Package: dnaEPICO (via r-universe)

November 25, 2026

Title dnaEPICO: Analysis Pipeline for Illumina DNA Methylation Array Data

Version 1.0.0

Date 2026-08-18

Description A modular and reproducible workflow for preprocessing and analysing Illumina DNA methylation array data from the EPICv2, EPIC, and 450K platforms. It supports quality control, probe filtering, cell-type deconvolution, phenotype preparation, CpG-wise generalised linear models with optional term-level omnibus F tests, longitudinal mixed-effects models using 'lmerTest'/'lme4' or 'nlme', optional 'nlme' residual correlation structures, and automated reports. The workflow uses established Bioconductor tools, including 'minfi', 'ENmix', and 'wateRmelon', and runs locally or in high-performance computing environments.

License AGPL-3 + file LICENSE

URL https://github.com/paulYRP/dnaEPICO

BugReports https://github.com/paulYRP/dnaEPICO/issues

Depends R (>= 4.4.0)

Imports base64enc, AnnotationHub, Biobase, BiocGenerics, car, RColorBrewer, data.table, ggplot2, grid, methods, minfi, openxlsx, limma, wateRmelon, ENmix, ggrepel, glm2, GenomeInfoDb, GenomicRanges, IRanges, parallel, parallelly, lme4, lmerTest, nlme, quadprog, S4Vectors, SummarizedExperiment, utils, stats

Suggests IlluminaHumanMethylation450kmanifest, IlluminaHumanMethylation450kanno.ilmn12.hg19, IlluminaHumanMethylationEPICv2anno.20a1.hg38, IlluminaHumanMethylationEPICv2manifest, IlluminaHumanMethylationEPICmanifest, IlluminaHumanMethylationEPICanno.ilm10b4.hg19, minfiData, remotes, BiocManager, BiocStyle, RefManageR, sessioninfo, devtools, testthat, ggVennDiagram, tiff, magick, knitr,

rmarkdown, covr, withr, tinytex, FlowSorted.Blood.EPIC, FlowSorted.Blood.450k, BeadSorted.Saliva.EPIC, pbkrtest

VignetteBuilder knitr

SystemRequirements Quarto command-line interface, GNU make (>= 4.3)

biocViews Software, Preprocessing, MethylationArray, QualityControl, Epigenetics, Microarray, StatisticalMethod, ChipOnChip

Encoding UTF-8

Roxygen list(markdown = TRUE)

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

Config/pak/sysreqs cmake make libbz2-dev libicu-dev liblzma-dev libpng-dev libxml2-dev libssl-dev libx11-dev xz-utils zlib1g-dev

Repository Bioconductor 3.24

Date/Publication 2026-11-23 22:56:07 UTC

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

RemoteRef RELEASE_3_24

RemoteSha 68f5f35eaa527998fc297576a2e80265e9a4199c

Contents

analyzeSvaEnmix . . . 4
annotateMethylationGLMSummaries . . . 5
annotateMethylationLMESummaries . . . 6
assessSamplesMinfiEwasWater . . . 7
buildClockFoundationInputsPreprocessingPheno . . . 8
buildRawMinfiEwasWater . . . 10
collectSignificantCpGsMethylationGLM . . . 11
collectSignificantInteractionsMethylationLME . . . 12
combineTimepointsPreprocessingPheno . . . 13
dnaEPICO . . . 14
dnaEPICO_dnamReport-class . . . 15
dnaEPICO_dnamReport_prepared-class . . . 16
dnaEPICO_dnamReport_render-class . . . 16
dnaEPICO_methylationGLM-class . . . 17
dnaEPICO_methylationLME-class . . . 17
dnaEPICO_preprocessingMinfiEwasWater-class . . . 18
dnaEPICO_preprocessingPheno-class . . . 19
dnaEPICO_svaEnmix-class . . . 19
dnamReport . . . 20
estimateLC . . . 23
estimateLCMinfiEwasWater . . . 24
estimateSvaEnmixControls . . . 26
extractMake . . . 27
extractMetricsMinfiEwasWater . . . 28

filterProbesMinfiEwasWater 29
filterSamplesMinfiEwasWater 31
fitMethylationGLMModels 32
fitMethylationLMEModels 33
loadMetricsPreprocessingPheno 35
mergeSvaTargetsEnmix 36
methylationGLM 37
methylationLME 42
normalizeMinfiEwasWater 48
plotAssessmentMinfiEwasWater 49
plotCtrlMinfiEwasWater 50
plotMethylationGLMDiagnostics 51
plotMethylationGLMDistributions 53
plotMethylationLMEDiagnostics 54
plotMetricsMinfiEwasWater 55
plotNormalizationMinfiEwasWater 57
plotRawDensityMinfiEwasWater 58
plotSexMinfiEwasWater 59
plotSvaEnmix 61
predictSexMinfiEwasWater 62
prepareDnamReportInputs 63
prepareMethylationGLMData 65
prepareMethylationLMEData 66
preprocessingMinfiEwasWater 68
preprocessingPheno 72
print.dnaEPICO_dnamReport 76
readPhenotypeTargets 76
readRGSetMinfiEwasWater 77
renderDnamReport 79
splitTimepointsPreprocessingPheno 80
summarizeMethylationGLMModels 81
summarizeMethylationLMEModels 83
summarizeTimepointsMethylationLME 84
svaEnmix 85
writeMethylationGLMOutputs 88
writeMethylationLMEOutputs 90
writePhenoLCMinfiEwasWater 92
writePreprocessingPhenoOutputs 93
writeSvaEnmixOutputs 94

Index **97**

`analyzeSvaEnmix` *Analyze surrogate variables against Sentrix chip and position factors*

Description

Fit linear models for each surrogate variable against informative Sentrix chip and position factors, perform backward elimination with `stats::drop1()`, and return the in-memory analysis objects. A constant technical factor is omitted; if both are constant, an intercept-only model is retained.

Usage

```
analyzeSvaEnmix(
  sva,
  RGSet,
  SentrixIDColumn = "Sentrix_ID",
  SentrixPositionColumn = "Sentrix_Position",
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_analyzeSvaEnmix.txt"
)
```

Arguments

`sva`	Numeric matrix of surrogate variables with samples in rows.
`RGSet`	An `RGChannelSet` aligned with `sva`.
`SentrixIDColumn`	Character. Name of the chip identifier column in `SummarizedExperiment::colData(RGSet)`.
`SentrixPositionColumn`	Character. Name of the chip position column in `SummarizedExperiment::colData(RGSet)`.
`verbose`	Logical. If `TRUE`, emit progress messages with `message()`.
`logs`	Logical. If `TRUE`, write the same messages to a log file.
`log_dir`	Character or `NULL`. Directory used for the log file when `logs = TRUE`.
`log_file`	Character. File name used when `logs = TRUE`.

Value

A list with class `'dnaEPICO_svaEnmix_analysis'` containing the aligned Sentrix factors, full and reduced linear models, and ANOVA tables.

Examples

```
ex <- dnaEPICO:::exampleSvaAnalysisStateDnaEpico()
analysis_data <- analyzeSvaEnmix(
  sva = ex$sva,
  RGSet = ex$RGSet,
```

```
  SentrixIDColumn = "Sentrix_ID",
  SentrixPositionColumn = "Sentrix_Position",
  verbose = FALSE,
  logs = FALSE
)
analysis_data$K
```

`annotateMethylationGLMSummaries` *Annotate one-timepoint GLM summary tables with array annotation metadata*

Description

Merge phenotype-specific CpG summary tables with probe annotation metadata and return a single annotated result table.

Usage

```
annotateMethylationGLMSummaries(
  modelSummaries,
  annotationObject,
 annotationCols = c("Name", "chr", "pos", "UCSC_RefGene_Group", "UCSC_RefGene_Name",
    "Relation_to_Island", "GencodeV41_Group"),
  gencodeHub = FALSE,
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_methylationGLM.txt"
)
```

Arguments

`modelSummaries`	Object returned by `summarizeMethylationGLMModels()` or a named list of CpG summary data frames.
`annotationObject`	Character package/object name, annotation data frame, or annotation object understood by `minfi::getAnnotation()`.
`annotationCols`	Character vector or comma-separated string of annotation columns to append.
`gencodeHub`	Logical. If `TRUE`, retrieve the package-managed GENCODE gene resource from AnnotationHub and append direct gene-body and nearest transcription-start-site annotations. This requires GRCh38 coordinates.
`verbose`	Logical. If `TRUE`, emit progress messages with `message()`.
`logs`	Logical. If `TRUE`, write the same messages to a log file.
`log_dir`	Character or `NULL`. Directory used for the log file when `logs = TRUE`.
`log_file`	Character. File name used when `logs = TRUE`.

Value

A list with class 'dnaEPICO_methylationGLM_annotation' containing the annotated summary table and any requested annotation columns that were unavailable in the chosen annotation object.

Examples

```
ex <- dnaEPICO:::exampleMethylationGLMStateDnaEpico()
annotation_data <- annotateMethylationGLMSummaries(
  modelSummaries = ex$modelSummaries,
  annotationObject = ex$annotationData,
  annotationCols = "Name,chr,pos",
  verbose = FALSE,
  logs = FALSE
)
names(annotation_data)
```

---

annotateMethylationLMESummaries

Annotate longitudinal mixed-effects summary tables with array annotation metadata

---

Description

Merge phenotype-specific longitudinal summary tables with probe annotation metadata and return a single annotated result table.

Usage

```
annotateMethylationLMESummaries(
  modelSummaries,
  annotationObject,
  annotationCols = c("Name", "chr", "pos", "UCSC_RefGene_Group", "UCSC_RefGene_Name",
    "Relation_to_Island", "GencodeV41_Group"),
  gencodeHub = FALSE,
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_methylationLME.txt"
)
```

Arguments

modelSummaries	Object returned by summarizeMethylationLMEModels() or a named list of summary data frames.
annotationObject	Character package/object name, annotation data frame, or annotation object understood by minfi::getAnnotation().

annotationCols	Character vector or comma-separated string of annotation columns to append.
gencodeHub	Logical. If TRUE, retrieve the package-managed GENCODE gene resource from AnnotationHub and append direct gene-body and nearest transcription-start-site annotations. This requires GRCh38 coordinates.
verbose	Logical. If TRUE, emit progress messages with message().
logs	Logical. If TRUE, write the same messages to a log file.
log_dir	Character or NULL. Directory used for the log file when logs = TRUE.
log_file	Character. File name used when logs = TRUE.

Value

A list with class 'dnaEPICO_methylationLME_annotation' containing the annotated summary table and any requested annotation columns that were unavailable in the chosen annotation object.

Examples

```
ex <- dnaEPICO:::exampleMethylationLMEStateDnaEpico()
annotation_data <- annotateMethylationLMESummaries(
  modelSummaries = ex$modelSummaries,
  annotationObject = ex$annotationData,
  annotationCols = "Name,chr,pos",
  verbose = FALSE,
  logs = FALSE
)
names(annotation_data)
```

---

assessSamplesMinfiEwasWater

Assess sample quality before sample filtering

Description

Compute minfi QC metrics and detection P values, identify failed samples using the requested threshold, and return the assessment as a single object.

Usage

```
assessSamplesMinfiEwasWater(
  rawData,
  RGSet,
  qcCutoff = 10.5,
  detPtype = "m+u",
  detPThreshold = 0.05,
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_assessSamplesMinfiEwasWater.txt"
)
```

Arguments

rawData	Object returned by buildRawMinfiEwasWater().
RGSet	An RGChannelSet aligned with rawData.
qcCutoff	Numeric. Cutoff passed to minfi::plotQC() when the QC plot is drawn.
detPtype	Character. Detection P-value mode passed to minfi::detectionP(). Common values in minfi workflows are 'm+u' and 'negative'. The default used here is 'm+u'.
detPThreshold	Numeric. Samples with mean detection P value above this threshold are marked as failed.
verbose	Logical. If TRUE, emit progress messages with message().
logs	Logical. If TRUE, write the same messages to a log file.
log_dir	Character or NULL. Directory used for the log file when logs = TRUE.
log_file	Character. File name used when logs = TRUE.

Value

A list with class 'dnaEPICO_minfiEwasWater_assessment' containing the QC object, detection P matrix, mean detection P values, and failed sample identifiers.

Examples

```
ex <- dnaEPICO:::exampleMinfiBaseDataDnaEpico()
raw_data <- buildRawMinfiEwasWater(ex$RGSet, verbose = FALSE, logs = FALSE)
assessment <- assessSamplesMinfiEwasWater(
  rawData = raw_data,
  RGSet = ex$RGSet,
  detPThreshold = 1,
  verbose = FALSE,
  logs = FALSE
)
names(assessment)
```

---

buildClockFoundationInputsPreprocessingPheno

Build Clock Foundation input tables from preprocessingPheno data

---

Description

Prepare the beta and phenotype tables commonly exported for Clock Foundation style downstream workflows, without writing them to disk. The sex column is passed through without validation or recoding.

Usage

```
buildClockFoundationInputsPreprocessingPheno(
  beta,
  pheno,
  SampleID = "Sample_Name",
  sexColumn = "Sex",
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_buildClockFoundationInputsPreprocessingPheno.txt"
)
```

Arguments

beta	Numeric matrix of beta values with probes in rows and samples in columns.
pheno	Phenotype data frame aligned with the beta matrix columns.
SampleID	Character. Name of the phenotype sample identifier column.
sexColumn	Character. Name of the phenotype sex column. Its values are preserved as supplied, including missing, blank, unknown, or other strings; PredSex is not substituted.
verbose	Logical. If TRUE, emit progress messages with message().
logs	Logical. If TRUE, write the same messages to a log file.
log_dir	Character or NULL. Directory used for the log file when logs = TRUE.
log_file	Character. File name used when logs = TRUE.

Value

A list with class 'dnaEPICO_preprocessingPheno_clock' containing betaCSV and phenoCF.

Examples

```
ex <- dnaEPICO:::examplePreprocessingPhenoStateDnaEpico()
clock_inputs <- buildClockFoundationInputsPreprocessingPheno(
  beta = ex$timepointData$data[["1"]]$beta,
  pheno = ex$timepointData$data[["1"]]$pheno,
  SampleID = "Sample_Name",
  sexColumn = "Sex",
  verbose = FALSE,
  logs = FALSE
)
names(clock_inputs)
```

buildRawMinfiEwasWater *Build raw minfi preprocessing objects*

Description

Create a raw MethylSet, RatioSet, and genome-mapped object from an RGChannelSet, and return them together in a single structured object.

Usage

```
buildRawMinfiEwasWater(
  RGSet,
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_buildRawMinfiEwasWater.txt"
)
```

Arguments

RGSet	An RGChannelSet.
verbose	Logical. If TRUE, emit progress messages with message().
logs	Logical. If TRUE, write the same messages to a log file.
log_dir	Character or NULL. Directory used for the log file when logs = TRUE.
log_file	Character. File name used when logs = TRUE.

Value

A list with class 'dnaEPICO_minfiEwasWater_raw' containing MSet, RatioSet, and GSet.

Examples

```
ex <- dnaEPICO:::exampleMinfiBaseDataDnaEpico()
raw_data <- buildRawMinfiEwasWater(
  RGSet = ex$RGSet,
  verbose = FALSE,
  logs = FALSE
)
names(raw_data)
```

`collectSignificantCpGsMethylationGLM` *Collect significant CpG coefficient tables from fitted one-timepoint GLMs*

Description

Collect coefficient tables for CpGs selected by the configured target test. Selection uses target-term omnibus p-values when omnibus testing was enabled during fitting, and phenotype main-effect or interaction coefficient p-values otherwise.

Usage

```
collectSignificantCpGsMethylationGLM(
  modelResults,
  pvalThreshold = 0.05,
  interactionTerm = NULL,
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_methylationGLM.txt"
)
```

Arguments

`modelResults`	Object returned by `fitMethylationGLMModels()`.
`pvalThreshold`	Numeric. Threshold applied to omnibus p-values when omnibus testing was enabled during model fitting, or to phenotype main-effect or interaction coefficient p-values otherwise.
`interactionTerm`	Character or `NULL`. Optional interaction term.
`verbose`	Logical. If `TRUE`, emit progress messages with `message()`.
`logs`	Logical. If `TRUE`, write the same messages to a log file.
`log_dir`	Character or `NULL`. Directory used for the log file when `logs = TRUE`.
`log_file`	Character. File name used when `logs = TRUE`.

Value

A list with class `'dnaEPICO_methylationGLM_significant_cpgs'`.

Examples

```
ex <- dnaEPICO:::exampleMethylationGLMStateDnaEpico()
significant_cpgs <- collectSignificantCpGsMethylationGLM(
  modelResults = ex$modelResults,
  pvalThreshold = 1,
```

```
  verbose = FALSE,
  logs = FALSE
)
names(significant_cpgs)
```

`collectSignificantInteractionsMethylationLME` *Collect significant longitudinal terms from compact mixed-effects results*

Description

Collect raw coefficient tables for CpGs whose phenotype main effect or requested interaction p-value passes the chosen threshold.

Usage

```
collectSignificantInteractionsMethylationLME(
  modelResults,
  pvalThreshold = 0.05,
  interactionTerm = NULL,
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_methylationLME.txt"
)
```

Arguments

`modelResults`	Object returned by `fitMethylationLMEModels()`.
`pvalThreshold`	Numeric. Threshold applied to the extracted phenotype or interaction p-values.
`interactionTerm`	Character or `NULL`. Optional interaction term. When `NULL`, phenotype main effects are used.
`verbose`	Logical. If `TRUE`, emit progress messages with `message()`.
`logs`	Logical. If `TRUE`, write the same messages to a log file.
`log_dir`	Character or `NULL`. Directory used for the log file when `logs = TRUE`.
`log_file`	Character. File name used when `logs = TRUE`.

Value

A list with class `'dnaEPICO_methylationLME_significant'` containing the retained coefficient tables for each phenotype.

Examples

```
ex <- dnaEPICO:::exampleMethylationLMEStateDnaEpico()
significant_hits <- collectSignificantInteractionsMethylationLME(
  modelResults = ex$modelResults,
  pvalThreshold = 1,
  verbose = FALSE,
  logs = FALSE
)
names(significant_hits)
```

---

combineTimepointsPreprocessingPheno

Combine selected timepoints for downstream longitudinal modeling

---

Description

Combine selected timepoints that were already aligned by `splitTimepointsPreprocessingPheno()` into the wide phenotype-plus-beta objects used by downstream longitudinal models.

Usage

```
combineTimepointsPreprocessingPheno(
  timepointData,
  combineTimepoints = "1,2",
  methylationScale = "beta",
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_combineTimepointsPreprocessingPheno.txt"
)
```

Arguments

`timepointData`	Object returned by `splitTimepointsPreprocessingPheno()`.
`combineTimepoints`	Character vector or comma-separated string of timepoints to combine.
`methylationScale`	Character. Methylation metric used in the merged modeling table. One of `'Beta'`, `'M'`, or `'CN'`, case-insensitively.
`verbose`	Logical. If `TRUE`, emit progress messages with `message()`.
`logs`	Logical. If `TRUE`, write the same messages to a log file.
`log_dir`	Character or `NULL`. Directory used for the log file when `logs = TRUE`.
`log_file`	Character. File name used when `logs = TRUE`.

Value

A list with class 'dnaEPICO_preprocessingPheno_combined' containing the combined phenotype table, merged phenotype-plus-beta table, selected timepoints, and output suffix.

Examples

```
ex <- dnaEPICO:::examplePreprocessingPhenoStateDnaEpico()
combined_data <- combineTimepointsPreprocessingPheno(
  timepointData = ex$timepointData,
  combineTimepoints = "1,2",
  verbose = FALSE,
  logs = FALSE
)
combined_data$suffix
```

---

dnaEPICO *dnaEPICO: DNA methylation preprocessing and modeling workflows*

Description

dnaEPICO provides workflows for preprocessing and analyzing Illumina DNA methylation arrays. It supports quality control, normalization, cell-type estimation, surrogate-variable analysis, phenotype preparation, CpG-wise generalised linear models, longitudinal mixed-effects models using lmerTest/lme4 or nlme, and web reporting.

Details

The package supports two usage styles:

- interactive use, where functions return structured in-memory result objects for inspection and composition; and
- file-based pipeline use, where the same functions can write logs, plots, tables, and saved R objects when saveOutputs = TRUE.

The main high-level entry points are:

- preprocessingMinfiEwasWater()
- svaEnmix()
- preprocessingPheno()
- methylationGLM()
- methylationLME()
- dnamReport()

Author(s)

Maintainer: Paul Ruiz Pinto <ruizpint@qut.edu.au> (ORCID)

Authors:

- Paul Ruiz Pinto <ruizpint@qut.edu.au> (ORCID)

See Also

Useful links:

- https://github.com/paulYRP/dnaEPICO
- Report bugs at https://github.com/paulYRP/dnaEPICO/issues

---

dnaEPICO_dnamReport-class

Result class returned by dnamReport

Description

Objects of class 'dnaEPICO_dnamReport' are list-based results returned by dnamReport(). They combine the prepared report inputs, render result, and final status metadata into one convenience object.

Structure

preparedReport Object returned by prepareDnamReportInputs().

renderResult Structured render metadata created by dnamReport().

status Final status string such as 'rendered', 'skipped', or 'failed'.

outputFile Path to docs/index.html.

resultTableSources Named character vector identifying whether each model table used a report data file read in batches or an XLSX workbook.

errorMessage Final render error or skip message when available, otherwise NULL.

logFile Resolved path to the optional log file, or NULL when logging was disabled.

See Also

dnamReport()

dnaEPICO_dnamReport_prepared-class

Result class returned by prepareDnamReportInputs

Description

Objects of class `'dnaEPICO_dnamReport_prepared'` are list-based results returned by `prepareDnamReportInputs()`. They capture normalized report paths, available figures, and logging metadata before rendering.

Structure

output Requested output file name.

outputDir Normalized report output directory.

outputFile Normalized full path to the intended report output file.

figDir Normalized directory used by the report template for copied figures.

figureInventory Named list describing the available figures for each report section.

missingFigureDirectories Character vector of expected figure directories that were not present at preparation time.

logFile Resolved path to the optional log file, or `NULL` when logging was disabled.

See Also

`prepareDnamReportInputs()`

dnaEPICO_dnamReport_render-class

Result class returned by renderDnamReport

Description

Objects of class `'dnaEPICO_dnamReport_render'` are list-based results returned by `renderDnamReport()`. They describe whether a prepared report was rendered, skipped, or failed.

Structure

preparedReport The input object supplied to `renderDnamReport()`.

status Render status string such as `'rendered'`, `'skipped'`, or `'failed'`.

renderedFile Normalized path to the rendered HTML file when rendering succeeded, otherwise `NULL`.

errorMessage Render error or skip message when available, otherwise `NULL`.

logFile Resolved path to the optional log file, or `NULL` when logging was disabled.

See Also

`renderDnamReport()`

dnaEPICO_methylationGLM-class

Result class returned by methylationGLM

Description

Objects of class 'dnaEPICO_methylationGLM' are list-based results returned by methylationGLM(). They collect the prepared analysis table, compact model results, summaries, diagnostics, annotations, and optional saved files.

Structure

preparedData Object returned by prepareMethylationGLMData().

distributionPlots Object returned by plotMethylationGLMDistributions().

modelFits Object returned by fitMethylationGLMModels().

modelSummaries Object returned by summarizeMethylationGLMModels().

significantCpGs Object returned by collectSignificantCpGsMethylationGLM().

diagnosticPlots Object returned by plotMethylationGLMDiagnostics().

annotation Object returned by annotateMethylationGLMSummaries().

manhattanPlots Versioned circular and rectangular Manhattan plots for every raw p-value column in the annotated results.

vennDPlots Requested coefficient and omnibus model-level Venn plots, worksheet tables, and label mappings.

savedFiles Object returned by writeMethylationGLMOutputs() when saveOutputs = TRUE, otherwise NULL.

runSettings High-level settings recorded for the analysis.

See Also

methylationGLM()

dnaEPICO_methylationLME-class

Result class returned by methylationLME

Description

Objects of class 'dnaEPICO_methylationLME' are list-based results returned by methylationLME(). They collect the prepared longitudinal analysis table, compact mixed-model results, summaries, diagnostics, annotations, and optional saved files.

Structure

preparedData Object returned by `prepareMethylationLMEData()`.

modelFits Object returned by `fitMethylationLMEModels()`.

modelSummaries Object returned by `summarizeMethylationLMEModels()`.

significantInteractions Object returned by `collectSignificantInteractionsMethylationLME()`.

diagnosticPlots Object returned by `plotMethylationLMEDiagnostics()`.

annotation Object returned by `annotateMethylationLMESummaries()`.

manhattanPlots Versioned circular and rectangular Manhattan plots for every raw p-value column in the annotated results.

vennDPlots Requested coefficient and omnibus model-level Venn plots, worksheet tables, and label mappings.

savedFiles Object returned by `writeMethylationLMEOutputs()` when `saveOutputs = TRUE`, otherwise `NULL`.

runSettings High-level settings recorded for the analysis.

See Also

`methylationLME()`

---

dnaEPICO_preprocessingMinfiEwasWater-class

Result class returned by preprocessingMinfiEwasWater

Description

Objects of class `'dnaEPICO_preprocessingMinfiEwasWater'` are list-based results returned by `preprocessingMinfiEwasWater()`. They are lightweight S3-style containers rather than formal S4 classes.

Structure

targets Filtered phenotype table aligned to the retained samples.

RGSet Filtered `RGChannelSet` used in downstream preprocessing.

rawData Object returned by `buildRawMinfiEwasWater()`.

assessment Object returned by `assessSamplesMinfiEwasWater()`.

sexData Object returned by `predictSexMinfiEwasWater()`, including predicted-sex values and normalization fallback provenance.

normData Object returned by `normalizeMinfiEwasWater()`, including the per-sample sex-resolution audit.

filterData Object returned by `filterProbesMinfiEwasWater()`.

metricsData Object returned by `extractMetricsMinfiEwasWater()`.

lcData Object returned by `estimateLCMinfiEwasWater()`.

logFile Resolved path to the optional log file, or `NULL` when logging was disabled.

See Also

preprocessingMinfiEwasWater()

---

dnaEPICO_preprocessingPheno-class
Result class returned by preprocessingPheno

Description

Objects of class 'dnaEPICO_preprocessingPheno' are list-based results returned by preprocessingPheno(). They describe the phenotype data, methylation matrices, timepoint splits, longitudinal merges, and optional exported files.

Structure

pheno Phenotype table read from phenoFile.

metricsData Object returned by loadMetricsPreprocessingPheno().

timepointData Object returned by splitTimepointsPreprocessingPheno().

combinedData Object returned by combineTimepointsPreprocessingPheno(), or NULL when the combined object was skipped.

clockFoundation Object returned by buildClockFoundationInputsPreprocessingPheno(), with original phenotype sex values preserved for export.

savedFiles Object returned by writePreprocessingPhenoOutputs() when saveOutputs = TRUE, otherwise NULL.

logFile Resolved path to the optional log file, or NULL when logging was disabled.

See Also

preprocessingPheno()

---

dnaEPICO_svaEnmix-class
Result class returned by svaEnmix

Description

Objects of class 'dnaEPICO_svaEnmix' are list-based results returned by svaEnmix(). They collect the loaded inputs, surrogate-variable results, association-analysis summaries, and optional file outputs.

Structure

targets Phenotype table read from `phenoFile` after any optional row subsetting.

RGSet Loaded `RGChannelSet` with sample names reset to match `targets`.

svaData Object returned by `estimateSvaEnmixControls()`.

mergedPheno Phenotype table returned by `mergeSvaTargetsEnmix()` after surrogate variables were appended.

analysisData Object returned by `analyzeSvaEnmix()`.

plotFiles Named list of TIFF output paths for the SVA figures when `saveOutputs = TRUE`, otherwise `NULL` entries for plots not written.

savedFiles Object returned by `writeSvaEnmixOutputs()` when `saveOutputs = TRUE`, otherwise `NULL`.

logFile Resolved path to the optional log file, or `NULL` when logging was disabled.

See Also

`svaEnmix()`

---

dnamReport *Generate an HTML report from saved pipeline outputs*

Description

Combine phenotype data, quality-control figures, annotated model results and logs in an HTML report. `modelSections` selects the GLM and LME pages; `character(0)` produces a preprocessing-only report. The returned object includes the report path and rendering status.

Usage

```
dnamReport(
  outputDir = "reports",
  phenoTab = NULL,
  enmixTab = file.path("figures", "preprocessingMinfiEwasWater", "enmix"),
  qcTab = file.path("figures", "preprocessingMinfiEwasWater", "qc"),
  svaTab = file.path("figures", "svaEnmix"),
  metricTab = file.path("figures", "preprocessingMinfiEwasWater", "metrics"),
  glmTab = NULL,
  lmeTab = NULL,
  modelSections = c("glm", "lme"),
  logTab = outputDir,
  verbose = FALSE,
  logs = FALSE,
  projectName = "dnaEPICO",
  detPPath = NULL,
  detPThreshold = 0.01,
```

```
  cpgDetectionPath = NULL,
  sampleDetectionPath = NULL,
  logoPath = system.file("extdata", "dnaEPICORM.svg", package = "dnaEPICO"),
  imagePattern = "\\.(png|jpg|jpeg|gif|webp|svg|tif|tiff)$",
  recursive = TRUE
)
```

Arguments

outputDir	Character. Directory where the Quarto project is written.
phenoTab	Character or NULL. CSV file shown in the Data tab. When NULL, the path is inferred from the Makefile output layout.
enmixTab	Character. Directory containing ENmix quality-control figures.
qcTab	Character. Directory containing Quality Control figures.
svaTab	Character. Directory containing Batch Effect or SVA figures.
metricTab	Character. Directory containing Metrics figures.
glmTab	Character or NULL. XLSX workbook shown in the GLM Analysis tab. When NULL, the path is inferred from the Makefile output layout. Report data files produced by methylationGLM() are read from this report project's assets/results/glm_results directory when available.
lmeTab	Character or NULL. XLSX workbook shown in the LME Analysis tab. When NULL, the path is inferred from the Makefile output layout. Report data files produced by methylationLME() are read from this report project's assets/results/lme_results directory when available.
modelSections	Character vector containing any of 'glm' and 'lme'. The report includes the corresponding model pages, logs, and summary sections. Use character(0) for a preprocessing-only report. The default preserves the complete GLM-and-LME report.
logTab	Character. Directory containing workflow logs shown in the Logs tab.
verbose	Logical. If TRUE, emit progress messages.
logs	Logical. If TRUE, write a report log.
projectName	Character. Name used for the generated Quarto project.
detPPath	Character or NULL. RData file containing the detection P-value matrix object detP, used to build the quality-control tables. When NULL, the path is inferred from the Makefile output layout.
detPThreshold	Numeric. Detection P-value threshold used when summarising the detP matrix.
cpgDetectionPath	Character or NULL. Optional fallback CpG detection summary CSV.
sampleDetectionPath	Character or NULL. Optional fallback sample detection summary CSV.
logoPath	Character. Path to the navigation-panel logo. Defaults to the packaged inst/extdata/dnaEPICORM.svg asset.
imagePattern	Character. Regular expression used to identify image files inside the section directories.
recursive	Logical. If TRUE, search section directories recursively.

Details

The Quarto command-line interface is required to render the website. It is not required to install or load dnaEPICO, or to use the package's preprocessing and statistical-modeling functions. When magick is installed, each readable figure is exported as PNG, JPEG and TIFF at the source image dimensions. Existing raster images are not upscaled. Exports are independent of the report's light/dark theme; JPEG transparency is flattened onto white. Without magick, original downloads remain available, but additional formats cannot be generated. The report opens at an 80% presentation scale. Browser zoom changes the on-screen display; downloaded figures retain their image dimensions.

Value

A list with class 'dnaEPICO_dnamReport'.

Examples

```
report_root <- file.path(tempdir(), "dnaepico-dnam-report")
pheno_file <- file.path(
  report_root,
  "data",
  "model1",
  "preprocessingMinfiEwasWater",
  "phenoLC.csv"
)
dir.create(dirname(pheno_file), recursive = TRUE, showWarnings = FALSE)
utils::write.csv(
  data.frame(
    UID = c("sample1", "sample2"),
    Timepoint = c(1, 2),
    Sex = c("F", "M")
  ),
  pheno_file,
  row.names = FALSE
)

result <- dnamReport(
  outputDir = file.path(report_root, "reports", "model1"),
  phenoTab = pheno_file,
  enmixTab = file.path(
    report_root,
    "figures",
    "model1",
    "preprocessingMinfiEwasWater",
    "enmix"
  ),
  qcTab = file.path(
    report_root,
    "figures",
    "model1",
    "preprocessingMinfiEwasWater",
    "qc"
```

```
  ),
  svaTab = file.path(report_root, "figures", "model1", "svaEnmix"),
  metricTab = file.path(
    report_root,
    "figures",
    "model1",
    "preprocessingMinfiEwasWater",
    "metrics"
  ),
  logTab = file.path(report_root, "logs", "model1")
)
result$status
```

`estimateLC` *Estimate saliva cell proportions from DNA methylation beta values*

Description

Estimate cell-type proportions with the saliva reference panels bundled in `dnaEPICO`.

Usage

```
estimateLC(meth, ref, constrained = FALSE)
```

Arguments

`meth`	Numeric matrix of beta values with CpGs in rows and samples in columns. Row names must contain probe identifiers compatible with the selected reference.
`ref`	Character. Reference panel name. Supported values are `'saliva'` and `'salivaEPIC'`.
`constrained`	Logical. If `TRUE`, estimated cell proportions are constrained to sum to one.

Value

A `data.table` with one row per sample and one column per estimated cell type.

References

Murat K, et al. Ewastools: Infinium Human Methylation BeadChip pipeline for population epigenetics integrated into Galaxy. *GigaScience*. 2020;9(5):giaa049. Houseman EA, Accomando WP, Koestler DC, et al. DNA methylation arrays as surrogate measures of cell mixture distribution. *BMC Bioinformatics*. 2012;13:86. Reinius LE, Acevedo N, Joerink M, et al. Differential DNA methylation in purified human blood cells: implications for cell lineage and studies on disease susceptibility. *PLoS One*. 2012;7(7):e41361. Bakulski KM, Feinberg JI, Andrews SV, et al. DNA methylation of cord blood cell types: applications for mixed cell birth studies. *Epigenetics*. 2016;11(5):354-362. de Goede OM, Razzaghian HR, Price EM, et al. Nucleated red blood cells impact DNA methylation and expression analyses of cord blood hematopoietic cells. *Clinical Epigenetics*. 2015;7:95. Gervin K, Salas LA, Bakulski KM, et al. Cell type specific DNA methylation in cord blood: a 450K

reference data set and cell count-based validation of estimated cell type composition. *Epigenetics*. 2016;11(9):690-698. Gervin K, Salas LA, Bakulski KM, et al. Systematic evaluation and validation of reference and library selection methods for deconvolution of cord blood DNA methylation data. *bioRxiv*. 2019. doi:10.1101/570457. Salas LA, Koestler DC, Butler RA, et al. An optimized library for reference-based deconvolution of whole-blood biospecimens assayed using the Illumina HumanMethylationEPIC BeadArray. *Genome Biology*. 2018;19:64. Heiss JA, Just AC, Brenner H. Training a model for estimating leukocyte composition using whole-blood DNA methylation and cell counts as reference. *Epigenomics*. 2017;9(1):13-20. Middleton LYM, Dou J, Mill J, et al. Saliva cell type DNA methylation reference panel for epidemiology studies in children. 2020.

Examples

```
ref_file <- system.file("extdata", "saliva.txt", package = "dnaEPICO")
ref_panel <- as.matrix(utils::read.table(ref_file))
meth <- ref_panel[1:20, , drop = FALSE]
colnames(meth) <- c("sample1", "sample2")
estimateLC(
    meth = meth,
    ref = "saliva",
    constrained = FALSE
)
```

`estimateLCMinfiEwasWater` *Estimate cell composition for preprocessingMinfiEwasWater*

Description

Estimate cell proportions from beta values using `estimateLC()` for saliva reference panels or `ENmix::estimateCellProp()` for other supported references, then merge the estimates into the phenotype table.

Usage

```
estimateLCMinfiEwasWater(
  beta,
  targets,
  SampleID = "Sample_Name",
  lcRef = "salivaEPIC",
  phenoOrder = paste0("Sample_Name;Timepoint;Sex;PredSex;Basename;",
    "Sentrix_ID;Sentrix_Position"),
  constrained = FALSE,
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_estimateLCMinfiEwasWater.txt"
)
```

Arguments

beta	Numeric matrix of beta values with probes in rows and samples in columns.
targets	Phenotype data frame aligned with the columns of beta.
SampleID	Character. Phenotype column containing sample identifiers.
lcRef	Character. Cell-composition reference. Internal saliva-based references supported through estimateLC() are 'saliva' and 'salivaEPIC'. Other references are passed to ENmix::estimateCellProp().
phenoOrder	Character vector or semicolon-separated phenotype columns to place first in the merged phenoLC output.
constrained	Logical. Passed to estimateLC() when an internal saliva reference is used. If TRUE, estimated proportions are constrained to sum to one.
verbose	Logical. If TRUE, emit progress messages with message().
logs	Logical. If TRUE, write the same messages to a log file.
log_dir	Character or NULL. Directory used for the log file when logs = TRUE.
log_file	Character. File name used when logs = TRUE.

Value

A list with class 'dnaEPICO_minfiEwasWater_lc' containing the cell proportion matrix, merged phenotype table, reference name, and method used.

Examples

```
ref_file <- system.file("extdata", "saliva.txt", package = "dnaEPICO")
beta <- as.matrix(utils::read.table(ref_file))[1:20, , drop = FALSE]
colnames(beta) <- c("sample1", "sample2")
targets <- data.frame(
    Sample_Name = colnames(beta),
    Timepoint = c("T1", "T2"),
    stringsAsFactors = FALSE
)
lc_data <- estimateLCMinfiEwasWater(
    beta = beta,
    targets = targets,
    lcRef = "saliva",
    phenoOrder = "Sample_Name;Timepoint"
)
stopifnot(is.data.frame(lc_data$phenoLC))
```

`estimateSvaEnmixControls` *Estimate surrogate variables from ENmix control probes*

Description

Run `ENmix::ctrlsva()` on an `RGChannelSet` and return the surrogate variable matrix as an in-memory object.

Usage

```
estimateSvaEnmixControls(
  RGSet,
  ctrlSvaPercVar = 0.9,
  ctrlSvaFlag = 1,
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_estimateSvaEnmixControls.txt"
)
```

Arguments

`RGSet`	An `RGChannelSet`.
`ctrlSvaPercVar`	Numeric. Proportion of variance explained by control probes, passed to `ENmix::ctrlsva()`.
`ctrlSvaFlag`	Integer. Control-probe flag passed to `ENmix::ctrlsva()`.
`verbose`	Logical. If `TRUE`, emit progress messages with `message()`.
`logs`	Logical. If `TRUE`, write the same messages to a log file.
`log_dir`	Character or `NULL`. Directory used for the log file when `logs = TRUE`.
`log_file`	Character. File name used when `logs = TRUE`.

Value

A list with class `'dnaEPICO_svaEnmix_sva'` containing the surrogate variable matrix and the parameters used to estimate it.

Examples

```
ex <- dnaEPICO:::exampleMinfiBaseDataDnaEpico()
sva_data <- estimateSvaEnmixControls(
  RGSet = ex$RGSet,
  ctrlSvaPercVar = 0.5,
  ctrlSvaFlag = 1,
  verbose = FALSE,
  logs = FALSE
)
```

```
sva_data$K
```

`extractMake` *Export the dnaEPICO Makefile to a user directory*

Description

Export the example Makefile supplied with `dnaEPICO` to a user directory.

Usage

```
extractMake(destDir, overwrite = FALSE)
```

Arguments

`destDir`	Character. Destination directory for the Makefile.
`overwrite`	Logical. Whether to overwrite an existing `Makefile` in `destDir`. The default is `FALSE`.

Details

The exported workflow requires GNU Make 4.3 or later. When the Makefile is created, `extractMake()` records the absolute path to the `Rscript` executable from the current R installation in the `RSCRIPT` Make variable. This value can be overridden when running Make, for example `make f4 RSCRIPT=/path/to/Rscript`, when another R installation or an HPC module should be used.

Quarto is required only for targets that render reports. The rules use relative project paths and support project directories containing spaces. GNU Make and Quarto are not required to install or load `dnaEPICO`.

`MODEL` selects one analysis configuration: its input files, settings and output directory. `MODELS` lists configurations for a grouped run. Each analysis can fit many CpG-wise statistical models.

`N_CORES` limits the total CPU cores used by one Make workflow. The rules share the available cores among the analyses in `MODELS`. An R worker is a R process that fits a subset of CpGs. With 8 available cores and 3 analyses, the maximum R workers per analysis are 3, 3 and 2. Analyses wait when all available workers are in use. Steps within each analysis run in order.

The shared rules set the OpenMP, OpenBLAS, MKL, Accelerate and NumExpr thread limits to one before starting R. This limits the threads used by numerical libraries within each R process. `make status` displays the available cores and maximum R workers per analysis. Fitting can use fewer workers for small CpG sets or limited memory. Memory requirements apply separately to each analysis.

Value

Character scalar containing the path to the copied `Makefile`.

Examples

```
tmp <- file.path(tempdir(), "dnaEPICO-make-example")
dir.create(tmp, recursive = TRUE, showWarnings = FALSE)
makefile_path <- extractMake(
    destDir = tmp,
    overwrite = TRUE
)
stopifnot(file.exists(makefile_path))
```

---

extractMetricsMinfiEwasWater

Extract beta, M, and copy-number matrices from a filtered object

---

Description

Extract beta, M, and copy-number matrices from a filtered object

Usage

```
extractMetricsMinfiEwasWater(
  filteredData,
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_extractMetricsMinfiEwasWater.txt"
)
```

Arguments

filteredData	Object returned by filterProbesMinfiEwasWater().
verbose	Logical. If TRUE, emit progress messages with message().
logs	Logical. If TRUE, write the same messages to a log file.
log_dir	Character or NULL. Directory used for the log file when logs = TRUE.
log_file	Character. File name used when logs = TRUE.

Value

A list with class 'dnaEPICO_minfiEwasWater_metrics' containing beta, m, and cn.

Examples

```
ex <- dnaEPICO:::exampleMinfiMetricsStateDnaEpico()
metrics_data <- extractMetricsMinfiEwasWater(
  filteredData = ex$filteredData,
  verbose = FALSE,
  logs = FALSE
```

```
)
names(metrics_data)
```

`filterProbesMinfiEwasWater` *Filter probes from a normalized methylation object*

Description

Apply detection P-value, chromosome, SNP, and probe-exclusion filters to the primary normalized object and return the filtered result.

Usage

```
filterProbesMinfiEwasWater(
  normData,
  RGSet,
  pvalThreshold = 0.01,
  chrToRemove = "chrX,chrY",
  snpsToRemove = "SBE,CpG",
  mafThreshold = 0.1,
  probeExclusionPath,
  probeExclusionIdColumn = NULL,
  useEpicV2Manifest = FALSE,
 epicV2ManifestFlags = c(CH_WGBS_evidence = TRUE, CH_BLAT = TRUE, MissingPos = TRUE,
    MismatchPos = FALSE),
  detPtype = "m+u",
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_filterProbesMinfiEwasWater.txt",
  crossReactivePath = NULL,
  crossReactiveIdColumn = NULL
)
```

Arguments

`normData`	Object returned by `normalizeMinfiEwasWater()`.
`RGSet`	Filtered `RGChannelSet` aligned with `normData`.
`pvalThreshold`	Numeric. Probes must have detection P values below this threshold in all samples to be retained.
`chrToRemove`	Character vector or comma-separated string of chromosome names to remove, for example `'chrX,chrY'`.
`snpsToRemove`	Character vector or comma-separated string of SNP probe types to remove, for example `'SBE,CpG'`.

`mafThreshold` Numeric. Minor allele frequency threshold passed to `minfi::dropLociWithSnps()`.

`probeExclusionPath`
: Character vector or semicolon-separated string of CSV files containing probe IDs to remove.

`probeExclusionIdColumn`
: Character or `NULL`. Column containing probe IDs. When `NULL` or `''`, each file is auto-detected using `ProbeID`, `TargetID`, `IlmnID`, or `Name`, then falling back to an unlabeled or probe-like first column.

`useEpicV2Manifest`
: Logical. If `TRUE`, also remove EPICv2 probes flagged in the Peters et al. expanded manifest from AnnotationHub resource `AH116484`.

`epicV2ManifestFlags`
: Named logical vector controlling which EPICv2 manifest flags are removed. Defaults remove `CH_WGBS_evidence`, `CH_BLAT`, and `MissingPos`, but not `MismatchPos`.

`detPtype` Character. Detection P-value mode passed to `minfi::detectionP()` for the probe filter. Common values in minfi workflows are `'m+u'` and `'negative'`.

`verbose` Logical. If `TRUE`, emit progress messages with `message()`.

`logs` Logical. If `TRUE`, write the same messages to a log file.

`log_dir` Character or `NULL`. Directory used for the log file when `logs = TRUE`.

`log_file` Character. File name used when `logs = TRUE`.

`crossReactivePath`
: Deprecated alias for `probeExclusionPath`.

`crossReactiveIdColumn`
: Deprecated alias for `probeExclusionIdColumn`.

Value

A list with class `'dnaEPICO_minfiEwasWater_filter'` containing the filtered object and counts for each filtering stage.

Examples

```
ex <- dnaEPICO:::exampleMinfiWorkflowStateDnaEpico()
filtered_data <- filterProbesMinfiEwasWater(
  normData = ex$normData,
  RGSet = ex$sampleData$RGSet,
  pvalThreshold = 1,
  chrToRemove = "chrY",
  snpsToRemove = "SBE",
  mafThreshold = 1,
  probeExclusionPath = ex$probeExclusionPath,
  detPtype = "m+u",
  verbose = FALSE,
  logs = FALSE
)
filtered_data$counts[["probeExclusion"]]
```

`filterSamplesMinfiEwasWater` *Filter failed samples from an RGSet and phenotype table*

Description

Remove failed samples identified during sample assessment and return the filtered `RGChannelSet` together with the aligned phenotype table.

Usage

```
filterSamplesMinfiEwasWater(
  RGSet,
  targets,
  failedSamples = character(0),
  SampleID = "Sample_Name",
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_filterSamplesMinfiEwasWater.txt"
)
```

Arguments

`RGSet`	An `RGChannelSet`.
`targets`	Data frame containing phenotype information.
`failedSamples`	Character vector of sample identifiers to remove.
`SampleID`	Character. Name of the sample identifier column in `targets`.
`verbose`	Logical. If `TRUE`, emit progress messages with `message()`.
`logs`	Logical. If `TRUE`, write the same messages to a log file.
`log_dir`	Character or `NULL`. Directory used for the log file when `logs = TRUE`.
`log_file`	Character. File name used when `logs = TRUE`.

Value

A list with class `'dnaEPICO_minfiEwasWater_samples'` containing the filtered `RGSet`, aligned phenotype table, and failed sample identifiers.

Examples

```
ex <- dnaEPICO:::exampleMinfiBaseDataDnaEpico()
filtered_samples <- filterSamplesMinfiEwasWater(
  RGSet = ex$RGSet,
  targets = ex$targets,
  failedSamples = ex$targets$Sample_Name[1],
  SampleID = "Sample_Name",
```

```
  verbose = FALSE,
  logs = FALSE
)
nrow(filtered_samples$targets)
```

`fitMethylationGLMModels` *Fit CpG-wise Gaussian GLMs for one-timepoint methylation analyses*

Description

Fit one Gaussian GLM per CpG for each phenotype requested in the object returned by `prepareMethylationGLMData()`. Each fitted model is reduced to compact numerical results and discarded before the next batch is returned.

Usage

```
fitMethylationGLMModels(
  preparedData,
  nCores = 1L,
  libPath = NULL,
  glmLibs = "glm2",
  summaryDir = NULL,
  omnibusTest = FALSE,
  resumeFromSummary = TRUE,
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_methylationGLM.txt"
)
```

Arguments

`preparedData`	Object returned by `prepareMethylationGLMData()`.
`nCores`	Integer. Maximum number of R worker processes for fitting. Automatic fitting uses one process for small CpG sets. Multiple workers are limited by the CpG count, available CPU cores and detected memory.
`libPath`	Character vector or `NULL`. Optional library paths forwarded to worker processes.
`glmLibs`	Character vector or comma-separated string of package names to check on worker processes. The default is `'glm2'`.
`summaryDir`	Character or `NULL`. Directory used for one complete compact summary per phenotype. `NULL` skips saving summaries.
`omnibusTest`	Logical. If `TRUE`, use `car::linearHypothesis()` to test the complete phenotype-by-interaction term, or the phenotype main effect when no interaction is specified, once per CpG.

`resumeFromSummary`
: Logical. If `TRUE`, reuse a complete summary in `summaryDir` when it was generated from the same input file and model configuration.

`verbose`
: Logical. If `TRUE`, emit progress messages with `message()`.

`logs`
: Logical. If `TRUE`, write the same messages to a log file.

`log_dir`
: Character or `NULL`. Directory used for the log file when `logs = TRUE`.

`log_file`
: Character. File name used when `logs = TRUE`.

Value

A list with class `'dnaEPICO_methylationGLM_models'` containing compact coefficient matrices, complete phenotype summaries, formulas, model messages, fitting errors, and saved phenotype summaries.

Examples

```
ex <- dnaEPICO:::exampleMethylationGLMStateDnaEpico()
model_results <- fitMethylationGLMModels(
  preparedData = ex$preparedData,
  nCores = 1,
  verbose = FALSE,
  logs = FALSE
)
names(model_results$fits)
```

`fitMethylationLMEModels` *Fit CpG-wise mixed-effects models for longitudinal methylation analyses*

Description

Fit one linear mixed-effects model per CpG for each phenotype requested in the object returned by `prepareMethylationLMEData()`. Each fitted model is reduced to compact numerical results and discarded before the next batch is returned.

Usage

```
fitMethylationLMEModels(
  preparedData,
  nCores = 1L,
  libPath = NULL,
  lmeLibs = "lme4,lmerTest",
  correlationStructure = "none",
  correlationVar = NULL,
  omnibusTest = FALSE,
```

```
  omnibusDdf = "Satterthwaite",
  summaryDir = NULL,
  resumeFromSummary = TRUE,
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_methylationLME.txt"
)
```

Arguments

preparedData	Object returned by prepareMethylationLMEData().
nCores	Integer. Maximum number of R worker processes for fitting. Automatic fitting uses one process for small CpG sets. Multiple workers are limited by the CpG count, available CPU cores and detected memory.
libPath	Character vector or NULL. Optional library paths forwarded to worker processes.
lmeLibs	Character vector or comma-separated string of package names to check on worker processes. The default is 'lme4,lmerTest'.
correlationStructure	Character. Residual correlation structure used when lmeLibs selects 'nlme'. One of 'none', 'AR1', or 'CAR1'.
correlationVar	Character or NULL. Variable used to order repeated observations within personVar for AR1 or CAR1 residual correlation structures. Must be supplied explicitly for AR1 or CAR1.
omnibusTest	Logical. If TRUE, calculate one joint fixed-effect test per CpG for the phenotype-by-interaction term, or the phenotype main effect when no interaction is requested.
omnibusDdf	Character. Denominator degrees-of-freedom method used by lmerTest::contestMD(): 'Satterthwaite' or 'Kenward-Roger'.
summaryDir	Character or NULL. Directory used for one complete compact summary per phenotype. NULL skips saving summaries.
resumeFromSummary	Logical. If TRUE, reuse a complete summary in summaryDir when it was generated from the same input file and model configuration.
verbose	Logical. If TRUE, emit progress messages with message().
logs	Logical. If TRUE, write the same messages to a log file.
log_dir	Character or NULL. Directory used for the log file when logs = TRUE.
log_file	Character. File name used when logs = TRUE.

Value

A list with class 'dnaEPICO_methylationLME_models' containing compact coefficient matrices, complete phenotype summaries, formulas, model messages, omnibus results, fitting errors, and phenotype saved summaries.

Examples

```
ex <- dnaEPICO:::exampleMethylationLMEStateDnaEpico()
model_results <- fitMethylationLMEModels(
  preparedData = ex$preparedData,
  nCores = 1,
  verbose = FALSE,
  logs = FALSE
)
names(model_results$fits)
```

---

loadMetricsPreprocessingPheno

Load methylation metric matrices for preprocessingPheno

Description

Load the metric matrices generated by `preprocessingMinfiEwasWater()` and return them as a single in-memory object for downstream phenotype alignment.

Usage

```
loadMetricsPreprocessingPheno(
  betaPath,
  mPath,
  cnPath,
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_loadMetricsPreprocessingPheno.txt"
)
```

Arguments

`betaPath`	Character. Path to the saved beta-value object.
`mPath`	Character. Path to the saved M-value object.
`cnPath`	Character. Path to the saved copy-number object.
`verbose`	Logical. If `TRUE`, emit progress messages with `message()`.
`logs`	Logical. If `TRUE`, write the same messages to a log file.
`log_dir`	Character or `NULL`. Directory used for the log file when `logs = TRUE`.
`log_file`	Character. File name used when `logs = TRUE`.

Value

A list with class `'dnaEPICO_preprocessingPheno_metrics'` containing `beta`, `m`, and `cn`.

Examples

```
ex <- dnaEPICO:::examplePreprocessingPhenoStateDnaEpico()
metrics_data <- loadMetricsPreprocessingPheno(
  betaPath = ex$betaPath,
  mPath = ex$mPath,
  cnPath = ex$cnPath,
  verbose = FALSE,
  logs = FALSE
)
names(metrics_data)
```

`mergeSvaTargetsEnmix` *Merge surrogate variables into the phenotype table*

Description

Merge the surrogate variable matrix back into the phenotype table while preserving the original row order of `targets`.

Usage

```
mergeSvaTargetsEnmix(
  targets,
  sva,
  SampleID = "Sample_Name",
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_mergeSvaTargetsEnmix.txt"
)
```

Arguments

`targets`	Phenotype data frame aligned with the samples in `sva`.
`sva`	Numeric matrix of surrogate variables with samples in rows.
`SampleID`	Character. Name of the phenotype sample identifier column.
`verbose`	Logical. If `TRUE`, emit progress messages with `message()`.
`logs`	Logical. If `TRUE`, write the same messages to a log file.
`log_dir`	Character or `NULL`. Directory used for the log file when `logs = TRUE`.
`log_file`	Character. File name used when `logs = TRUE`.

Value

A phenotype data frame with the surrogate variables appended.

Examples

```
ex <- dnaEPICO:::exampleSvaAnalysisStateDnaEpico()
merged_pheno <- mergeSvaTargetsEnmix(
  targets = ex$targets,
  sva = ex$sva,
  SampleID = "Sample_Name",
  verbose = FALSE,
  logs = FALSE
)
colnames(merged_pheno)[seq_len(4)]
```

`methylationGLM` *Fit CpG-wise GLMs for one-timepoint methylation analyses*

Description

`methylationGLM()` prepares phenotype-plus-methylation data, fits one Gaussian GLM per CpG and phenotype, summarizes and annotates the results, and creates optional coefficient tables and diagnostic plots. It writes outputs only when `saveOutputs = TRUE`. CpG methylation is the response; each variable in `phenotypes` is a predictor of interest. Saved workbooks, figures and logs can be included in a report with `dnamReport()`. Numeric CpG columns are passed to `glm2::glm2()` without a separate methylation-domain filter. Model messages, warnings, and errors are recorded in one phenotype-specific `Model.Message` field. Annotated outputs contain CpGs with at least one returned coefficient or omnibus p-value; aggregate availability and condition counts are recorded in workbook metadata.

Usage

```
methylationGLM(
  inputPheno = "rData/preprocessingPheno/mergeData/phenoBT1.RData",
  outputLogs = "logs",
  outputRData = "rData/methylationGLM/models",
  outputPlots = "figures/methylationGLM",
 phenotypes = c("DASS_Depression", "DASS_Anxiety", "DASS_Stress", "PCL5_TotalScore",
    "MHCSF_TotalScore", "BRS_TotalScore"),
  covariates = paste0("Sex,Age,Ethnicity,TraumaDefinition,Leukocytes,",
    "Epithelial.cells"),
  factorVars = "Sex,Ethnicity,TraumaDefinition",
  scaleVars = NULL,
  cpgPrefix = "cg",
  cpgLimit = NA,
  methylationScale = "beta",
  nCores = 32,
  plotWidth = 2000,
  plotHeight = 1000,
  plotDPI = 150,
```

```
  interactionTerm = NULL,
  omnibusTest = FALSE,
  vennDPhenotypes = NULL,
  vennDLabels = NULL,
  vennDOmnibusPhenotypes = NULL,
  vennDOmnibusLabels = NULL,
  libPath = NULL,
  glmLibs = "glm2",
  prsMap = NULL,
  summaryPval = NA,
  summaryResidualSD = TRUE,
  saveSignificantCpGs = FALSE,
  significantCpGDir = "preliminaryResults/cpgs/methylationGLM",
  significantCpGPval = 0.05,
  saveTxtSummaries = TRUE,
  chunkSize = NULL,
  summaryTxtDir = "preliminaryResults/summary/methylationGLM",
  fdrThreshold = 0.05,
  padjmethod = "fdr",
  annotationPackage = "IlluminaHumanMethylationEPICv2anno.20a1.hg38",
 annotationCols = c("Name", "chr", "pos", "UCSC_RefGene_Group", "UCSC_RefGene_Name",
    "Relation_to_Island", "GencodeV41_Group"),
  gencodeHub = FALSE,
  annotatedGLMOut = "data/methylationGLM",
  reportAssetsDir = NULL,
  display = FALSE,
  verbose = FALSE,
  logs = FALSE,
  saveOutputs = FALSE,
  resumeFromSummary = TRUE
)
```

Arguments

inputPheno	Character. Path to the merged phenotype-plus-methylation `.RData` or `.rds` object created by `preprocessingPheno()`. The default points to the timepoint-1 object produced by the package workflow.
outputLogs	Character. Directory used for optional log files.
outputRData	Character. Directory used for compact, resumable phenotype summaries.
outputPlots	Character. Directory used for optional TIFF plots.
phenotypes	Character vector or comma-separated phenotype variables used as predictors of interest in separate CpG-wise models.
covariates	Character. Comma-separated covariate variables included in each GLM.
factorVars	Character. Comma-separated variables to convert to factors before modeling.
scaleVars	Character vector, comma-separated variable names, or `NULL`. Numeric fixed-effect variables to centre and divide by their sample standard deviations before model fitting.

`cpgPrefix` Character. Prefix used to identify methylation columns in the merged phenotype-plus-methylation input object. The default is `'cg'`.

`cpgLimit` Integer or `NA`. Maximum number of CpGs to analyse. Use `NA` to keep all CpGs matching `cpgPrefix`.

`methylationScale` Character. Methylation metric represented by the CpG columns. One of `'Beta'`, `'M'`, or `'CN'`, in any combination of upper- and lower-case letters. The default is `'beta'`.

`nCores` Integer. Maximum number of R worker processes for fitting. Automatic fitting uses one process for small CpG sets. Multiple workers are limited by the CpG count, available CPU cores and detected memory.

`plotWidth` Integer. TIFF width in pixels when plots are written to disk.

`plotHeight` Integer. TIFF height in pixels when plots are written to disk.

`plotDPI` Integer. TIFF resolution in DPI when plots are written to disk.

`interactionTerm` Character or `NULL`. Optional interaction term. When supplied and present in the input data, the phenotype is modeled together with its interaction against this variable.

`omnibusTest` Logical. If `TRUE`, use `car::linearHypothesis()` to test the complete phenotype-by-interaction term, or the phenotype main effect when `interactionTerm = NULL`, once per CpG. One-degree-of-freedom terms are tested and therefore reproduce the corresponding coefficient p-value.

`vennDPhenotypes` Character vector, comma-separated phenotype names, or `NULL`. Selected phenotypes are expanded to all coefficient p-value columns for model-level Venn diagrams and workbook tables.

`vennDLabels` Character vector, comma-separated display labels, or `NULL`. Labels follow the resolved coefficient order and preserve case.

`vennDOmnibusPhenotypes` Character vector, comma-separated phenotype names, or `NULL`. These use only omnibus p-value columns and require `omnibusTest = TRUE`.

`vennDOmnibusLabels` Character vector, comma-separated display labels, or `NULL`, supplied in the resolved omnibus phenotype order.

`libPath` Character vector or `NULL`. Optional library paths forwarded to worker processes. By default, the current `.libPaths()` are used.

`glmLibs` Character. Comma-separated package names to validate on worker processes. The default is `'glm2'`.

`prsMap` Character or `NULL`. Optional phenotype-to-PRS mapping in the form `'Phenotype1:PRS_1,Phenotype2:P`

`summaryPval` Numeric or `NA`. Optional p-value threshold applied to the returned CpG summary tables. Use `NA` to keep all summary rows.

`summaryResidualSD` Logical. If `TRUE`, append residual standard deviations to the CpG summary tables and residual diagnostic plots.

saveSignificantCpGs
: Logical. If TRUE, collect significant CpG coefficient tables in the returned object and optionally write them to disk when saveOutputs = TRUE.

significantCpGDir
: Character. Directory used for optional significant CpG coefficient tables.

significantCpGPval
: Numeric. P-value threshold used to collect or write significant CpG coefficient tables. The threshold is applied to omnibus p-values when omnibusTest = TRUE, and to target coefficient p-values otherwise.

saveTxtSummaries
: Logical. If TRUE and saveOutputs = TRUE, write tab-delimited summary tables to summaryTxtDir.

chunkSize
: Integer or NULL. Number of CpGs processed per summary extraction chunk. NULL chooses a value automatically.

summaryTxtDir
: Character. Directory used for optional tab-delimited GLM summary tables.

fdrThreshold
: Numeric. False-discovery-rate threshold used to highlight CpGs in the residual-significance diagnostic plots.

padjmethod
: Character. P-value adjustment method passed to stats::p.adjust(). The default is 'fdr'.

annotationPackage
: Character. Annotation package or object name passed to minfi::getAnnotation(), for example 'IlluminaHumanMethylationEPICv2anno.20a1.hg38'.

annotationCols
: Character vector or comma-separated annotation columns to append to the combined GLM summary table. Available columns depend on the selected annotation package.

gencodeHub
: Logical. If TRUE, append release-aware GENCODE gene-body and nearest-TSS annotations obtained through AnnotationHub. The selected array annotation must use GRCh38 coordinates.

annotatedGLMOut
: Character. Directory used for the optional annotated GLM summary XLSX workbook.

reportAssetsDir
: Character or NULL. Report results directory used for the compressed TSV table and its compact metadata files. NULL writes only the model outputs and annotated workbook.

display
: Logical. If TRUE, draw exploratory and diagnostic plots on the active graphics device.

verbose
: Logical. If TRUE, emit progress messages with message(). The default is FALSE.

logs
: Logical. If TRUE, write the same progress messages to file.path(outputLogs, 'log_methylationGLM.txt').

saveOutputs
: Logical. If TRUE, write compact phenotype summaries, text summaries, significant-CpG tables, annotated results, and TIFF plots. The default is FALSE.

resumeFromSummary
: Logical. If TRUE and saveOutputs = TRUE, reuse a complete phenotype summary when its input file and model configuration match the current analysis. If processing stops before a phenotype summary is complete, that phenotype is fitted again from its first CpG.

Value

A list with class 'dnaEPICO_methylationGLM'.

preparedData Object returned by prepareMethylationGLMData() containing the merged phenotype-plus-methylation analysis table and modeling metadata.

distributionPlots Object returned by plotMethylationGLMDistributions() describing any exploratory plots that were generated or written.

designPlots Missingness and numeric-correlation plots for the variables used in the model.

modelFits Object returned by fitMethylationGLMModels() containing compact per-phenotype coefficient, omnibus, and condition results.

modelSummaries Object returned by summarizeMethylationGLMModels() containing the combined CpG summary tables used for reporting and annotation.

significantCpGs Object returned by collectSignificantCpGsMethylationGLM() containing optional phenotype-specific significant-CpG tables.

diagnosticPlots Object returned by plotMethylationGLMDiagnostics() describing the diagnostic plot objects and any written TIFF files.

annotation Object returned by annotateMethylationGLMSummaries() containing the annotated combined summary table.

manhattanPlots Versioned circular and rectangular Manhattan plots for every raw p-value column in the annotated results.

vennDPlots Requested coefficient and omnibus model-level Venn plots, worksheet tables, and label mappings.

savedFiles Object returned by writeMethylationGLMOutputs() when saveOutputs = TRUE, otherwise NULL.

runSettings High-level run metadata including the generic analysis label, methylation scale, display label, selected merged-object prefix, and internal response-column name.

See dnaEPICO_methylationGLM for a class-level overview.

See Also

dnaEPICO_methylationGLM

Examples

```
if (requireNamespace(
    "IlluminaHumanMethylation450kanno.ilmn12.hg19",
    quietly = TRUE
)) {
    tmp <- tempdir()
    toy_path <- file.path(tmp, "phenoBT1.RData")
    phenoBT1 <- data.frame(
        Sample_Name = c("S1", "S2", "S3", "S4"),
        status = factor(c("Case", "Case", "Control", "Control")),
        sex = factor(c("F", "M", "F", "M")),
        cg00000029 = c(0.20, 0.25, 0.22, 0.27),
        cg00000108 = c(0.60, 0.55, 0.52, 0.58),
```

```
        check.names = FALSE
    )
    save(phenoBT1, file = toy_path)

    result <- methylationGLM(
        inputPheno = toy_path,
        phenotypes = "status",
        covariates = "sex",
        factorVars = "status,sex",
        cpgLimit = 2,
        nCores = 1,
        summaryPval = 1,
        annotationPackage = "IlluminaHumanMethylation450kanno.ilmn12.hg19",
        annotationCols = "Name,chr,pos",
        display = FALSE,
        verbose = FALSE,
        logs = FALSE,
        saveOutputs = FALSE
    )

    class(result)
}
```

`methylationLME` *Fit CpG-wise linear mixed-effects models for longitudinal methylation analyses*

Description

`methylationLME()` prepares longitudinal phenotype-plus-methylation data, fits one mixed-effects model per CpG and phenotype, summarizes and annotates the results, and creates optional interaction tables and diagnostic plots. It writes outputs only when `saveOutputs = TRUE`. CpG methylation is the response; each variable in `phenotypes` is a predictor of interest. The participant identifier defines the random intercept. Saved workbooks, figures and logs can be included in a report with `dnamReport()`. Numeric CpG columns are passed to the selected lmerTest/lme4 or nlme engine without a separate methylation-domain filter. Model messages, warnings, and errors are recorded in one phenotype-specific `Model.Message` field. Annotated outputs contain CpGs with at least one returned coefficient or omnibus p-value; aggregate availability and condition counts are recorded in workbook metadata.

Usage

```
methylationLME(
  inputPheno = "rData/preprocessingPheno/mergeData/phenoBT1T2.RData",
  outputLogs = "logs",
  outputRData = "rData/methylationLME/models",
  outputPlots = "figures/methylationLME",
  personVar = "person",
```

```
  timeVar = "Timepoint",
  phenotypes = c("DASS_Depression", "DASS_Anxiety", "DASS_Stress", "PCL5_TotalScore",
    "MHCSF_TotalScore", "BRS_TotalScore"),
  covariates = paste0("Sex,Age,Ethnicity,TraumaDefinition,Leukocytes,",
    "Epithelial.cells"),
  factorVars = "Sex,Ethnicity,TraumaDefinition",
  scaleVars = NULL,
  lmeLibs = "lme4,lmerTest",
  correlationStructure = "none",
  correlationVar = NULL,
  prsMap = NULL,
  libPath = NULL,
  cpgPrefix = "cg",
  cpgLimit = NA,
  methylationScale = "beta",
  nCores = 32,
  summaryPval = NA,
  plotWidth = 2000,
  plotHeight = 1000,
  plotDPI = 150,
  interactionTerm = NULL,
  omnibusTest = FALSE,
  omnibusDdf = "Satterthwaite",
  vennDPhenotypes = NULL,
  vennDLabels = NULL,
  vennDOmnibusPhenotypes = NULL,
  vennDOmnibusLabels = NULL,
  saveSignificantInteractions = TRUE,
  significantInteractionDir = "preliminaryResults/cpgs/methylationLME",
  significantInteractionPval = 0.05,
  saveTxtSummaries = TRUE,
  chunkSize = NULL,
  summaryTxtDir = "preliminaryResults/summary/methylationLME",
  fdrThreshold = 0.05,
  padjmethod = "fdr",
  annotationPackage = "IlluminaHumanMethylationEPICv2anno.20a1.hg38",
  annotationCols = c("Name", "chr", "pos", "UCSC_RefGene_Group", "UCSC_RefGene_Name",
    "Relation_to_Island", "GencodeV41_Group"),
  gencodeHub = FALSE,
  annotatedLMEOut = "data/methylationLME",
  reportAssetsDir = NULL,
  display = FALSE,
  verbose = FALSE,
  logs = FALSE,
  saveOutputs = FALSE,
  resumeFromSummary = TRUE,
  SampleID = "SID"
)
```

Arguments

inputPheno	Character. Path to the merged longitudinal phenotype-plus-methylation .RData or .rds object created by preprocessingPheno(). The default points to the combined timepoint object produced by the package workflow.
outputLogs	Character. Directory used for optional log files.
outputRData	Character. Directory used for compact, resumable phenotype summaries.
outputPlots	Character. Directory used for optional TIFF diagnostic plots.
personVar	Character. Subject identifier variable used for the random intercept. When this column is missing, it is derived from SampleID using the package's terminal A/B visit naming convention.
timeVar	Character. Name of the longitudinal time variable used for timepoint summaries and preprocessing checks.
phenotypes	Character vector or comma-separated phenotype variables used as predictors of interest in separate CpG-wise models.
covariates	Character. Comma-separated fixed-effect covariates included in every mixed model.
factorVars	Character. Comma-separated variables to convert to factors before modeling, including categorical phenotypes, covariates, or interaction variables.
scaleVars	Character vector, comma-separated variable names, or NULL. Numeric fixed-effect variables to centre and divide by their sample standard deviations before fitting.
lmeLibs	Character. Comma-separated package names to validate on worker processes and select the model engine. Use 'lme4,lmerTest' or 'lme4' for lmerTest/lme4, or 'nlme' for nlme::lme().
correlationStructure	Character. Residual correlation structure used when lmeLibs = 'nlme'. One of 'none', 'AR1', or 'CAR1'. The default is 'none'.
correlationVar	Character or NULL. Variable used to order repeated observations within personVar for AR1 or CAR1 residual correlation structures. Must be supplied explicitly for AR1 or CAR1.
prsMap	Character or NULL. Optional phenotype-to-PRS mapping in the form 'Phenotype1:PRS_1,Phenotype2:P
libPath	Character vector or NULL. Optional library paths forwarded to worker processes. By default, the current .libPaths() are used.
cpgPrefix	Character. Prefix used to identify methylation columns in the merged phenotype-plus-methylation input object. The default is 'cg'.
cpgLimit	Integer or NA. Maximum number of CpGs to analyse. Use NA to keep all CpGs matching cpgPrefix.
methylationScale	Character. Methylation metric represented by the CpG columns. One of 'Beta', 'M', or 'CN', in any combination of upper- and lower-case letters. The default is 'beta'.
nCores	Integer. Maximum number of R worker processes for fitting. Automatic fitting uses one process for small CpG sets. Multiple workers are limited by the CpG count, available CPU cores and detected memory.

summaryPval — Numeric or NA. Optional p-value threshold applied to the returned longitudinal CpG summary tables. Use NA to keep all summary rows.

plotWidth — Integer. TIFF width in pixels when plots are written to disk.

plotHeight — Integer. TIFF height in pixels when plots are written to disk.

plotDPI — Integer. TIFF resolution in DPI when plots are written to disk.

interactionTerm — Character or NULL. Optional interaction term. When supplied and present in the input data, the phenotype is modeled together with its interaction against this variable.

omnibusTest — Logical. If TRUE, use lmerTest::contestMD() to test the complete phenotype-by-interaction term, or the phenotype main effect when interactionTerm = NULL, once per CpG. This is available only for the lmerTest/lme4 engine.

omnibusDdf — Character. Denominator degrees-of-freedom method for the omnibus F test. One of 'Satterthwaite' or 'Kenward-Roger'.

vennDPhenotypes — Character vector, comma-separated phenotype names, or NULL. Selected phenotypes are expanded to all coefficient p-value columns for model-level Venn diagrams and workbook tables.

vennDLabels — Character vector, comma-separated display labels, or NULL. Labels follow the resolved coefficient order and preserve case.

vennDOmnibusPhenotypes — Character vector, comma-separated phenotype names, or NULL. These use only omnibus p-value columns and require omnibusTest = TRUE.

vennDOmnibusLabels — Character vector, comma-separated display labels, or NULL, supplied in the resolved omnibus phenotype order.

saveSignificantInteractions — Logical. If TRUE, collect coefficient tables for CpGs passing significantInteractionPval in the returned object and optionally write them to disk when saveOutputs = TRUE.

significantInteractionDir — Character. Directory used for optional significant-interaction coefficient tables.

significantInteractionPval — Numeric. P-value threshold used to collect or write significant interaction coefficient tables.

saveTxtSummaries — Logical. If TRUE and saveOutputs = TRUE, write tab-delimited summary tables to summaryTxtDir.

chunkSize — Integer or NULL. Number of CpGs processed per summary extraction chunk. NULL chooses a value automatically.

summaryTxtDir — Character. Directory used for optional tab-delimited LME summary tables.

fdrThreshold — Numeric. False-discovery-rate threshold used to highlight CpGs in the residual-significance diagnostic plots.

padjmethod — Character. P-value adjustment method passed to stats::p.adjust(). The default is 'fdr'.

annotationPackage
: Character. Annotation package or object name passed to `minfi::getAnnotation()`, for example `'IlluminaHumanMethylationEPICv2anno.20a1.hg38'`.

annotationCols
: Character vector or comma-separated annotation columns to append to the combined LME summary table. Available columns depend on the selected annotation package.

gencodeHub
: Logical. If `TRUE`, append release-aware GENCODE gene-body and nearest-TSS annotations obtained through AnnotationHub. The selected array annotation must use GRCh38 coordinates.

annotatedLMEOut
: Character. Directory used for the optional annotated LME summary XLSX workbook.

reportAssetsDir
: Character or `NULL`. Report results directory used for the compressed TSV table and its compact metadata files. `NULL` writes only the model outputs and annotated workbook.

display
: Logical. If `TRUE`, draw diagnostic plots on the active graphics device.

verbose
: Logical. If `TRUE`, emit progress messages with `message()`. The default is `FALSE`.

logs
: Logical. If `TRUE`, write the same progress messages to `file.path(outputLogs, 'log_methylationLME.txt')`.

saveOutputs
: Logical. If `TRUE`, write compact phenotype summaries, text summaries, significant-interaction tables, annotated results, and TIFF plots. The default is `FALSE`.

resumeFromSummary
: Logical. If `TRUE` and `saveOutputs = TRUE`, reuse a complete phenotype summary when its input file and model configuration match the current analysis. If processing stops before a phenotype summary is complete, that phenotype is fitted again from its first CpG.

SampleID
: Character. Name of the sample identifier column used to derive `personVar` when the subject identifier column is missing. The default, `"SID"`, preserves the package's earlier behavior.

Value

A list with class `'dnaEPICO_methylationLME'`.

preparedData Object returned by `prepareMethylationLMEData()` containing the merged longitudinal phenotype-plus-methylation analysis table and modeling metadata.

distributionPlots Distributions of the model variables, longitudinal observation counts, time values, and numeric phenotype trajectories.

designPlots Missingness and numeric-correlation plots for the variables used in the model.

modelFits Object returned by `fitMethylationLMEModels()` containing compact per-phenotype coefficient, omnibus, and condition results.

modelSummaries Object returned by `summarizeMethylationLMEModels()` containing the combined CpG summary tables used for reporting and annotation.

significantInteractions Object returned by `collectSignificantInteractionsMethylationLME()` containing optional phenotype-specific significant-interaction tables.

diagnosticPlots Object returned by `plotMethylationLMEDiagnostics()` describing the diagnostic plot objects and any written TIFF files.

annotation Object returned by `annotateMethylationLMESummaries()` containing the annotated combined summary table.

manhattanPlots Versioned circular and rectangular Manhattan plots for every raw p-value column in the annotated results.

vennDPlots Requested coefficient and omnibus model-level Venn plots, worksheet tables, and label mappings.

savedFiles Object returned by `writeMethylationLMEOutputs()` when `saveOutputs = TRUE`, otherwise `NULL`.

runSettings High-level run metadata including the generic analysis label, methylation scale, display label, selected merged-object prefix, internal response-column name, and longitudinal time variable.

See dnaEPICO_methylationLME for a class-level overview.

See Also

dnaEPICO_methylationLME

Examples

```
if (
    requireNamespace(
        "IlluminaHumanMethylation450kanno.ilmn12.hg19",
        quietly = TRUE
    ) &&
        requireNamespace("lmerTest", quietly = TRUE)
) {
    tmp <- tempdir()
    toy_path <- file.path(tmp, "phenoBT1T2.RData")
    phenoBT1T2 <- data.frame(
        SID = c("P1A", "P1B", "P2A", "P2B", "P3A", "P3B", "P4A", "P4B"),
        person = c(1, 1, 2, 2, 3, 3, 4, 4),
        Timepoint = factor(c("1", "2", "1", "2", "1", "2", "1", "2")),
        score = c(10, 12, 9, 11, 13, 14, 8, 9),
        sex = factor(c("F", "F", "M", "M", "F", "F", "M", "M")),
        cg00000029 = c(0.25, 0.27, 0.20, 0.22, 0.30, 0.31, 0.18, 0.20),
        cg00000108 = c(0.50, 0.53, 0.55, 0.57, 0.48, 0.49, 0.60, 0.61),
        check.names = FALSE
    )
    save(phenoBT1T2, file = toy_path)

    result <- methylationLME(
        inputPheno = toy_path,
        phenotypes = "score",
        covariates = "sex",
        factorVars = "sex",
        cpgLimit = 2,
        nCores = 1,
```

```
        summaryPval = 1,
        annotationPackage = "IlluminaHumanMethylation450kanno.ilmn12.hg19",
        annotationCols = "Name,chr,pos",
        display = FALSE,
        verbose = FALSE,
        logs = FALSE,
        saveOutputs = FALSE
    )

    class(result)
}
```

`normalizeMinfiEwasWater` *Normalize filtered samples with minfi and wateRmelon methods*

Description

Apply one or more supported normalization methods to a filtered `RGSet` and return all normalized objects together in a single result object.

Usage

```
normalizeMinfiEwasWater(
  sampleData,
  sexColumn = "Sex",
  normMethods = "adjustedfunnorm",
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_normalizeMinfiEwasWater.txt"
)
```

Arguments

`sampleData`	Object returned by `filterSamplesMinfiEwasWater()`.
`sexColumn`	Character. Name of the phenotype column used as the optional sex covariate for normalization methods that support it. Missing or unsupported values use `PredSex` when available, and each substitution is recorded in the returned `sexResolution` table.
`normMethods`	Character vector or semicolon-separated string of normalization methods. Supported values are `'adjustedfunnorm'`, `'funnorm'`, `'illumina'`, `'quantile'`, and `'swan'`.
`verbose`	Logical. If `TRUE`, emit progress messages with `message()`.
`logs`	Logical. If `TRUE`, write the same messages to a log file.
`log_dir`	Character or `NULL`. Directory used for the log file when `logs = TRUE`.
`log_file`	Character. File name used when `logs = TRUE`.

Value

A list with class 'dnaEPICO_minfiEwasWater_norm' containing the requested normalized objects, the first method as primary, and a sexResolution audit table describing reported-sex and PredSex use.

Examples

```
ex <- dnaEPICO:::exampleMinfiBaseDataDnaEpico()
sample_data <- filterSamplesMinfiEwasWater(
  RGSet = ex$RGSet,
  targets = ex$targets,
  failedSamples = character(0),
  SampleID = "Sample_Name",
  verbose = FALSE,
  logs = FALSE
)
norm_data <- normalizeMinfiEwasWater(
  sampleData = sample_data,
  sexColumn = "Sex",
  normMethods = "quantile",
  verbose = FALSE,
  logs = FALSE
)
names(norm_data$normalized)
```

---

plotAssessmentMinfiEwasWater

Plot quality-assessment outputs for preprocessingMinfiEwasWater

---

Description

Draw either the minfi QC plot or the detection P-value plot from an assessment object returned by assessSamplesMinfiEwasWater(). Detection plots show one grey bar per sample in input order, with sample labels. Output width expands for large sample sets to keep labels readable.

Usage

```
plotAssessmentMinfiEwasWater(
  assessment,
  plot = c("qc", "detection"),
  display = FALSE,
  file = NULL,
  width = 2000L,
  height = 1000L,
  res = 150L,
  verbose = FALSE,
  logs = FALSE,
```

```
  log_dir = NULL,
  log_file = "log_plotAssessmentMinfiEwasWater.txt"
)
```

Arguments

assessment	Object returned by assessSamplesMinfiEwasWater().
plot	Character. Plot type: 'qc' or 'detection'.
display	Logical. If TRUE, draw the plot on the active graphics device.
file	Character or NULL. TIFF file written when supplied.
width	Integer. TIFF width in pixels when file is supplied.
height	Integer. TIFF height in pixels when file is supplied.
res	Integer. TIFF resolution in DPI when file is supplied.
verbose	Logical. If TRUE, emit progress messages with message().
logs	Logical. If TRUE, write the same messages to a log file.
log_dir	Character or NULL. Directory used for the log file when logs = TRUE.
log_file	Character. File name used when logs = TRUE.

Value

Invisibly returns the saved TIFF path when file is supplied, otherwise NULL.

Examples

```
assessment <- list(
  meanDetP = c(S1 = 0.01, S2 = 0.02, S3 = 0.04),
  detPThreshold = 0.05
)
plotAssessmentMinfiEwasWater(
  assessment = assessment,
  plot = "detection",
  display = FALSE,
  verbose = FALSE,
  logs = FALSE
)
```

---

plotCtrlMinfiEwasWater

Plot ENmix control images from an RGSet

Description

Call ENmix::plotCtrl() for a supplied RGSet. This function only writes files when output_dir is provided because ENmix::plotCtrl() produces JPG files on disk rather than returning a plot object.

Usage

```
plotCtrlMinfiEwasWater(
  RGSet,
  output_dir = NULL,
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_plotCtrlMinfiEwasWater.txt"
)
```

Arguments

RGSet	An RGChannelSet.
output_dir	Character or NULL. Directory where ENmix control JPG files should be written. If NULL, the function returns without writing files.
verbose	Logical. If TRUE, emit progress messages with message().
logs	Logical. If TRUE, write the same messages to a log file.
log_dir	Character or NULL. Directory used for the log file when logs = TRUE.
log_file	Character. File name used when logs = TRUE.

Value

Invisibly returns output_dir.

Examples

```
ex <- dnaEPICO:::exampleMinfiBaseDataDnaEpico()
output_dir <- file.path(tempdir(), "enmix-control-plots")
plotCtrlMinfiEwasWater(
  RGSet = ex$RGSet,
  output_dir = output_dir,
  verbose = FALSE,
  logs = FALSE
)
dir.exists(output_dir)
```

---

plotMethylationGLMDiagnostics

Plot diagnostic summaries for one-timepoint methylation GLMs

Description

Create Q-Q and residual-diagnostic plots from the CpG summary tables returned by summarizeMethylationGLMModels().

Usage

```
plotMethylationGLMDiagnostics(
  modelSummaries,
  preparedData,
  fdrThreshold = 0.05,
  padjmethod = "fdr",
  outputDir = NULL,
  plotWidth = 2000L,
  plotHeight = 1000L,
  plotDPI = 150L,
  display = FALSE,
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_methylationGLM.txt"
)
```

Arguments

modelSummaries	Object returned by summarizeMethylationGLMModels().
preparedData	Object returned by prepareMethylationGLMData().
fdrThreshold	Numeric. False-discovery-rate threshold used to highlight CpGs in the diagnostic plots.
padjmethod	Character. P-value adjustment method passed to stats::p.adjust().
outputDir	Character or NULL. Directory used for TIFF files. When NULL, plots are returned in memory only.
plotWidth	Integer. TIFF width in pixels when plots are written to disk.
plotHeight	Integer. TIFF height in pixels when plots are written to disk.
plotDPI	Integer. TIFF resolution in DPI when plots are written to disk.
display	Logical. If TRUE, draw plots on the active graphics device.
verbose	Logical. If TRUE, emit progress messages with message().
logs	Logical. If TRUE, write the same messages to a log file.
log_dir	Character or NULL. Directory used for the log file when logs = TRUE.
log_file	Character. File name used when logs = TRUE.

Value

A list with class 'dnaEPICO_methylationGLM_diagnostic_plots' containing the generated ggplot2 objects, genomic inflation factors, and any saved TIFF file paths.

Examples

```
ex <- dnaEPICO:::exampleMethylationGLMStateDnaEpico()
diagnostic_plots <- plotMethylationGLMDiagnostics(
  modelSummaries = ex$modelSummaries,
```

```
  preparedData = ex$preparedData,
  display = FALSE,
  verbose = FALSE,
  logs = FALSE
)
names(diagnostic_plots$plots)
```

`plotMethylationGLMDistributions` *Plot phenotype and covariate distributions for one-timepoint GLM analyses*

Description

Create phenotype, factor-variable, and numeric-covariate distribution plots from the object returned by `prepareMethylationGLMData()`.

Usage

```
plotMethylationGLMDistributions(
  preparedData,
  plotWidth = 2000L,
  plotHeight = 1000L,
  plotDPI = 150L,
  outputDir = NULL,
  display = FALSE,
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_methylationGLM.txt"
)
```

Arguments

`preparedData`	Object returned by `prepareMethylationGLMData()`.
`plotWidth`	Integer. TIFF width in pixels when plots are written to disk.
`plotHeight`	Integer. TIFF height in pixels when plots are written to disk.
`plotDPI`	Integer. TIFF resolution in DPI when plots are written to disk.
`outputDir`	Character or `NULL`. Directory used for TIFF files. When `NULL`, plots are returned in memory only.
`display`	Logical. If `TRUE`, draw plots on the active graphics device.
`verbose`	Logical. If `TRUE`, emit progress messages with `message()`.
`logs`	Logical. If `TRUE`, write the same messages to a log file.
`log_dir`	Character or `NULL`. Directory used for the log file when `logs = TRUE`.
`log_file`	Character. File name used when `logs = TRUE`.

Value

A list with class 'dnaEPICO_methylationGLM_distribution_plots' containing the generated ggplot2 objects and any saved TIFF file paths.

Examples

```
ex <- dnaEPICO:::exampleMethylationGLMStateDnaEpico()
distribution_plots <- plotMethylationGLMDistributions(
  preparedData = ex$preparedData,
  display = FALSE,
  verbose = FALSE,
  logs = FALSE
)
names(distribution_plots)
```

---

plotMethylationLMEDiagnostics

Plot longitudinal mixed-effects model diagnostics

Description

Create Q-Q and standard-error diagnostic plots from the mixed-effects summary tables returned by summarizeMethylationLMEModels().

Usage

```
plotMethylationLMEDiagnostics(
  modelSummaries,
  preparedData,
  fdrThreshold = 0.05,
  padjmethod = "fdr",
  outputDir = NULL,
  plotWidth = 2000L,
  plotHeight = 1000L,
  plotDPI = 150L,
  display = FALSE,
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_methylationLME.txt"
)
```

Arguments

modelSummaries	Object returned by summarizeMethylationLMEModels().
preparedData	Object returned by prepareMethylationLMEData().

fdrThreshold	Numeric. False-discovery-rate threshold used to highlight CpGs in the diagnostic plots.
padjmethod	Character. P-value adjustment method passed to `stats::p.adjust()`.
outputDir	Character or `NULL`. Directory used for TIFF files. When `NULL`, plots are returned in memory only.
plotWidth	Integer. TIFF width in pixels when plots are written to disk.
plotHeight	Integer. TIFF height in pixels when plots are written to disk.
plotDPI	Integer. TIFF resolution in DPI when plots are written to disk.
display	Logical. If `TRUE`, draw plots on the active graphics device.
verbose	Logical. If `TRUE`, emit progress messages with `message()`.
logs	Logical. If `TRUE`, write the same messages to a log file.
log_dir	Character or `NULL`. Directory used for the log file when `logs = TRUE`.
log_file	Character. File name used when `logs = TRUE`.

Value

A list with class `'dnaEPICO_methylationLME_diagnostic_plots'` containing the generated ggplot2 objects, genomic inflation factors, and any saved TIFF file paths.

Examples

```
ex <- dnaEPICO:::exampleMethylationLMEStateDnaEpico()
diagnostic_plots <- plotMethylationLMEDiagnostics(
  modelSummaries = ex$modelSummaries,
  preparedData = ex$preparedData,
  display = FALSE,
  verbose = FALSE,
  logs = FALSE
)
names(diagnostic_plots$plots)
```

---

plotMetricsMinfiEwasWater

Plot multidimensional scaling or density summaries from final metrics

---

Description

Plot multidimensional scaling or density summaries from final metrics

Usage

```
plotMetricsMinfiEwasWater(
  metricsData,
  targets,
  plot = c("mds", "density"),
  plotGroupVar = "Sex",
  sexColumn = "Sex",
  display = FALSE,
  file = NULL,
  width = 2000L,
  height = 1000L,
  res = 150L,
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_plotMetricsMinfiEwasWater.txt"
)
```

Arguments

metricsData	Object returned by extractMetricsMinfiEwasWater().
targets	Filtered phenotype data aligned with metricsData.
plot	Character. Plot type: 'mds' or 'density'.
plotGroupVar	Character. Phenotype column used for the main grouping.
sexColumn	Character. Phenotype column used for the sex grouping in the MDS plot.
display	Logical. If TRUE, draw the plot on the active graphics device.
file	Character or NULL. TIFF file written when supplied.
width	Integer. TIFF width in pixels when file is supplied.
height	Integer. TIFF height in pixels when file is supplied.
res	Integer. TIFF resolution in DPI when file is supplied.
verbose	Logical. If TRUE, emit progress messages with message().
logs	Logical. If TRUE, write the same messages to a log file.
log_dir	Character or NULL. Directory used for the log file when logs = TRUE.
log_file	Character. File name used when logs = TRUE.

Value

Invisibly returns the saved TIFF path when file is supplied, otherwise NULL.

Examples

```
ex <- dnaEPICO:::exampleMinfiMetricsStateDnaEpico()
plotMetricsMinfiEwasWater(
  metricsData = ex$metricsData,
  targets = ex$targets,
```

```
  plot = "density",
  plotGroupVar = "Sex",
  sexColumn = "Sex",
  display = FALSE,
  verbose = FALSE,
  logs = FALSE
)
```

`plotNormalizationMinfiEwasWater` *Plot raw and normalized methylation distributions*

Description

Draw the density comparison plot used to inspect raw versus normalized data.

Usage

```
plotNormalizationMinfiEwasWater(
  RGSet,
  normData,
  targets,
  sexColumn = "Sex",
  display = FALSE,
  file = NULL,
  width = 2000L,
  height = 1000L,
  res = 150L,
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_plotNormalizationMinfiEwasWater.txt"
)
```

Arguments

`RGSet`	An `RGChannelSet` aligned with `targets`.
`normData`	Object returned by `normalizeMinfiEwasWater()`.
`targets`	Filtered phenotype data aligned with `RGSet`.
`sexColumn`	Character. Name of the phenotype column used to colour the density curves when `normData` does not contain resolved normalization sex. Otherwise the reported-sex/`PredSex` normalization vector is used.
`display`	Logical. If `TRUE`, draw the plot on the active graphics device.
`file`	Character or `NULL`. TIFF file written when supplied.
`width`	Integer. TIFF width in pixels when `file` is supplied.

height	Integer. TIFF height in pixels when file is supplied.
res	Integer. TIFF resolution in DPI when file is supplied.
verbose	Logical. If TRUE, emit progress messages with message().
logs	Logical. If TRUE, write the same messages to a log file.
log_dir	Character or NULL. Directory used for the log file when logs = TRUE.
log_file	Character. File name used when logs = TRUE.

Value

Invisibly returns the saved TIFF path when file is supplied, otherwise NULL.

Examples

```
ex <- dnaEPICO:::exampleMinfiMetricsStateDnaEpico()
plotNormalizationMinfiEwasWater(
  RGSet = ex$beta,
  normData = ex$normData,
  targets = ex$targets,
  sexColumn = "Sex",
  display = FALSE,
  verbose = FALSE,
  logs = FALSE
)
```

---

plotRawDensityMinfiEwasWater

Plot raw beta-value density from a raw preprocessing object

Description

Draw the pre-normalization beta density plot from a raw minfi object and a grouping variable in the phenotype table.

Usage

```
plotRawDensityMinfiEwasWater(
  rawData,
  targets,
  plotGroupVar = "Sex",
  display = FALSE,
  file = NULL,
  width = 2000L,
  height = 1000L,
  res = 150L,
  verbose = FALSE,
  logs = FALSE,
```

```
  log_dir = NULL,
  log_file = "log_plotRawDensityMinfiEwasWater.txt"
)
```

Arguments

rawData	Object returned by buildRawMinfiEwasWater().
targets	Filtered phenotype data aligned with rawData.
plotGroupVar	Character. Phenotype column used to group samples in the density plot.
display	Logical. If TRUE, draw the plot on the active graphics device.
file	Character or NULL. TIFF file written when supplied.
width	Integer. TIFF width in pixels when file is supplied.
height	Integer. TIFF height in pixels when file is supplied.
res	Integer. TIFF resolution in DPI when file is supplied.
verbose	Logical. If TRUE, emit progress messages with message().
logs	Logical. If TRUE, write the same messages to a log file.
log_dir	Character or NULL. Directory used for the log file when logs = TRUE.
log_file	Character. File name used when logs = TRUE.

Value

Invisibly returns the saved TIFF path when file is supplied, otherwise NULL.

Examples

```
ex <- dnaEPICO:::exampleMinfiMetricsStateDnaEpico()
plotRawDensityMinfiEwasWater(
  rawData = ex$rawData,
  targets = ex$targets,
  plotGroupVar = "Sex",
  display = FALSE,
  verbose = FALSE,
  logs = FALSE
)
```

---

plotSexMinfiEwasWater *Plot predicted or clinical sex from* predictSexMinfiEwasWater()

Description

Plot predicted or clinical sex from predictSexMinfiEwasWater()

Usage

```
plotSexMinfiEwasWater(
  sexData,
  type = c("predicted", "clinical"),
  display = FALSE,
  file = NULL,
  width = 2000L,
  height = 1000L,
  res = 70L,
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_plotSexMinfiEwasWater.txt"
)
```

Arguments

sexData	Object returned by predictSexMinfiEwasWater().
type	Character. Plot type: 'predicted' for methylation-predicted sex or 'clinical' for reported sex.
display	Logical. If TRUE, draw the plot on the active graphics device.
file	Character or NULL. TIFF file written when supplied.
width	Integer. TIFF width in pixels when file is supplied.
height	Integer. TIFF height in pixels when file is supplied.
res	Integer. TIFF resolution in DPI when file is supplied.
verbose	Logical. If TRUE, emit progress messages with message().
logs	Logical. If TRUE, write the same messages to a log file.
log_dir	Character or NULL. Directory used for the log file when logs = TRUE.
log_file	Character. File name used when logs = TRUE.

Value

Invisibly returns the saved TIFF path when file is supplied, otherwise NULL.

Examples

```
ex <- dnaEPICO:::exampleSexPlotStateDnaEpico()
plotSexMinfiEwasWater(
  sexData = ex,
  type = "predicted",
  display = FALSE,
  verbose = FALSE,
  logs = FALSE
)
```

plotSvaEnmix *Plot surrogate variables for svaEnmix*

Description

Draw one of the standard surrogate-variable plots used by `svaEnmix()`. An empty association result is recorded in the log without creating a placeholder figure.

Usage

```
plotSvaEnmix(
  analysisData,
  plot = c("sentrix_id", "sentrix_position", "matrix", "association"),
  display = FALSE,
  file = NULL,
  width = 2000L,
  height = 1000L,
  res = 150L,
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_plotSvaEnmix.txt"
)
```

Arguments

`analysisData`	Object returned by `analyzeSvaEnmix()`.
`plot`	Character. Plot type: `'sentrix_id'`, `'sentrix_position'`, or `'matrix'`, or `'association'`.
`display`	Logical. If `TRUE`, draw the plot on the active graphics device.
`file`	Character or `NULL`. TIFF file path used for saved output.
`width`	Integer. Plot width in pixels when `file` is supplied.
`height`	Integer. Plot height in pixels when `file` is supplied.
`res`	Integer. TIFF resolution in DPI when `file` is supplied.
`verbose`	Logical. If `TRUE`, emit progress messages with `message()`.
`logs`	Logical. If `TRUE`, write the same messages to a log file.
`log_dir`	Character or `NULL`. Directory used for the log file when `logs = TRUE`.
`log_file`	Character. File name used when `logs = TRUE`.

Value

Invisibly returns saved TIFF path(s), otherwise `NULL`. Association output is omitted when no technical-factor association is estimable.

Examples

```
ex <- dnaEPICO:::exampleSvaAnalysisStateDnaEpico()
plotSvaEnmix(
  analysisData = ex$analysisData,
  plot = "sentrix_id",
  display = FALSE,
  verbose = FALSE,
  logs = FALSE
)
```

`predictSexMinfiEwasWater` *Predict biological sex from a filtered raw-data object*

Description

Predict sample sex from a genome-mapped methylation object, align the predictions with phenotype data, and return a structured object that can be plotted or merged into downstream phenotype tables.

Usage

```
predictSexMinfiEwasWater(
  rawData,
  targets,
  SampleID = "Sample_Name",
  sexColumn = "Sex",
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_predictSexMinfiEwasWater.txt"
)
```

Arguments

`rawData`	Object returned by `buildRawMinfiEwasWater()`.
`targets`	Filtered phenotype data frame aligned with `rawData`.
`SampleID`	Character. Name of the sample identifier column in `targets`.
`sexColumn`	Character. Name of the phenotype column containing reported sex.
`verbose`	Logical. If `TRUE`, emit progress messages with `message()`.
`logs`	Logical. If `TRUE`, write the same messages to a log file.
`log_dir`	Character or `NULL`. Directory used for the log file when `logs = TRUE`.
`log_file`	Character. File name used when `logs = TRUE`.

Value

A list with class 'dnaEPICO_minfiEwasWater_sex' containing the sex prediction result, aligned phenotype data, plotting data, mismatch table, and a sexResolution audit showing where PredSex would replace an unusable reported value during sex-aware normalization.

Examples

```
ex <- dnaEPICO:::exampleMinfiWorkflowStateDnaEpico()
sex_data <- predictSexMinfiEwasWater(
  rawData = ex$rawFiltered,
  targets = ex$sampleData$targets,
  SampleID = "Sample_Name",
  sexColumn = "Sex",
  verbose = FALSE,
  logs = FALSE
)
names(sex_data)
```

---

prepareDnamReportInputs

Prepare inputs for a DNA methylation report

Description

Prepare inputs for a DNA methylation report

Usage

```
prepareDnamReportInputs(
  outputDir = "reports",
  qcDir = file.path("figures", "preprocessingMinfiEwasWater", "enmix"),
  preprocessingDir = file.path("figures", "preprocessingMinfiEwasWater", "qc"),
 postprocessingDir = file.path("figures", "preprocessingMinfiEwasWater", "metrics"),
  svaDir = file.path("figures", "svaEnmix"),
  glmDir = file.path("figures", "methylationGLM"),
  lmeDir = file.path("figures", "methylationLME"),
  figDir = file.path(outputDir, "assets", "figures"),
  verbose = FALSE,
  logs = FALSE,
  logDir = outputDir
)
```

Arguments

outputDir	Character. Directory where the report project is written.
qcDir	Character. Directory containing ENmix quality-control figures.

`preprocessingDir`
: Character. Directory containing preprocessing quality-control figures.

`postprocessingDir`
: Character. Directory containing postprocessing metric figures.

`svaDir`
: Character. Directory containing SVA or batch-effect figures.

`glmDir`
: Character. Directory containing GLM figures.

`lmeDir`
: Character. Directory containing LME figures.

`figDir`
: Character. Directory used for generated report figure assets.

`verbose`
: Logical. If `TRUE`, emit progress messages with `message()`.

`logs`
: Logical. If `TRUE`, write progress messages to `file.path(logDir, 'log_dnamReport.txt')`.

`logDir`
: Character. Directory for optional log files.

Value

A list with class `'dnaEPICO_dnamReport_prepared'`.

Examples

```
report_root <- file.path(tempdir(), "dnaepico-report-inputs")
prepared <- prepareDnamReportInputs(
  outputDir = file.path(report_root, "reports"),
  qcDir = file.path(
    report_root,
    "figures",
    "preprocessingMinfiEwasWater",
    "enmix"
  ),
  preprocessingDir = file.path(
    report_root,
    "figures",
    "preprocessingMinfiEwasWater",
    "qc"
  ),
  postprocessingDir = file.path(
    report_root,
    "figures",
    "preprocessingMinfiEwasWater",
    "metrics"
  ),
  svaDir = file.path(report_root, "figures", "svaEnmix")
)
inherits(prepared, "dnaEPICO_dnamReport_prepared")
```

prepareMethylationGLMData *Prepare phenotype-plus-methylation data for one-timepoint GLM analyses*

Description

Load the merged phenotype-plus-methylation input object, validate the requested modeling variables, convert selected variables to factors, and return a single in-memory object for downstream helpers.

Usage

```
prepareMethylationGLMData(
  inputPheno,
  phenotypes,
  covariates,
  factorVars,
  scaleVars = NULL,
  cpgPrefix = "cg",
  cpgLimit = NA,
  methylationScale = "beta",
  interactionTerm = NULL,
  prsMap = NULL,
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_methylationGLM.txt"
)
```

Arguments

inputPheno	Character. Path to the merged phenotype-plus-methylation object created by `preprocessingPheno()`.
phenotypes	Character vector or comma-separated phenotype variables used as predictors of interest in separate CpG-wise models.
covariates	Character vector or comma-separated string of covariate variables to adjust for.
factorVars	Character vector or comma-separated string of variables that should be converted to factors before modeling.
scaleVars	Character vector, comma-separated variable names, or `NULL`. Numeric fixed-effect variables to standardize before model fitting.
cpgPrefix	Character. Prefix used to identify methylation columns.
cpgLimit	Integer or `NA`. Maximum number of CpGs to retain. `NA` keeps all matching CpGs.
methylationScale	Character. Methylation metric represented by the CpG columns. One of `'Beta'`, `'M'`, or `'CN'`, case-insensitively.

interactionTerm	Character or NULL. Optional interaction term.
prsMap	Character vector or comma-separated string of phenotype-to-PRS mappings in the form 'Phenotype:PRS'.
verbose	Logical. If TRUE, emit progress messages with message().
logs	Logical. If TRUE, write the same messages to a log file.
log_dir	Character or NULL. Directory used for the log file when logs = TRUE.
log_file	Character. File name used when logs = TRUE.

Value

A list with class 'dnaEPICO_methylationGLM_data' containing the prepared analysis data, parsed variable selections, CpG columns, and exploratory summaries.

Examples

```
ex <- dnaEPICO:::exampleMethylationGLMStateDnaEpico()
prepared_data <- prepareMethylationGLMData(
  inputPheno = ex$inputPath,
  phenotypes = "status",
  covariates = "sex,age",
  factorVars = "status,sex",
  cpgLimit = 2,
  verbose = FALSE,
  logs = FALSE
)
names(prepared_data)
```

---

prepareMethylationLMEData

Prepare longitudinal phenotype-plus-methylation data for mixed-effects analyses

---

Description

Load the merged longitudinal phenotype-plus-methylation object, ensure that a subject identifier column is available, validate the requested modeling variables, convert selected variables to factors, and return a single in-memory object for downstream mixed-effects modeling helpers.

Usage

```
prepareMethylationLMEData(
  inputPheno,
  personVar = "person",
  timeVar = "Timepoint",
  phenotypes,
```

```
  covariates,
  factorVars,
  scaleVars = NULL,
  prsMap = NULL,
  cpgPrefix = "cg",
  cpgLimit = NA,
  methylationScale = "beta",
  interactionTerm = NULL,
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_methylationLME.txt",
  SampleID = "SID"
)
```

Arguments

inputPheno	Character. Path to the merged longitudinal phenotype-plus- methylation object created by preprocessingPheno().
personVar	Character. Name of the subject identifier column.
timeVar	Character. Name of the time variable.
phenotypes	Character vector or comma-separated phenotype variables used as predictors of interest in separate CpG-wise models.
covariates	Character vector or comma-separated string of covariate variables to adjust for.
factorVars	Character vector or comma-separated string of variables that should be converted to factors before modeling.
scaleVars	Character vector, comma-separated variable names, or NULL. Numeric fixed-effect variables to standardize before model fitting.
prsMap	Character vector or comma-separated string of phenotype-to-PRS mappings in the form 'Phenotype:PRS'.
cpgPrefix	Character. Prefix used to identify methylation columns.
cpgLimit	Integer or NA. Maximum number of CpGs to retain. NA keeps all matching CpGs.
methylationScale	Character. Methylation metric represented by the CpG columns. One of 'Beta', 'M', or 'CN', case-insensitively.
interactionTerm	Character or NULL. Optional interaction term.
verbose	Logical. If TRUE, emit progress messages with message().
logs	Logical. If TRUE, write the same messages to a log file.
log_dir	Character or NULL. Directory used for the log file when logs = TRUE.
log_file	Character. File name used when logs = TRUE.
SampleID	Character. Name of the sample identifier column used to derive personVar when it is absent. Values must end in A or B for automatic derivation. The default is "SID".

Value

A list with class 'dnaEPICO_methylationLME_data' containing the prepared analysis data, parsed variable selections, CpG columns, timepoint summaries, and subject-ID diagnostics.

Examples

```
ex <- dnaEPICO:::exampleMethylationLMEStateDnaEpico()
prepared_data <- prepareMethylationLMEData(
  inputPheno = ex$inputPath,
  personVar = "person",
  timeVar = "Timepoint",
  phenotypes = "score",
  covariates = "sex",
  factorVars = "sex",
  cpgLimit = 2,
  verbose = FALSE,
  logs = FALSE
)
names(prepared_data)
```

---

preprocessingMinfiEwasWater

Preprocess and filter Illumina methylation-array data

Description

Read IDAT files, assess sample quality, normalise methylation values, filter probes and estimate cell composition using minfi, ENmix and wateRmelon. With saveOutputs = TRUE, save the methylation matrices, quality-control figures and phenoLC.csv for the next pipeline steps. Use the saved RGSet and phenotype table with svaEnmix(), then prepare modelling tables with preprocessingPheno().

Usage

```
preprocessingMinfiEwasWater(
  phenoFile = "data/preprocessingMinfiEwasWater/pheno.csv",
  idatFolder = "data/preprocessingMinfiEwasWater/idats",
  outputLogs = "logs",
  nSamples = NA,
  SampleID = "Sample_Name",
  arrayType = "IlluminaHumanMethylationEPICv2",
  annotationVersion = "20a1.hg38",
  idatForce = FALSE,
  scriptLabel = "preprocessingMinfiEwasWater",
  baseDataFolder = "rData",
  figureBaseDir = "figures",
  sepType = NULL,
  tiffWidth = 2000,
```

```
  tiffHeight = 1000,
  tiffRes = 150,
  qcCutoff = 10.5,
  detPtype = "m+u",
  detPThreshold = 0.05,
  normMethods = "adjustedfunnorm",
  sexColumn = "Sex",
  removeSexMismatch = FALSE,
  pvalThreshold = 0.01,
  chrToRemove = "chrX,chrY",
  snpsToRemove = "SBE,CpG",
  mafThreshold = 0.1,
  probeExclusionPath = paste0("data/preprocessingMinfiEwasWater/",
    "12864_2024_10027_MOESM8_ESM.csv"),
  probeExclusionIdColumn = NULL,
  useEpicV2Manifest = FALSE,
 epicV2ManifestFlags = c(CH_WGBS_evidence = TRUE, CH_BLAT = TRUE, MissingPos = TRUE,
    MismatchPos = FALSE),
  plotGroupVar = "Sex",
  lcRef = "salivaEPIC",
  phenoOrder = paste0("Sample_Name;Timepoint;Sex;PredSex;Basename;",
    "Sentrix_ID;Sentrix_Position"),
  lcPhenoDir = "data/preprocessingMinfiEwasWater",
  display = FALSE,
  verbose = FALSE,
  logs = FALSE,
  saveOutputs = FALSE,
  crossReactivePath = NULL,
  crossReactiveIdColumn = NULL
)
```

Arguments

phenoFile	Character. Path to the phenotype CSV file.
idatFolder	Character. Directory containing the IDAT files.
outputLogs	Character. Directory used for log files when logs = TRUE.
nSamples	Integer or NA. Number of rows to keep from the phenotype table. Use NA to keep all samples.
SampleID	Character. Name of the phenotype column containing sample identifiers.
arrayType	Character. Illumina array identifier passed to Biobase::annotation(), for example 'IlluminaHumanMethylationEPICv2'.
annotationVersion	Character. Annotation build passed to Biobase::annotation(), for example '20a1.hg38' or 'ilmn12.hg19'.
idatForce	Logical. Passed to minfi::read.metharray.exp() when reading IDAT files. Use TRUE only after confirming that the selected IDAT files should be read together.

scriptLabel Character. Label used to name output folders when saveOutputs = TRUE.

baseDataFolder Character. Base directory used for saved .RData outputs when saveOutputs = TRUE.

figureBaseDir Character. Base directory used for saved figure outputs when saveOutputs = TRUE.

sepType Character or NULL. Field separator used in phenoFile. Use NULL for a comma-separated file, '\\t' for a tab-delimited file, or another separator accepted by utils::read.csv().

tiffWidth Integer. Width of saved TIFF plots in pixels.

tiffHeight Integer. Height of saved TIFF plots in pixels.

tiffRes Integer. Resolution in DPI for saved TIFF plots.

qcCutoff Numeric. QC cutoff passed to minfi::plotQC().

detPtype Character. Detection P-value mode passed to minfi::detectionP(). Common values in minfi workflows are 'm+u' and 'negative'. The default here is 'm+u'.

detPThreshold Numeric. Samples with mean detection P value above this threshold are removed.

normMethods Character vector or semicolon-separated string of normalization methods. Supported values are 'adjustedfunnorm', 'funnorm', 'illumina', 'quantile', and 'swan'.

sexColumn Character. Name of the phenotype column containing reported sex. Sex-aware normalization methods use PredSex when this column is missing, blank, unknown, unsupported, or otherwise not coded as female or male. Every substitution is recorded without overwriting this column.

removeSexMismatch
Logical. If TRUE, remove samples whose reported sex and methylation-predicted sex are both known and disagree. Missing or unknown sex values are retained. The default is FALSE.

pvalThreshold Numeric. Probe-level detection P-value threshold used in the probe filter.

chrToRemove Character vector or comma-separated string of chromosome names to remove, for example 'chrX,chrY'.

snpsToRemove Character vector or comma-separated string of SNP probe types to remove, for example 'SBE,CpG'.

mafThreshold Numeric. Minor allele frequency threshold passed to minfi::dropLociWithSnps().

probeExclusionPath
Character vector or semicolon-separated string of CSV files containing probe IDs to remove.

probeExclusionIdColumn
Character or NULL. Column containing probe IDs. When NULL or '', each file is auto-detected using ProbeID, TargetID, IlmnID, or Name, then falling back to an unlabeled or probe-like first column.

useEpicV2Manifest
Logical. If TRUE, also remove EPICv2 probes flagged in the Peters et al. expanded manifest from AnnotationHub resource AH116484.

`epicV2ManifestFlags`
: Named logical vector controlling which EPICv2 manifest flags are removed. Defaults remove `CH_WGBS_evidence`, `CH_BLAT`, and `MissingPos`, but not `MismatchPos`.

`plotGroupVar`
: Character. Phenotype column used for density and MDS grouping plots.

`lcRef`
: Character. Reference panel used for cell composition estimation. `'saliva'` and `'salivaEPIC'` use `estimateLC()`. Other values are passed to `ENmix::estimateCellProp()`.

`phenoOrder`
: Character vector or semicolon-separated phenotype columns to place first in the merged `phenoLC` table.

`lcPhenoDir`
: Character. Directory used for the saved `phenoLC.csv` file when `saveOutputs = TRUE`.

`display`
: Logical. If `TRUE`, draw plots on the active graphics device.

`verbose`
: Logical. If `TRUE`, emit progress messages with `message()`. The default is `FALSE`.

`logs`
: Logical. If `TRUE`, write log messages to `outputLogs`. The default is `FALSE`.

`saveOutputs`
: Logical. If `TRUE`, write the `.RData`, figure, and `phenoLC.csv` outputs to disk. The default is `FALSE`.

`crossReactivePath`
: Deprecated alias for `probeExclusionPath`.

`crossReactiveIdColumn`
: Deprecated alias for `probeExclusionIdColumn`.

Value

A list with class `'dnaEPICO_preprocessingMinfiEwasWater'`.

targets Filtered phenotype table aligned to the retained samples.

RGSet Filtered `RGChannelSet` used in downstream preprocessing and available for direct interactive inspection.

rawData Object returned by `buildRawMinfiEwasWater()` containing the raw `MSet`, `RatioSet`, and genome-mapped object derived from `RGSet`.

assessment Object returned by `assessSamplesMinfiEwasWater()` containing detection P values, QC summaries, and failed-sample tracking.

sexData Object returned by `predictSexMinfiEwasWater()` containing predicted sex labels, mismatch summaries, normalization-sex provenance, plotting data, and the sample IDs removed when `removeSexMismatch = TRUE`.

normData Object returned by `normalizeMinfiEwasWater()` containing the selected normalized objects, method metadata, and the reported-sex or `PredSex` value used for each sample.

filterData Object returned by `filterProbesMinfiEwasWater()` containing the probe-filtered methylation objects at each filtering stage.

metricsData Object returned by `extractMetricsMinfiEwasWater()` containing the beta-value, M-value, and copy-number matrices used by later workflow steps.

lcData Object returned by `estimateLCMinfiEwasWater()` containing the estimated cell-type proportions and the phenotype table augmented with those proportions.

logFile Resolved path to the optional log file, or `NULL` when logging was disabled.

See dnaEPICO_preprocessingMinfiEwasWater for a class-level overview.

See Also

dnaEPICO_preprocessingMinfiEwasWater

Examples

```
if (requireNamespace("minfiData", quietly = TRUE) &&
  requireNamespace(
    "IlluminaHumanMethylation450kmanifest",
    quietly = TRUE
  ) &&
  requireNamespace(
    "IlluminaHumanMethylation450kanno.ilmn12.hg19",
    quietly = TRUE
  )) {
  ex <- dnaEPICO:::exampleMinfiIdatInputsDnaEpico(n = 4)
  result <- preprocessingMinfiEwasWater(
    phenoFile = ex$phenoFile,
    idatFolder = ex$idatFolder,
    outputLogs = file.path(ex$tempDir, "logs"),
    nSamples = 4,
    SampleID = "Sample_Name",
    arrayType = ex$arrayType,
    annotationVersion = ex$annotationVersion,
    scriptLabel = "preprocessingMinfiEwasWater",
    baseDataFolder = file.path(ex$tempDir, "rData"),
    figureBaseDir = file.path(ex$tempDir, "figures"),
    detPThreshold = 1,
    normMethods = "quantile",
    sexColumn = "Sex",
    pvalThreshold = 1,
    chrToRemove = "",
    snpsToRemove = "SBE",
    mafThreshold = 1,
    probeExclusionPath = ex$probeExclusionPath,
    plotGroupVar = "Sex",
    lcRef = "saliva",
    phenoOrder = "Sample_Name;Sex;Basename;Sentrix_ID;Sentrix_Position",
    lcPhenoDir = ex$tempDir,
    saveOutputs = FALSE,
    verbose = FALSE,
    logs = FALSE
  )
  inherits(result, "dnaEPICO_preprocessingMinfiEwasWater")
}
```

`preprocessingPheno` *Prepare phenotype and methylation data for modelling*

Description

Match samples in the phenotype table to the beta-value, M-value and copy-number matrices. Prepare tables for individual timepoints and, when selected, combine timepoints for longitudinal analysis. Use the resulting tables with `methylationGLM()` or `methylationLME()`. The function also prepares Clock Foundation export tables and saves outputs when `saveOutputs = TRUE`.

Usage

```
preprocessingPheno(
  phenoFile = "data/preprocessingMinfiEwasWater/phenoLC.csv",
  sepType = NULL,
  betaPath = paste0("rData/preprocessingMinfiEwasWater/metrics/",
    "beta_NomFilt_MSetF_Flt_Rxy_Ds_Rc.RData"),
  mPath = paste0("rData/preprocessingMinfiEwasWater/metrics/",
    "m_NomFilt_MSetF_Flt_Rxy_Ds_Rc.RData"),
  cnPath = paste0("rData/preprocessingMinfiEwasWater/metrics/",
    "cn_NomFilt_MSetF_Flt_Rxy_Ds_Rc.RData"),
  SampleID = "Sample_Name",
  timeVar = "Timepoint",
  timepoints = "1,2",
  combineTimepoints = "1,2",
  methylationScale = "beta",
  outputPheno = "data/preprocessingPheno",
  outputRData = "rData/preprocessingPheno/metrics",
  outputRDataMerge = "rData/preprocessingPheno/mergeData",
  sexColumn = "Sex",
  outputLogs = "logs",
  outputDir = "data/preprocessingPheno",
  verbose = FALSE,
  logs = FALSE,
  saveOutputs = FALSE
)
```

Arguments

`phenoFile`	Character. Path to the phenotype CSV file.
`sepType`	Character or `NULL`. Field separator used in `phenoFile`. Use `NULL` for a comma-separated file, `'\\t'` for a tab-delimited file, or another separator accepted by `utils::read.csv()`.
`betaPath`	Character. Path to the saved beta-value object. Both `.RData` and `.rds` files are supported.
`mPath`	Character. Path to the saved M-value object. Both `.RData` and `.rds` files are supported.
`cnPath`	Character. Path to the saved copy-number object. Both `.RData` and `.rds` files are supported.
`SampleID`	Character. Name of the phenotype column containing sample identifiers used to align phenotype and methylation data.

`timeVar` Character. Name of the phenotype column containing timepoint labels.

`timepoints` Character vector or comma-separated string of timepoints to retain and split into separate in-memory subsets.

`combineTimepoints` Character vector or comma-separated string of timepoints to combine into the longitudinal phenotype-plus-methylation object, or `NULL` to skip the combined object.

`methylationScale` Character. Methylation metric to use in merged modeling tables. One of `'Beta'`, `'M'`, or `'CN'`, in any combination of upper- and lower-case letters. The default is `'beta'`. Beta values are always used for Clock Foundation exports.

`outputPheno` Character. Directory used for saved phenotype CSV files when `saveOutputs = TRUE`.

`outputRData` Character. Directory used for saved metric `.RData` files when `saveOutputs = TRUE`.

`outputRDataMerge` Character. Directory used for saved merged phenotype-plus-methylation `.RData` files when `saveOutputs = TRUE`.

`sexColumn` Character. Name of the phenotype sex column used when building Clock Foundation exports. Values are preserved as supplied, including missing, blank, unknown, or other character values; `PredSex` is not substituted.

`outputLogs` Character. Directory used for log files when `logs = TRUE`.

`outputDir` Character. Directory used for Clock Foundation export files when `saveOutputs = TRUE`.

`verbose` Logical. If `TRUE`, emit progress messages with `message()`. The default is `FALSE`.

`logs` Logical. If `TRUE`, write the same progress messages to `outputLogs`. The default is `FALSE`.

`saveOutputs` Logical. If `TRUE`, write the CSV, ZIP, and `.RData` outputs to disk. The default is `FALSE`.

Value

A list with class `'dnaEPICO_preprocessingPheno'`.

pheno Phenotype table read from `phenoFile`.

metricsData Object returned by `loadMetricsPreprocessingPheno()` containing the beta-value, M-value, and copy-number matrices loaded from `betaPath`, `mPath`, and `cnPath`.

timepointData Object returned by `splitTimepointsPreprocessingPheno()` containing per-timepoint phenotype tables and methylation matrices.

combinedData Object returned by `combineTimepointsPreprocessingPheno()` containing the merged longitudinal phenotype-plus-methylation object and the timepoint combination metadata, or `NULL` when `combineTimepoints = NULL`.

clockFoundation Object returned by `buildClockFoundationInputsPreprocessingPheno()` containing the beta table and phenotype table prepared for Clock Foundation export, with the sex column preserved as supplied.

savedFiles Object returned by `writePreprocessingPhenoOutputs()` when `saveOutputs = TRUE`, otherwise `NULL`.

logFile Resolved path to the optional log file, or `NULL` when logging was disabled.

See dnaEPICO_preprocessingPheno for a class-level overview.

See Also

dnaEPICO_preprocessingPheno

Examples

```
tmp <- tempdir()
pheno <- data.frame(
    Sample_Name = c("S1", "S2", "S3"),
    Timepoint = c("1", "1", "2"),
    Sex = c(0, 1, 0),
    stringsAsFactors = FALSE
)
beta <- matrix(
    c(0.10, 0.20, 0.30, 0.40, 0.50, 0.60),
    nrow = 2,
    dimnames = list(c("cg1", "cg2"), pheno$Sample_Name)
)
m <- beta * 10
cn <- beta * 100
pheno_file <- file.path(tmp, "pheno.csv")
beta_path <- file.path(tmp, "beta.RData")
m_path <- file.path(tmp, "m.RData")
cn_path <- file.path(tmp, "cn.RData")
utils::write.csv(pheno, pheno_file, row.names = FALSE)
save(beta, file = beta_path)
save(m, file = m_path)
save(cn, file = cn_path)
result <- preprocessingPheno(
    phenoFile = pheno_file,
    betaPath = beta_path,
    mPath = m_path,
    cnPath = cn_path,
    SampleID = "Sample_Name",
    timeVar = "Timepoint",
    timepoints = "1,2",
    combineTimepoints = "1,2",
    outputPheno = file.path(tmp, "data", "preprocessingPheno"),
    outputRData = file.path(tmp, "rData", "preprocessingPheno", "metrics"),
    outputRDataMerge = file.path(
        tmp, "rData", "preprocessingPheno", "mergeData"
    ),
    sexColumn = "Sex",
    outputLogs = file.path(tmp, "logs"),
    outputDir = file.path(tmp, "clockFoundation"),
    saveOutputs = FALSE
```

```
)
stopifnot(inherits(result, "dnaEPICO_preprocessingPheno"))
```

`print.dnaEPICO_dnamReport` *Print a DNA methylation report result*

Description

Print a DNA methylation report result

Usage

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

Arguments

`x`	Object returned by `dnamReport()`.
`...`	Additional arguments ignored.

Value

Invisibly returns `x`.

`readPhenotypeTargets` *Read phenotype targets for shared dnaEPICO workflows*

Description

Read the phenotype table used by shared `dnaEPICO` workflows, validate the sample identifier column, optionally subset the first `nSamples`, and return the targets as a base `data.frame`.

Usage

```
readPhenotypeTargets(
  phenoFile,
  sepType = NULL,
  nSamples = NA,
  SampleID = "Sample_Name",
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_readPhenotypeTargets.txt"
)
```

Arguments

phenoFile	Character. Path to the phenotype table on disk.
sepType	Character or NULL. Field separator used in phenoFile. Use NULL for a standard comma-separated file, '\\t' for a tab-delimited file, or another single-character separator accepted by utils::read.csv().
nSamples	Integer or NA. Number of rows to keep from the start of the phenotype table. The default NA reads and returns all rows.
SampleID	Character. Name of the column containing sample identifiers that will later be used to name methylation-array samples.
verbose	Logical. If TRUE, emit progress and preview messages with message(). The default is FALSE.
logs	Logical. If TRUE, write the same progress messages to a log file. The default is FALSE.
log_dir	Character or NULL. Directory where the log file should be written when logs = TRUE. If NULL, the current working directory is used.
log_file	Character. File name used when logs = TRUE. The default is 'log_readPhenotypeTargets.txt'.

Value

A data.frame containing the phenotype targets.

Examples

```
tmp <- tempdir()
pheno <- data.frame(
    Sample_Name = c("S1", "S2"),
    Sex = c("F", "M"),
    stringsAsFactors = FALSE
)
pheno_file <- file.path(tmp, "pheno.csv")
utils::write.csv(pheno, pheno_file, row.names = FALSE)
targets <- readPhenotypeTargets(
    phenoFile = pheno_file,
    SampleID = "Sample_Name"
)
stopifnot(is.data.frame(targets))
stopifnot(nrow(targets) == 2L)
```

---

readRGSetMinfiEwasWater

Read IDAT files into an annotated RGChannelSet

Description

Read methylation-array IDAT files with minfi::read.metharray.exp(), set sample names from the phenotype table, apply the requested annotation, and return the resulting RGChannelSet.

Usage

```
readRGSetMinfiEwasWater(
  idatFolder,
  targets,
  SampleID = "Sample_Name",
  arrayType = "IlluminaHumanMethylationEPICv2",
  annotationVersion = "20a1.hg38",
  force = FALSE,
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_readRGSetMinfiEwasWater.txt"
)
```

Arguments

idatFolder	Character. Directory containing the IDAT files.
targets	Data frame returned by readPhenotypeTargets() or an equivalent phenotype table.
SampleID	Character. Name of the phenotype column containing sample identifiers used to label the RGChannelSet.
arrayType	Character. Array name passed to Biobase::annotation(RGSet), for example 'IlluminaHumanMethylationEPICv2'.
annotationVersion	Character. Annotation build passed to Biobase::annotation(RGSet), for example '20a1.hg38' for EPIC v2 hg38 annotations or 'ilmn12.hg19' for 450K hg19 annotations.
force	Logical. Passed to minfi::read.metharray.exp(). Use TRUE only after confirming that the selected IDAT files should be read together.
verbose	Logical. If TRUE, emit progress messages with message().
logs	Logical. If TRUE, write the same messages to a log file.
log_dir	Character or NULL. Directory used for the log file when logs = TRUE.
log_file	Character. File name used when logs = TRUE.

Value

An annotated RGChannelSet.

Examples

```
if (requireNamespace("minfiData", quietly = TRUE) &&
  requireNamespace(
    "IlluminaHumanMethylation450kmanifest",
    quietly = TRUE
  ) &&
  requireNamespace(
```

```
    "IlluminaHumanMethylation450kanno.ilmn12.hg19",
    quietly = TRUE
  )) {
  ex <- dnaEPICO:::exampleMinfiIdatInputsDnaEpico(n = 4)
  rgset <- readRGSetMinfiEwasWater(
    idatFolder = ex$idatFolder,
    targets = ex$targets,
    SampleID = "Sample_Name",
    arrayType = ex$arrayType,
    annotationVersion = ex$annotationVersion
  )
  class(rgset)
}
```

`renderDnamReport` *Render a prepared DNA methylation report*

Description

Render a prepared DNA methylation report

Usage

```
renderDnamReport(
  preparedReport,
  verbose = FALSE,
  logs = FALSE,
  logDir = NULL,
  clean = TRUE
)
```

Arguments

`preparedReport`	Object returned by `prepareDnamReportInputs()`.
`verbose`	Logical. If `TRUE`, emit progress messages.
`logs`	Logical. If `TRUE`, write progress messages to a log file.
`logDir`	Character or `NULL`. Directory for optional log files.
`clean`	Logical. Retained for backwards compatibility.

Value

A list with class `'dnaEPICO_dnamReport_render'`.

Examples

```
report_root <- file.path(tempdir(), "dnaepico-render-example")
prepared <- prepareDnamReportInputs(
  outputDir = file.path(report_root, "reports")
)
rendered <- renderDnamReport(prepared)
rendered$status
```

---

splitTimepointsPreprocessingPheno

Split phenotype and methylation data by timepoint

Description

Align phenotype rows and metric matrices for each requested timepoint, and precompute the per-timepoint phenotype-plus-beta objects used by downstream modeling functions.

Usage

```
splitTimepointsPreprocessingPheno(
  pheno,
  metricsData,
  SampleID = "Sample_Name",
  timeVar = "Timepoint",
  timepoints = "1,2",
  methylationScale = "beta",
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_splitTimepointsPreprocessingPheno.txt"
)
```

Arguments

pheno	Data frame containing phenotype information.
metricsData	Object returned by `loadMetricsPreprocessingPheno()`.
SampleID	Character. Name of the sample identifier column in `pheno`.
timeVar	Character. Name of the timepoint column in `pheno`.
timepoints	Character vector or comma-separated string of timepoints to retain.
methylationScale	Character. Methylation metric to use for the merged modeling table. One of `'Beta'`, `'M'`, or `'CN'`, case-insensitively.
verbose	Logical. If `TRUE`, emit progress messages with `message()`.
logs	Logical. If `TRUE`, write the same messages to a log file.
log_dir	Character or `NULL`. Directory used for the log file when `logs = TRUE`.
log_file	Character. File name used when `logs = TRUE`.

Value

A list with class 'dnaEPICO_preprocessingPheno_timepoints' containing the parsed timepoints and aligned per-timepoint subsets.

Examples

```
ex <- dnaEPICO:::examplePreprocessingPhenoStateDnaEpico()
timepoint_data <- splitTimepointsPreprocessingPheno(
  pheno = ex$pheno,
  metricsData = ex$metricsData,
  SampleID = "Sample_Name",
  timeVar = "Timepoint",
  timepoints = "1,2",
  verbose = FALSE,
  logs = FALSE
)
timepoint_data$timepoints
```

---

summarizeMethylationGLMModels

Summarize CpG-wise Gaussian GLM results for one-timepoint analyses

---

Description

Return phenotype-specific CpG coefficient tables from the compact fit-time results produced by `fitMethylationGLMModels()`.

Usage

```
summarizeMethylationGLMModels(
  modelResults,
  preparedData,
  summaryResidualSD = TRUE,
  summaryPval = NA,
  padjmethod = "fdr",
  nCores = 1L,
  libPath = NULL,
  glmLibs = "glm2",
  chunkSize = NULL,
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_methylationGLM.txt"
)
```

Arguments

modelResults	Object returned by fitMethylationGLMModels().
preparedData	Object returned by prepareMethylationGLMData().
summaryResidualSD	Logical. If TRUE, add residual standard deviations to each CpG summary row.
summaryPval	Numeric or NA. Optional p-value filter applied to the returned summary tables. NA keeps all rows.
padjmethod	Character. Adjustment method passed to stats::p.adjust() for omnibus p-values across CpGs within each phenotype and tested term.
nCores	Integer. Maximum number of R worker processes used to extract summary rows, limited by available CPU cores. With the serial setting, summarisation uses one process.
libPath	Character vector or NULL. Optional library paths forwarded to worker processes.
glmLibs	Character vector or comma-separated string of package names to check on worker processes. The default is 'glm2'.
chunkSize	Integer or NULL. Number of CpGs to process per parallel chunk. NULL chooses a value automatically.
verbose	Logical. If TRUE, emit progress messages with message().
logs	Logical. If TRUE, write the same messages to a log file.
log_dir	Character or NULL. Directory used for the log file when logs = TRUE.
log_file	Character. File name used when logs = TRUE.

Value

A list with class 'dnaEPICO_methylationGLM_summaries' containing the optionally filtered summary tables in summaries and the complete CpG-level tables in diagnosticSummaries. Diagnostics, annotation, and report output use the complete tables so summaryPval does not remove CpGs from those outputs. modelMessages retains model messages, warnings, and errors for every attempted CpG.

Examples

```
ex <- dnaEPICO:::exampleMethylationGLMStateDnaEpico()
summary_results <- summarizeMethylationGLMModels(
  modelResults = ex$modelResults,
  preparedData = ex$preparedData,
  summaryResidualSD = TRUE,
  summaryPval = NA,
  nCores = 1,
  verbose = FALSE,
  logs = FALSE
)
names(summary_results$summaries)
```

summarizeMethylationLMEModels
Summarize CpG-wise mixed-effects results for longitudinal analyses

Description

Return phenotype-specific fixed-effect tables from the compact fit-time results produced by `fitMethylationLMEModels()`.

Usage

```
summarizeMethylationLMEModels(
  modelResults,
  preparedData,
  summaryPval = NA,
  padjmethod = "fdr",
  nCores = 1L,
  chunkSize = NULL,
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_methylationLME.txt"
)
```

Arguments

`modelResults`	Object returned by `fitMethylationLMEModels()`.
`preparedData`	Object returned by `prepareMethylationLMEData()`.
`summaryPval`	Numeric or `NA`. Optional p-value filter applied to the returned summary tables. `NA` keeps all rows.
`padjmethod`	Character. Adjustment method passed to `stats::p.adjust()` for omnibus p-values across CpGs within each phenotype and tested term.
`nCores`	Integer. Maximum number of R worker processes used to extract summary rows, limited by available CPU cores. With the `serial` setting, summarisation uses one process.
`chunkSize`	Integer or `NULL`. Number of CpGs processed per parallel chunk. `NULL` chooses a value automatically.
`verbose`	Logical. If `TRUE`, emit progress messages with `message()`.
`logs`	Logical. If `TRUE`, write the same messages to a log file.
`log_dir`	Character or `NULL`. Directory used for the log file when `logs = TRUE`.
`log_file`	Character. File name used when `logs = TRUE`.

Value

A list with class 'dnaEPICO_methylationLME_summaries' containing the optionally filtered summary tables in summaries and the complete CpG-level tables in diagnosticSummaries. Diagnostics, annotation, and report output use the complete tables so summaryPval does not remove CpGs from those outputs. modelMessages retains model messages, warnings, and errors for every attempted CpG.

Examples

```
ex <- dnaEPICO:::exampleMethylationLMEStateDnaEpico()
summary_results <- summarizeMethylationLMEModels(
  modelResults = ex$modelResults,
  preparedData = ex$preparedData,
  summaryPval = NA,
  nCores = 1,
  verbose = FALSE,
  logs = FALSE
)
names(summary_results$summaries)
```

---

summarizeTimepointsMethylationLME

Summarize phenotype values by timepoint for longitudinal methylation analyses

---

Description

Summarize the requested phenotype variables by timepoint. Numeric phenotypes are reported with mean, standard deviation, and non-missing counts; non-numeric phenotypes are reported with non-missing counts and the observed levels.

Usage

```
summarizeTimepointsMethylationLME(
  data,
  timeVar = "Timepoint",
  phenotypes,
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_methylationLME.txt"
)
```

Arguments

data	Data frame containing the longitudinal phenotype-plus-beta data.
timeVar	Character. Name of the time variable.
phenotypes	Character vector or comma-separated string of phenotype variables to summarize.
verbose	Logical. If TRUE, emit progress messages with message().
logs	Logical. If TRUE, write the same messages to a log file.
log_dir	Character or NULL. Directory used for the log file when logs = TRUE.
log_file	Character. File name used when logs = TRUE.

Value

A data frame with one row per timepoint and summary columns for each requested phenotype.

Examples

```
ex <- dnaEPICO:::exampleMethylationLMEStateDnaEpico()
timepoint_summary <- summarizeTimepointsMethylationLME(
  data = ex$preparedData$data,
  timeVar = "Timepoint",
  phenotypes = "score",
  verbose = FALSE,
  logs = FALSE
)
nrow(timepoint_summary)
```

---

svaEnmix *Estimate surrogate variables from ENmix control probes*

Description

Estimate surrogate variables from control probes using ENmix and assess their associations with Sentrix chip and position. With saveOutputs = TRUE, add the surrogate-variable columns to phenoFile and save the SVA results and figures. The updated phenotype table is used by preprocessingPheno().

Usage

```
svaEnmix(
  phenoFile = "data/preprocessingMinfiEwasWater/phenoLC.csv",
  rgsetData = "rData/preprocessingMinfiEwasWater/objects/RGSet.RData",
  sepType = NULL,
  outputLogs = "logs",
  nSamples = NA,
  SampleID = "Sample_Name",
  arrayType = "IlluminaHumanMethylationEPICv2",
```

```
  annotationVersion = "20a1.hg38",
  SentrixIDColumn = "Sentrix_ID",
  SentrixPositionColumn = "Sentrix_Position",
  ctrlSvaPercVar = 0.9,
  ctrlSvaFlag = 1,
  scriptLabel = "svaEnmix",
  tiffWidth = 2000,
  tiffHeight = 1000,
  tiffRes = 150,
  figureBaseDir = "figures",
  dataBaseDir = "data",
  rBaseDir = "rData",
  display = FALSE,
  verbose = FALSE,
  logs = FALSE,
  saveOutputs = FALSE
)
```

Arguments

phenoFile	Character. Path to the phenotype file with cell-composition data. When saveOutputs = TRUE, add the surrogate-variable columns to this file. The original file is restored if saving the replacement fails.
rgsetData	Character. Path to a saved RGChannelSet object. Both .RData and .rds files are supported.
sepType	Character or NULL. Field separator used in phenoFile. Use NULL for a comma-separated file, '\\t' for a tab-delimited file, or another separator accepted by utils::read.csv().
outputLogs	Character. Directory used for log files when logs = TRUE.
nSamples	Integer or NA. Number of rows to keep from the phenotype table. Use NA to keep all samples.
SampleID	Character. Name of the phenotype column containing sample identifiers.
arrayType	Character. Illumina array identifier assigned to Biobase::annotation(RGSet).
annotationVersion	Character. Annotation build assigned to Biobase::annotation(RGSet).
SentrixIDColumn	Character. Name of the chip identifier column in the phenotype data.
SentrixPositionColumn	Character. Name of the chip position column in the phenotype data.
ctrlSvaPercVar	Numeric. Proportion of control-probe variance explained when running ENmix::ctrlsva().
ctrlSvaFlag	Integer. Control-probe flag passed to ENmix::ctrlsva().
scriptLabel	Character. Label used to name output folders when saveOutputs = TRUE.
tiffWidth	Integer. Width of saved TIFF plots in pixels.
tiffHeight	Integer. Height of saved TIFF plots in pixels.
tiffRes	Integer. Resolution in DPI for saved TIFF plots.

figureBaseDir	Character. Base directory used for saved figure outputs when saveOutputs = TRUE.
dataBaseDir	Character. Base directory used for saved CSV and text outputs when saveOutputs = TRUE.
rBaseDir	Character. Base directory used for saved .RData outputs when saveOutputs = TRUE.
display	Logical. If TRUE, draw plots on the active graphics device.
verbose	Logical. If TRUE, emit progress messages with message(). The default is FALSE.
logs	Logical. If TRUE, write the same progress messages to outputLogs. The default is FALSE.
saveOutputs	Logical. If TRUE, write the CSV, .RData, text, and TIFF outputs to disk. The default is FALSE.

Value

A list with class 'dnaEPICO_svaEnmix'.

targets Phenotype table read from phenoFile after any optional row subsetting.

RGSet Loaded RGChannelSet with sample names realigned to targets[[SampleID]].

svaData Object returned by estimateSvaEnmixControls() containing the surrogate-variable matrix and the control-probe settings used to estimate it.

mergedPheno Phenotype table returned by mergeSvaTargetsEnmix() after the surrogate variables were appended as additional columns.

analysisData Object returned by analyzeSvaEnmix() containing the surrogate-variable association models, ANOVA tables, and Sentrix metadata.

plotFiles Named list describing the plot file paths requested for the SVA figures. When saveOutputs = FALSE, the entries are typically NULL.

savedFiles Object returned by writeSvaEnmixOutputs() when saveOutputs = TRUE, otherwise NULL.

logFile Resolved path to the optional log file, or NULL when logging was disabled.

See dnaEPICO_svaEnmix for a class-level overview.

See Also

dnaEPICO_svaEnmix

Examples

```
tmp <- tempdir()
stopifnot(dir.exists(tmp))

if (requireNamespace("minfiData", quietly = TRUE)) {
    ex <- dnaEPICO:::exampleMinfiBaseDataDnaEpico()
    pheno_file <- file.path(tmp, "pheno.csv")
    rgset_path <- file.path(tmp, "RGSet.RData")
```

```
    RGSet <- ex$RGSet
    utils::write.csv(ex$targets, pheno_file, row.names = FALSE)
    save(RGSet, file = rgset_path)
    sva_result <- svaEnmix(
        phenoFile = pheno_file,
        rgsetData = rgset_path,
        SampleID = "Sample_Name",
        arrayType = "IlluminaHumanMethylation450k",
        annotationVersion = "ilmn12.hg19",
        SentrixIDColumn = "Sentrix_ID",
        SentrixPositionColumn = "Sentrix_Position",
        outputLogs = file.path(tmp, "logs"),
        figureBaseDir = file.path(tmp, "figures"),
        dataBaseDir = file.path(tmp, "data"),
        rBaseDir = file.path(tmp, "rData"),
        saveOutputs = FALSE
    )
    stopifnot(inherits(sva_result, "dnaEPICO_svaEnmix"))
}
```

`writeMethylationGLMOutputs` *Write optional disk outputs for one-timepoint GLM analyses*

Description

Write compact phenotype summaries, optional text and significant-CpG tables, and annotated results from the one-timepoint GLM workflow.

Usage

```
writeMethylationGLMOutputs(
  modelResults,
  modelSummaries,
  annotatedResults,
  significantCpGs = NULL,
  outputRData,
  summaryTxtDir,
  significantCpGDir,
  annotatedGLMOut,
  reportAssetsDir = NULL,
  vennDResults = NULL,
  saveTxtSummaries = TRUE,
  saveSignificantCpGs = FALSE,
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_methylationGLM.txt"
)
```

Arguments

modelResults	Object returned by fitMethylationGLMModels().
modelSummaries	Object returned by summarizeMethylationGLMModels().
annotatedResults	Object returned by annotateMethylationGLMSummaries() or a compatible data frame.
significantCpGs	Object returned by collectSignificantCpGsMethylationGLM() or NULL.
outputRData	Character. Directory used for complete compact phenotype summaries.
summaryTxtDir	Character. Directory used for tab-delimited summary tables.
significantCpGDir	Character. Directory used for significant-CpG coefficient tables.
annotatedGLMOut	Character. Directory used for the annotated summary XLSX workbook.
reportAssetsDir	Character or NULL. Report results directory used for the compressed TSV table and compact metadata files.
vennDResults	Optional model-level Venn result. Its configuration metadata and threshold tables are added before the workbook dictionary.
saveTxtSummaries	Logical. If TRUE, write tab-delimited summary tables.
saveSignificantCpGs	Logical. If TRUE, write significant-CpG coefficient tables.
verbose	Logical. If TRUE, emit progress messages with message().
logs	Logical. If TRUE, write the same messages to a log file.
log_dir	Character or NULL. Directory used for the log file when logs = TRUE.
log_file	Character. File name used when logs = TRUE.

Value

A list with class 'dnaEPICO_methylationGLM_paths' containing the paths of the files written to disk, including the annotated workbook and any requested report data files.

Examples

```
ex <- dnaEPICO:::exampleMethylationGLMStateDnaEpico()
annotation_data <- annotateMethylationGLMSummaries(
  modelSummaries = ex$modelSummaries,
  annotationObject = ex$annotationData,
  annotationCols = "Name,chr,pos",
  verbose = FALSE,
  logs = FALSE
)
significant_cpgs <- collectSignificantCpGsMethylationGLM(
  modelResults = ex$modelResults,
```

```
  pvalThreshold = 1,
  verbose = FALSE,
  logs = FALSE
)
output_paths <- writeMethylationGLMOutputs(
  modelResults = ex$modelResults,
  modelSummaries = ex$modelSummaries,
  annotatedResults = annotation_data,
  significantCpGs = significant_cpgs,
  outputRData = file.path(ex$tempDir, "models"),
  summaryTxtDir = file.path(ex$tempDir, "summary"),
  significantCpGDir = file.path(ex$tempDir, "significant"),
  annotatedGLMOut = file.path(ex$tempDir, "annotated"),
  saveTxtSummaries = TRUE,
  saveSignificantCpGs = TRUE,
  verbose = FALSE,
  logs = FALSE
)
names(output_paths)
```

`writeMethylationLMEOutputs` *Write optional disk outputs for longitudinal mixed-effects analyses*

Description

Write compact phenotype summaries, optional text and significant-interaction tables, and annotated results from the longitudinal mixed-effects workflow.

Usage

```
writeMethylationLMEOutputs(
  modelResults,
  modelSummaries,
  annotatedResults,
  significantInteractions = NULL,
  outputRData,
  summaryTxtDir,
  significantInteractionDir,
  annotatedLMEOut,
  reportAssetsDir = NULL,
  vennDResults = NULL,
  saveTxtSummaries = TRUE,
  saveSignificantInteractions = FALSE,
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_methylationLME.txt"
)
```

Arguments

modelResults	Object returned by fitMethylationLMEModels().
modelSummaries	Object returned by summarizeMethylationLMEModels().
annotatedResults	Object returned by annotateMethylationLMESummaries() or a compatible data frame.
significantInteractions	Object returned by collectSignificantInteractionsMethylationLME() or NULL.
outputRData	Character. Directory used for complete compact phenotype summaries.
summaryTxtDir	Character. Directory used for tab-delimited summary tables.
significantInteractionDir	Character. Directory used for significant interaction coefficient tables.
annotatedLMEOut	Character. Directory used for the annotated summary XLSX workbook.
reportAssetsDir	Character or NULL. Report results directory used for the compressed TSV table and compact metadata files.
vennDResults	Optional model-level Venn result. Its configuration metadata and threshold tables are added before the workbook dictionary.
saveTxtSummaries	Logical. If TRUE, write tab-delimited summary tables.
saveSignificantInteractions	Logical. If TRUE, write significant interaction coefficient tables.
verbose	Logical. If TRUE, emit progress messages with message().
logs	Logical. If TRUE, write the same messages to a log file.
log_dir	Character or NULL. Directory used for the log file when logs = TRUE.
log_file	Character. File name used when logs = TRUE.

Value

A list with class 'dnaEPICO_methylationLME_paths' containing the paths of the files written to disk, including the annotated workbook and any requested report data files.

Examples

```
ex <- dnaEPICO:::exampleMethylationLMEStateDnaEpico()
annotation_data <- annotateMethylationLMESummaries(
  modelSummaries = ex$modelSummaries,
  annotationObject = ex$annotationData,
  annotationCols = "Name,chr,pos",
  verbose = FALSE,
  logs = FALSE
)
significant_hits <- collectSignificantInteractionsMethylationLME(
```

```
  modelResults = ex$modelResults,
  pvalThreshold = 1,
  verbose = FALSE,
  logs = FALSE
)
output_paths <- writeMethylationLMEOutputs(
  modelResults = ex$modelResults,
  modelSummaries = ex$modelSummaries,
  annotatedResults = annotation_data,
  significantInteractions = significant_hits,
  outputRData = file.path(ex$tempDir, "models"),
  summaryTxtDir = file.path(ex$tempDir, "summary"),
  significantInteractionDir = file.path(ex$tempDir, "significant"),
  annotatedLMEOut = file.path(ex$tempDir, "annotated"),
  saveTxtSummaries = TRUE,
  saveSignificantInteractions = TRUE,
  verbose = FALSE,
  logs = FALSE
)
names(output_paths)
```

`writePhenoLCMinfiEwasWater` *Write the merged phenotype plus cell-composition table*

Description

Write the merged phenotype plus cell-composition table

Usage

```
writePhenoLCMinfiEwasWater(
  lcData,
  file,
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_writePhenoLCMinfiEwasWater.txt"
)
```

Arguments

`lcData`	Object returned by `estimateLCMinfiEwasWater()`.
`file`	Character. Path to the CSV file to write.
`verbose`	Logical. If `TRUE`, emit progress messages with `message()`.
`logs`	Logical. If `TRUE`, write the same messages to a log file.
`log_dir`	Character or `NULL`. Directory used for the log file when `logs = TRUE`.
`log_file`	Character. File name used when `logs = TRUE`.

Value

Invisibly returns `file`.

Examples

```
ref_file <- system.file("extdata", "saliva.txt", package = "dnaEPICO")
beta <- as.matrix(utils::read.table(ref_file))[1:20, , drop = FALSE]
colnames(beta) <- c("sample1", "sample2")
targets <- data.frame(
    Sample_Name = colnames(beta),
    Timepoint = c("T1", "T2"),
    stringsAsFactors = FALSE
)
lc_data <- estimateLCMinfiEwasWater(
    beta = beta,
    targets = targets,
    lcRef = "saliva",
    phenoOrder = "Sample_Name;Timepoint"
)
output_file <- file.path(tempdir(), "phenoLC.csv")
writePhenoLCMinfiEwasWater(lcData = lc_data, file = output_file)
file.exists(output_file)
```

`writePreprocessingPhenoOutputs` *Write preprocessingPheno outputs to disk*

Description

Write the CSV, ZIP, and `.RData` outputs produced by `preprocessingPheno()`. This helper keeps file writing separate from the in-memory preprocessing steps.

Usage

```
writePreprocessingPhenoOutputs(
  preprocessingData,
  outputPheno = "data/preprocessingPheno",
  outputRData = "rData/preprocessingPheno/metrics",
  outputRDataMerge = "rData/preprocessingPheno/mergeData",
  outputDir = "data/preprocessingPheno",
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_writePreprocessingPhenoOutputs.txt"
)
```

Arguments

preprocessingData	Object returned by `preprocessingPheno()` or a list with the same components.
outputPheno	Character. Directory used for saved phenotype CSV files.
outputRData	Character. Directory used for saved metric `.RData` files.
outputRDataMerge	Character. Directory used for saved merged phenotype-plus-beta `.RData` files.
outputDir	Character. Directory used for the Clock Foundation export files.
verbose	Logical. If `TRUE`, emit progress messages with `message()`.
logs	Logical. If `TRUE`, write the same messages to a log file.
log_dir	Character or `NULL`. Directory used for the log file when `logs = TRUE`.
log_file	Character. File name used when `logs = TRUE`.

Value

A list with class `'dnaEPICO_preprocessingPheno_paths'` containing the paths written to disk.

Examples

```
ex <- dnaEPICO:::examplePreprocessingPhenoStateDnaEpico()
output_paths <- writePreprocessingPhenoOutputs(
  preprocessingData = ex$preprocessingData,
  outputPheno = file.path(ex$tempDir, "pheno"),
  outputRData = file.path(ex$tempDir, "metrics"),
  outputRDataMerge = file.path(ex$tempDir, "merge"),
  outputDir = file.path(ex$tempDir, "clock"),
  verbose = FALSE,
  logs = FALSE
)
names(output_paths)
```

---

writeSvaEnmixOutputs *Write svaEnmix outputs to disk*

Description

Write the SVA matrix, phenotype, and model-summary outputs.

Usage

```
writeSvaEnmixOutputs(
  svaData,
  mergedPheno,
  analysisData = NULL,
  phenoFile = NULL,
```

```
  SampleID = "Sample_Name",
  sepType = NULL,
  dataBaseDir = "data",
  rBaseDir = "rData",
  scriptLabel = "svaEnmix",
  verbose = FALSE,
  logs = FALSE,
  log_dir = NULL,
  log_file = "log_writeSvaEnmixOutputs.txt"
)
```

Arguments

svaData	Object returned by `estimateSvaEnmixControls()`.
mergedPheno	Phenotype data frame returned by `mergeSvaTargetsEnmix()`.
analysisData	Optional object returned by `analyzeSvaEnmix()`.
phenoFile	Character or `NULL`. Existing phenotype file replaced with `mergedPheno` after validation.
SampleID	Character. Phenotype column containing sample identifiers.
sepType	Character or `NULL`. Field separator used by `phenoFile`.
dataBaseDir	Character. Base directory used for saved data outputs.
rBaseDir	Character. Base directory used for saved `.RData` outputs.
scriptLabel	Character. Label used to create the output subdirectory.
verbose	Logical. If `TRUE`, emit progress messages with `message()`.
logs	Logical. If `TRUE`, write the same messages to a log file.
log_dir	Character or `NULL`. Directory used for the log file when `logs = TRUE`.
log_file	Character. File name used when `logs = TRUE`.

Value

A list with class `'dnaEPICO_svaEnmix_paths'` containing the paths written to disk.

Examples

```
ex <- dnaEPICO:::exampleSvaAnalysisStateDnaEpico()
temp_dir <- tempdir()
pheno_file <- file.path(temp_dir, "phenoLC.csv")
utils::write.csv(ex$targets, pheno_file, row.names = FALSE)
output_paths <- writeSvaEnmixOutputs(
  svaData = list(sva = ex$sva),
  mergedPheno = ex$mergedPheno,
  analysisData = ex$analysisData,
  phenoFile = pheno_file,
  SampleID = "Sample_Name",
  dataBaseDir = file.path(temp_dir, "data"),
  rBaseDir = file.path(temp_dir, "rData"),
```

```
  scriptLabel = "svaEnmixExample",
  verbose = FALSE,
  logs = FALSE
)
names(output_paths)
```

Index

∗ **package**
dnaEPICO, 14

analyzeSvaEnmix, 4
analyzeSvaEnmix(), *20*, *87*
annotateMethylationGLMSummaries, 5
annotateMethylationGLMSummaries(), *17*, *41*
annotateMethylationLMESummaries, 6
annotateMethylationLMESummaries(), *18*, *47*
assessSamplesMinfiEwasWater, 7
assessSamplesMinfiEwasWater(), *18*, *71*

buildClockFoundationInputsPreprocessingPheno, 8
buildClockFoundationInputsPreprocessingPheno(), *19*, *74*
buildRawMinfiEwasWater, 10
buildRawMinfiEwasWater(), *18*, *71*

collectSignificantCpGsMethylationGLM, 11
collectSignificantCpGsMethylationGLM(), *17*, *41*
collectSignificantInteractionsMethylationLME, 12
collectSignificantInteractionsMethylationLME(), *18*, *46*
combineTimepointsPreprocessingPheno, 13
combineTimepointsPreprocessingPheno(), *19*, *74*

dnaEPICO, 14
dnaEPICO-package *(*dnaEPICO*)*, 14
dnaEPICO_dnamReport *(*dnaEPICO_dnamReport-class*)*, 15
dnaEPICO_dnamReport-class, 15
dnaEPICO_dnamReport_prepared *(*dnaEPICO_dnamReport_prepared-class*)*, 16
dnaEPICO_dnamReport_prepared-class, 16
dnaEPICO_dnamReport_render *(*dnaEPICO_dnamReport_render-class*)*, 16
dnaEPICO_dnamReport_render-class, 16
dnaEPICO_methylationGLM, *41*
dnaEPICO_methylationGLM *(*dnaEPICO_methylationGLM-class*)*, 17
dnaEPICO_methylationGLM-class, 17
dnaEPICO_methylationLME, *47*
dnaEPICO_methylationLME *(*dnaEPICO_methylationLME-class*)*, 17
dnaEPICO_methylationLME-class, 17
dnaEPICO_preprocessingMinfiEwasWater, *71*, *72*
dnaEPICO_preprocessingMinfiEwasWater *(*dnaEPICO_preprocessingMinfiEwasWater-class*)*, 18
dnaEPICO_preprocessingMinfiEwasWater-class, 18
dnaEPICO_preprocessingPheno, *75*
dnaEPICO_preprocessingPheno *(*dnaEPICO_preprocessingPheno-class*)*, 19
dnaEPICO_preprocessingPheno-class, 19
dnaEPICO_svaEnmix, *87*
dnaEPICO_svaEnmix *(*dnaEPICO_svaEnmix-class*)*, 19
dnaEPICO_svaEnmix-class, 19
dnamReport, 20
dnamReport(), *14*, *15*, *37*, *42*, *76*

estimateLC, 23
estimateLCMinfiEwasWater, 24
estimateLCMinfiEwasWater(), *18*, *71*

estimateSvaEnmixControls, 26
estimateSvaEnmixControls(), *20*, *87*
extractMake, 27
extractMetricsMinfiEwasWater, 28
extractMetricsMinfiEwasWater(), *18*, *71*

filterProbesMinfiEwasWater, 29
filterProbesMinfiEwasWater(), *18*, *71*
filterSamplesMinfiEwasWater, 31
fitMethylationGLMModels, 32
fitMethylationGLMModels(), *17*, *41*
fitMethylationLMEModels, 33
fitMethylationLMEModels(), *18*, *46*

loadMetricsPreprocessingPheno, 35
loadMetricsPreprocessingPheno(), *19*, *74*

mergeSvaTargetsEnmix, 36
mergeSvaTargetsEnmix(), *20*, *87*
methylationGLM, 37
methylationGLM(), *14*, *17*, *73*
methylationLME, 42
methylationLME(), *14*, *17, 18*, *73*

normalizeMinfiEwasWater, 48
normalizeMinfiEwasWater(), *18*, *71*

plotAssessmentMinfiEwasWater, 49
plotCtrlMinfiEwasWater, 50
plotMethylationGLMDiagnostics, 51
plotMethylationGLMDiagnostics(), *17*, *41*
plotMethylationGLMDistributions, 53
plotMethylationGLMDistributions(), *17*, *41*
plotMethylationLMEDiagnostics, 54
plotMethylationLMEDiagnostics(), *18*, *47*
plotMetricsMinfiEwasWater, 55
plotNormalizationMinfiEwasWater, 57
plotRawDensityMinfiEwasWater, 58
plotSexMinfiEwasWater, 59
plotSvaEnmix, 61
predictSexMinfiEwasWater, 62
predictSexMinfiEwasWater(), *18*, *71*
prepareDnamReportInputs, 63
prepareDnamReportInputs(), *15, 16*
prepareMethylationGLMData, 65
prepareMethylationGLMData(), *17*, *41*
prepareMethylationLMEData, 66
prepareMethylationLMEData(), *18*, *46*

preprocessingMinfiEwasWater, 68
preprocessingMinfiEwasWater(), *14*, *18*, *19*
preprocessingPheno, 72
preprocessingPheno(), *14*, *19*, *68*, *85*
print.dnaEPICO_dnamReport, 76

readPhenotypeTargets, 76
readRGSetMinfiEwasWater, 77
renderDnamReport, 79
renderDnamReport(), *16*

splitTimepointsPreprocessingPheno, 80
splitTimepointsPreprocessingPheno(), *19*, *74*
summarizeMethylationGLMModels, 81
summarizeMethylationGLMModels(), *17*, *41*
summarizeMethylationLMEModels, 83
summarizeMethylationLMEModels(), *18*, *46*
summarizeTimepointsMethylationLME, 84
svaEnmix, 85
svaEnmix(), *14*, *19, 20*, *68*

writeMethylationGLMOutputs, 88
writeMethylationGLMOutputs(), *17*, *41*
writeMethylationLMEOutputs, 90
writeMethylationLMEOutputs(), *18*, *47*
writePhenoLCMinfiEwasWater, 92
writePreprocessingPhenoOutputs, 93
writePreprocessingPhenoOutputs(), *19*, *75*
writeSvaEnmixOutputs, 94
writeSvaEnmixOutputs(), *20*, *87*