Package 'DAssemble'

November 12, 2026

Type Package

Title Ensemble Models for Differential Analysis

Version 0.99.5

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 (>= 4.6.0)

Imports stats, utils, dplyr, MultiAssayExperiment, SummarizedExperiment

Suggests DESeq2, edgeR, limma, dearseq, metagenomeSeq, MAST, cplm, brglm2, dfadjust, glmmTMB, logging, MASS, pbapply, preprocessCore, Maaslin2, maaslin3, LOCOM2, MicrobiomeStat, ANCOMBC, ALDEx2, airway, scran, SingleCellExperiment, knitr, rmarkdown, BiocStyle, phyloseq, S4Vectors, testthat (>= 3.0.0), boot

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

git_url https://git.bioconductor.org/packages/DAssemble

git_branch devel

git_last_commit cb82319

git_last_commit_date 2026-11-12

Repository Bioconductor 3.24

Date/Publication 2026-11-12

Author Ziyu Liu [aut],
Nalin Arora [aut, cre] (ORCID: <https://orcid.org/0009-0009-1340-688X>),
Chuxuan Gao [aut],
Himel Mallick [aut]

Maintainer Nalin Arora <naa4050@med.cornell.edu>

Contents

DAssemble-package 2
DAssemble 3

Index **5**

`DAssemble-package` *DAssemble: Ensemble Models for Differential Analysis*

Description

DAssemble provides an ensemble framework for differential-abundance and differential-expression analysis across bulk RNA-seq, single-cell RNA-seq, and microbiome studies. The package combines a user-selected core method with optional enhancer tests and aggregates evidence using the Cauchy Combination Test.

Details

The main user-facing function is `DAssemble()`. It accepts either a `MultiAssayExperiment` object or a feature table paired with sample metadata. Core methods can be combined with enhancer tests including Wilcoxon rank-sum, Kolmogorov-Smirnov, and presence-absence logistic regression.

Author(s)

Maintainer: Nalin Arora <naa4050@med.cornell.edu> (ORCID)

Authors:

- Nalin Arora <naa4050@med.cornell.edu> (ORCID)
- Ziyu Liu <lclcencen@gmail.com>
- Chuxuan Gao <chg4016@alumni.weill.cornell.edu>
- Himel Mallick <him4004@med.cornell.edu>

See Also

`DAssemble()`

DAssemble *Differential analysis using DAssemble*

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,
  random_effects = NULL,
  method_args = NULL,
  assay_name = NULL,
  p_adj = "BY",
  enhancer_norm = "TSS",
  return_components = TRUE,
  return_subensembles = FALSE
)
```

Arguments

features	A MultiAssayExperiment object, or a data.frame with samples in rows and features in columns.
metadata	A data.frame with sample metadata. Leave as NULL when features is a MultiAssayExperiment; sample metadata is then taken from colData(features).
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.
random_effects	Optional character vector naming grouping variables for random intercepts in longitudinal-compatible methods.
method_args	Optional named list of method-specific control arguments. Use method_args$core for arguments shared by the selected core method, method_args$enhancer for arguments shared by enhancers, and/or method names such as method_args$Maaslin3 or method_args$LR for method-specific overrides. Matching arguments override DAssemble wrapper defaults before the underlying function is called. For the LR enhancer, the wrapper-specific option method_argsLRseparation_method

= "firth" switches the cross-sectional presence/absence logistic fit from the default MaAsLin3-style augmented logistic fit to bias-reduced fitting via brglm2. For the Maaslin2 core, method_args$Maaslin2$median_comparison = TRUE applies the internal median-comparison adjustment to raw fit$results before the exposure-specific result table is standardized.

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 stats::p.adjust.

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 the CCT-ranked joint results in res, the analyzed feature table in features, elapsed time in Time.min, and a named components list when return_components = TRUE. Each component is the standardized result table from one selected core method or enhancer. Optional CCT sub-ensemble results are returned in 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 = "LR",
  expVar = "group",
  method_args = list(
    LR = list(control = glm.control(maxit = 50))
  )
)
head(result$res)
```

Index

∗ **package**
DAssemble-package, 2

DAssemble, 3
DAssemble(), *2*
DAssemble-package, 2

stats::p.adjust, *4*