Package ‘PathwayVote’

April 16, 2025

Title Robust Pathway Enrichment for DNA Methylation Studies Using Ensemble Voting

Version 0.1.0

Description Implements a robust, voting-based pathway enrichment method designed for DNA methylation data analysis. The algorithm allows the input of expression quantitative trait methylation (eQTM) data to aggregates pathway signals across multiple parameter settings, and selects pathways supported across combinations using a vote-pruning strategy.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.0.0)

Imports dplyr, org.Hs.eg.db, ReactomePA, clusterProfiler, AnnotationDbi, methods, future, furrr, parallelly

Suggests testthat

RoxygenNote 7.3.2

NeedsCompilation no

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

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

Repository CRAN

Date/Publication 2025-04-16 14:20:12 UTC

Contents

`create_eQTM` *Create an eQTM object*

Description

Create an 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` *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 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 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 Vote Algorithm for eQTM Data (Auto Parallel)*

Description

Performs pathway enrichment analysis using a voting-based approach on eQTM data.

Usage

```
pathway_vote(
  ewas_data,
  eQTM,
  k_values,
  stat_grid,
  distance_grid,
  overlap_threshold,
  databases = c("Reactome"),
  rank_column = "p_value",
  rank_decreasing = FALSE,
  use_abs = FALSE,
  prune_strategy = "cuberoot",
  fixed_value = 3,
  min_genes_per_hit = 3,
  readable = FALSE,
```

```
  workers = NULL,
  verbose = FALSE
)
```

Arguments

`ewas_data`	A data.frame with columns: cpg and a ranking column (e.g., p_value, score).
`eQTM`	An eQTM object containing eQTM data.
`k_values`	A numeric vector of top k CpGs to select (e.g., c(10, 50, 100)).
`stat_grid`	A numeric vector of statistics thresholds.
`distance_grid`	A numeric vector of distance thresholds.
`overlap_threshold`	A numeric value for gene list overlap threshold.
`databases`	A character vector of pathway databases (e.g., "Reactome").
`rank_column`	A character string indicating which column in ‘ewas_data‘ to use for ranking.
`rank_decreasing`	Logical. If TRUE (default), sorts CpGs from high to low based on ‘rank_column‘.
`use_abs`	Logical. Whether to apply ‘abs()‘ to the ranking column before sorting CpGs.
`prune_strategy`	Character, either "cuberoot" or "fixed". If "cuberoot", the minimum vote support is computed as (N)^(1/3) where N is the number of enrichment combinations. If "fixed", uses the value provided by ‘fixed_value‘.
`fixed_value`	Integer, used only if ‘prune_strategy = "fixed"‘.
`min_genes_per_hit`	Minimum number of genes (‘Count‘) a pathway must include to be considered.
`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 by default.
`verbose`	Logical. whether to print progress messages.

Value

A named list of data.frames, each containing enrichment results (pathway ID, p.adjust, Description, geneID) for one database (e.g., Reactome, KEGG).

Examples

```
data <- data.frame(
  cpg = c("cg000001", "cg000002", "cg000003"),
  statistics = c(2.5, -1.8, 3.2),
  p_value = c(0.01, 0.03, 0.005),
  distance = c(50000, 80000, 30000),
  entrez = c("673", "1956", "5290")
)
eqtm_obj <- create_eQTM(data)

results <- pathway_vote(
```

```
  ewas_data = data,
  eQTM = eqtm_obj,
  k_values = c(2),
  stat_grid = c(1.5),
  distance_grid = c(1e5),
  overlap_threshold = 0.3,
  databases = c("KEGG"),
  rank_column = "p_value",
  rank_decreasing = FALSE,
  use_abs = FALSE,
  worker = 1, # If not specified, will use 2 cores by default
  verbose = FALSE
)
```

Index