Package 'CauMedi'

August 4, 2026

Type Package

Title Cell Type-Specific Causal Mediation Models for Single-Cell Data

Version 0.1.1

Description A causal mediation framework for cell type-specific single-cell data based on joint mediator multilevel models. The framework jointly models overdispersed counts and zero inflation for the mediators.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Imports glmnet

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Seungjun Ahn [aut],
Donald Porchia [aut],
Weijia Fu [cre, aut] (ORCID: <https://orcid.org/0000-0002-4816-8924>),
Zhigang Li [ctb]

Maintainer Weijia Fu <weijia.fu@mountsinai.org>

Repository CRAN

Date/Publication 2026-08-04 13:50:12 UTC

Contents

`CauMedi` *Main function*

Description

A joint-mediator multilevel modeling framework for causal mediation analysis of cell type-specific single-cell data. Use prepare_CauMedi_data() to prepare inputs before running the function.

Usage

```
CauMedi(
  data,
  feature_meta = NULL,
  fdr_threshold = 0.05,
  mediator_screen_thres = 0.05,
  num_keep_multiplier = 10
)
```

Arguments

`data` A data frame containing the following:

- Y: Values of outcome.
- X: Values of exposure.
- M mediators: Columns whose names begin with "M_".
- F mediators: Columns whose names begin with "F_".

`feature_meta` A data frame containing the following:

- feature_name: The prefixed feature_id.
- cell_type: Character. Cell-type label.
- gene: Character. Gene name.

`fdr_threshold` Numeric scalar. BH-FDR threshold for declaring significance. Default 0.05.

`mediator_screen_thres` The p-value threshold for exposure-mediator association screening. Default 0.05.

`num_keep_multiplier` Numeric. Controls how many mediators per side (M and F) are kept for the final model. The rule is floor(n / (num_keep_multiplier) - 1). Default 10 reproduces the original n/10 - 1 rule.

Value

A named list with the following components:

- coef_outcome: Numeric vector of final outcome model coefficients, or NULL if no mediators were selected.
- se_outcome: Numeric vector of standard errors for coef_outcome.

- gamma_coef_vec: Numeric vector of X→M coefficients for the selected M mediators.
- gamma_se_vec: Numeric vector of standard erros for gamma_se_vec.
- alpha_coef_vec: Numeric vector of X→F coefficients for the selected F mediators.
- alpha_se_vec: Numeric vector of standard erros for alpha_se_vec.
- signif_M: Indices (into M_summary) of rows passing fdr_threshold.
- signif_F: Indices (into F_summary) of rows passing fdr_threshold.
- M_summary: Data frame of M-mediator results with columns cell_type, mediator, gene, gamma (X→M coefficient), p_gamma, beta_M (M→Y coefficient), p_beta_M, joint_pval, adj_pval.
- F_summary: Data frame of F mediators with columns cell_type, mediator, gene, alpha (X→F coefficients), p_alpha, beta_F, p_beta_F, joint_pval, adj_pval.

Examples

```
set.seed(278)
n_subj <- 100
n_gene <- 50
gene_ids <- paste0("gene", seq_len(n_gene))
M_mat <- matrix(rgamma(n_subj * n_gene, shape = 2, rate = 1),
                nrow = n_subj, ncol = n_gene,
                dimnames = list(paste0("subj", seq_len(n_subj)), gene_ids))
F_mat <- matrix(runif(n_subj * n_gene, min = 0, max = 1),
                nrow = n_subj, ncol = n_gene,
                dimnames = list(paste0("subj", seq_len(n_subj)), gene_ids))
Y <- rnorm(n_subj)
X <- rbinom(n_subj, 1, 0.5)
feature_meta <- data.frame(
 feature_id = gene_ids,
 cell_type = "Vasculature_cells",
 gene = gene_ids
)
pd <- prep_CauMedi_data(M_mat = M_mat, F_mat = F_mat, feature_meta = feature_meta, Y = Y, X = X)
res <- CauMedi(pd$data, feature_meta = pd$feature_meta)
print(res)
```

`prep_CauMedi_data` *Prepare inputs for CauMedi*

Description

Use prepare_CauMedi_data() first to prepare inputs before running main function.

Usage

```
prep_CauMedi_data(M_mat = NULL, F_mat = NULL, feature_meta, Y, X, scale = TRUE)
```

Arguments

M_mat	An n by p matrix of average expression levels across cells for the mediators (i.e., M_mediators). May be NULL if not available.
F_mat	An n by p matrix of the proportions of cells with zero counts for the mediators (i.e., F_mediators). May be NULL if not available.
feature_meta	Data frame describing the mediator features. Must contain the following columns: • feature_id: Character. A unique identifier of feature names in M_mat and F_mat. • cell_type: Character. Cell-type label of the feature. • gene: Character. Gene name of the feature.
Y	Numeric vector of outcome values.
X	Numeric vector of exposure values.
scale	Logical. If TRUE (default), each mediator column is standardised to mean 0, SD 1 before analysis. Set to FALSE if you have already scaled your data.

Value

A list with two elements:

- data: A data frame with columns subject_id, Y, X, followed by all M columns (prefixed M_) then all F columns (prefixed F_). Column names are M_<feature_id> and F_<feature_id>.
- feature_meta: A metadata table with type (M or F), feature_name, feature_id, cell_type and gene.

Examples

```
set.seed(278)
n_subj <- 100
n_gene <- 50
gene_ids <- paste0("gene", seq_len(n_gene))
M_mat <- matrix(rgamma(n_subj * n_gene, shape = 2, rate = 1),
                nrow = n_subj, ncol = n_gene,
                dimnames = list(paste0("subj", seq_len(n_subj)), gene_ids))
F_mat <- matrix(runif(n_subj * n_gene, min = 0, max = 1),
                nrow = n_subj, ncol = n_gene,
                dimnames = list(paste0("subj", seq_len(n_subj)), gene_ids))
Y <- rnorm(n_subj)
X <- rbinom(n_subj, 1, 0.5)
feature_meta <- data.frame(
 feature_id = gene_ids,
 cell_type = "Vasculature_cells",
 gene = gene_ids
)
pd <- prep_CauMedi_data(M_mat = M_mat, F_mat = F_mat, feature_meta = feature_meta, Y = Y, X = X)
```

Index