Package ‘PathwayVote’

July 21, 2025

Title Robust Pathway Enrichment for DNA Methylation Studies Using Ensemble Voting

Version 0.1.1

Description Performs pathway enrichment analysis using a voting-based framework that integrates CpG–gene regulatory information from expression quantitative trait methylation (eQTM) data. For a grid of top-ranked CpGs and filtering thresholds, gene sets are generated and refined using an entropy-based pruning strategy that balances information richness, stability, and probe bias correction. In particular, gene lists dominated by genes with disproportionately high numbers of CpG mappings are penalized to mitigate active probe bias—a common artifact in methylation data analysis. Enrichment results across parameter combinations are then aggregated using a voting scheme, prioritizing pathways that are consistently recovered under diverse settings and robust to parameter perturbations.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.0.0)

Imports AnnotationDbi, clusterProfiler, future, furrr, GO.db, methods, org.Hs.eg.db, parallelly, reactome.db

Suggests testthat

RoxygenNote 7.3.2

NeedsCompilation no

Author Yinan Zheng [aut, cre] (ORCID: <https://orcid.org/0000-0002-2006-7320>)

Maintainer Yinan Zheng <y-zheng@northwestern.edu>

Repository CRAN

Date/Publication 2025-06-25 12:20:34 UTC

Contents

`create_eQTM` *Create an expression quantitative trait methylation (eQTM) object*

Description

Create an expression quantitative trait methylation (eQTM) object

Usage

```
create_eQTM(data, metadata = list())
```

Arguments

`data` A data.frame containing eQTM data with columns: cpg, statistics, p_value, distance, and at least one of entrez or ensembl.

- **cpg** Character. CpG probe ID (e.g., "cg00000029"), representing a methylation site.
- **statistics** Numeric. Test statistic from eQTM association analysis (e.g., correlation coefficient, r-square, regression coefficient, or t-statistic). Can be positive or negative.
- **p_value** Numeric. P-value associated with the test statistic, must be between 0 and 1.
- **distance** Numeric. Genomic distance (in base pairs) between the CpG and the associated gene's transcription start site (TSS). Must be non-negative.
- **entrez** Character. Entrez gene ID of the associated gene. At least one of entrez or ensembl must be provided.
- **ensembl** Character. Ensembl gene ID of the associated gene. At least one of entrez or ensembl must be provided.

`metadata` A list of metadata (optional).

Value

An eQTM object.

Examples

```
data <- data.frame(
  cpg = c("cg000001", "cg000002"),
  statistics = c(2.5, -1.8),
  p_value = c(0.01, 0.03),
  distance = c(50000, 80000),
  entrez = c("673", "1956")
)
eqtm_obj <- create_eQTM(data)
```

eQTM-class *Expression quantitative trait methylation (eQTM) Class*

Description

A class to store eQTM data for pathway analysis. eQTM stands for Expression Quantitative Trait Methylation.

Slots

`data` A data.frame containing eQTM data with columns: cpg, statistics, p_value, distance, and at least one of entrez or ensembl.

`metadata` A list of metadata (e.g., data source, time point). Reserved for future use.

getData *Get expression quantitative trait methylation (eQTM) Data*

Description

Retrieve the eQTM data.frame from an eQTM object.

Usage

```
getData(object)

## S4 method for signature 'eQTM'
getData(object)
```

Arguments

`object`	An eQTM object.

Value

A data.frame stored in the object.

getMetadata *Get expression quantitative trait methylation (eQTM) Metadata*

Description

Retrieve the metadata list from an eQTM object.

Usage

```
getMetadata(object)

## S4 method for signature 'eQTM'
getMetadata(object)
```

Arguments

`object`	An eQTM object.

Value

A list containing metadata.

pathway_vote *Pathway Voting-Based Enrichment Analysis*

Description

Performs pathway enrichment analysis using a voting-based framework that integrates CpG–gene regulatory information from expression quantitative trait methylation (eQTM) data. For a grid of top-ranked CpGs and filtering thresholds, gene sets are generated and refined using an entropy-based pruning strategy that balances information richness, stability, and probe bias correction. In particular, gene lists dominated by genes with disproportionately high numbers of CpG mappings are penalized to mitigate active probe bias—a common artifact in methylation data analysis. Enrichment results across parameter combinations are then aggregated using a voting scheme, prioritizing pathways that are consistently recovered under diverse settings and robust to parameter perturbations.

Usage

```
pathway_vote(
  ewas_data,
  eQTM,
  databases = c("Reactome"),
  k_grid = NULL,
  stat_grid = NULL,
  distance_grid = NULL,
  fixed_prune = NULL,
  grid_size = 5,
  min_genes_per_hit = 2,
  overlap_threshold = 0.7,
  readable = FALSE,
  workers = NULL,
  verbose = FALSE
)
```

Arguments

`ewas_data` A data.frame containing CpG-level association results. The first column must contain CpG probe IDs, which will be matched against the eQTM object. The second column should contain a numeric ranking metric, such as a p-value, t-statistic, or feature importance score.

`eQTM` An `eQTM` object containing CpG–gene linkage information, created by the `create_eQTM()` function. This object provides the CpG-to-gene mapping used for pathway inference.

`databases` A character vector of pathway databases. Supporting: "Reactome", "KEGG", and "GO".

`k_grid` A numeric vector of top-k CpGs used for gene set construction. If NULL, the grid is automatically inferred using a log-scaled range guided by the number of CpGs passing FDR < 0.05. Note: This requires that `ewas_data` contains raw p-values (second column) from an association analysis; for other metrics (e.g., t-statistic or importance scores), `k_grid` must be provided manually.

`stat_grid` A numeric vector of eQTM statistic thresholds. If NULL, generated based on quantiles of the observed distribution.

`distance_grid` A numeric vector of CpG-gene distance thresholds (in base pairs). If NULL, generated similarly.

`fixed_prune` Integer or NULL. Minimum number of votes to retain a pathway. If NULL, will use cuberoot(N) where N is the number of enrichment runs.

`grid_size` Integer. Number of values in each grid when auto-generating. Default is 5.

`min_genes_per_hit`
Minimum number of genes (‘Count‘) a pathway must include to be considered. Default is 2.

`overlap_threshold`
Numeric between 0 and 1. Controls the maximum allowed Jaccard similarity between gene lists during redundancy filtering. Defualt is 0.7, which provides robust and stable results across a variety of simulation scenarios.

`readable` Logical. whether to convert Entrez IDs to gene symbols in enrichment results.

`workers` Optional integer. Number of parallel workers. If NULL, use 2 logical cores.

`verbose` Logical. whether to print progress messages.

Value

A named list of data.frames, each corresponding to a selected pathway database (e.g., ‘Reactome‘, ‘KEGG‘, ‘GO‘). Each data.frame contains enriched pathways with columns: ‘ID‘, ‘p.adjust‘, ‘Description‘, and ‘geneID‘.

Examples

```
set.seed(123)

# Simulated EWAS result: a mix of signal and noise
n_cpg <- 500
ewas <- data.frame(
  cpg = paste0("cg", sprintf("%08d", 1:n_cpg)),
 p_value = c(runif(n_cpg*0.1, 1e-9, 1e-5), runif(n_cpg*0.2, 1e-3, 0.05), runif(n_cpg*0.7, 0.05, 1))
)

# Corresponding eQTM mapping (some of these CpGs have gene links)
signal_genes <- c("5290", "673", "1956", "7157", "7422")
background_genes <- as.character(1000:9999)
entrez_signal <- sample(signal_genes, n_cpg * 0.1, replace = TRUE)
entrez_background <- sample(setdiff(background_genes, signal_genes), n_cpg * 0.9, replace = TRUE)

eqtm_data <- data.frame(
  cpg = ewas$cpg,
  statistics = rnorm(n_cpg, mean = 2, sd = 1),
  p_value = runif(n_cpg, min = 0.001, max = 0.05),
  distance = sample(1000:100000, n_cpg, replace = TRUE),
  entrez = c(entrez_signal, entrez_background),
  stringsAsFactors = FALSE
)
eqtm_obj <- create_eQTM(eqtm_data)

# Run pathway voting with minimal settings
## Not run:
results <- pathway_vote(
  ewas_data = ewas,
  eQTM = eqtm_obj,
  databases = c("GO", "KEGG", "Reactome"),
  readable = TRUE,
  verbose = TRUE
)
head(results$GO)
head(results$KEGG)
head(results$Reactome)

## End(Not run)
```

Index