

Package: DAssemble (via r-universe)

June 9, 2026

Type Package

Title Ensemble Models for Differential Analysis

Version 0.99.0

Description The DAssemble package implements an ensemble framework for differential-abundance and differential-expression analysis across bulk RNA-seq, single-cell RNA-seq, and microbiome studies. It wraps a collection of popular DA/DE methods as core methods and combines them via Cauchy Combination Tests (CCT), optionally augmented by simple enhancers such as Wilcoxon, Kolmogorov–Smirnov, and presence–absence logistic regression.

License MIT + file LICENSE

Encoding UTF-8

biocViews Software, StatisticalMethod, DifferentialExpression, RNASeq, SingleCell, Microbiome, Normalization, Regression

Depends R (>= 4.5.0)

Imports stats, utils, dplyr, MultiAssayExperiment, SummarizedExperiment

Suggests DESeq2, edgeR, limma, dearseq, metagenomeSeq, MAST, cplm, dfadjust, glmmTMB, logging, MASS, pbapply, preprocessCore, Maaslin2, maaslin3, LOCOM2, MicrobiomeStat, ANCOMBC, ALDEx2, parallel, airway, scran, SingleCellExperiment, knitr, rmarkdown, BiocStyle, phyloseq, Seurat, S4Vectors, testthat (>= 3.0.0), TreeSummarizedExperiment, KernSmooth, foreign, nnet, rpart, spatial, boot, class

Roxygen list(markdown = TRUE)

VignetteBuilder knitr

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

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

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

Config/pak/sysreqs libicu-dev zlib1g-dev
Repository https://biocstaging.r-universe.dev
Date/Publication 2026-05-30 18:08:02 UTC
RemoteUrl https://github.com/BiocStaging/DAssemble
RemoteRef HEAD
RemoteSha dcf0c31c018d543448f0044cd562417822a2d206

Contents

DAssemble	2
Index	4

DAssemble	<i>Differential analysis using DAssemble</i>
-----------	--

Description

The `DAssemble` function implements an ensemble strategy for differential analysis. Users can select a single *core* model and optionally augment it with up to three *enhancer* models. Enhancers are simple, often nonparametric tests that provide complementary evidence. P-values from the core and enhancer models are combined using the Cauchy Combination Test (CCT).

Usage

```
DAssemble(
  features,
  metadata = NULL,
  core_method = NULL,
  enhancers = NULL,
  expVar = "group",
  coVars = NULL,
  assay_name = NULL,
  p_adj = "BY",
  enhancer_norm = "TSS",
  return_components = TRUE,
  return_subensembles = FALSE
)
```

Arguments

<code>features</code>	A <code>MultiAssayExperiment</code> object, or a <code>data.frame</code> with samples in rows and features in columns.
<code>metadata</code>	A <code>data.frame</code> with sample metadata. Leave as <code>NULL</code> when <code>features</code> is a <code>MultiAssayExperiment</code> ; sample metadata is then taken from <code>colData(features)</code> .

core_method	Character scalar naming the core method to run, or NULL / "none" for enhancer-only analysis.
enhancers	Character vector containing any of "WLX", "LR", and "KS".
expVar	Character scalar naming the binary exposure variable in metadata or colData(features).
coVars	Optional character vector naming covariates to include in covariate-aware core methods and logistic-regression enhancers.
assay_name	Name of the assay/experiment to use when features is a MultiAssayExperiment. Required when the object contains multiple experiments.
p_adj	Multiple-testing correction method passed to <code>stats::p.adjust</code> .
enhancer_norm	Normalization method for enhancer tests. One of "TSS", "CLR", "TMM", or "SCRAN".
return_components	Logical; if TRUE, return per-method component result tables.
return_subensembles	Logical; if TRUE, return CCT sub-ensemble results for all available method subsets.

Value

A list containing combined results in `res`, the analyzed feature table in `features`, elapsed time in `Time.min`, and optionally components and ensembles.

Examples

```
features <- data.frame(
  taxon1 = c(12, 9, 11, 8, 25, 21, 23, 26),
  taxon2 = c(30, 28, 31, 29, 10, 12, 11, 9),
  taxon3 = c(5, 6, 4, 7, 5, 4, 6, 5),
  taxon4 = c(8, 7, 9, 6, 12, 13, 11, 14),
  row.names = paste0("sample", seq_len(8))
)
metadata <- data.frame(
  group = factor(rep(c("control", "case"), each = 4)),
  row.names = rownames(features)
)
result <- DAssemble(
  features = features,
  metadata = metadata,
  core_method = NULL,
  enhancers = "WLX",
  expVar = "group"
)
head(result$res)
```

Index

DAssemble, [2](#)

stats::p.adjust, [3](#)