, facnum)

ewasfun_lmer(cg, ff, cov, facnum)

ewasfun_cox(cg, ff, cov)
```

Arguments

`cg`	A row vector representing one CpG site's beta values across samples.
`ff`	A model formula object (e.g., cpg ~ var1 + var2).
`cov`	A data.frame containing the covariates for the model.
`facnum`	(Only for lm/lmer) Number of factor levels for exposure variable.

Value

A numeric vector with model coefficients, standard errors, and p-values.

initEWAS *Initialize the EWAS module*

Description

This function generates an R6 class for storing EWAS analysis data and results. By default, results are kept in memory and no files are written. If file export is requested, users must supply an explicit output directory.

Usage

```
initEWAS(outpath = NULL, export = FALSE)
```

Arguments

`outpath`	Optional path to an existing directory. When `export = TRUE`, a subdirectory named `"EWASresult"` is created under `outpath` for exported result files.
`export`	Logical. If `TRUE`, create an output folder and export result files. If `FALSE` (default), do not write files to disk; keep results in the returned object.

Value

input, an R6 class object integrating all information.

Examples

```
res <- initEWAS(export = FALSE)
res
```

loadEWAS *Load all data files for EWAS module*

Description

Upload sample data and methylation data for EWAS analysis.

Usage

```
loadEWAS(input, ExpoPath = NULL, MethyPath = NULL, ExpoData = "default",
MethyData = "default")
```

Arguments

`input`	An R6 class integrated with all the information obtained from the initEWAS function.
`ExpoPath`	The path to store the user's sample data. Each row represents a sample, and each column represents a variable (exposure variable or covariate). Both .csv and .xlsx file types are supported. The first column must be the sample ID, which must be consistent with the IDs in the methylation data.
`MethyPath`	The path to store the user's methylation data. Each row represents a CpG site, and each column represents a sample. Both .csv and .xlsx file types are supported. The first column must be the CpG probes. The sample IDs must be consistent with the IDs in the sample data.
`ExpoData`	The data.frame of the user-supplied sample data that has been loaded into the R environment. If default, the example data inside the package is used. The first column must be the sample name.
`MethyData`	The data.frame of the user-supplied methylation data that has been loaded into the R environment. If default, an example of methylation data inside the package is loaded. The first column must be the CpG site name.

Value

input, an R6 class object integrating all information.

Examples

```
res <- initEWAS(export = FALSE)
res <- loadEWAS(input = res, ExpoData = "default", MethyData = "default")
dim(res$Data$Expo)
dim(res$Data$Methy)
```

`methydata` *Example Methylation Matrix*

Description

Example CpG methylation matrix used by `easyEWAS`.

Usage

```
methydata
```

Format

A data frame with 1232 rows and 101 columns containing CpG probe identifiers and sample-level methylation values.

Source

Simulated example data bundled with the package.

plotEWAS *Visualize the results of EWAS analysis*

Description

Visualize EWAS results based on the CMplot package, including Manhattan plots, QQ plots, etc. Please note that this function only supports plotting a single-layer circular Manhattan plot. Additionally, the meaning of each parameter in this function is exactly the same as in CMplot For more detailed information or to create multi-layer circular Manhattan plots, please refer to CMplot (https://cran.r-project.org/web/packages/CMplot/index.html).

Usage

```
plotEWAS(
  input,
  p = "PVAL",
  threshold = NULL,
  file = c("jpg", "pdf", "tiff", "png"),
 col = c("#4197d8", "#f8c120", "#413496", "#495226", "#d60b6f", "#e66519", "#d581b7",
    "#83d3ad", "#7c162c", "#26755d"),
  bin.size = 1e+06,
  bin.breaks = NULL,
  LOG10 = TRUE,
  pch = 19,
  type = "p",
  band = 1,
  H = 1.5,
  ylim = NULL,
  axis.cex = 1,
  axis.lwd = 1.5,
  lab.cex = 1.5,
  lab.font = 2,
  plot.type = c("m", "c", "q", "d"),
  multracks = FALSE,
  multracks.xaxis = FALSE,
  multraits = FALSE,
  points.alpha = 100L,
  r = 0.3,
  cex = c(0.5, 1, 1),
  outward = FALSE,
  ylab = expression(-log[10](italic(p))),
  ylab.pos = 3,
  xticks.pos = 1,
  mar = c(3, 6, 3, 3),
  mar.between = 0,
  threshold.col = "red",
  threshold.lwd = 1,
```

```
  threshold.lty = 2,
  amplify = FALSE,
  signal.cex = 1.5,
  signal.pch = 19,
  signal.col = NULL,
  signal.line = 2,
  highlight = NULL,
  highlight.cex = 1,
  highlight.pch = 19,
  highlight.type = "p",
  highlight.col = "red",
  highlight.text = NULL,
  highlight.text.col = "black",
  highlight.text.cex = 1,
  highlight.text.font = 3,
  chr.labels = NULL,
  chr.border = FALSE,
  chr.labels.angle = 0,
  chr.den.col = "black",
  chr.pos.max = FALSE,
  cir.band = 1,
  cir.chr = TRUE,
  cir.chr.h = 1.5,
  cir.axis = TRUE,
  cir.axis.col = "black",
  cir.axis.grid = TRUE,
  conf.int = TRUE,
  conf.int.col = NULL,
  file.output = TRUE,
  file.name = "",
  dpi = 300,
  height = NULL,
  width = NULL,
  main = "",
  main.cex = 1.5,
  main.font = 2,
  legend.ncol = NULL,
  legend.cex = 1,
  legend.pos = c("left", "middle", "right"),
  box = FALSE,
  verbose = FALSE
)
```

Arguments

input	An R6 class integrated with all the information obtained from the startEWAS function.
p	The user needs to specify the name of the p value selected for the result visual-

ization.

threshold — The significant threshold.If threshold = 0 or NULL, then the threshold line will not be added.

file — The format of the output image file, including "jpg","pdf","tiff", and "png".

col — A vector specifies the colors for the chromosomes. If the length of col is shorter than the number of chromosomes, the colors will be applied cyclically.

bin.size — a integer, the size of bin in bp for marker density plot.

bin.breaks — a vector, set the breaks for the legend of density plot, e.g., seq(min, max, step), the windows in which the number of markers is out of the this range will be plotted in the same colors with the min or max value.

LOG10 — logical, whether to change the p-value into log10(p-value) scale.

pch — a integer, the shape for the points, is the same with "pch" in plot.

type — a character, could be "p" (point), "l" (cross line), "h" (vertical lines) and so on, is the same with "type" in plot.

band — a number, the size of space between chromosomes, the default is 1.

H — A number controlling the height of the circular Manhattan track.

ylim — vector (c(min, max)), CMplot will only plot the points among this interval.

axis.cex — a number, controls the size of ticks labels of X/Y-axis and the ticks labels of axis for circle plot.

axis.lwd — a number, controls the thickness of X/Y-axis lines and the thickness of axis for circle plot.

lab.cex — a number, controls the size of labels of X/Y-axis and the labels of chromosomes for circle plot.

lab.font — a number, controls the font of labels of all axis.

plot.type — a character or vector, only "d", "c", "m", "q" can be used. if plot.type="d", CpG density will be plotted; if plot.type="c", only circle-Manhattan plot will be plotted; if plot.type="m",only Manhattan plot will be plotted; if plot.type="q",only Q-Q plot will be plotted; if plot.type=c("m","q"), Both Manhattan and Q-Q plots will be plotted.

multracks — Logical. Whether to use multi-track mode in CMplot.

multracks.xaxis — Logical. Whether to draw x-axis in multi-track mode.

multraits — Logical. Whether to use multi-trait mode in CMplot.

points.alpha — Integer. Transparency (alpha) value for points (0–255). Default is 100.

r — a number, the radius for the circle (the inside radius), the default is 1.

cex — a number or a vector, the size for the points, is the same with "size" in plot, and if it is a vector, the first number controls the size of points in circle plot(the default is 0.5), the second number controls the size of points in Manhattan plot (the default is 1), the third number controls the size of points in Q-Q plot (the default is 1)

outward — logical, if TRUE, all points will be plotted from inside to outside for circular Manhattan plot.

`ylab` a character, the labels for y axis.

`ylab.pos` the distance between ylab and yaxis.

`xticks.pos` the distance between labels of x ticks and x axis.

`mar` the size of white gaps around the plot, 4 values should be provided, indicating the direction of bottom, left, up, and right.

`mar.between` Space between tracks for multi-track plotting.

`threshold.col` a character or vector, the color for the line of threshold levels, it can also control the color of the diagonal line of QQplot.

`threshold.lwd` a number or vector, the width for the line of threshold levels, it can also control the thickness of the diagonal line of QQplot.

`threshold.lty` a number or vector, the type for the line of threshold levels, it can also control the type of the diagonal line of QQplot

`amplify` logical, CMplot can amplify the significant points, if TRUE, then the points bigger than the minimal significant level will be amplified, the default: amplify=TRUE.

`signal.cex` a number, if amplify=TRUE, users can set the size of significant points.

`signal.pch` a number, if amplify=TRUE, users can set the shape of significant points.

`signal.col` a character, if amplify=TRUE, users can set the colour of significant points, if signal.col=NULL, then the colors of significant points will not be changed.

`signal.line` a number, the thickness of the lines of significant CpGs cross the circle.

`highlight` a vector, names of CpGs which need to be highlighted.

`highlight.cex` a vector, the size of points for CpGs which need to be highlighted.

`highlight.pch` a vector, the pch of points for CpGs which need to be highlighted.

`highlight.type` a vector, the type of points for CpGs which need to be highlighted.

`highlight.col` a vector, the col of points for CpGs which need to be highlighted.

`highlight.text` a vector, the text which would be added around the highlighted CpGs.

`highlight.text.col`
a vector, the color for added text.

`highlight.text.cex`
a value, the size for added text.

`highlight.text.font`
text font for the highlighted CpGs

`chr.labels` a vector, the labels for the chromosomes of density plot and Manhattan plot.

`chr.border` a logical, whether to plot the dot line between chromosomes.

`chr.labels.angle`
a value, rotate tick labels of x-axis for Manhattan plot (-90 < chr.labels.angle < 90).

`chr.den.col` a character or vector or NULL, the colour for the CpG density. If the length of parameter 'chr.den.col' is bigger than 1, CpG density that counts the number of CpG within given size ('bin.size') will be plotted around the circle. If chr.den.col=NULL, the density bar will not be attached on the bottom of manhattan plot.

chr.pos.max	logical, whether the physical positions of each chromosome contain the maximum length of the chromosome.
cir.band	A number controlling the spacing between circular tracks.
cir.chr	logical, a boundary that represents chromosomes will be plotted on the periphery of a circle, the default is TRUE.
cir.chr.h	a number, the width for the boundary, if cir.chr=FALSE, then this parameter will be useless.
cir.axis	a logical, whether to add the axis of circle Manhattan plot.
cir.axis.col	a character, the color of the axis for circle.
cir.axis.grid	logical, whether to add axis grid line in circles.
conf.int	logical, whether to plot confidence interval on QQ-plot.
conf.int.col	character or vector, the color of confidence interval of QQplot.
file.output	a logical, users can choose whether to output the plot results.
file.name	a character or vector, the names of output files.
dpi	a number, the picture resolution for '.jpg', '.npg', and '.tiff' files. The default is 300.
height	the height of output files.
width	the width of output files.
main	character of vector, the title of the plot for manhattan plot and qqplot.
main.cex	size of title.
main.font	font of title.
legend.ncol	Number of columns used in the legend.
legend.cex	A numeric value controlling legend text size.
legend.pos	A character value specifying legend position.
box	logical, this function draws a box around the current plot.
verbose	whether to print the log information.

Value

The updated input object, including CMplot-ready data stored in `input$CMplot` and plotting arguments stored in `input$plot_args`. When files are not exported, a replayable plot may also be stored in `input$plot_record` for the current R session.

Examples

```
res <- initEWAS(export = FALSE)
res$result <- data.frame(
  probe = paste0("cg", seq_len(50)),
  chr = rep(c("1", "2"), each = 25),
  pos = seq_len(50) * 1000,
  PVAL = seq(0.001, 0.05, length.out = 50)
)
res <- plotEWAS(input = res, p = "PVAL", plot.type = "m", file.output = FALSE)
```

sampledata — *Example Sample Metadata*

Description

Example phenotype and covariate data used by `easyEWAS`.

Usage

```
sampledata
```

Format

A data frame with 100 rows and 5 variables:

SampleName Sample identifier.

var Example exposure variable.

cov1 Example covariate 1.

cov2 Example covariate 2.

batch Example batch variable.

Source

Simulated example data bundled with the package.

startEWAS — *Perform EWAS Analysis*

Description

Perform EWAS analysis to obtain the coefficient value, standard deviation and significance p value (or adjust p value) of each site.

Usage

```
startEWAS(
  input,
  filename = "default",
  model = "lm",
  expo = NULL,
  cov = NULL,
  random = NULL,
  time = NULL,
  status = NULL,
  adjustP = TRUE,
```

```
  chipType = "EPICV2",
  core = "default",
  annotation_cache = NULL,
  auto_download_annotation = FALSE,
  annotation_base_url = getOption("easyEWAS.annotation_base_url",
    "https://github.com/ytwangZero/easyEWAS_materials/raw/main/annotation")
)
```

Arguments

`input` An R6 class integrated with all the information obtained from the loadEWAS or transEWAS function.

`filename` filename Name of the output CSV file to store EWAS results. If set to "default", the file will be named "ewasresult.csv" and saved in the specified output directory.

`model` Statistical model to use for EWAS analysis. Options include:

- "lm": Linear regression (default)
- "lmer": Linear mixed-effects model
- "cox": Cox proportional hazards model

`expo` Name of the exposure variable used in the EWAS analysis.

`cov` Comma-separated list of covariate variable names to include in the model (e.g., "age,sex,bmi"). Do not include spaces between names. Optional.

`random` Name of the grouping variable for the random intercept, required only when using the "lmer" model.

`time` Name of the time-to-event variable, required only when using the "cox" model.

`status` Name of the event/censoring indicator variable, required only when using the "cox" model.

`adjustP` Logical. If TRUE (default), adjusts p-values using both FDR (Benjamini-Hochberg) and Bonferroni correction methods.

`chipType` Illumina array platform used for DNA methylation measurement. Available options:

- "27K"
- "450K"
- "EPICV1"
- "EPICV2" (default)
- "MSA"
- NULL (skip annotation)

`core` Number of CPU cores to use for parallel processing. If set to "default", uses the number of available physical cores minus one.

`annotation_cache` Local directory for cached annotation files. If NULL, uses `tools::R_user_dir("easyEWAS", "cache")/annotation`.

`auto_download_annotation` Logical. If TRUE, annotation files are downloaded automatically when missing.

`annotation_base_url` Base URL for annotation `.rds` files.

Value

input, An R6 class object integrating all information.

Examples

```
res <- initEWAS(export = FALSE)
res <- loadEWAS(input = res, ExpoData = "default", MethyData = "default")
res <- transEWAS(input = res, Vars = "cov1", TypeTo = "factor")
res$Data$Methy <- res$Data$Methy[seq_len(50), , drop = FALSE]
res <- startEWAS(
  input = res,
  model = "lm",
  expo = "var",
  cov = "cov1,cov2",
  chipType = NULL,
  core = 1
)
head(res$result)
```

---

transEWAS *Convert variable type of sample data*

Description

Transform the variable types of sample data to the types specified by users.

Usage

```
transEWAS(input, Vars = "default", TypeTo = "factor")
```

Arguments

input	An R6 class integrated with all the information obtained from the loadEWAS function.
Vars	Variable names that the user wants to convert types for, with each variable name separated by a comma. Ensure there are no spaces. e.g. "var1,var2,var3".
TypeTo	The type of variable that the function allows to be converted, including numeric and factor.

Value

input, An R6 class object integrating all information.

Examples

```
res <- initEWAS(export = FALSE)
res <- loadEWAS(input = res, ExpoData = "default", MethyData = "default")
res <- transEWAS(input = res, Vars = "cov1", TypeTo = "factor")
class(res$Data$Expo$cov1)
```

Index

∗ **datasets**
    methydata, 11
    sampledata, 17

batchEWAS, 2
bootEWAS, 3

CMplot, *12*
ComBat, *2*

dmrEWAS, 4
downloadAnnotEWAS, 6

enrichEWAS, 7
ewasfun_cox *(*ewasfun_lm*)*, 9
ewasfun_lm, 9
ewasfun_lmer *(*ewasfun_lm*)*, 9

initEWAS, 10

loadEWAS, 10

methydata, 11

plot, *14*
plotEWAS, 12

sampledata, 17
startEWAS, 17

transEWAS, 19