Package 'ALDEx3'

January 31, 2026

Title Linear Models for Sequence Count Data

Version 1.0.0

Description Provides scalable generalized linear and mixed effects models tailored for sequence count data analysis (e.g., analysis of 16S or RNA-seq data). Uses Dirichlet-multinomial sampling to quantify uncertainty in relative abundance or relative expression conditioned on observed count data.
Implements scale models as a generalization of normalizations which account for uncertainty in scale (e.g., total abundances) as described in Nixon et al. (2025) <doi:10.1186/s13059-025-03609-3> and McGovern et al. (2025) <doi:10.1101/2025.08.05.668734>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Suggests rBeta2009, testthat (>= 3.0.0), lmtest, sandwich, knitr, rmarkdown

Config/testthat/edition 3

Imports purrr, lme4, lmerTest, parallel, MASS, nlme, abind, matrixStats, methods, stats

Depends R (>= 3.5)

LazyData true

VignetteBuilder knitr

NeedsCompilation no

Author Justin Silverman [aut, cre], Greg Gloor [aut], Kyle McGovern [aut, ctb]

Maintainer Justin Silverman <JustinSilverman@psu.edu>

Repository CRAN

Date/Publication 2026-01-31 19:10:16 UTC

Contents

`aldex` *ALDEx3 Linear Models*

Description

ALDEx3 Linear Models

Usage

```
aldex(
  Y,
  X,
  data = NULL,
  method = "lm",
  nsample = 2000,
  scale = NULL,
  streamsize = 8000,
  n.cores = detectCores() - 1,
  return.pars = c("X", "estimate", "std.error", "p.val", "p.val.adj", "logComp",
    "logScale"),
  p.adjust.method = "BH",
  test = "t.HC3",
  onesided = FALSE,
  ...
)
```

Arguments

Y	an (D x N) matrix of sequence count, N is number of samples, D is number of taxa or genes
X	either a formula (in which case DATA must be non-null) or a model matrix of dimension P x N (P is number of linear model covariates). If a formula is passed it should not include the target Y, e.g., should simply be "~condition-1" (note the lack of the left hand side). If using lme4, this should be the formula including random effects.
data	a data frame for use with formula, must have N rows
method	(default lm) The regression method; "lm": linear regression, "lme4": linear mixed effects regression with lme4; "nlme": linear mixed effects models with nlme, REQUIRES "random" argument representing random effects be passed into aldex function, can also pass "correlation" as argument (see nlme documentation for how to use "correlation" argument).
nsample	number of monte carlo replicates
scale	the scale model, can be a function or an N x nsample matrix (samples should be given on log2-scale; e.g., samples should be of log of system scale). The API for writing your own scale models is documented below in examples.
streamsize	(default 8000) memory footprint (approximate) at which to use streaming. This should be thought of as the number of Mb for each streaming chunk. If D*N*nsample*8/1000000 is less than streamsize then no streaming will be performed. Note, to conserve memory, samples from the Dirichlet and scale models will not be returned if streaming is used. Streaming can be turned off by setting streamsize=Inf.
n.cores	(default detectCores()-1) If method is 'lme4', use this many cores for running mixed effects models in parallel.
return.pars	what results should be returned, see return section below.
p.adjust.method	(default BH) The method for multiple hypothesis test correction. See p.adjust for all available methods.
test	(default t.HC3), "t", t test is performed for each covariate (fast); "t.HC0" Heteroskedasticsity-Robust Standard Errors used (HC0; White's; slower); "t.HC3" (default) Heteroskedasticsity-Robust Standard Errors used (HC3; unlike HC0, this includes a leverage adjustment and is better for small sample sizes or when there are data with high leverage; slowest). To learn more about these, loko at Long and Ervin (2000) Using Heteroscedasticity Consistent Standard Errors in the Linear Regression Model, The American Statistician.
onesided	(default: FALSE) if sided return p-values for two-sided test. Otherwise if "lower" or "upper" return one-sided test corresponding to test that estimate is negative or positive respectively.
...	parameters to be passed to the scale model (if a function is provided), may also be random or correlation arguments for nlme.

Value

a list with elements controled by parameter resturn.pars. Options include: - X: P x N covariate matrix - estimate: (P x D x nsample) array of linear model estimates - std.error: (P x D x nsample)

array of standard error for the estimates - p.val: (P x D) matrix, unadjusted p-value for two-sided t-test - p.val.adj: (P x D) matrix, p-value for two-sided t-test adjusted according to `p.adj.method` - logScale: (N x S) matrix, samples of the log scale from the scale model - logComp: (D x N x S) array, samples of the log composition from the multinomial-Dirichlet - streaming: boolean, detnote if streaming was used. - random.effects (Pr x N x S): if using mixed effects models, return all Pr random effects. Note, logScale and logComp are not returned if streaming is active.

Author(s)

Justin Silverman, Kyle McGovern

Examples

```
Y <- matrix(1:110, 10, 11)
condition <- c(rep(0, 5), rep(1, 6))
data <- data.frame(condition=condition)
## demonstrate formula interface and passing optional argument (gamma) to
## the scale model (clr)
res <- aldex(Y, ~condition, data, nsample=2000, scale=clr.sm, gamma=0.5)

## demonstrating how to write a custom scale model, I will write a model
## that generalizes total sum scaling (where we assume no change between
## conditions)
## Functions can include parameters X (model matrix), Y, and logComp.
## If included in the function definition, those parameters will be passed
## dynamically when aldex is running. Other optional parameters (gamma)
## can be passed as additional arguments to the aldex function
tss <- function(X, logComp, gamma=0.5) {
  P <- nrow(X)
  nsample <- dim(logComp)[3]
  LambdaScale <- matrix(rnorm(P*nsample,0,gamma), P, nsample)
  logScale <- t(X)%*% LambdaScale
  return(logScale)
}
res <- aldex(Y, ~condition, data, nsample=2000, scale=tss, gamma=0.75)
```

`aldex.lm.sim.clr` *Simulation function soley for testing and exploring ALDEx3, Truth is in CLR Coordinates.*

Description

Not designed to create realistic data. Does not add any noise to linear regression! True W is in CLR Corrdinates

Usage

```
aldex.lm.sim.clr(D = 10, N = 11, P = 2, depth = 10000)
```

Arguments

D	number of taxa/genes
N	number of samples
P	number of covariates
depth	sum of counts for each multinomial draw

Value

a list with elements Y, X, W, and Lambda

Author(s)

Justin Silverman

---

aldex.mem.sim *Simulation for testing mixed effects models.*

Description

Includes random intercpt, option to include random slope, second random intercept, and time-correlation.

Usage

```
aldex.mem.sim(
  D,
  days,
  subjects,
  depth = 10000,
  location = 1,
  random_slope = FALSE,
  corr = 0,
  rho_ar1 = 0,
  sd_resid = 0.1
)
```

Arguments

D	number of taxa/genes
days	num days (i.e., repeated measurements) for each subject
subjects	num of subjects to simulate
depth	sum of counts for each multinomial draw
location	second random intercept, if 0 ignore, else simulate num of locations
random_slope	If true, simulate a random slope for each subject.
corr	The correlation between slope/random intercept
rho_ar1	The ar1 time-correlation, if 0, don't simulate
sd_resid	The residual error time-correlation.

Author(s)

Kyle McGovern

`aldex.pvals` *Calculate p-values adjusting for changes in sign as described by Nixon et al. (2024) in Beyond Normalization: Incorporating Scale Uncertainty in Microbiome and Gene Expression Analysis (internal only)*

Description

Calculate p-values adjusting for changes in sign as described by Nixon et al. (2024) in Beyond Normalization: Incorporating Scale Uncertainty in Microbiome and Gene Expression Analysis (internal only)

Usage

```
aldex.pvals(p.lower, p.upper, p.adjust.method, onesided = FALSE)
```

Arguments

`p.lower`	A P x D x S matrix for P covariates, D taxa/genes, and S monte carlo samples representing the lower tail p.values
`p.upper`	A P x D x S matrix for P covariates, D taxa/genes, and S monte carlo samples representing the upper tail p.values
`p.adjust.method`	An adjutment method for p.adjust
`onesided`	(default: FALSE) if sided return p-values for two-sided test. Otherwise if "lower" or "upper" return one-sided test corresponding to test that estimate is negative or positive respectively.

Value

A list with P x D matrices with the non-adjusted and adjusted p-values.

Author(s)

Justin Silverman, Kyle McGovern

References

Nixon G, Gloor GB, Silverman JD (2025). "Incorporating scale uncertainty in microbiome and gene expression analysis as an extension of normalization". Genome Biology. doi:10.1186/s13059-025036093

clr.sm *Default CLR-based scale model (with optional scale uncertainty)*

Description

Implements the default scale model described in Nixon et al. (Beyond Normalizations / scale-uncertainty framework). This model generalizes the centered log-ratio (CLR) normalization by treating the (log) scale as a latent random variable and allowing additive uncertainty around the CLR-implied scale differences via a Gaussian term with standard deviation gamma.

Usage

```
clr.sm(X, logComp, gamma = 0.5)
```

Arguments

X	A numeric design matrix used to model scale variation across samples. This is the covariate/design matrix passed internally by aldex() to the scale model. Rows correspond to regression coefficients (e.g., intercept and covariates after contrasts/encoding) and columns correspond to samples. If the analysis includes only an intercept (no covariates), X is typically a 1 x N matrix of ones. (This parameter is automatically passed by aldex)
logComp	A numeric array of Monte Carlo log-compositions with dimensions features x samples x nsample. This is produced internally by ALDEx3 from Dirichlet-multinomial Monte Carlo sampling and log-ratio representation. ##' (This parameter is automatically passed by aldex)
gamma	Non-negative scalar. Standard deviation of the Gaussian perturbation that relaxes the CLR assumption about scale. gamma = 0 yields the pure CLR assumption; recommended default values in the scale-uncertainty literature are often around 0.5, but appropriate values depend on how strongly you trust the CLR scale assumption in the current study.

Details

In the limit gamma = 0, this reduces to the CLR assumption (no scale uncertainty beyond the CLR-implied scale). Larger gamma values represent increasing uncertainty about the CLR-implied scale differences.

Value

A numeric matrix of dimension N x nsample giving Monte Carlo samples of the log-scale for each sample (rows) and each Monte Carlo draw (columns).

Author(s)

Justin Silverman

References

Nixon G, Gloor GB, Silverman JD (2025). "Incorporating scale uncertainty in microbiome and gene expression analysis as an extension of normalization". Genome Biology. doi:10.1186/s13059-025036093

`coefficient.sm` *Coefficient-based scale model with user-specified prior on fixed effects*

Description

Draws Monte Carlo samples of the log2 scale by sampling fixed-effect coefficients from a multivariate normal distribution and mapping them through the design matrix `X`. This scale model is useful when you want to encode prior information about how covariates (e.g., treatment, batch, time) affect scale, rather than specifying scale moments directly per sample.

Usage

```
coefficient.sm(X, logComp, c.mu = NULL, c.cor = NULL)
```

Arguments

`X`	A numeric design matrix passed internally by `aldex()` to the scale model. Rows correspond to fixed-effect coefficients/covariates (`P = nrow(X)`) and columns correspond to samples (`N = ncol(X)`). (Automatically supplied by `aldex()`.)
`logComp`	A numeric array of Monte Carlo log-compositions with dimensions `features x samples x nsample`. This scale model uses `nsample` (the number of Monte Carlo draws) but does not otherwise use `logComp`. (Automatically supplied by `aldex()`.)
`c.mu`	Numeric vector of length `P` giving the mean of the fixed effect coefficients in log2-scale space. Must not be `NULL`.
`c.cor`	Numeric `P x P` covariance matrix for the fixed effect coefficients in log2-scale space. Must not be `NULL`.

Details

Specifically, for each Monte Carlo draw $b^{(m)} \sim N(c.mu, c.cor)$, the per-sample log2 scale is computed as $b^{(m)T} X$, producing an `N x nsample` matrix of log2-scale draws.

For example, with an intercept and a treatment indicator where treatment is expected to increase log2 scale by ~1 on average, one might use `c.mu = c(0, 1)` and `c.cor = diag(c(0.25, 0.25))` (i.e., SD 0.5 for each coefficient, independent).

Value

A numeric matrix of dimension `N x nsample` giving Monte Carlo draws of the log2 scale for each sample (rows) across `nsample` draws (columns).

Author(s)

Kyle McGovern

`cohensd` *Cohen's D*

Description

Function to compute cohensd on the results provided by the aldex function

Usage

```
cohensd(m, var)
```

Arguments

`m`	the output of a call to `aldex`
`var`	if aldex was called with X being a pre-computed model matrix, this var should be an integer corresponding to a binary covariate indicateing the desired effect size to calculate (an effect size between two groups indicated by the binary covariate). For example, if the third covariate in the model is an indicator denoting health (0) and disease (1) then set `var=3`. In contrast, if X was a formula (in which case the `data` argument should have been specified) then `var` can be set to the unquoted name of the binary condition variable (e.g., `var=condition`).

Details

WARNING: this function is experimental and requires users read the documetation fully.

Value

A (D x nsample)-matrix of Cohen's D statistics for the variable of interest

Author(s)

Justin Silverman

combine.streams *Combine output from aldex streams (internal only)*

Description

Takes a list l, where every element is itself a list of 3D arrays. Each array should have matching first and second dimentions. This function combines those arrays along the third dimension.

Usage

```
combine.streams(l)
```

Arguments

l	a list, where every element is itself a list of 3D arrays.

Value

a list of 3D arrays

Author(s)

Justin Silverman

fflm *Freaking Fask Linear Models*

Description

Tailored for ALDEx3 where covariates are shared between massive numbers of linear regressions where only Y is changing.

Usage

```
fflm(Y, X, test = "t.HC3")
```

Arguments

Y	a numeric array (N x D X S) where D is the number of taxa/genes, N is the number of samples, and S is the number of posterior samples
X	a numeric matrix (N x P) where P is number of covariates
test	(default t.HC3), "t", t test is performed for each covariate (fast); "t.HC0" Heteroskedasticsity-Robust Standard Errors used (HC0; White's; slower); "t.HC3" (default) Heteroskedasticsity-Robust Standard Errors used (HC3; unlike HC0, this includes a leverage adjustment and is better for small sample sizes or when there are data with high leverage; slowest). To learn more about these, loko at Long and Ervin (2000) Using Heteroscedasticity Consistent Standard Errors in the Linear Regression Model, The American Statistician.

Value

A list of (P x D x S)-arrays with the OLS point estimates, the standard errors, and the two-sided p-values for each coefficient (P), of each model fit to each taxa (D) and each posterior sample (S)

Author(s)

Justin Silverman

`gut_crohns_data` *Gut Crohn's microbiome dataset (list)*

Description

This dataset contains a matrix and a data frame: genus-level microbiome profiles and corresponding sample metadata from a Crohn's disease case–control cohort. The dataset is used in examples and vignettes throughout the package.

Usage

```
gut_crohns_data
```

Format

A named list of one matrix and a dataframe:

`counts` A matrix with read counts with 195 rows and 95 columns)

`metadata` A data frame with 95 rows and 7 columns containing subject-level covariates.

Details

The `counts` matrix has one row per sample and one column per genus. The `metadata` data frame has one row per sample with metadata, critically Health.status either CD or Control, and Average cell count per gram frozen feces.

Source

Vandeputte D, Falony G, Vieira-Silva S, Wang J, Sailer M, Theis S, Raes J (2017). "Quantitative microbiome profiling links gut community variation to microbial load". Nature, 551, 507–511. doi:10.1038/nature24460

miniclo *Closure operation*

Description

Closure operation

Usage

```
miniclo(X)
```

Arguments

X	should be a D x N matrix, closes along the D dimension

Value

D x N matrix with columns summing to 1

Author(s)

Justin Silverman

oral_mouthwash_data *Oral microbiome perturbation dataset (list)*

Description

This dataset contains a matrix and a data frame: genus-level microbiome profiles and corresponding sample metadata from an oral microbiome perturbation study. 28 participants' oral microbiomes were measured before, 15 minutes after, and 2 hours after perturbation with either a water control, antiseptic mouthwash, alchohol-free mouthwash, or soda. The dataset is used in examples and vignettes throughout the package.

Usage

```
oral_mouthwash_data
```

Format

A named list of one matrix and a dataframe:

`counts` A matrix with read counts with 116 rows and 81 columns)

`metadata` A data frame with 81 rows and 38 columns containing sample-level covariates.

Details

The counts matrix has one row per sample and one column per genus. The metadata data frame has one row per sample with metadata, critically the participant ID, flow cytometry average (and replicate) cells, time_c (time points), and treat (the perturbations)

Source

Marotz C, Morton JT, Navarro P, Coker J, Belda-Ferre P, Knight R (2021). "Quantifying live microbial load in human saliva samples over time reveals stable composition and dynamic load". mSystems, 6(3), e01182-21. doi:10.1128/mSystems.0118220

---

req *Test object contains elements*

Description

Test object contains elements

Usage

```
req(obj, names)
```

Arguments

obj	object (list type)
names	names of elements required to be in the list

Author(s)

Justin Silverman

---

rLogDirichlet *Function for sampling Dirichlet random variables (base-2 normalized via log-sum-exp for stability)*

Description

Function for sampling Dirichlet random variables (base-2 normalized via log-sum-exp for stability)

Usage

```
rLogDirichlet(n, alpha)
```

Arguments

n	number of samples
alpha	D-vector of Dirichlet parameters

Value

a D x n matrix of samples

Author(s)

Justin Silverman

---

sample.sm *Sample-specific scale model with user-specified mean and variance/covariance*

Description

Draws Monte Carlo samples of the log2 scale for each sample using user-supplied moments. This scale model is useful when external measurements (e.g., qPCR, flow cytometry, spike-ins) provide information about absolute scale, or when you want to encode prior information about scale on a per-sample basis.

Usage

```
sample.sm(X, logComp, s.mu = NULL, s.var = NULL, s.cor = NULL)
```

Arguments

X	A numeric design matrix passed internally by `aldex()` to the scale model. Columns correspond to samples (`N = ncol(X)`). This scale model does not use `X` directly, but `N` is inferred from it. (Automatically supplied by `aldex()`.)
logComp	A numeric array of Monte Carlo log-compositions with dimensions `features x samples x nsample`. This scale model uses `nsample` to determine the number of Monte Carlo draws, but does not otherwise use `logComp`. (Automatically supplied by `aldex()`.)
s.mu	Numeric vector of length `N` giving the mean of the log2 scale for each sample. Must not be `NULL`.
s.var	Numeric vector of length `N` giving the marginal variance of the log2 scale for each sample. Use this when assuming samples' log2 scales are independent. Must be `NULL` if `s.cor` is provided.
s.cor	Numeric `N x N` covariance matrix for the log2 scale across samples. Use this when encoding correlations between samples' log2 scales. Must be `NULL` if `s.var` is provided.

Details

Exactly one of `s.var` or `s.cor` must be provided:

- `s.var`: independent per-sample log2-scale variance (diagonal covariance)
- `s.cor`: full `N` x `N` log2-scale covariance matrix

The returned matrix has `N` rows (samples) and `nsample` columns (Monte Carlo draws), consistent with the ALDEx3 scale-model interface.

Value

A numeric matrix of dimension `N` x `nsample` giving Monte Carlo draws of the log2 scale for each sample (rows) across `nsample` draws (columns).

Author(s)

Kyle McGovern

`sr.mem` *Implementation of SR-MEM: scale-reliant mixed effects models.*

Description

Implementation of SR-MEM: scale-reliant mixed effects models.

Usage

```
sr.mem(logW, formula, data, n.cores, method, mem.args)
```

Arguments

`logW`	a numeric array (N x D X S) where D is the number of taxa, N is the number of samples, and S is the number of posterior samples
`formula`	an lme4 formula with fixed and random effects for lmer
`data`	A data.frame with the random and fixed effects items in formula
`n.cores`	The number of cores for parallelization. If n.cores=1, no parallelization is used
`method`	The mem method to use: either nlme or lme4
`mem.args`	Additional arguments including random or correlation

Value

A list of (P x D x S)-arrays with the fixed effect point estimates, the standard errors, degrees of freedom and the lower and upper p-values for each coefficient (P), of each model fit to each taxa (D) and each posterior sample (S) and a (Pr x D x S)-array with (Pr) random effects.

Author(s)

Kyle McGovern

summary.aldex *Summary Method for ALDEx3 Objects*

Description

Summarize an ALDEx3 result object

Usage

```
## S3 method for class 'aldex'
summary(object, ignore.intercept = TRUE, ...)
```

Arguments

object	An object of class aldex
ignore.intercept	(default=TRUE), ignore intercept when creating summary table
...	Additional arguments (currently ignored).

Details

Provides a summary of the adjusted p-values, estimates, and standard errors from an ALDEx3 result object.

This method extracts adjusted p-values from object$p.val.adj, along with posterior estimates and standard errors averaged across Monte Carlo samples. The result is returned as a long-format data.frame suitable for downstream analysis or visualization.

Value

A data.frame with columns parameter, entity, p.val.adjusted, estimate, and std.error.

Author(s)

Justin Silverman

tss.sm *TSS-centered scale model (with optional scale uncertainty)*

Description

Implements a total-sum-scaling (TSS)-centered variant of the default scale-uncertainty model described in Nixon et al. (Beyond Normalizations / scale-uncertainty framework). Unlike clr.sm, which is centered on the CLR-implied scale, this model is centered on the TSS assumption that there is no systematic change in scale across.

Usage

```
tss.sm(X, logComp, gamma = 0.5)
```

Arguments

X	A numeric design matrix passed internally by aldex() to the scale model. Rows correspond to fixed-effect coefficients/covariates (P = nrow(X)) and columns correspond to samples (N = ncol(X)). (Automatically supplied by aldex().)
logComp	A numeric array of Monte Carlo log-compositions with dimensions features x samples x nsample. This scale model uses nsample (the number of Monte Carlo draws) but does not otherwise use logComp. (Automatically supplied by aldex().)
gamma	Non-negative scalar. Standard deviation of the Gaussian perturbation applied to the scale-model coefficients (in log2 space). gamma = 0 implies no scale uncertainty (all draws are centered at zero effect); larger values allow greater departures from the TSS-centered assumption.

Details

Scale uncertainty is introduced via an additive Gaussian perturbation on the (log2) fixed effects. For each Monte Carlo draw, a coefficient vector is sampled as $b^{(m)} \sim N(0, \gamma^2 I)$, and the per-sample log2 scale is computed as $b^{(m)T} X$. Larger values of gamma correspond to weaker confidence in the TSS-centered assumption (more allowed scale variation); gamma = 0 yields no scale variation beyond the model center.

Note: logComp is included to match the ALDEx3 scale-model interface and to determine nsample, but it is not otherwise used by this model.

Value

A numeric matrix of dimension N x nsample giving Monte Carlo draws of the log2 scale for each sample (rows) across nsample draws (columns).

Author(s)

Justin Silverman

References

Nixon G, Gloor GB, Silverman JD (2025). "Incorporating scale uncertainty in microbiome and gene expression analysis as an extension of normalization". Genome Biology. doi:10.1186/s13059-025036093

Index