

Package: Tweedieverse (via r-universe)

September 28, 2026

Title Differential Analysis of Omics Data Based on the Tweedie Distribution

Version 0.99.0

Date 2026-05-07

Description A toolkit for differential analysis of omics data. Implements a CPLM-based statistical workflow built on the Tweedie distribution. Unlike traditional single-omics tools, Tweedieverse is technology-agnostic and can be applied to both count and continuous measurements arising from diverse high-throughput technologies (e.g. transcript abundances from bulk and single-cell RNA-Seq studies in the form of UMI counts or non-UMI counts, microbiome taxonomic and functional profiles in the form of counts or relative abundances, and compound abundance levels or peak intensities from metabolomics and other mass spectrometry-based experiments, among others). The software includes multiple customization options such as the inclusion of random effects and multiple covariates along with several data exploration capabilities and visualization modules in a unified estimation umbrella.

Depends R (>= 4.5.0)

Imports cplm, glmmTMB, BiocParallel, logging, dplyr, ggplot2, pheatmap, cowplot, SummarizedExperiment, statmod, Maaslin2, edgeR, scan, SingleCellExperiment, TreeSummarizedExperiment, MultiAssayExperiment

Suggests airway, data.table, knitr, rmarkdown, BiocStyle, muscat, testthat

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

VignetteBuilder knitr

URL <https://github.com/himelmallick/Tweedieverse>

BugReports <https://github.com/himelmallick/Tweedieverse/issues>

biocViews DifferentialExpression, Software, Visualization,
CellBasedAssays, ATACSeq, Microbiome, MassSpectrometry,
SingleCell, Metabolomics, Metagenomics, WorkflowStep,
MultipleComparison, Normalization, Regression, RNASeq
Config/pak/sysreqs cmake libglpk-dev make libicu-dev libuv1-dev libxml2-dev zlib1g-dev
Repository <https://biocstaging.r-universe.dev>
Date/Publication 2026-09-28 16:58:52 UTC
RemoteUrl <https://github.com/BiocStaging/Tweedieverse>
RemoteRef HEAD
RemoteSha a7e3890c120c8e4b166e1c558f9066798620335c

Contents

entropy	2
median_comparison_tweedie	3
theme_omicsEye	4
Tweedieverse	5
Tweedieverse_heatmap	9
Index	11

entropy	<i>Entropy of a Vector</i>
---------	----------------------------

Description

Compute Shannon entropy for a vector.

Usage

entropy(target)

Arguments

target A vector of values.

Value

A numeric entropy value.

Examples

entropy(c("A", "A", "B", "B", "C"))

median_comparison_tweedie

Median Comparison for Compositionality Adjustment

Description

Adjust Tweedieverse(or any other differential analysis methods) coefficient estimates and p-values by testing each taxon against the **median** effect for the same metadata variable - a simple post-hoc strategy to curb false discoveries driven by the compositional nature of microbiome count data (after the approach adopted in ***MaAsLin 3***).

Usage

```
median_comparison_tweedie(
  df,
  p_cutoff = 0.95,
  subtract_median = FALSE,
  n_sims = 10000,
  median_threshold = 0
)
```

Arguments

df	A 'data.frame' returned by <i>**Tweedieverse**</i> or any other differential analysis methods containing (at minimum) the columns 'taxon', 'metadata', 'effect_size', 'pval', 'stderr', and 'qval'.
p_cutoff	Numeric. Upper bound on the original p-value to include an effect in the median calculation (default '0.95' = all non-missing).
subtract_median	Logical. If 'TRUE', subtracts the group median from every coefficient before returning it in 'coef_median'; otherwise the original 'effect_size' is copied unchanged (default 'FALSE').
n_sims	Integer. Number of Monte-Carlo simulations used to estimate the covariance between each coefficient and the group median (default '10 000').
median_threshold	Numeric. Absolute difference below which a coefficient is considered effectively equal to the median and assigned 'pval_median = 1' (default '0').

Details

For every distinct value in 'metadata' the algorithm

1. keeps coefficients with 'pval < p_cutoff' and computes their median;
2. simulates 'n_sims' draws of coefficients using a normal approximation ('N(effect_size, stderr^2)') and records the empirical distribution of the simulated medians;

3. derives a variance-inflated **offset** that accounts for the covariance between each coefficient and the group median;
4. performs a two-sided Z-test of ' $H_0 : \beta_i = \text{offset}_i$ ', returning the resulting p-value in '*pval_median*'.

Value

The input '*df*' with two new columns:

'coef_median' Median-centred coefficient (or the original '*effect_size*' if '*subtract_median* = FALSE').

'pval_median' Two-sided p-value from the median-comparison test.

Warning

The procedure is heuristic and relies on normal approximations. Results may be unstable for very small sample sizes or when '*stderr*' values are zero or missing.

See Also

[MaAsLin 3 GitHub](<https://github.com/biobakery/maaslin3>)

Examples

```
toy <- data.frame(
  taxon = c("tax1", "tax2"),
  metadata = c("grp", "grp"),
  effect_size = c(0.4, 0.2),
  pval = c(0.01, 0.2),
  stderr = c(0.1, 0.1),
  qval = c(0.02, 0.25)
)
median_comparison_tweedie(toy, n_sims = 100)
```

theme_omicsEye

Omics-Eye Plot Theme

Description

A compact ggplot2 theme used by Tweedieverse visualization functions.

Usage

```
theme_omicsEye()
```

Value

A list of ggplot2 theme elements.

Examples

```
theme_omicsEye()
```

Tweedieverse

Differential analysis of multi-omics data using Tweedie GLMs

Description

Fit a per-feature Tweedie generalized linear model to omics features.

Usage

```
Tweedieverse(
  input_features,
  input_metadata = NULL,
  output = NULL,
  assay_name = "counts",
  abd_threshold = 0,
  prev_threshold = 0.1,
  var_threshold = 0,
  entropy_threshold = 0,
  base_model = "CPLM",
  link = "log",
  tweedie_p = NULL,
  p0_transform = "NONE",
  p0_transform_pseudocount = 1,
  fixed_effects = NULL,
  random_effects = NULL,
  domain = "microbiome",
  normalization = NULL,
  adjust_offset = TRUE,
  scale_factor = NULL,
  max_significance = 0.05,
  correction = "BH",
  Maaslin2_run = TRUE,
  median_comparison = FALSE,
  median_subtraction = FALSE,
  run_presence_absence_model = FALSE,
  method_args = NULL,
  method.args = NULL,
  standardize = TRUE,
  BPPARAM = BiocParallel::SerialParam(),
  optimizer = "nlminb",
  na.action = na.exclude,
  plot_heatmap = FALSE,
  plot_scatter = FALSE,
```

```

heatmap_first_n = 50,
reference = NULL
)

```

Arguments

- | | |
|-------------------|--|
| input_features | A domain-appropriate Bioconductor container, such as <code>TreeSummarizedExperiment</code> for domain = "microbiome", <code>SingleCellExperiment</code> for domain = "single_cell", and <code>SummarizedExperiment</code> for domain = "bulk_rnaseq". The assays slot contains the expression or abundance matrix and defaults to an assay named "counts". A <code>MultiAssayExperiment</code> object is also accepted. In that case, <code>Tweedieverse()</code> iterates over each experiment/omics layer and returns a named list of omics-specific results. This matrix should have one row for each feature and one sample for each column. The <code>colData</code> slot should contain a data frame with one row per sample and columns that contain metadata for each sample. Additional information about the experiment can be contained in the metadata slot as a list. |
| input_metadata | Optional R data frame of metadata. Samples are expected to have matching sample names with <code>input_features</code> . If <code>NULL</code> , metadata are taken from the Bioconductor container's <code>colData</code> . |
| output | The output folder to write results. |
| assay_name | If the input is provided as one of the accepted Bioconductor objects, this argument selects the name of the assay slot in the input object that contains the omics measurements. For <code>MultiAssayExperiment</code> input, most arguments can be supplied as a single value used for every omics layer, or as a named list/vector keyed by experiment name. |
| abd_threshold | If prevalence-abundance filtering is desired, only features that are present (or detected) in at least <code>prev_threshold</code> percent of samples at <code>abd_threshold</code> minimum abundance (read count or proportion) are retained. Default value for <code>abd_threshold</code> is 0.0. To disable prevalence-abundance filtering, set <code>abd_threshold = -Inf</code> . |
| prev_threshold | If prevalence-abundance filtering is desired, only features that are present (or detected) in at least <code>prev_threshold</code> percent of samples at <code>abd_threshold</code> minimum abundance (read count or proportion) are retained. Default value for <code>prev_threshold</code> is 0.1. |
| var_threshold | If variance filtering is desired, only features that have variances greater than <code>var_threshold</code> are retained. This step is done after the prevalence-abundance filtering. Default value for <code>var_threshold</code> is 0.0 (i.e. no variance filtering). |
| entropy_threshold | If entropy-based filtering is desired for metadata, only features that have entropy greater than <code>entropy_threshold</code> are retained. Default value for <code>entropy_threshold</code> is 0.0 (i.e. no entropy filtering). |
| base_model | The per-feature base model. Only "CPLM" is supported. |
| link | A specification of the GLM link function. Default is "log". Must be one of "log", "identity", "sqrt", or "inverse". |

tweedie_p	Numeric Tweedie variance power. Default is NULL, which estimates the compound-Poisson Tweedie index in the usual $1 < p < 2$ range when all filtered feature values are non-negative. If tweedie_p is NULL or NA and filtered features contain negative values, it is resolved to 0, the Gaussian case. Values between 0 and 1 are undefined and rejected. Other supplied values pin the model to that fixed index. Negative supplied powers are allowed with a warning for non-negative data, but negative feature values are supported only at $p = 0$. Random-effect models support fixed $p = 0$, $p = 1$, $1 < p < 2$, $p = 2$, and $p = 3$.
p0_transform	Transformation to apply to a copy of the filtered feature table when the resolved Tweedie index is 0. The original input data are not modified. Must be one of "NONE", "CLR", "RCLR", "LOG", or "ARC_SIGNED_SQRT". The transformed copy is used for the $p = 0$ abundance model and written to transformed_features.tsv when output is provided. Default is "NONE".
p0_transform_pseudocount	Numeric pseudocount used for <code>p0_transform = "CLR"</code> and <code>p0_transform = "LOG"</code> . Default is 1.
fixed_effects	Metadata variable(s) describing the fixed effects coefficients.
random_effects	Metadata variable(s) describing the random effects part of the model.
domain	Domain used to choose the default size-factor normalization. Must be one of "microbiome", "single_cell", "bulk_rnaseq", or "custom". The default is "microbiome".
normalization	Size-factor strategy used only to compute the model offset. The input feature table is never normalized or transformed. If NULL, the domain default is used: "TSS" for microbiome, "SCRAN" for single-cell, "TMM" for bulk RNA-seq, and "MEDIAN" for custom/other omics. Supported strategies are "TSS", "GMPR", "CSS", "SCRAN", "TMM", "RLE" / "DESEQ2", "CPM", "MEDIAN", and "NONE".
adjust_offset	If TRUE (default), an offset term will be included as the logarithm of the size factor estimated by normalization, or by scale_factor when supplied.
scale_factor	Name of a numerical metadata variable containing user-supplied sample size factors. When supplied, this overrides domain and normalization for offset calculation.
max_significance	The q-value threshold for significance. Default is 0.05.
correction	The correction method for computing the q-value (see p.adjust for options, default is 'BH').
Maaslin2_run	Logical. For <code>tweedie_p = 0</code> , run the MaAsLin2 linear-model path instead of the Tweedie GLM path. Set to FALSE to fit <code>tweedie_p = 0</code> with the Tweedie GLM. Default is TRUE. When the resolved Tweedie index is 0, non-identity links are changed to "identity" and <code>adjust_offset</code> is disabled.
median_comparison	If TRUE, coefficients will be tested against a null value corresponding to the median coefficient for a covariate in the metadata. Default is FALSE. Should only be used for relative abundance data.
median_subtraction	If TRUE, coefficients minus median will be used for compositionality adjustment.

<code>run_presence_absence_model</code>	If TRUE, also fit a DAssemble-style presence-absence logistic regression model alongside the selected abundance model, including the MaAsLin2 path used by default for <code>tweedie_p = 0</code> . Retains individual abundance and presence-absence model results, and ranks features by a Cauchy combination test of the two p-values. Default is FALSE.
<code>method_args</code>	Optional named list of method-specific arguments. For <code>tweedie_p = 0</code> , use <code>method_args = list(Maaslin2 = list(...))</code> to pass arguments to Maaslin2 ; user-supplied values override Tweedieverse defaults.
<code>method.args</code>	Alias for <code>method_args</code> .
<code>standardize</code>	Should continuous metadata be standardized? Default is TRUE. Bypassed for categorical variables.
<code>BPPARAM</code>	BiocParallelParam object controlling Bioconductor-native parallel evaluation. Default is SerialParam .
<code>optimizer</code>	The optimization routine to be used for estimating the parameters of the Tweedie model. Possible choices are "nlminb" (the default, see nlminb), "bobyqa" (bobyqa), and "L-BFGS-B" (optim). Ignored for random effects modeling which uses an alternative Template Model Builder (TMB) approach (glmmTMB).
<code>na.action</code>	How to handle missing values? See na.action . Default is na.exclude .
<code>plot_heatmap</code>	Logical. If TRUE (default is FALSE), generate a heatmap of the (top <code>heatmap_first_n</code>) significant associations.
<code>plot_scatter</code>	Logical. If TRUE (default is FALSE), generate scatter/box plots of individual associations.
<code>heatmap_first_n</code>	In heatmap, plot top N features with significant associations (default is 50).
<code>reference</code>	The factor to use as a reference for a variable with more than two levels provided as a string of 'variable,reference' semi-colon delimited for multiple variables (default is NULL).

Value

For single-omics input, a data frame containing coefficient estimates, p-values, and q-values (multiplicity-adjusted p-values) is returned. For `MultiAssayExperiment` input, a named list of omics-specific result data frames is returned.

Author(s)

Himel Mallick, <him4004@med.cornell.edu>

Examples

```
set.seed(123)
features <- as.data.frame(matrix(rpois(24, lambda = 5), nrow = 12, ncol = 2))
colnames(features) <- c("feature1", "feature2")
rownames(features) <- paste0("sample", seq_len(nrow(features)))
metadata <- data.frame(
  group = rep(c("A", "B"), each = 6),
```

```

    row.names = rownames(features)
  )
  experiment <- SummarizedExperiment::SummarizedExperiment(
    assays = list(counts = t(as.matrix(features))),
    colData = metadata
  )
  fit <- Tweedieverse(
    input_features = experiment,
    output = NULL,
    fixed_effects = "group",
    median_comparison = FALSE
  )
  head(fit)

  maaslin2_median_fit <- Tweedieverse(
    input_features = experiment,
    output = NULL,
    fixed_effects = "group",
    tweedie_p = 0,
    Maaslin2_run = TRUE,
    median_comparison = TRUE,
    median_subtraction = TRUE
  )

  # For Bioconductor-container workflow examples, see:
  # vignette("Tweedieverse", package = "Tweedieverse")

```

Tweedieverse_heatmap *Plot a Tweedieverse Heatmap*

Description

Create a heatmap summarizing association statistics from Tweedieverse results.

Usage

```

Tweedieverse_heatmap(
  output_results,
  title = NA,
  cell_value = "qval",
  data_label = "data",
  metadata_label = "metadata",
  border_color = "grey93",
  color = colorRampPalette(c("darkblue", "grey90", "darkred")),
  col_rotate = 90,
  first_n = 50,
  write_to = NA
)

```

Arguments

<code>output_results</code>	Tweedieverse result data frame or path to a TSV file.
<code>title</code>	Optional plot title.
<code>cell_value</code>	Which value to display ("qval", "pval", or "coef").
<code>data_label</code>	Label for feature axis.
<code>metadata_label</code>	Label for metadata axis.
<code>border_color</code>	Heatmap border color.
<code>color</code>	Color palette function.
<code>col_rotate</code>	Rotation angle for column labels.
<code>first_n</code>	Number of top features to include.
<code>write_to</code>	Optional output directory for serialized plot objects.

Value

A 'pheatmap' object or 'NULL' when insufficient data are available.

Examples

```
toy <- data.frame(
  feature = c("tax1", "tax2", "tax1", "tax2"),
  metadata = c("grp", "grp", "time", "time"),
  value = c("A", "B", "A", "B"),
  pval = c(0.01, 0.05, 0.02, 0.03),
  qval = c(0.02, 0.08, 0.03, 0.04),
  coef = c(1.2, -0.7, 0.9, -0.4)
)
Tweedieverse_heatmap(toy, first_n = 2)
```

Index

- * **metagenomics**
 - Tweedieverse, [5](#)
- * **microbiome**
 - Tweedieverse, [5](#)
- * **multiomics**
 - Tweedieverse, [5](#)
- * **scRNASeq**
 - Tweedieverse, [5](#)
- * **singlecell**
 - Tweedieverse, [5](#)
- * **tweedie**
 - Tweedieverse, [5](#)
- BiocParallelParam, [8](#)
- bobyqa, [8](#)
- entropy, [2](#)
- glmmTMB, [8](#)
- Maaslin2, [8](#)
- median_comparison_tweedie, [3](#)
- na.action, [8](#)
- na.exclude, [8](#)
- nlimb, [8](#)
- optim, [8](#)
- p.adjust, [7](#)
- SerialParam, [8](#)
- theme_omicsEye, [4](#)
- Tweedieverse, [5](#)
- Tweedieverse_heatmap, [9](#)