Package 'corncob'

March 29, 2025

Title Count Regression for Correlated Observations with the Beta-Binomial

Version 0.4.2

Description Statistical modeling for correlated count data using the beta-binomial distribution, described in Martin et al. (2020) <doi:10.1214/19-AOAS1283>. It allows for both mean and overdispersion covariates.

URL `https://github.com/statdivlab/corncob`,
`https://statdivlab.github.io/corncob/`

BugReports `https://github.com/statdivlab/corncob/issues`

Depends R (>= 3.2)

License GPL (>= 2)

Imports stats, utils, VGAM, numDeriv, ggplot2, trust, dplyr, magrittr, detectseparation, scales, rlang

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Suggests knitr, rmarkdown, testthat, covr, limma, slam, R.rsp, optimx, phyloseq

VignetteBuilder knitr

NeedsCompilation no

Author Bryan D Martin [aut],
Daniela Witten [aut],
Sarah Teichman [ctb],
Amy D Willis [aut, cre],
Thomas W Yee [ctb] (VGAM library),
Xiangjie Xue [ctb] (VGAM library)

Maintainer Amy D Willis <`adwillis@uw.edu`>

Repository CRAN

Date/Publication 2025-03-29 15:30:19 UTC

Contents

`corncob-package` *Corncob package documentation.*

Description

Corncob provides methods for estimating and plotting count data. Specifically, corncob is designed to account for the challenges of modeling sequencing data from microbial abundance studies.

Details

For details on the model implemented in this package, see Martin et al. (2020) <doi:10.1214/19-AOAS1283>.

The development version of the package will be maintained on `https://github.com/statdivlab/corncob`.

Value

No return value. Created for documentation.

`bbdml` *Maximum Likelihood for the Beta-binomial Distribution*

Description

Maximum Likelihood for the Beta-binomial Distribution

Usage

```
bbdml(
  formula,
  phi.formula,
  data,
  link = "logit",
  phi.link = "logit",
  method = "trust",
```

```
  control = list(maxit = 1000, reltol = 1e-14),
  numerical = FALSE,
  nstart = 1,
  inits = NULL,
  allow_noninteger = FALSE,
  robust = FALSE,
  ...
)
```

Arguments

`formula`	an object of class `formula`: a symbolic description of the model to be fitted to the abundance
`phi.formula`	an object of class `formula` without the response: a symbolic description of the model to be fitted to the dispersion
`data`	a data frame or `phyloseq` object containing the variables in the models
`link`	link function for abundance covariates, defaults to `"logit"`
`phi.link`	link function for dispersion covariates, defaults to `"logit"`
`method`	optimization method, defaults to `"trust"`, or see `optimr` for other options
`control`	optimization control parameters (see `optimr`)
`numerical`	Boolean. Defaults to `FALSE`. Indicator of whether to use the numeric Hessian (not recommended).
`nstart`	Integer. Defaults to `1`. Number of starts for optimization.
`inits`	Optional initializations as rows of a matrix. Defaults to `NULL`.
`allow_noninteger`	Boolean. Defaults to `FALSE`. Should noninteger W's and M's be allowed? This behavior was not permitted prior to v4.1, needs to be explicitly allowed.
`robust`	Should robust standard errors be returned? If not, model-based standard arras are used. Logical, defaults to `FALSE`.
`...`	Optional additional arguments for `optimr` or `trust`

Value

An object of class `bbdml`.

Examples

```
# data frame example
data(soil_phylum_small_otu1)
bbdml(formula = cbind(W, M - W) ~ DayAmdmt,
phi.formula = ~ DayAmdmt,
data = soil_phylum_small_otu1)

# phyloseq example (only run this if you have phyloseq installed)
## Not run:
data(soil_phylum_small_sample)
```

```
data(soil_phylum_small_otu)
data_phylo <- phyloseq::phyloseq(phyloseq::sample_data(soil_phylum_small_sample),
phyloseq::otu_table(soil_phylum_small_otu, taxa_are_rows = TRUE))
bbdml(formula = Proteobacteria ~ DayAmdmt,
phi.formula = ~ DayAmdmt,
data = data_phylo)

## End(Not run)
```

`checkNested` *Check for nested models*

Description

Check for nested models

Usage

```
checkNested(mod, mod_null)
```

Arguments

`mod`	an object of class `bbdml`
`mod_null`	an object of class `bbdml`

Value

`TRUE` if `mod_null` is nested within `mod`, otherwise it throws an error.

Examples

```
data(soil_phylum_small_otu1)
mod1 <- bbdml(formula = cbind(W, M - W) ~ DayAmdmt,
phi.formula = ~ DayAmdmt,
data = soil_phylum_small_otu1)

mod2 <- bbdml(formula = cbind(W, M - W) ~ 1,
phi.formula = ~ 1,
data = soil_phylum_small_otu1)

checkNested(mod1, mod2)
```

`clean_taxa_names` *Rename taxa*

Description

Renames taxa to have short human-readable names

Usage

```
clean_taxa_names(x, name = "OTU")
```

Arguments

`x`	Object of class `phyloseq`
`name`	Character, defaults to `"OTU"`. Optional. String to use in every taxa name.

Details

The original taxa names are saved as the `original_names` attribute. See the example for an example of how to access the original names.

Value

Object of class `phyloseq`, with taxa renamed (defaults to OTU1, OTU2, ...), with the original taxa names saved as an attribute.

`contrastsTest` *Identify differentially-abundant and differentially-variable taxa using contrasts*

Description

Identify differentially-abundant and differentially-variable taxa using contrasts

Usage

```
contrastsTest(
  formula,
  phi.formula,
  contrasts_DA = NULL,
  contrasts_DV = NULL,
  data,
  link = "logit",
  phi.link = "logit",
  sample_data = NULL,
  taxa_are_rows = TRUE,
```

```
  filter_discriminant = TRUE,
  fdr_cutoff = 0.05,
  fdr = "fdr",
  inits = NULL,
  try_only = NULL,
  ...
)
```

Arguments

`formula`	an object of class `formula` without the response: a symbolic description of the model to be fitted to the abundance
`phi.formula`	an object of class `formula` without the response: a symbolic description of the model to be fitted to the dispersion
`contrasts_DA`	List. Optional. Constructs a contrast matrix. List elements should be characters specifying contrasts in the parameters within `formula`. Note that this is only available with `"Wald"` value for `test`. Must include at least one of `contrasts_DA` or `contrasts_DV`.
`contrasts_DV`	List. Optional. Constructs a contrast matrix. List elements should be characters specifying contrasts in the parameters within `phi.formula`. Note that this is only available with `"Wald"` value for `test`. Must include at least one of `contrasts_DA` or `contrasts_DV`.
`data`	a data frame containing the OTU table, or `phyloseq` object containing the variables in the models
`link`	link function for abundance covariates, defaults to `"logit"`
`phi.link`	link function for dispersion covariates, defaults to `"logit"`
`sample_data`	Data frame or matrix. Defaults to `NULL`. If `data` is a data frame or matrix, this must be included as covariates/sample data.
`taxa_are_rows`	Boolean. Optional. If `data` is a data frame or matrix, this indicates whether taxa are rows. Defaults to `TRUE`.
`filter_discriminant`	Boolean. Defaults to `TRUE`. If `FALSE`, discriminant taxa will not be filtered out.
`fdr_cutoff`	Integer. Defaults to `0.05`. Desired type 1 error rate
`fdr`	Character. Defaults to `"fdr"`. False discovery rate control method, see `p.adjust` for more options.
`inits`	Optional initializations for model fit using `formula` and `phi.formula` as rows of a matrix. Defaults to `NULL`.
`try_only`	Optional numeric. Will try only the `try_only` taxa, specified either via numeric input or character taxa names. Useful for speed when troubleshooting. Defaults to `NULL`, testing all taxa.
`...`	Optional additional arguments for `bbdml`

Details

This function uses contrast matrices to test for differential abundance and differential variability using a Wald-type chi-squared test. To use a formula implementation, see `differentialTest`.

Value

An object of class `contrastsTest`. List with elements `p` containing the p-values for each contrast, `p_fdr` containing the p-values after false discovery rate control, `significant_taxa` containing the taxa names of the statistically significant taxa, `contrasts_DA` containing the contrast matrix for parameters associated with the abundance, `contrasts_DV` containing the contrast matrix for parameters associated with the dispersion, `discriminant_taxa_DA` containing the taxa for which at least one covariate associated with the abundance was perfectly discriminant, `discriminant_taxa_DV` containing the taxa for which at least one covariate associated with the dispersion was perfectly discriminant, and `data` containing the data used to fit the models.

Examples

```
# data frame example
# note that this function will only run if the `limma` package is installed
limma_install <- try(find.package("limma"), silent = TRUE)
if (!(inherits(limma_install, "try-error"))) {
  data(soil_phylum_contrasts_sample)
  data(soil_phylum_contrasts_otu)
  da_analysis <- contrastsTest(formula = ~ DayAmdmt,
                             phi.formula = ~ DayAmdmt,
                             contrasts_DA = list("DayAmdmt21 - DayAmdmt11",
                                                 "DayAmdmt22 - DayAmdmt21"),
                             data = soil_phylum_contrasts_otu,
                              sample_data = soil_phylum_contrasts_sample,
                              fdr_cutoff = 0.05,
                             try_only = 1:5)
}

# phyloseq example (only run if you have phyloseq installed)
## Not run:
contrasts_phylo <- phyloseq::phyloseq(phyloseq::sample_data(soil_phylum_contrasts_sample),
phyloseq::otu_table(soil_phylum_contrasts_otu, taxa_are_rows = TRUE))
da_analysis <- contrastsTest(formula = ~ DayAmdmt,
                             phi.formula = ~ DayAmdmt,
                             contrasts_DA = list("DayAmdmt21 - DayAmdmt11",
                                                 "DayAmdmt22 - DayAmdmt21"),
                             data = contrasts_phylo,
                             fdr_cutoff = 0.05,
                             try_only = 1:5)

## End(Not run)
```

`convert_phylo` *Function to subset and convert phyloseq data*

Description

Function to subset and convert phyloseq data

Usage

```
convert_phylo(data, select)
```

Arguments

`data`	a `phyloseq` object
`select`	Name of OTU or taxa to select, must match taxa name in `data`

Value

A `data.frame` object, with elements `W` as the observed counts, `M` as the sequencing depth, and the sample data with their original names.

`coth` *Hyperbolic cotangent transformation*

Description

Hyperbolic cotangent transformation

Usage

```
coth(x)
```

Arguments

`x`	data

Value

Hyperbolic cotangent transformation of x

Examples

```
x <- .5
coth(x)
```

`dbetabin` *Betabinomial density*

Description

Betabinomial density

Usage

```
dbetabin(theta, W, M, X, X_star, np, npstar, link, phi.link, logpar = TRUE)
```

Arguments

`theta`	Numeric vector. Parameters associated with `X` and `X_star`
`W`	Numeric vector of counts
`M`	Numeric vector of sequencing depth
`X`	Matrix of covariates associated with abundance (including intercept)
`X_star`	Matrix of covariates associated with dispersion (including intercept)
`np`	Number of covariates associated with abundance (including intercept)
`npstar`	Number of covariates associated with dispersion (including intercept)
`link`	ink function for abundance covariates
`phi.link`	ink function for dispersion covariates
`logpar`	Boolean. Defaults to `TRUE`. Indicator of whether to return log-likelihood.

Value

Negative beta-binomial (log-)likelihood

`dbetabinom_cts` *Densities of beta binomial distributions, permitting non integer x and size*

Description

In some cases we may not have integer W and M's. In these cases, we can still use corncob to estimate parameters, but we need to think of them as no longer coming from the specific beta binomial parametric model, and instead from an estimating equations framework.

Usage

```
dbetabinom_cts(x, size, prob, rho = 0, log = FALSE)
```

Arguments

`x`	the value at which defined the density
`size`	number of trials
`prob`	the probability of success
`rho`	the correlation parameter
`log`	if TRUE, log-densities p are given

Author(s)

Thomas W Yee

Xiangjie Xue

Amy D Willis

`dbetabin_neg` *Negative betabinomial density*

Description

Created as a convenient helper function for optimization. Not intended for users.

Usage

```
dbetabin_neg(theta, W, M, X, X_star, np, npstar, link, phi.link, logpar = TRUE)
```

Arguments

`theta`	Numeric vector. Parameters associated with `X` and `X_star`
`W`	Numeric vector of counts
`M`	Numeric vector of sequencing depth
`X`	Matrix of covariates associated with abundance (including intercept)
`X_star`	Matrix of covariates associated with dispersion (including intercept)
`np`	Number of covariates associated with abundance (including intercept)
`npstar`	Number of covariates associated with dispersion (including intercept)
`link`	ink function for abundance covariates
`phi.link`	ink function for dispersion covariates
`logpar`	Boolean. Defaults to `TRUE`. Indicator of whether to return log-likelihood.

Value

Negative beta-binomial (log-)likelihood

`differentialTest` *Identify differentially-abundant and differentially-variable taxa*

Description

Identify differentially-abundant and differentially-variable taxa

Usage

```
differentialTest(
  formula,
  phi.formula,
  formula_null,
  phi.formula_null,
  data,
  link = "logit",
  phi.link = "logit",
  test,
  boot = FALSE,
  B = 1000,
  sample_data = NULL,
  taxa_are_rows = TRUE,
  filter_discriminant = TRUE,
  fdr_cutoff = 0.05,
  fdr = "fdr",
  full_output = FALSE,
  inits = NULL,
  inits_null = NULL,
  try_only = NULL,
  verbose = FALSE,
  robust = FALSE,
  ...
)
```

Arguments

`formula`	an object of class `formula` without the response: a symbolic description of the model to be fitted to the abundance
`phi.formula`	an object of class `formula` without the response: a symbolic description of the model to be fitted to the dispersion
`formula_null`	Formula for mean under null, without response
`phi.formula_null`	Formula for overdispersion under null, without response
`data`	a data frame containing the OTU table, or `phyloseq` object containing the variables in the models
`link`	link function for abundance covariates, defaults to `"logit"`

`phi.link`	link function for dispersion covariates, defaults to `"logit"`
`test`	Character. Hypothesis testing procedure to use. One of `"Wald"`, `"LRT"` (likelihood ratio test), or `"Rao"`.
`boot`	Boolean. Defaults to `FALSE`. Indicator of whether or not to use parametric bootstrap algorithm. (See `pbWald` and `pbLRT`).
`B`	Optional integer. Number of bootstrap iterations. Ignored if `boot` is `FALSE`. Otherwise, defaults to `1000`.
`sample_data`	Data frame or matrix. Defaults to `NULL`. If `data` is a data frame or matrix, this must be included as covariates/sample data.
`taxa_are_rows`	Boolean. Optional. If `data` is a data frame or matrix, this indicates whether taxa are rows. Defaults to `TRUE`.
`filter_discriminant`	Boolean. Defaults to `TRUE`. If `FALSE`, discriminant taxa will not be filtered out.
`fdr_cutoff`	Integer. Defaults to `0.05`. Desired false discovery rate.
`fdr`	Character. Defaults to `"fdr"`. False discovery rate control method, see `p.adjust` for more options.
`full_output`	Boolean. Opetional. Defaults to `FALSE`. Indicator of whether to include full `bbdml` model output for all taxa.
`inits`	Optional initializations for model fit using `formula` and `phi.formula` as rows of a matrix. Defaults to `NULL`.
`inits_null`	Optional initializations for model fit using `formula_null` and `phi.formula_null` as rows of a matrix. Defaults to `NULL`.
`try_only`	Optional numeric. Will try only the `try_only` taxa, specified either via numeric input or character taxa names. Useful for speed when troubleshooting. Defaults to `NULL`, testing all taxa.
`verbose`	Boolean. Defaults to `FALSE`; print status updates for long-running analyses
`robust`	Should robust standard errors be used? If not, model-based standard errors are used. Logical, defaults to `FALSE`.
`...`	Optional additional arguments for `bbdml`

Details

See package vignette for details and example usage. Make sure the number of columns in all of the initializations are correct! `inits` probably shouldn't match `inits_null`. To use a contrast matrix, see `contrastsTest`.

Value

An object of class `differentialTest`. List with elements `p` containing the p-values, `p_fdr` containing the p-values after false discovery rate control, `significant_taxa` containing the taxa names of the statistically significant taxa, `significant_models` containing a list of the model fits for the significant taxa, `all_models` containing a list of the model fits for all taxa, `restrictions_DA` containing a list of covariates that were tested for differential abundance, `restrictions_DV` containing a list of covariates that were tested for differential variability, `discriminant_taxa_DA` containing the taxa for which at least one covariate associated with the abundance was perfectly discriminant, `discriminant_taxa_DV` containing the taxa for which at least one covariate associated with the dispersion was perfectly discriminant, `data` containing the data used to fit the models. If `full_output = TRUE`, it will also include `full_output`, a list of all model output from `bbdml`.

Examples

```
# data frame example
data(soil_phylum_small_sample)
data(soil_phylum_small_otu)
da_analysis <- differentialTest(formula = ~ DayAmdmt,
                                phi.formula = ~ DayAmdmt,
                                formula_null = ~ 1,
                                phi.formula_null = ~ DayAmdmt,
                                test = "Wald", boot = FALSE,
                                data = soil_phylum_small_otu,
                                sample_data = soil_phylum_small_sample,
                                fdr_cutoff = 0.05,
                                try_only = 1:5)

# phyloseq example (only run if you have phyloseq installed)
## Not run:
data_phylo <- phyloseq::phyloseq(phyloseq::sample_data(soil_phylum_small_sample),
phyloseq::otu_table(soil_phylum_small_otu, taxa_are_rows = TRUE))
da_analysis <- differentialTest(formula = ~ DayAmdmt,
                               phi.formula = ~ DayAmdmt,
                               formula_null = ~ 1,
                               phi.formula_null = ~ DayAmdmt,
                               test = "Wald", boot = FALSE,
                               data = data_phylo,
                               fdr_cutoff = 0.05,
                               try_only = 1:5)

## End(Not run)
```

`doBoot` *Function to run a bootstrap iteration*

Description

Internal function. Not intended for users.

Usage

```
doBoot(mod, mod_null, test, robust = FALSE)
```

Arguments

`mod`	an object of class `bbdml`
`mod_null`	an object of class `bbdml`
`test`	Character. Hypothesis testing procedure to use. One of `"Wald"` or `"LRT"` (likelihood ratio test).
`robust`	Should robust standard errors be used? If not, model-based standard arras are used. Logical, defaults to `FALSE`.

Value

test statistic from one bootstrap iteration

`fishZ` *Fisher's z transformation*

Description

Fisher's z transformation

Usage

```
fishZ(x)
```

Arguments

`x`	data

Value

Fisher's z transformation of x

Examples

```
x <- .5
fishZ(x)
```

`genInits` *Generate initialization for optimization*

Description

Generate initialization for optimization

Usage

```
genInits(W, M, X, X_star, np, npstar, link, phi.link, nstart = 1, use = TRUE)
```

Arguments

`W`	Numeric vector of counts
`M`	Numeric vector of sequencing depth
`X`	Matrix of covariates associated with abundance (including intercept)
`X_star`	Matrix of covariates associated with dispersion (including intercept)
`np`	Number of covariates associated with abundance (including intercept)
`npstar`	Number of covariates associated with dispersion (including intercept)
`link`	ink function for abundance covariates
`phi.link`	ink function for dispersion covariates
`nstart`	Integer. Defaults to `1`. Number of starts for optimization.
`use`	Boolean. Defaults to `TRUE`. Indicator of whether to use deterministic intialization.

Value

Matrix of initializations

Examples

```
set.seed(1)
seq_depth <- rpois(20, lambda = 10000)
my_counts <- rbinom(20, size = seq_depth, prob = 0.001) * 10
my_covariate <- cbind(rep(c(0,1), each = 10))
colnames(my_covariate) <- c("X1")

genInits(W = my_counts, M = seq_depth,
       X = cbind(1, my_covariate), X_star = cbind(1, my_covariate),
       np = 2, npstar = 2,
       link = "logit",
       phi.link = "logit", nstart = 2, use = TRUE)
```

`getRestrictionTerms` *Get index of restricted terms for Wald test*

Description

Created as a convenient helper function. Not intended for users.

Usage

```
getRestrictionTerms(
  mod,
  mod_null = NULL,
  restrictions = NULL,
  restrictions.phi = NULL
)
```

Arguments

`mod`	an object of class `bbdml`
`mod_null`	Optional. An object of class `bbdml`. Defaults to `NULL`
`restrictions`	Optional. Defaults to `NULL`. Numeric vector indicating the parameters associated with the abundance to test, or character vector with name of variable to test. Note that `1` is the intercept associated with the abundance.
`restrictions.phi`	Optional. Defaults to `NULL`. Numeric vector indicating the parameters associated with the dispersion to test, or character vector with name of variable to test. Note that `1` is the intercept associated with the dispersion.

Value

A list with `mu` representing the index of the restricted covariates associated with abundance and `phi` representing the index of the restricted covarates associated with the dispersion

---

`gr_full` *Parameter Gradient Vector*

Description

Used for internal optimization. Not intended for users.

Usage

```
gr_full(theta, W, M, X, X_star, np, npstar, link, phi.link, logpar = TRUE)
```

Arguments

`theta`	Numeric vector. Parameters associated with `X` and `X_star`
`W`	Numeric vector of counts
`M`	Numeric vector of sequencing depth
`X`	Matrix of covariates associated with abundance (including intercept)
`X_star`	Matrix of covariates associated with dispersion (including intercept)
`np`	Number of covariates associated with abundance (including intercept)
`npstar`	Number of covariates associated with dispersion (including intercept)
`link`	ink function for abundance covariates
`phi.link`	ink function for dispersion covariates
`logpar`	Boolean. Defaults to `TRUE`. Indicator of whether to return log-likelihood.

Value

Gradient of likelihood with respect to parameters

`HDIbetabinom` *Get highest density interval of beta-binomial*

Description

Get highest density interval of beta-binomial

Usage

```
HDIbetabinom(percent, M, mu, phi)
```

Arguments

`percent`	Numeric. Percent interval desired.
`M`	Numeric vector of sequencing depth
`mu`	Numeric vector of abundance parameter
`phi`	Numeric vector of dispersion parameter

Value

List where `lower` represents the lower bound and `upper` represents the upper bound

Examples

```
data(soil_phylum_small_otu1)
mod <- bbdml(formula = cbind(W, M - W) ~ DayAmdmt,
phi.formula = ~ DayAmdmt,
data = soil_phylum_small_otu1)
HDIbetabinom(.95, M = mod$M[1], mu = mod$mu.resp[1], phi = mod$phi.resp[1])
```

`hessian` *Compute Hessian matrix at the MLE*

Description

Compute Hessian matrix at the MLE

Usage

```
hessian(mod, numerical = FALSE)
```

Arguments

`mod`	an object of class `bbdml`
`numerical`	Boolean. Defaults to `FALSE`. Indicator of whether to use the numeric Hessian (not recommended).

Value

Hessian matrix at the MLE. In this setting, it's hard to compute expectations to calculate the information matrix, so we return the consistent estimate using sample moments: $\hat{A}(\hat{\theta}) = \sum_i \frac{\partial^2}{\partial\theta\partial\theta^T} l(\theta, W_i)$ evaluated at $\theta = \hat{\theta}$.

Examples

```
data(soil_phylum_small_otu1)
mod <- bbdml(formula = cbind(W, M - W) ~ DayAmdmt,
phi.formula = ~ DayAmdmt,
data = soil_phylum_small_otu1)
hessian(mod)
```

`ibd_phylo_otu` *IBD data, OTU count data frame.*

Description

OTU data frame from a phyloseq object from an IBD microbiome study.

Usage

```
ibd_phylo_otu
```

Format

A data frame of OTU counts.

References

Papa, E., Docktor, M., Smillie, C., Weber, S., Preheim, S. P., Gevers, D., Giannoukos, G., Ciulla, D., Tabbaa, D., Ingram, J., Schauer, D. B., Ward, D. V., Korzenik, J. R., Xavier, R. J., Bousvaros, A., Alm, E. J. & Schauer, D. B. (2012). *Non-invasive mapping of the gastrointestinal microbiota identifies children with inflammatory bowel disease*. PloS One, 7(6), e39242. <doi.org/10.1371/journal.pone.0039242>.

Duvallet, C., Gibbons, S., Gurry, T., Irizarry, R., & Alm, E. (2017). *MicrobiomeHD: the human gut microbiome in health and disease [Data set]*. Zenodo. <doi.org/10.5281/zenodo.1146764>.

`ibd_phylo_sample` *IBD data, sample data frame.*

Description

Sample data from a phyloseq object from an IBD microbiome study.

Usage

```
ibd_phylo_sample
```

Format

A data frame of sample data.

References

Papa, E., Docktor, M., Smillie, C., Weber, S., Preheim, S. P., Gevers, D., Giannoukos, G., Ciulla, D., Tabbaa, D., Ingram, J., Schauer, D. B., Ward, D. V., Korzenik, J. R., Xavier, R. J., Bousvaros, A., Alm, E. J. & Schauer, D. B. (2012). *Non-invasive mapping of the gastrointestinal microbiota identifies children with inflammatory bowel disease*. PloS One, 7(6), e39242. <doi.org/10.1371/journal.pone.0039242>.

Duvallet, C., Gibbons, S., Gurry, T., Irizarry, R., & Alm, E. (2017). *MicrobiomeHD: the human gut microbiome in health and disease [Data set]*. Zenodo. <doi.org/10.5281/zenodo.1146764>.

`ibd_phylo_taxa` *IBD data, taxonomy data frame.*

Description

Taxonomy data from a phyloseq object from an IBD microbiome study.

Usage

```
ibd_phylo_taxa
```

Format

A data frame of taxonomy data.

References

Papa, E., Docktor, M., Smillie, C., Weber, S., Preheim, S. P., Gevers, D., Giannoukos, G., Ciulla, D., Tabbaa, D., Ingram, J., Schauer, D. B., Ward, D. V., Korzenik, J. R., Xavier, R. J., Bousvaros, A., Alm, E. J. & Schauer, D. B. (2012). *Non-invasive mapping of the gastrointestinal microbiota identifies children with inflammatory bowel disease*. PloS One, 7(6), e39242. <doi.org/10.1371/journal.pone.0039242>.

Duvallet, C., Gibbons, S., Gurry, T., Irizarry, R., & Alm, E. (2017). *MicrobiomeHD: the human gut microbiome in health and disease [Data set]*. Zenodo. <doi.org/10.5281/zenodo.1146764>.

`invfishZ` *Inverse Fisher's z transformation*

Description

Inverse Fisher's z transformation

Usage

```
invfishZ(x)
```

Arguments

`x`	data

Value

Inverse Fisher's z transformation of x

Examples

```
x <- .5
invfishZ(x)
```

`invlogit` *Inverse logit transformation*

Description

Inverse logit transformation

Usage

```
invlogit(x)
```

Arguments

`x`	data

Value

Inverse logit transformation of x

Examples

```
x <- .5
invlogit(x)
```

`logit` *Logit transformation*

Description

Logit transformation

Usage

```
logit(x)
```

Arguments

`x`	data

Value

logit of x

Examples

```
x <- .5
logit(x)
```

`lrtest` *Likelihood ratio test*

Description

Likelihood ratio test

Usage

```
lrtest(mod, mod_null)
```

Arguments

`mod`	an object of class `bbdml`
`mod_null`	an object of class `bbdml`, should be nested within `mod`

Value

P-value from likelihood ratio test.

Examples

```
data(soil_phylum_small_otu1)
mod1 <- bbdml(formula = cbind(W, M - W) ~ DayAmdmt,
phi.formula = ~ DayAmdmt,
data = soil_phylum_small_otu1)

mod2 <- bbdml(formula = cbind(W, M - W) ~ 1,
phi.formula = ~ 1,
data = soil_phylum_small_otu1)
lrtest(mod1, mod2)
```

`objfun` *Objective function*

Description

Used for internal optimization. Not intended for users.

Usage

```
objfun(theta, W, M, X, X_star, np, npstar, link, phi.link)
```

Arguments

`theta`	Numeric vector. Parameters associated with `X` and `X_star`
`W`	Numeric vector of counts
`M`	Numeric vector of sequencing depth
`X`	Matrix of covariates associated with abundance (including intercept)
`X_star`	Matrix of covariates associated with dispersion (including intercept)
`np`	Number of covariates associated with abundance (including intercept)
`npstar`	Number of covariates associated with dispersion (including intercept)
`link`	ink function for abundance covariates
`phi.link`	ink function for dispersion covariates

Value

List of negative log-likelihood, gradient, and hessian

`otu_to_taxonomy` *Transform OTUs to their taxonomic label*

Description

Transform OTUs to their taxonomic label

Usage

```
otu_to_taxonomy(OTU, data, level = NULL)
```

Arguments

`OTU`	String vector. Names of OTU labels in `data`
`data`	`phyloseq` object with a taxonomy table
`level`	(Optional). Character vector. Desired taxonomic levels for output.

Value

String vector. Names of taxonomic labels matching labels of `OTU`.

---

pbLRT *Parametric bootstrap likelihood ratio test*

Description

Parametric bootstrap likelihood ratio test

Usage

```
pbLRT(mod, mod_null, B = 1000)
```

Arguments

`mod`	an object of class `bbdml`
`mod_null`	an object of class `bbdml`, should be nested within `mod`
`B`	Integer. Defaults to `1000`. Number of bootstrap iterations.

Value

P-value from parametric bootstrap likelihood ratio test.

Examples

```
data(soil_phylum_small_otu1)
mod1 <- bbdml(formula = cbind(W, M - W) ~ DayAmdmt,
phi.formula = ~ DayAmdmt,
data = soil_phylum_small_otu1)

mod2 <- bbdml(formula = cbind(W, M - W) ~ 1,
phi.formula = ~ 1,
data = soil_phylum_small_otu1)
pbLRT(mod1, mod2, B = 50)
```

---

pbRao *Parametric bootstrap Rao test*

Description

Parametric bootstrap Rao test

Usage

```
pbRao(mod, mod_null, B = 1000)
```

Arguments

`mod`	an object of class `bbdml`
`mod_null`	an object of class `bbdml`, should be nested within `mod`
`B`	Integer. Defaults to `1000`. Number of bootstrap iterations.

Value

P-value from parametric bootstrap Rao test.

Examples

```
data(soil_phylum_small_otu1)
mod1 <- bbdml(formula = cbind(W, M - W) ~ DayAmdmt,
phi.formula = ~ DayAmdmt,
data = soil_phylum_small_otu1)

mod2 <- bbdml(formula = cbind(W, M - W) ~ 1,
phi.formula = ~ 1,
data = soil_phylum_small_otu1)
pbRao(mod1, mod2, B = 10)
```

`pbWald` *Parametric bootstrap Wald test*

Description

Parametric bootstrap Wald test

Usage

```
pbWald(mod, mod_null, B = 1000, robust = FALSE)
```

Arguments

`mod`	an object of class `bbdml`
`mod_null`	an object of class `bbdml`, should be nested within `mod`
`B`	Integer. Defaults to `1000`. Number of bootstrap iterations.
`robust`	Should robust standard errors be used? If not, model-based standard arras are used. Logical, defaults to `FALSE`.

Value

P-value from parametric bootstrap Wald test.

Examples

```
data(soil_phylum_small_otu1)
mod1 <- bbdml(formula = cbind(W, M - W) ~ DayAmdmt,
phi.formula = ~ DayAmdmt,
data = soil_phylum_small_otu1)

mod2 <- bbdml(formula = cbind(W, M - W) ~ 1,
phi.formula = ~ 1,
data = soil_phylum_small_otu1)
pbWald(mod1, mod2, B = 50)
```

`plot.bbdml` *Plotting function*

Description

Plotting function

Usage

```
## S3 method for class 'bbdml'
plot(
  x,
  total = FALSE,
  color = NULL,
  shape = NULL,
  facet = NULL,
  title = NULL,
  B = 1000,
  sample_names = TRUE,
  data_only = FALSE,
  ...
)
```

Arguments

`x`	Object of class `bbdml`.
`total`	(Optional). Default `FALSE`. Boolean indicator for whether to plot on total counts scale
`color`	(Optional). Default `NULL`. The sample variable to map to different colors. Can be a single character string of the variable name in `sample_data` or a custom supplied vector with length equal to the number of samples. Use a character vector to have `ggplot2` default.
`shape`	(Optional). Default `NULL`. The sample variable to map to different shapes. Can be a single character string of the variable name in `sample_data` or a custom supplied vector with length equal to the number of samples.

`facet`	(Optional). Default `NULL`. The sample variable to map to different panels in a facet grid. Must be a single character string of a variable name in `sample_data`.
`title`	(Optional). Default `NULL`. Character string. The main title for the graphic.
`B`	(Optional). Default `1000`. Integer. Number of bootstrap simulations for prediction intervals. Use `B = 0` for no prediction intervals.
`sample_names`	(Optional). Default `TRUE`. Boolean. If `FALSE`, remove sample names from the plot.
`data_only`	(Optional). Default `FALSE`. Boolean. If `TRUE`, only returns data frame.
`...`	There are no optional parameters at this time.

Value

Object of class `ggplot`. Plot of `bbdml` model fit with 95

Examples

```
data(soil_phylum_small_otu1)
mod <- bbdml(formula = cbind(W, M - W) ~ DayAmdmt,
phi.formula = ~ DayAmdmt,
data = soil_phylum_small_otu1)
# Here we use B = 50 for quick demonstration purposes.
# In practice, we recommend a higher value for B for more accurate intervals
plot(mod, color = "DayAmdmt", B = 50)
```

`plot.differentialTest` *differentialTest plot function*

Description

differentialTest plot function

Usage

```
## S3 method for class 'differentialTest'
plot(x, level = NULL, data_only = FALSE, ...)
```

Arguments

`x`	Object of class `differentialTest`
`level`	(Optional). Character vector. Desired taxonomic levels for taxa labels.
`data_only`	(Optional). Default `FALSE`. Boolean. If `TRUE`, only returns data frame.
`...`	No optional arguments are accepted at this time.

Value

Object of class `ggplot`. Plot of coefficients from models for significant taxa from `differentialTest`

Examples

```
# phyloseq example
data(soil_phylum_small_sample)
data(soil_phylum_small_otu)
da_analysis <- differentialTest(formula = ~ DayAmdmt,
                                phi.formula = ~ DayAmdmt,
                                formula_null = ~ 1,
                                phi.formula_null = ~ DayAmdmt,
                                test = "Wald", boot = FALSE,
                                data = soil_phylum_small_otu,
                                sample_data = soil_phylum_small_sample,
                                fdr_cutoff = 0.05,
                                try_only = 1:5)
plot(da_analysis, level = "Phylum")
```

`print.bbdml` *Print function*

Description

Print function

Usage

```
## S3 method for class 'bbdml'
print(
  x,
  digits = max(3L, getOption("digits") - 3L),
  signif.stars = getOption("show.signif.stars"),
  ...
)
```

Arguments

`x`	Object of class `bbdml`
`digits`	the number of significant digits to use when printing.
`signif.stars`	logical. If `TRUE`, ‘significance stars’ are printed for each coefficient.
`...`	No optional arguments are accepted at this time.

Value

`NULL`. Displays printed model summary.

Examples

```
data(soil_phylum_small_otu1)
mod <- bbdml(formula = cbind(W, M - W) ~ DayAmdmt,
phi.formula = ~ DayAmdmt,
data = soil_phylum_small_otu1)
print(mod)
```

`print.differentialTest` *differentialTest print function*

Description

differentialTest print function

Usage

```
## S3 method for class 'differentialTest'
print(x, ...)
```

Arguments

`x`	Object of class `bbdml`
`...`	No optional arguments are accepted at this time.

Value

`NULL`. Displays printed `differentialTest` summary.

Examples

```
# phyloseq example
data(soil_phylum_small_sample)
data(soil_phylum_small_otu)
da_analysis <- differentialTest(formula = ~ DayAmdmt,
                                phi.formula = ~ DayAmdmt,
                                formula_null = ~ 1,
                                phi.formula_null = ~ DayAmdmt,
                                test = "Wald", boot = FALSE,
                                data = soil_phylum_small_otu,
                                sample_data = soil_phylum_small_sample,
                                fdr_cutoff = 0.05,
                                try_only = 1:5)
print(da_analysis)
```

`print.summary.bbdml` *Print summary function*

Description

Print summary function

Usage

```
## S3 method for class 'summary.bbdml'
print(
  x,
  digits = max(3L, getOption("digits") - 3L),
  signif.stars = getOption("show.signif.stars"),
  ...
)
```

Arguments

`x`	Object of class `bbdml`
`digits`	the number of significant digits to use when printing.
`signif.stars`	logical. If `TRUE`, ‘significance stars’ are printed for each coefficient.
`...`	No optional arguments are accepted at this time.

Value

`NULL`. Displays printed model summary.

Examples

```
data(soil_phylum_small_otu1)
mod <- bbdml(formula = cbind(W, M - W) ~ DayAmdmt,
phi.formula = ~ DayAmdmt,
data = soil_phylum_small_otu1)
print(summary(mod))
```

`qbetabinom` *Get quantiles of beta binom*

Description

Get quantiles of beta binom

Usage

```
qbetabinom(p, M, mu, phi)
```

Arguments

`p`	Numeric. Probability for quantile
`M`	Numeric vector of sequencing depth
`mu`	Numeric vector of abundance parameter
`phi`	Numeric vector of dispersion parameter

Value

quantile

Examples

```
data(soil_phylum_small_otu1)
mod <- bbdml(formula = cbind(W, M - W) ~ DayAmdmt,
phi.formula = ~ DayAmdmt,
data = soil_phylum_small_otu1)
qbetabinom(.5, M = mod$M[1], mu = mod$mu.resp[1], phi = mod$phi.resp[1])
```

`raotest` *Rao-type chi-squared test (model-based or robust)*

Description

Rao-type chi-squared test (model-based or robust)

Usage

```
raotest(mod, mod_null)
```

Arguments

`mod`	an object of class `bbdml`
`mod_null`	an object of class `bbdml`, should be nested within `mod`

Value

P-value from likelihood ratio test.

Examples

```
data(soil_phylum_small_otu1)
mod1 <- bbdml(formula = cbind(W, M - W) ~ DayAmdmt,
phi.formula = ~ DayAmdmt,
data = soil_phylum_small_otu1)

mod2 <- bbdml(formula = cbind(W, M - W) ~ 1,
phi.formula = ~ 1,
data = soil_phylum_small_otu1)
raotest(mod1, mod2)
```

`sandSE` *Compute sandwich standard errors. Legacy function. Use sand_vcov instead.*

Description

Compute sandwich standard errors. Legacy function. Use sand_vcov instead.

Usage

```
sandSE(mod, numerical = FALSE)
```

Arguments

`mod`	an object of class `bbdml`
`numerical`	Boolean. Defaults to `FALSE`. Indicator of whether to use the numeric Hessian and score (not recommended).

Value

Sandwich variance-covariance matrix

Examples

```
data(soil_phylum_small_otu1)
mod <- bbdml(formula = cbind(W, M - W) ~ DayAmdmt,
phi.formula = ~ DayAmdmt,
data = soil_phylum_small_otu1)
sandSE(mod)
```

`sand_vcov` *Compute sandwich estimate of variance-covariance matrix*

Description

Compute sandwich estimate of variance-covariance matrix

Usage

```
sand_vcov(mod, numerical = FALSE)
```

Arguments

`mod`	an object of class `bbdml`
`numerical`	Boolean. Defaults to `FALSE`. Indicator of whether to use the numeric Hessian and score (not recommended).

Value

Sandwich variance-covariance matrix. $\hat{A}^{-1}\hat{B}\hat{A}^{-1}$.

Examples

```
data(soil_phylum_small_otu1)
mod <- bbdml(formula = cbind(W, M - W) ~ DayAmdmt,
phi.formula = ~ DayAmdmt,
data = soil_phylum_small_otu1)
sand_vcov(mod)
```

`score` *Compute score at the MLE*

Description

Compute score at the MLE

Usage

```
score(mod, numerical = FALSE, get_score_covariance = FALSE)
```

Arguments

`mod`	an object of class `bbdml`
`numerical`	Boolean. Defaults to `FALSE`. Indicator of whether to use the numeric Hessian and score (not recommended).
`get_score_covariance`	Boolean. Defaults to `FALSE`. Should we return a robust estimate of variance of score: $\hat{B}(\hat{\theta}) = \sum_i G(\hat{\theta}; W_i)G(\hat{\theta}; W_i)^T$. This parameter is not intended for users.

Value

Score at the MLE. For $G(\theta, w)$ score function, returns $\sum_i G(\hat{\theta}, W_i)$ if get_score_covariance = FALSE.

Examples

```
data(soil_phylum_small_otu1)
mod <- bbdml(formula = cbind(W, M - W) ~ DayAmdmt,
phi.formula = ~ DayAmdmt,
data = soil_phylum_small_otu1)
score(mod)
```

`simulate.bbdml` *Simulate from beta-binomial model*

Description

Simulate from beta-binomial model

Usage

```
## S3 method for class 'bbdml'
simulate(object, nsim, seed = NULL, ...)
```

Arguments

`object`	an object of class `bbdml`
`nsim`	Integer. Number of simulations
`seed`	Optional integer to set a random seed
`...`	There are no additional parameters at this time.

Value

`nsim` simulations from `object`

`soil_phylo_otu` *Soil data, otu table as data frame.*

Description

A data frame made from a soil ‘phyloseq‘ object with only otu count data.

Usage

```
soil_phylo_otu
```

Format

A phyloseq-class experiment-level object with an OTU table.

otu_table OTU table with 7,770 taxa and 119 samples

References

Whitman, T., Pepe-Ranney, C., Enders, A., Koechli, C., Campbell, A., Buckley, D. H., Lehmann, J. (2016). *Dynamics of microbial community composi-tion and soil organic carbon mineralization in soil following addition of pyrogenic andfresh organic matter*. The ISME journal, 10(12):2918. <doi: 10.1038/ismej.2016.68>.

`soil_phylo_sample` *Soil data, sample data.*

Description

A data frame made from a soil ‘phyloseq‘ object with only sample data.

Usage

```
soil_phylo_sample
```

Format

A phyloseq-class experiment-level object with sample data.

sam_data sample data with the following covariates:

- `Plants`, values `0` and `1`. Index for different plants
- `Day`, values `0` (initial sampling point), `1` (12 days after treatment additions), and `2` (82 days after treatment additions). Index for different days of measurement
- `Amdmt`, values `0` (no additions), `1` (biochar additions), and `2` (fresh biomass additions). Index for different soil additives.
- `DayAmdmt`, values `00`, `01`, `02`, `10`, `11`, `12`, `20`, `21`, and `22`. A single index for the combination of `Day` and `Amdmt` with `Day` as the first digit and `Amdmt` as the second digit.
- `ID`, values `A`, `B`, `C`, `D`, and `F`. Index for different soil plots.

References

Whitman, T., Pepe-Ranney, C., Enders, A., Koechli, C., Campbell, A., Buckley, D. H., Lehmann, J. (2016). *Dynamics of microbial community composi-tion and soil organic carbon mineralization in soil following addition of pyrogenic andfresh organic matter*. The ISME journal, 10(12):2918. <doi: 10.1038/ismej.2016.68>.

`soil_phylo_taxa` *Soil data, taxa table as data frame.*

Description

A data frame made from a soil ‘phyloseq‘ object with only taxonomy data.

Usage

```
soil_phylo_taxa
```

Format

A phyloseq-class experiment-level object with an OTU table.

tax_table taxonomy table

References

Whitman, T., Pepe-Ranney, C., Enders, A., Koechli, C., Campbell, A., Buckley, D. H., Lehmann, J. (2016). *Dynamics of microbial community composi-tion and soil organic carbon mineralization in soil following addition of pyrogenic andfresh organic matter*. The ISME journal, 10(12):2918. <doi: 10.1038/ismej.2016.68>.

`soil_phylum_contrasts_otu` *Small soil phylum data for contrasts examples, otu table as data frame*

Description

A small subset of `soil_phylo_otu` used for examples of testing contrasts. A data frame made from the ‘phyloseq‘ object with only otu counts.

Usage

```
soil_phylum_contrasts_otu
```

Format

A phyloseq-class experiment-level object with an OTU table.

otu_table OTU table with 39 taxa and 56 samples

References

Whitman, T., Pepe-Ranney, C., Enders, A., Koechli, C., Campbell, A., Buckley, D. H., Lehmann, J. (2016). *Dynamics of microbial community composi-tion and soil organic carbon mineralization in soil following addition of pyrogenic andfresh organic matter*. The ISME journal, 10(12):2918. <doi: 10.1038/ismej.2016.68>.

`soil_phylum_contrasts_sample` *Small soil phylum data for contrasts examples, sample data as data frame*

Description

A small subset of `soil_phylo_sample` used for examples of testing contrasts. A data frame made from the ‘phyloseq‘ object with only sample data.

Usage

```
soil_phylum_contrasts_sample
```

Format

A phyloseq-class experiment-level object with sample data.

sam_data sample data with the following covariates:

- `Plants`, values `0` and `1`. Index for different plants
- `Day`, values `0` (initial sampling point), `1` (12 days after treatment additions), and `2` (82 days after treatment additions). Index for different days of measurement
- `Amdmt`, values `0` (no additions), `1` (biochar additions), and `2` (fresh biomass additions). Index for different soil additives.
- `DayAmdmt`, values `00`, `01`, `02`, `10`, `11`, `12`, `20`, `21`, and `22`. A single index for the combination of `Day` and `Amdmt` with `Day` as the first digit and `Amdmt` as the second digit.
- `ID`, values `A`, `B`, `C`, `D`, and `F`. Index for different soil plots.

References

Whitman, T., Pepe-Ranney, C., Enders, A., Koechli, C., Campbell, A., Buckley, D. H., Lehmann, J. (2016). *Dynamics of microbial community composi-tion and soil organic carbon mineralization in soil following addition of pyrogenic andfresh organic matter*. The ISME journal, 10(12):2918. <doi: 10.1038/ismej.2016.68>.

`soil_phylum_small_otu` *Small soil phylum data for examples, otu table as data frame*

Description

A small subset of `soil_phylo_otu` used for examples. A data frame made from the ‘phyloseq‘ object with only otu counts.

Usage

```
soil_phylum_small_otu
```

Format

A phyloseq-class experiment-level object with an OTU table.

otu_table OTU table with 39 taxa and 32 samples

References

Whitman, T., Pepe-Ranney, C., Enders, A., Koechli, C., Campbell, A., Buckley, D. H., Lehmann, J. (2016). *Dynamics of microbial community composi-tion and soil organic carbon mineralization in soil following addition of pyrogenic andfresh organic matter*. The ISME journal, 10(12):2918. <doi: 10.1038/ismej.2016.68>.

`soil_phylum_small_otu1` *Small soil phylum data for examples, sample data as data frame combined with counts for OTU 1 and sequencing depth.*

Description

A small subset of `soil_phylo_sample` used for examples. A data frame made from the 'phyloseq' object with only sample data and counts for OTU 1.

Usage

```
soil_phylum_small_otu1
```

Format

A phyloseq-class experiment-level object with sample data and OTU 1 counts.

sam_data sample data with the following covariates:

- `Plants`, values `0` and `1`. Index for different plants
- `Day`, values `0` (initial sampling point), `1` (12 days after treatment additions), and `2` (82 days after treatment additions). Index for different days of measurement
- `Amdmt`, values `0` (no additions), `1` (biochar additions), and `2` (fresh biomass additions). Index for different soil additives.
- `DayAmdmt`, values `00`, `01`, `02`, `10`, `11`, `12`, `20`, `21`, and `22`. A single index for the combination of `Day` and `Amdmt` with `Day` as the first digit and `Amdmt` as the second digit.
- `ID`, values `A`, `B`, `C`, `D`, and `F`. Index for different soil plots.
- `W`, counts for OTU1 in each sample. This OTU corresponds with the phylum *Proteobacteria*.
- `M`, the sequencing depth for each sample.

References

Whitman, T., Pepe-Ranney, C., Enders, A., Koechli, C., Campbell, A., Buckley, D. H., Lehmann, J. (2016). *Dynamics of microbial community composi-tion and soil organic carbon mineralization in soil following addition of pyrogenic andfresh organic matter*. The ISME journal, 10(12):2918. <doi: 10.1038/ismej.2016.68>.

`soil_phylum_small_sample`

Small soil phylum data for examples, sample data as data frame

Description

A small subset of `soil_phylo_sample` used for examples. A data frame made from the ‘phyloseq‘ object with only sample data.

Usage

```
soil_phylum_small_sample
```

Format

A phyloseq-class experiment-level object with sample data.

sam_data sample data with the following covariates:

- `Plants`, values `0` and `1`. Index for different plants
- `Day`, values `0` (initial sampling point), `1` (12 days after treatment additions), and `2` (82 days after treatment additions). Index for different days of measurement
- `Amdmt`, values `0` (no additions), `1` (biochar additions), and `2` (fresh biomass additions). Index for different soil additives.
- `DayAmdmt`, values `00`, `01`, `02`, `10`, `11`, `12`, `20`, `21`, and `22`. A single index for the combination of `Day` and `Amdmt` with `Day` as the first digit and `Amdmt` as the second digit.
- `ID`, values `A`, `B`, `C`, `D`, and `F`. Index for different soil plots.

References

Whitman, T., Pepe-Ranney, C., Enders, A., Koechli, C., Campbell, A., Buckley, D. H., Lehmann, J. (2016). *Dynamics of microbial community composi-tion and soil organic carbon mineralization in soil following addition of pyrogenic andfresh organic matter*. The ISME journal, 10(12):2918. <doi: 10.1038/ismej.2016.68>.

`summary.bbdml` *Summary function*

Description

Summary function

Usage

```
## S3 method for class 'bbdml'
summary(object, ...)
```

Arguments

`object`	Object of class `bbdml`
`...`	No optional arguments are accepted at this time.

Value

Object of class `summary.bbdml`. Displays printed model summary.

Examples

```
data(soil_phylum_small_otu1)
mod <- bbdml(formula = cbind(W, M - W) ~ DayAmdmt,
phi.formula = ~ DayAmdmt,
data = soil_phylum_small_otu1)
summary(mod)
```

`waldchisq` *Wald-type chi-squared test*

Description

Wald-type chi-squared test

Usage

```
waldchisq(
  mod,
  mod_null = NULL,
  restrictions = NULL,
  restrictions.phi = NULL,
  contrasts_DA = NULL,
  contrasts_DV = NULL,
  robust = FALSE
)
```

Arguments

`mod`	an object of class `bbdml`
`mod_null`	Optional. An object of class `bbdml`, should be nested within `mod`. If not included, need to include `restrictions` or `restrictions.phi`.
`restrictions`	Optional. Defaults to `NULL`. Numeric vector indicating the parameters associated with the abundance to test, or character vector with name of variable to test. Note that `1` is the intercept associated with the abundance.
`restrictions.phi`	Optional. Defaults to `NULL`. Numeric vector indicating the parameters associated with the dispersion to test, or character vector with name of variable to test. Note that `1` is the intercept associated with the dispersion.
`contrasts_DA`	List. Optional. Constructs a contrast matrix. List elements should be characters specifying contrasts in the parameters within `formula`. Note that this is only available with `"Wald"` value for `test`.
`contrasts_DV`	List. Optional. Constructs a contrast matrix. List elements should be characters specifying contrasts in the parameters within `phi.formula`. Note that this is only available with `"Wald"` value for `test`.
`robust`	Should robust standard errors be used? If not, model-based standard arras are used. Logical, defaults to `FALSE`.

Value

Matrix with wald test statistics and p-values. Only performs univariate tests.

P-value from Wald test.

Examples

```
data(soil_phylum_small_otu1)
mod1 <- bbdml(formula = cbind(W, M - W) ~ DayAmdmt,
phi.formula = ~ DayAmdmt,
data = soil_phylum_small_otu1)

mod2 <- bbdml(formula = cbind(W, M - W) ~ 1,
phi.formula = ~ 1,
data = soil_phylum_small_otu1)

# Example using mod_null
waldchisq(mod = mod1, mod_null = mod2)
waldchisq(mod = mod1, mod_null = mod2, robust = TRUE)

# Example using restrictions and restrictions.phi
waldchisq(mod = mod1, restrictions = 2, restrictions.phi = 2)
waldchisq(mod = mod1, restrictions = "DayAmdmt", restrictions.phi = "DayAmdmt")
waldchisq(mod = mod1, restrictions = 2, restrictions.phi = "DayAmdmt")
waldchisq(mod = mod1, restrictions = 2, restrictions.phi = 2, robust = TRUE)
```

`waldchisq_test` *Wald-type chi-squared test statistic (model-based or robust)*

Description

This is a helper function and not intended for users

Usage

```
waldchisq_test(
  mod,
  restrictions = NULL,
  restrictions.phi = NULL,
  contrasts_DA = NULL,
  contrasts_DV = NULL,
  robust = FALSE
)
```

Arguments

`mod`	an object of class `bbdml`
`restrictions`	Optional. Defaults to `NULL`. Numeric vector indicating the parameters associated with the abundance to test, or character vector with name of variable to test. Note that `1` is the intercept associated with the abundance.
`restrictions.phi`	Optional. Defaults to `NULL`. Numeric vector indicating the parameters associated with the dispersion to test, or character vector with name of variable to test. Note that `1` is the intercept associated with the dispersion.
`contrasts_DA`	List. Optional. Constructs a contrast matrix. List elements should be characters specifying contrasts in the parameters within `formula`. Note that this is only available with `"Wald"` value for `test`.
`contrasts_DV`	List. Optional. Constructs a contrast matrix. List elements should be characters specifying contrasts in the parameters within `phi.formula`. Note that this is only available with `"Wald"` value for `test`.
`robust`	Should robust standard errors be used? If not, model-based standard arras are used. Logical, defaults to `FALSE`.

Value

Test statistic for Wald test.

`waldt` *Wald-type t test (model-based or robust)*

Description

Wald-type t test (model-based or robust)

Usage

```
waldt(mod)
```

Arguments

`mod`	an object of class `bbdml`

Value

Matrix with wald test statistics and p-values. Only performs univariate tests.

Examples

```
data(soil_phylum_small_otu1)
mod <- bbdml(formula = cbind(W, M - W) ~ DayAmdmt,
phi.formula = ~ DayAmdmt,
data = soil_phylum_small_otu1)
waldt(mod)
```

`warn_phyloseq` *Function to throw error if the ‘phyloseq‘ package is called but it is not installed*

Description

Function to throw error if the ‘phyloseq‘ package is called but it is not installed

Usage

```
warn_phyloseq()
```

Index