

# Package: breakaway (via r-universe)

October 26, 2024

**Title** Species Richness Estimation and Modeling

**Version** 4.8.4

**Date** 2022-11-17

**Description** Understanding the drivers of microbial diversity is an important frontier of microbial ecology, and investigating the diversity of samples from microbial ecosystems is a common step in any microbiome analysis. 'breakaway' is the premier package for statistical analysis of microbial diversity. 'breakaway' implements the latest and greatest estimates of species richness, described in Willis and Bunge (2015) <doi:10.1111/biom.12332>, Willis et al. (2017) <doi:10.1111/rssc.12206>, and Willis (2016) <arXiv:1604.02598>, as well as the most commonly used estimates, including the objective Bayes approach described in Barger and Bunge (2010) <doi:10.1214/10-BA527>.

**License** GPL-2

**BugReports** <https://github.com/adw96/breakaway/issues>

**LazyData** true

**RoxygenNote** 7.2.3

**Depends** R (>= 3.5.0)

**Imports** ggplot2, graphics, lme4, magrittr, MASS, phyloseq, stats, tibble, utils

**Suggests** devtools, knitr, plyr, RCurl, rmarkdown, testthat, dplyr

**VignetteBuilder** knitr

**URL** <https://adw96.github.io/breakaway/>

**Encoding** UTF-8

**Repository** <https://adw96.r-universe.dev>

**RemoteUrl** <https://github.com/adw96/breakaway>

**RemoteRef** HEAD

**RemoteSha** 6d3c912963b1eb4feb23f469e147f6e95c67dbe2

## Contents

alpha_estimate . . . . .	3
alpha_estimates . . . . .	4
apples . . . . .	5
beta . . . . .	5
beta_lincom . . . . .	8
beta_pic . . . . .	10
beta_random . . . . .	11
breakaway . . . . .	13
breakaway_nof1 . . . . .	14
build_frequency_count_tables . . . . .	16
chao1 . . . . .	16
chao1_bc . . . . .	17
chao_bunge . . . . .	18
chao_shen . . . . .	19
check_format . . . . .	19
convert . . . . .	20
cutoff_wrap . . . . .	20
F_test . . . . .	21
get_F_stat . . . . .	22
good_turing . . . . .	23
hawaii . . . . .	23
kemp . . . . .	24
make_design_matrix . . . . .	25
make_frequency_count_table . . . . .	26
objective_bayes_geometric . . . . .	26
objective_bayes_mixedgeo . . . . .	28
objective_bayes_negbin . . . . .	29
objective_bayes_poisson . . . . .	31
pasolli_et_al . . . . .	33
physeq_wrap . . . . .	33
plot.alpha_estimates . . . . .	34
poisson_model . . . . .	35
poisson_model_nof1 . . . . .	35
proportions_instead . . . . .	36
rnbinomtable . . . . .	36
rztnbinomtable . . . . .	37
sample_inverse_simpson . . . . .	37
sample_richness . . . . .	38
sample_shannon . . . . .	38
sample_shannon_e . . . . .	39
sample_simpson . . . . .	39
sample_size_estimate . . . . .	40
sample_size_figure . . . . .	41
simulate_beta . . . . .	41
simulate_beta_random . . . . .	42
soil_phylo . . . . .	42

test_submodel . . . . .	43
toy_metadata . . . . .	45
toy_otu_table . . . . .	45
toy_taxonomy . . . . .	46
true_gini . . . . .	46
true_hill . . . . .	47
true_inverse_simpson . . . . .	47
true_shannon . . . . .	48
true_shannon_e . . . . .	48
true_simpson . . . . .	49
wlrm_transformed . . . . .	49
wlrm_untransformed . . . . .	50

## Index 52

---

alpha_estimate	<i>alpha_estimate</i>
----------------	-----------------------

---

### Description

Build objects of class `alpha_estimate` from their components. `alpha_estimate()` is a constructor method

### Usage

```
alpha_estimate(
  estimate = NULL,
  error = NULL,
  estimand = NULL,
  name = NULL,
  interval = NULL,
  interval_type = NULL,
  type = NULL,
  model = NULL,
  warnings = NULL,
  frequentist = NULL,
  parametric = NULL,
  plot = NULL,
  reasonable = NULL,
  other = NULL,
  ...
)
```

### Arguments

<code>estimate</code>	The estimate
<code>error</code>	The standard error in the estimate
<code>estimand</code>	What is the estimate trying to estimate? (richness, Shannon...)

name	The name of the method
interval	An interval estimate
interval_type	Type of interval estimate
type	TODO(Amy): Deprecate?
model	What model is fit
warnings	Any warnings?
frequentist	Logical. Frequentist or Bayesian?
parametric	Logical. Parametric or not?
plot	A ggplot associated with the estimate
reasonable	Is the estimate likely to be reasonable?
other	Any other relevant objects
...	Any other objects

**Value**

An object of class alpha\_estimate

---

alpha_estimates	<i>alpha_estimates</i>
-----------------	------------------------

---

**Description**

Build objects of class alpha\_estimates from their components. alpha\_estimates() is a constructor method

**Usage**

```
alpha_estimates(...)
```

**Arguments**

... Objects of class alpha\_estimate, or a list of objects of class alpha\_estimate

**Value**

An object of class alpha\_estimates

---

apples

(Data) Frequency count table of soil microbes in an apples orchard.

---

### Description

(Data) Frequency count table of soil microbes in an apples orchard.

### Usage

apples

### Format

A data frame with 88 rows and 2 variables:

**index** an index variable

**frequency** number of taxa that were observed with this frequency ...

### References

Willis, A. and Bunge, J. (2015). Estimating diversity via frequency ratios. *Biometrics*, **71**(4), 1042–1049. doi:[10.1111/biom.12332](https://doi.org/10.1111/biom.12332)

Walsh, F. et al. (2014). (2014). Restricted streptomycin use in apple orchards did not adversely alter the soil bacteria communities. *Frontiers in Microbiology* **4**, 383.

---

betta

*Modelling total diversity with betta*

---

### Description

This function tests for heterogeneity of total diversity (observed plus unobserved) across multiple sites. It can account or test for fixed effects that may explain diversity. It returns the significance of the covariates in explaining diversity and a hypothesis test for heterogeneity.

### Usage

```
betta(  
  chats = NULL,  
  ses,  
  X = NULL,  
  initial_est = NULL,  
  formula = NULL,  
  data = NULL,  
  p.digits = 3  
)
```

**Arguments**

chats	A vector of estimates of total diversity at different sampling locations. ‘breakaway’ estimates are suggested in the high-diversity case but not enforced.
ses	The standard errors in chats, the diversity estimates. This can either be a vector of standard errors (with the arguments chats and X), or the name of the variable in the dataframe data that contains the standard errors (with the arguments formula and data).
X	A numeric matrix of covariates. If not supplied, an intercept-only model will be fit. This is optional with the chats argument.
initial_est	(Optional) A vector of length $1 + \text{ncol}(X)$ giving the starting values for the likelihood maximisation search. The first element is the starting estimate for $\sigma^2_u$ , and the remaining elements are the starting elements for beta. Defaults to NULL, in which case the starting values outlined in the paper are used.
formula	A formula object of the form $y \sim x   group$ . Required with the data argument.
data	A dataframe containing the response, response standard errors, covariates, and grouping variable. Required with the formula argument.
p.digits	(Optional) A number that specifies the number of digits to which p-values will be rounded. The default value is 3 digits.

**Value**

table	A coefficient table for the model parameters. The columns give the parameter estimates, standard errors, and p-values, respectively. This model is only as effective as your diversity estimation procedure; for this reason please confirm that your estimates are appropriate and that your model is not misspecified. beta_pic may be useful for this purpose.
cov	Estimated covariance matrix of the parameter estimates.
ssq_u	The estimate of the heterogeneity variance.
homogeneity	The test statistic and p-value for the test of homogeneity.
global	The test statistic and p-value for the test of model explanatory power.
blups	The conditional expected values of the diversity estimates (conditional on the random effects). The authors propose that if the practitioner believes that information from one diversity estimator may inform the others, then using the ‘condfits’ as estimators of total diversity rather than ‘Chats’ may reduce variance of diversity estimates by “sharing strength” across the samples.
blupses	The estimated standard deviation (standard errors) in the blups.
loglikelihood	The log likelihood of the fitted model.
aic	The Akaike information criterion for the fitted model.
aicc	The finite sample correction of the Akaike information criterion for the fitted model.
r_squared_wls	The weighted $R^2$ statistic, appropriate for heteroskedastic linear models.
function.args	A list containing values initially passed to beta_random.

**Note**

Ecologists who are interested in the way species richness varies with covariate information often run a regression-type analysis on the observed diversity using their covariate information as predictors. However, in many settings (especially microbial), rare and unobserved taxa play a hugely important role in explaining the subtleties of the ecosystem, however, a regression analysis on the observed diversity level fails to account for these unobserved taxa. By predicting the total level of diversity (for example, via [breakaway](#)) and estimating the standard error in the estimate, one can take account of these unobserved, but important, taxa. In order to account for the estimated nature of the response, a mixed model approach is taken, whereby the varying levels of confidence in the estimates contributes to a diagonal but heteroscedastic covariance matrix. Given covariates constitute the fixed effects in the mixed model, and significance of the random effect term “`sigsq_u`” reflects heterogeneity in the sample, that is, variability that cannot be explained by only the covariates. The authors believe this to be the first attempt at modelling total diversity in a way that accounts for its estimated nature.

**Author(s)**

Amy Willis

**References**

- Willis, A., Bunge, J., and Whitman, T. (2015). Inference for changes in biodiversity. *arXiv preprint*.  
Willis, A. and Bunge, J. (2015). Estimating diversity via frequency ratios. *Biometrics*.

**See Also**

[breakaway](#); [breakaway\\_nof1](#); [apples](#)

**Examples**

```
df <- data.frame(chats = c(2000, 3000, 4000, 3000), ses = c(100, 200, 150, 180),  
                Cont_var = c(100, 150, 100, 50))  
  
# formula notation
```

```
beta(formula = chats ~ Cont_var, ses = ses, data = df)

# direct input
beta(c(2000, 3000, 4000, 3000), c(100, 200, 150, 180), cbind(1, c(100, 150, 100,
  50)))

## handles missing data
beta(c(2000, 3000, 4000, 3000), c(100, 200, 150, NA))

## A test for heterogeneity of apples diversity estimates vs butterfly estimates
beta(c(1552, 1500, 884), c(305, 675, 205), cbind(1, c(0, 0, 1)))
```

---

beta_lincom	<i>Confidence intervals and testing for linear combinations of fixed effects estimated via beta() or beta_random()</i>
-------------	------------------------------------------------------------------------------------------------------------------------

---

### Description

This function provides point estimates, standard errors, and equal-tailed confidence intervals for linear combinations of fixed effects estimated via `beta()` or `beta_random()`. A p-value for a Wald test of the null that the linear combination of effects is equal to zero (against a general alternative) is also returned.

### Usage

```
beta_lincom(fitted_beta, linear_com, signif_cutoff = 0.05)
```



**Arguments**

fitted_beta	A fitted beta object – i.e., the output of either beta() or beta_random() – containing fixed effect estimates of interest.
linear_com	The linear combination of fixed effects for which a point estimate, confidence interval, and hypothesis test are to be produced.
signif_cutoff	The type-I significance threshold for confidence intervals. Defaults to 0.05.

**Value**

table	A table containing a point estimate, standard error, lower and upper confidence bounds, and a p-value for the linear combination of fixed effects specified in input. The p-value is generated via a two-sided Wald test of the null that the linear combination of fixed effects is equal to zero.
-------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Author(s)**

David Clausen

**References**

Willis, A., Bunge, J., and Whitman, T. (2015). Inference for changes in biodiversity. *arXiv preprint*.

**See Also**

[beta](#);

**Examples**

```
# generate example data
df <- data.frame(chats = c(2000, 3000, 4000, 3000), ses = c(100, 200, 150, 180),
                 Cont_var = c(100, 150, 100, 50))

# fit beta()
example_fit <- beta(formula = chats ~ Cont_var, ses = ses, data = df)

# generate point estimate and 95% CI for mean richness at Cont_var = 125

beta_lincom(fitted_beta = example_fit,
            linear_com = c(1, 125)) # this tells beta_lincom to estimate value of beta_0 + 125*beta_1,
# where beta_0 is the intercept, and beta_1 is the (true value of the) coefficient on Cont_var
```

---

beta\_pic                      *function for plotting total diversity*

---

### Description

A simple plotting interface for comparing total diversity across samples or a covariate gradient.

### Usage

```
beta_pic(  
  y,  
  se,  
  x = 1:length(y),  
  ylimu = NULL,  
  myy = NULL,  
  mymain = NULL,  
  mycol = NULL,  
  labs = NULL,  
  mypch = NULL,  
  myxlim = NULL  
)
```

### Arguments

y	A vector of estimates of total diversity. Other parameter estimates are accessible; this method may be used for plotting any parameter estimates..
se	The standard errors in 'y', the diversity (or other parameter's) estimates.
x	A vector of covariates to form the x-coordinates of the intervals. If no argument is given, defaults to the order.
ylimu	The upper endpoint of the y-axis.
myy	Deprecated, for backwards compatibility
mymain	Deprecated, for backwards compatibility
mycol	Deprecated, for backwards compatibility
labs	Deprecated, for backwards compatibility
mypch	Deprecated, for backwards compatibility
myxlim	Deprecated, for backwards compatibility

### Value

A ggplot object.

### Author(s)

Amy Willis

**References**

Willis, A., Bunge, J., and Whitman, T. (2015). Inference for changes in biodiversity. arXiv preprint.

**See Also**

[beta](#)

**Examples**

```
beta_pic(c(1552, 1500, 884), c(305, 675, 205), mymain = "Example title")
```

---

beta_random	<i>modelling total diversity with random effects</i>
-------------	------------------------------------------------------

---

**Description**

This function extends `beta()` to permit random effects modelling.

**Usage**

```
beta_random(
  chats = NULL,
  ses,
  X = NULL,
  groups = NULL,
  formula = NULL,
  data = NULL,
  p.digits = 3
)
```

**Arguments**

chats	A vector of estimates of total diversity at different sampling locations. Required with the <code>groups</code> argument, and optionally with the <code>X</code> argument.
ses	The standard errors in <code>chats</code> , the diversity estimates. This can either be a vector of standard errors (with the arguments <code>chats</code> and <code>X</code> ), or the name of the variable in the dataframe <code>data</code> that contains the standard errors (with the arguments <code>formula</code> and <code>data</code> ).
X	A numeric matrix of covariates corresponding to fixed effects. If not supplied, an intercept-only model will be fit. Optional with the <code>chats</code> and <code>groups</code> arguments.
groups	A categorical variable representing the random-effects groups that each of the estimates belong to. Required with the <code>chats</code> argument and optionally with the <code>X</code> argument.
formula	A formula object of the form $y \sim x group$ . Required with the <code>data</code> argument.

data	A dataframe containing the response, response standard errors, covariates, and grouping variable. Required with the formula argument.
p.digits	(Optional) A number that specifies the number of digits to which p-values will be rounded. The default value is 3 digits.

**Value**

table	A coefficient table for the model parameters. The columns give the parameter estimates, standard errors, and p-values, respectively. This model is only as effective as your diversity estimation procedure; for this reason please confirm that your estimates are appropriate and that your model is not misspecified. beta_pic may be useful for this purpose.
cov	Estimated covariance matrix of the parameter estimates.
ssq_u	The estimate of the heterogeneity variance.
ssq_g	Estimates of within-group variance. The estimate will be zero for groups with only one observation.
homogeneity	The test statistic and p-value for the test of homogeneity.
global	The test statistic and p-value for the test of model explanatory power.
blups	The conditional expected values of the diversity estimates (conditional on the random effects). Estimates of variability for the random effects case are unavailable at this time; please contact the maintainer if needed.
loglikelihood	The log likelihood of the fitted model.
aic	The Akaike information criterion for the fitted model.
aicc	The finite sample correction of the Akaike information criterion for the fitted model.
function.args	A list containing values initially passed to beta_random.

**Author(s)**

Amy Willis

**References**

Willis, A., Bunge, J., and Whitman, T. (2015). Inference for changes in biodiversity. *arXiv preprint*.

**See Also**

[beta](#);

**Examples**

```
df <- data.frame(chats = c(2000, 3000, 4000, 3000),
  ses = c(100, 200, 150, 180),
  Cont_var = c(100, 150, 100, 50),
  groups = c("a", "a", "a", "b"))

# formula notation
```

```

beta_random(formula = chats ~ Cont_var | groups,
ses = ses,
data = df)

# direct input
beta_random(c(2000, 3000, 4000, 3000), c(100, 200, 150, 180),
X = cbind(Int = 1, Cont_var = c(100, 150, 100, 50)),
groups = c("a", "a", "a", "b"))

## handles missing data
beta_random(c(2000, 3000, 4000, 3000), c(100, 200, 150, NA),
groups = c("a", NA,
"b", "b"))

```

---

breakaway

*Species richness estimation with breakaway*


---

## Description

breakaway is a wrapper for modern species richness estimation for modern datasets

## Usage

```

breakaway(
  input_data,
  cutoff = NA,
  output = NULL,
  plot = NULL,
  answers = NULL,
  print = NULL,
  ...
)

```

## Arguments

input_data	An input type that can be processed by <code>convert()</code>
cutoff	The maximum frequency count to use for model fitting
output	Deprecated; only for backwards compatibility
plot	Deprecated; only for backwards compatibility
answers	Deprecated; only for backwards compatibility
print	Deprecated; only for backwards compatibility
...	Additional arguments will be ignored; this is for backward compatibility

## Value

An object of class `alpha_estimate`

**Note**

'breakaway' presents an estimator of species richness that is well-suited to the high-diversity/microbial setting. However, many microbial datasets display more diversity than the Kemp-type models can permit. In this case, the log-transformed WLRM diversity estimator of Rocchetti et. al. (2011) is returned. The authors' experience suggests that some datasets that require the log-transformed WLRM contain "false" diversity, that is, diversity attributable to sequencing errors (via an inflated singleton count). The authors encourage judicious use of diversity estimators when the dataset may contain these errors, and recommend the use of [breakaway\\_nof1](#) as an exploratory tool in this case.

**Author(s)**

Amy Willis

**References**

Willis, A. and Bunge, J. (2015). Estimating diversity via frequency ratios. *Biometrics*, **71**(4), 1042–1049.

**See Also**

[breakaway\\_nof1](#); [kemp](#); [apples](#)

**Examples**

```
breakaway(apples)
breakaway(apples, plot = FALSE, output = FALSE, answers = TRUE)
```

---

breakaway_nof1	<i>species richness estimation without singletons</i>
----------------	-------------------------------------------------------

---

**Description**

This function permits estimation of total diversity based on a sample frequency count table. Unlike [breakaway](#), it does not require an input for the number of species observed once, making it an excellent exploratory tool for microbial ecologists who believe that their sample may contain spurious singletons. The underlying estimation procedure is similar to that of [breakaway](#) and is outlined in Willis & Bunge (2014). The diversity estimate, standard error, estimated model coefficients and plot of the fitted model are returned.

**Usage**

```
breakaway_nof1(  
  input_data,  
  output = NULL,  
  plot = NULL,  
  answers = NULL,  
  print = NULL  
)
```

**Arguments**

input_data	An input type that can be processed by <code>convert()</code>
output	Deprecated; only for backwards compatibility
plot	Deprecated; only for backwards compatibility
answers	Deprecated; only for backwards compatibility
print	Deprecated; only for backwards compatibility

**Value**

An object of class `alpha_estimate`

code	A category representing algorithm behaviour. ‘code=1’ indicates no nonlinear models converged and the transformed WLRM diversity estimate of Rocchetti et. al. (2011) is returned. ‘code=2’ indicates that the iteratively reweighted model converged and was returned. ‘code=3’ indicates that iterative reweighting did not converge but a model based on a simplified variance structure was returned (in this case, the variance of the frequency ratios is assumed to be proportional to the denominator frequency index). Please peruse your fitted model before using your diversity estimate.
name	The “name” of the selected model. The first integer represents the numerator polynomial degree and the second integer represents the denominator polynomial degree. See Willis & Bunge (2014) for details.
para	Estimated model parameters and standard errors.
est	The estimate of total (observed plus unobserved) diversity.
seest	The standard error in the diversity estimate.
full	The chosen nonlinear model for frequency ratios.

**Note**

It is common for microbial ecologists to believe that their dataset contains false diversity. This often arises because sequencing errors result in classifying already observed organisms as new organisms. ‘breakaway\_nof1’ was developed as an exploratory tool in this case. Practitioners can run ‘breakaway’ on their dataset including the singletons, and ‘breakaway\_nof1’ on their dataset excluding the singletons, and assess if the estimated levels of diversity are very different. Great disparity may provide evidence of an inflated singleton count, or at the very least, that ‘breakaway’ is especially sensitive to the number of rare species observed. Note that ‘breakaway\_nof1’ may be less stable than ‘breakaway’ due to predicting based on a reduced dataset, and have greater standard errors.

**Author(s)**

Amy Willis

**References**

- Willis, A. (2015). Species richness estimation with high diversity but spurious singletons. *arXiv*.
- Willis, A. and Bunge, J. (2015). Estimating diversity via frequency ratios. *Biometrics*.

**See Also**

[breakaway](#); [apples](#)

**Examples**

```
breakaway_nof1(apples[-1, ])
breakaway_nof1(apples[-1, ], plot = FALSE, output = FALSE, answers = TRUE)
```

---

```
build_frequency_count_tables
```

*Build frequency count tables from an OTU table*

---

**Description**

Build frequency count tables from an OTU table

**Usage**

```
build_frequency_count_tables(the_table)
```

**Arguments**

<code>the_table</code>	An OTU table as a data frame or a matrix. Columns are the samples and rows give the taxa.
------------------------	-------------------------------------------------------------------------------------------

**Value**

A list of frequency count tables corresponding to the columns.

---

```
chao1
```

*Chao1 species richness estimator*

---

**Description**

This function implements the Chao1 richness estimate, which is often mistakenly referred to as an index.

**Usage**

```
chao1(input_data, output = NULL, answers = NULL)
```

**Arguments**

<code>input_data</code>	An input type that can be processed by <code>convert()</code> or a phyloseq object
<code>output</code>	Deprecated; only for backwards compatibility
<code>answers</code>	Deprecated; only for backwards compatibility



**Value**

An object of class `alpha_estimate`, or `alpha_estimates` for `phyloseq` objects

**Note**

The authors of this package strongly discourage the use of this estimator. It is only valid when you wish to assume that every taxa has equal probability of being observed. You don't really think that's possible, do you?

**Author(s)**

Amy Willis

**Examples**

```
chao1(apples)
```

---

chao1\_bc

*Bias-corrected Chao1 species richness estimator*

---

**Description**

This function implements the bias-corrected Chao1 richness estimate.

**Usage**

```
chao1_bc(input_data, output = NULL, answers = NULL)
```

**Arguments**

<code>input_data</code>	An input type that can be processed by <code>convert()</code> or a <code>phyloseq</code> object
<code>output</code>	Deprecated; only for backwards compatibility
<code>answers</code>	Deprecated; only for backwards compatibility

**Value**

An object of class `alpha_estimate`, or `alpha_estimates` for `phyloseq` objects

**Note**

The authors of this package strongly discourage the use of this estimator. It is only valid when you wish to assume that every taxa has equal probability of being observed. You don't really think that's possible, do you?

**Author(s)**

Amy Willis

**Examples**

```
chao1_bc(apples)
```

---

chao\_bunge

*Chao-Bunge species richness estimator*

---

**Description**

This function implements the species richness estimation procedure outlined in Chao & Bunge (2002).

**Usage**

```
chao_bunge(input_data, cutoff = 10, output = NULL, answers = NULL)
```

**Arguments**

input_data	An input type that can be processed by <code>convert()</code> or a phyloseq object
cutoff	The maximum frequency to use in fitting.
output	Deprecated; only for backwards compatibility
answers	Deprecated; only for backwards compatibility

**Value**

An object of class `alpha_estimate`, or `alpha_estimates` for phyloseq objects

**Author(s)**

Amy Willis

**Examples**

```
chao_bunge(apples)
```

---

chao_shen	<i>The Chao-Shen estimate of Shannon diversity</i>
-----------	----------------------------------------------------

---

**Description**

The Chao-Shen estimate of Shannon diversity

**Usage**

```
chao_shen(input_data)
```

**Arguments**

input\_data      An input type that can be processed by `convert()`

**Value**

An object of class `alpha_estimate`

---

check_format	<i>Run some basic checks on a possible frequency count table</i>
--------------	------------------------------------------------------------------

---

**Description**

Run some basic checks on a possible frequency count table

**Usage**

```
check_format(output_data)
```

**Arguments**

output\_data      A matrix to test

**Value**

A checked frequency count table

---

convert	<i>convert between different inputs for alpha-diversity estimates</i>
---------	-----------------------------------------------------------------------

---

**Description**

Inputs slated for development include phyloseq and otu\_table

**Usage**

```
convert(input_data)
```

**Arguments**

input\_data      Supported types include filenames, data frames, matrices, vectors...

**Value**

Frequency count table

---

cutoff_wrap	<i>Find a cut-off for estimates relying on contiguous counts</i>
-------------	------------------------------------------------------------------

---

**Description**

Find a cut-off for estimates relying on contiguous counts

**Usage**

```
cutoff_wrap(my_data, requested = NA)
```

**Arguments**

my\_data          Frequency count table  
requested        The user-requested cutoff

**Value**

Cutoff value

---

F_test	<i>Conduct F test of null hypothesis <math>LB = 0</math> using output from <code>betta()</code> or <code>betta_random()</code></i>
--------	------------------------------------------------------------------------------------------------------------------------------------

---

**Description**

This function performs an F-test of a null hypothesis  $LB = 0$  where B is a vector of p fixed effects returned by `betta()` or `betta_random()` and L is an m x p matrix with linearly independent rows.

**Usage**

```
F_test(fitted_beta, L, method = "bootstrap", nboot = 1000)
```

**Arguments**

fitted_beta	A fitted beta object – i.e., the output of either <code>betta()</code> or <code>betta_random()</code> – containing fixed effect estimates of interest.
L	An m x p matrix defining the null $LB = 0$ . L must have full row rank.
method	A character variable indicating which method should be used to estimate the distribution of the test statistic under the null.
nboot	Number of bootstrap samples to use if <code>method = "bootstrap"</code> . Ignored if <code>method = "asymptotic"</code> .

**Value**

A list containing

pval	The p-value
F_stat	The calculated F statistic
boot_F	A vector of bootstrapped F statistics if bootstrap has been used. Otherwise NULL.

**Author(s)**

David Clausen

**References**

Willis, A., Bunge, J., and Whitman, T. (2015). Inference for changes in biodiversity. *arXiv preprint*.

**See Also**

[betta](#);  
[betta\\_random](#);  
[betta](#);  
[get\\_F\\_stat](#)

**Examples**

```
# generate example data
df <- data.frame(chats = c(2000, 3000, 4000, 3000,
2000, 3000, 4000, 3000), ses = c(100, 200, 150, 180,
100, 200, 150, 180),
                Cont_var = c(100, 150, 100, 50,
100, 150, 100, 50),
                Cont_var_2 = c(50,200,25,125,
50,200,25,125))

# fit betta()
example_fit <- betta(formula = chats ~ Cont_var + Cont_var_2, ses = ses, data = df)

# construct L for hypothesis that B_cont_var = B_cont_var_2 = 0
L <- rbind(c(0,1,0),
           c(0,0,1))

F_test_results <- F_test(example_fit, L, nboot = 100)
```

---

get_F_stat	<i>Calculate F statistic under null hypothesis <math>LB = 0</math> using output from betta() or betta_random()</i>
------------	--------------------------------------------------------------------------------------------------------------------

---

**Description**

This function calculates an F statistic for a test of null hypothesis  $LB = 0$  (against an unrestricted alternative) where  $B$  is a vector of  $p$  fixed effects returned by `betta()` or `betta_random()` and  $L$  is an  $m \times p$  matrix with linearly independent rows.

**Usage**

```
get_F_stat(fitted_betta, L)
```

**Arguments**

`fitted_betta` A fitted betta object – i.e., the output of either `betta()` or `betta_random()` – containing fixed effect estimates of interest.

`L` An  $m \times p$  matrix defining the null  $LB = 0$ .  $L$  must have full row rank.

**Value**

A list containing

`F_stat` The calculated F statistic

---

good_turing	<i>The Good-Turing estimate of species richness</i>
-------------	-----------------------------------------------------

---

**Description**

The Good-Turing estimate of species richness

**Usage**

```
good_turing(input_data)
```

**Arguments**

`input_data` An input type that can be processed by `convert()` or a phyloseq object

**Value**

An object of class `alpha_estimate`, or `alpha_estimates` for phyloseq objects

---

hawaii	<i>(Data) Frequency count table of soil microbes in Hawaii.</i>
--------	-----------------------------------------------------------------

---

**Description**

(Data) Frequency count table of soil microbes in Hawaii.

**Usage**

```
hawaii
```

**Format**

A data frame with 198 rows and 2 variables:

**index** an index variable

**frequency** number of taxa that were observed with this frequency ...

**References**

Willis, A. and Bunge, J. (2015). Estimating diversity via frequency ratios. *Biometrics*, 71(4), 1042–1049. doi:10.1111/biom.12332

---

kemp

*Species richness estimation with Kemp-type models*


---

## Description

This function implements the species richness estimation procedure outlined in Willis & Bunge (2015). The diversity estimate, standard error, estimated model coefficients, model details and plot of the fitted model are returned.

## Usage

```
kemp(
  input_data,
  cutoff = NA,
  output = NULL,
  plot = NULL,
  answers = NULL,
  print = NULL,
  ...
)
```

## Arguments

<code>input_data</code>	An input type that can be processed by <code>convert()</code>
<code>cutoff</code>	The maximum frequency count to use for model fitting
<code>output</code>	Deprecated; only for backwards compatibility
<code>plot</code>	Deprecated; only for backwards compatibility
<code>answers</code>	Deprecated; only for backwards compatibility
<code>print</code>	Deprecated; only for backwards compatibility
<code>...</code>	Additional arguments will be ignored; this is for backward compatibility

## Value

An object of class `alpha_estimate`

<code>code</code>	A category representing algorithm behaviour. ‘code=1’ indicates no nonlinear models converged and the transformed WLRM diversity estimate of Rocchetti et. al. (2011) is returned. ‘code=2’ indicates that the iteratively reweighted model converged and was returned. ‘code=3’ indicates that iterative reweighting did not converge but a model based on a simplified variance structure was returned (in this case, the variance of the frequency ratios is assumed to be proportional to the denominator frequency index). Please peruse your fitted model before using your diversity estimate.
-------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------





**Value**

A matrix object giving the design matrix from the desired variables.

---

make\_frequency\_count\_table

*Draw frequency count subtables from an OTU table*

---

**Description**

Draw frequency count subtables from an OTU table

**Usage**

```
make_frequency_count_table(labels)
```

**Arguments**

labels            A vector of counts of the taxa; i.e. a vector giving the number of times each taxon was observed.

**Value**

A frequency count table.

---

objective\_bayes\_geometric

*Estimate species richness with an objective Bayes method using a geometric model*

---

**Description**

Estimate species richness with an objective Bayes method using a geometric model

**Usage**

```
objective_bayes_geometric(
  data,
  output = TRUE,
  plot = TRUE,
  answers = FALSE,
  tau = 10,
  burn.in = 100,
  iterations = 2500,
  Metropolis.stdev.N = 75,
  Metropolis.start.theta = 1,
  Metropolis.stdev.theta = 0.3
)
```

**Arguments**

data	TODO(Kathryn)(Kathryn)
output	TODO(Kathryn)(Kathryn)
plot	TODO(Kathryn)(Kathryn)
answers	TODO(Kathryn)(Kathryn)
tau	TODO(Kathryn)
burn.in	TODO(Kathryn)
iterations	TODO(Kathryn)
Metropolis.stdev.N	TODO(Kathryn)
Metropolis.start.theta	TODO(Kathryn)
Metropolis.stdev.theta	TODO(Kathryn)

**Value**

A list of results, including

est	the median of estimates of N
,	
ci	a confidence interval for N
,	
mean	the mean of estimates of N
,	
semeanest	the standard error of mean estimates
,	
dic	the DIC of the model
,	
fits	fitted values
, and	
diagnostics	model diagnostics
.	

---

objective\_bayes\_mixedgeo

*Objective Bayes species richness estimate with the mixed-geometric model*

---

## Description

Objective Bayes species richness estimate with the mixed-geometric model

## Usage

```
objective_bayes_mixedgeo(  
  data,  
  output = TRUE,  
  plot = TRUE,  
  answers = FALSE,  
  tau = 10,  
  burn.in = 100,  
  iterations = 2500,  
  Metropolis.stdev.N = 100,  
  Metropolis.start.T1 = 1,  
  Metropolis.stdev.T1 = 2,  
  Metropolis.start.T2 = 3,  
  Metropolis.stdev.T2 = 2,  
  bars = 3  
)
```

## Arguments

data	TODO(Kathryn)
output	TODO(Kathryn)
plot	TODO(Kathryn)
answers	TODO(Kathryn)
tau	TODO(Kathryn)
burn.in	TODO(Kathryn)
iterations	TODO(Kathryn)
Metropolis.stdev.N	TODO(Kathryn)
Metropolis.start.T1	TODO(Kathryn)
Metropolis.stdev.T1	TODO(Kathryn)
Metropolis.start.T2	TODO(Kathryn)

```

Metropolis.stdev.T2
                TODO(Kathryn)
bars            TODO(Kathryn)

```

**Value**

A list of results, including

```

est            the median of estimates of N
,
ci            a confidence interval for N
,
mean          the mean of estimates of N
,
semeanest     the standard error of mean estimates
,
dic           the DIC of the model
,
fits         fitted values
, and
diagnostics   model diagnostics
.

```

---

objective\_bayes\_negbin

*Objective Bayes species richness estimate with the Negative Binomial model*

---

**Description**

Objective Bayes species richness estimate with the Negative Binomial model

**Usage**

```
objective_bayes_negbin(
  data,
  output = TRUE,
  plot = TRUE,
  answers = FALSE,
  tau = 10,
  burn.in = 1000,
  iterations = 5000,
  Metropolis.stdev.N = 100,
  Metropolis.start.T1 = -0.8,
  Metropolis.stdev.T1 = 0.01,
  Metropolis.start.T2 = 0.8,
  Metropolis.stdev.T2 = 0.01,
  bars = 5
)
```

**Arguments**

data	TODO(Kathryn)
output	TODO(Kathryn)
plot	TODO(Kathryn)
answers	TODO(Kathryn)
tau	TODO(Kathryn)
burn.in	TODO(Kathryn)
iterations	TODO(Kathryn)
Metropolis.stdev.N	TODO(Kathryn)
Metropolis.start.T1	TODO(Kathryn)
Metropolis.stdev.T1	TODO(Kathryn)
Metropolis.start.T2	TODO(Kathryn)
Metropolis.stdev.T2	TODO(Kathryn)
bars	TODO(Kathryn)

**Value**

A list of results, including

est	the median of estimates of N
-----	------------------------------

,

ci                    a confidence interval for N  
 ,  
 mean                  the mean of estimates of N  
 ,  
 semeanest            the standard error of mean estimates  
 ,  
 dic                    the DIC of the model  
 ,  
 fits                   fitted values  
 , and  
 diagnostics          model diagnostics  
 .

---

objective\_bayes\_poisson

*Objective Bayes species richness estimate with the Poisson model*

---

### Description

Objective Bayes species richness estimate with the Poisson model

### Usage

```

objective_bayes_poisson(
  data,
  output = TRUE,
  plot = TRUE,
  answers = FALSE,
  tau = 10,
  burn.in = 100,
  iterations = 2500,
  Metropolis.stdev.N = 75,
  Metropolis.start.lambda = 1,
  Metropolis.stdev.lambda = 0.3,
  bars = 5
)

```

**Arguments**

data	TODO(Kathryn)
output	TODO(Kathryn)
plot	TODO(Kathryn)
answers	TODO(Kathryn)
tau	TODO(Kathryn)
burn.in	TODO(Kathryn)
iterations	TODO(Kathryn)
Metropolis.stdev.N	TODO(Kathryn)
Metropolis.start.lambda	TODO(Kathryn)
Metropolis.stdev.lambda	TODO(Kathryn)
bars	TODO(Kathryn)

**Value**

A list of results, including

est	the median of estimates of N
,	
ci	a confidence interval for N
,	
mean	the mean of estimates of N
,	
semeanest	the standard error of mean estimates
,	
dic	the DIC of the model
,	
fits	fitted values
, and	
diagnostics	model diagnostics
.	



---

pasolli\_et\_al                    *(Data) Data frame of covariate information about pasolli\_et\_al.*

---

**Description**

(Data) Data frame of covariate information about pasolli\_et\_al.

**Usage**

```
pasolli_et_al
```

**Format**

A data frame with 4930 rows and 9 variables:

**SGB.ID** sample ID  
**AverageAbundance** average abundance in the sample  
**std** standard deviation  
**q1** first quartile  
**MedianAbundance** median abundance  
**q3** third quartile  
**min** minimum  
**max** maximum  
**#.Samples** number of samples ...

---

physeq\_wrap                    *Wrapper for phyloseq objects*

---

**Description**

Wrapper for phyloseq objects

**Usage**

```
physeq_wrap(fn, physeq, ...)
```

**Arguments**

**fn**                    alpha diversity estimator function with breakaway to be applied to physeq  
**physeq**                A phyloseq object, or an object of class otu\_table  
**...**                    Additional arguments for fn

**Value**

Object of class alpha\_estimates

---

plot.alpha\_estimates *Plot function for alpha\_estimates class*

---

## Description

Plot function for alpha\_estimates class

## Usage

```
## S3 method for class 'alpha_estimates'
plot(
  x,
  physeq = NULL,
  measure = NULL,
  color = NULL,
  shape = NULL,
  title = NULL,
  trim_plot = FALSE,
  ...
)
```

## Arguments

x	Object of class alpha_estimates.
physeq	(Optional). Default NULL. Required object of class phyloseq if including a sample_data variable for color or shape.
measure	(Optional). If there are multiple richness measures included in x, this can be set to the the desired measure to be plotted. Defaults to the measure of the first alpha diversity estimate.
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.
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.
title	(Optional). Default NULL. Character string. The main title for the graphic.
trim_plot	(Optional). Default FALSE. Boolean indicator for whether you want the plot to include the full confidence intervals.
...	See details

## Details

... does not currently have any implemented options. Optional arguments currently include "trim\_plot", a Optional

**Value**

A ggplot object.

**Examples**

```
library(phyloseq)
data(GlobalPatterns)
alphas <- breakaway(GlobalPatterns)
plot(alphas)
```

---

poisson_model	<i>PoissonModel</i>
---------------	---------------------

---

**Description**

A model to estimate the number of missing taxa under a Poisson Model

**Usage**

```
poisson_model(input_data, cutoff = 10)
```

**Arguments**

input_data	A frequency count table
cutoff	The largest frequency to use for predicting f0. Defaults to 10.

**Value**

An object of class alpha\_estimate.

---

poisson_model_nof1	<i>PoissonModelNof1</i>
--------------------	-------------------------

---

**Description**

A model to estimate the number of missing taxa under a zero- and one-truncated Poisson Model

**Usage**

```
poisson_model_nof1(input_data, cutoff = 10)
```

**Arguments**

input_data	A frequency count table
cutoff	The largest frequency to use for predicting f0 and f1. Defaults to 10.

**Value**

An object of class alpha\_estimate.

---

proportions\_instead    *OTU table to relative abundances*

---

**Description**

OTU table to relative abundances

**Usage**

```
proportions_instead(the_table)
```

**Arguments**

the\_table    An OTU table

**Value**

A proportion table or vector.

---

rnbinomtable    *Negative binomially distributed frequency count tables.*

---

**Description**

Simulate a frequency count table based on a negative binomial model. Zero-truncated, obviously.

**Usage**

```
rnbinomtable(C, size, probability)
```

**Arguments**

C                    species richness  
size                 size parameter for the negative binomial distribution  
probability         probability parameter for the negative binomial distribution

**Value**

A simulated frequency count table.

**Author(s)**

Amy Willis

---

rztmbinomtable	<i>beta version: Zero-truncated negative binomially distributed frequency count tables.</i>
----------------	---------------------------------------------------------------------------------------------

---

**Description**

Simulate a frequency count table based on a negative binomial model. Zero-truncated, obviously.

**Usage**

```
rztmbinomtable(C, size, probability)
```

**Arguments**

C	species richness
size	size parameter for the negative binomial distribution
probability	probability parameter for the negative binomial distribution

**Value**

A simulated frequency count table.

**Author(s)**

Amy Willis

---

sample_inverse_simpson	<i>Plug-in Inverse Simpson diversity</i>
------------------------	------------------------------------------

---

**Description**

This function implements the plug-in Inverse Simpson diversity

**Usage**

```
sample_inverse_simpson(input_data)
```

**Arguments**

input_data	An input type that can be processed by <code>convert()</code> or a phyloseq object
------------	------------------------------------------------------------------------------------

**Value**

An object of class `alpha_estimate`, or `alpha_estimates` for phyloseq objects

**Examples**

```
sample_inverse_simpson(apples)
```

---

sample_richness	<i>Sample richness estimator</i>
-----------------	----------------------------------

---

**Description**

This function implements the sample richness estimate, which is the number of non-zero taxa per sample.

**Usage**

```
sample_richness(input_data)
```

**Arguments**

input\_data      An input type that can be processed by `convert()` or a phyloseq object

**Value**

An object of class `alpha_estimate`, or `alpha_estimates` for phyloseq objects

**Examples**

```
sample_richness(apples)
```

---

sample_shannon	<i>Plug-in Shannon diversity</i>
----------------	----------------------------------

---

**Description**

This function implements the plug-in Shannon diversity

**Usage**

```
sample_shannon(input_data)
```

**Arguments**

input\_data      An input type that can be processed by `convert()` or a phyloseq object

**Value**

An object of class alpha\_estimate, or alpha\_estimates for phyloseq objects

**Examples**

```
sample_shannon(apples)
```

---

sample_shannon_e	<i>Plug-in Shannon's E ("Equitability")</i>
------------------	---------------------------------------------

---

**Description**

This function implements the plug-in Shannon's E

**Usage**

```
sample_shannon_e(input_data)
```

**Arguments**

input\_data      An input type that can be processed by convert() or a phyloseq object

**Value**

An object of class alpha\_estimate, or alpha\_estimates for phyloseq objects

**Examples**

```
sample_shannon_e(apples)
```

---

sample_simpson	<i>Plug-in Simpson diversity</i>
----------------	----------------------------------

---

**Description**

This function implements the plug-in Simpson diversity

**Usage**

```
sample_simpson(input_data)
```

**Arguments**

input\_data      An input type that can be processed by convert() or a phyloseq object

**Value**

An object of class alpha\_estimate, or alpha\_estimates for phyloseq objects

**Examples**

```
sample_simpson(apples)
```

---

sample\_size\_estimate    *Estimate the sample size needed to do an unpaired one-way test using beta*

---

**Description**

Estimate the sample size needed to do an unpaired one-way test using beta

**Usage**

```
sample_size_estimate(
  control_group_est,
  se_est,
  diff = 5,
  alpha = 0.05,
  prop = 0.8,
  samples = 20,
  precision = 5
)
```

**Arguments**

control_group_est	An estimate of the alpha diversity parameter for the control group
se_est	An estimate of the (common) standard deviation
diff	An estimate of the difference between the control and non-control groups
alpha	Minimum significance level desired
prop	Desired power
samples	Number of bootstrap resamples used to estimate the sample size. Increase for a more accurate estimate.
precision	How much to increment the sample size as we try to increase the power

**Value**

An estimate of the necessary sample size and some details



---

sample\_size\_figure      *Plot the power obtained with sample size*

---

**Description**

Plot the power obtained with sample size

**Usage**

```
sample_size_figure(control_group_est, se_est, diff = 5, samples = 20)
```

**Arguments**

control_group_est	An estimate of the alpha diversity parameter for the control group
se_est	An estimate of the (common) standard deviation
diff	An estimate of the difference between the control and non-control groups
samples	Number of bootstrap resamples used to estimate the sample size. Increase for a more accurate estimate.

**Value**

A plot of the power with the sample size

---

simulate\_beta      *Simulate from a fitted beta model*

---

**Description**

Simulate from a fitted beta model

**Usage**

```
simulate_beta(fitted_beta, nsim)
```

**Arguments**

fitted_beta	A fitted beta object
nsim	Number of times to simulate

**Value**

A list of length nsim, each element of which is a vector of simulated Y-values under the fitted beta model

---

simulate\_beta\_random *Simulate from a fitted beta\_random model*

---

### Description

Simulate from a fitted beta\_random model

### Usage

```
simulate_beta_random(fitted_beta, nsim)
```

### Arguments

**fitted\_beta** A fitted beta\_random object  
**nsim** Number of times to simulate

### Value

A list of length nsim, each element of which is a vector of simulated Y-values under the fitted beta model

---

soil\_phylo *(Data) Data frame of soil data from whitman\_et\_al.*

---

### Description

A phyloseq object with an OTU table and sample data from a soil microbiome study.

### Usage

```
soil_phylo
```

### Format

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

**otu\_table** OTU table with 7,770 taxa and 119 samples

**tax\_table** taxonomy table

**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 composition and soil organic carbon mineralization in soil following addition of pyrogenic and fresh organic matter*. The ISME journal, 10(12):2918. <doi: 10.1038/ismej.2016.68>.

---

test_submodel	<i>Conduct F test of null hypothesis <math>LB = 0</math> using output from <code>betta()</code> or <code>betta_random()</code></i>
---------------	------------------------------------------------------------------------------------------------------------------------------------

---

## Description

This function performs an F-test of a null hypothesis  $LB = 0$  where B is a vector of p fixed effects returned by `betta()` or `betta_random()` and L is an m x p matrix with linearly independent rows.

## Usage

```
test_submodel(
  fitted_beta,
  submodel_formula,
  method = "bootstrap",
  nboot = 1000
)
```

## Arguments

<code>fitted_beta</code>	A fitted beta object – i.e., the output of either <code>betta()</code> or <code>betta_random()</code> – containing fixed effect estimates of interest.
<code>submodel_formula</code>	A formula defining which submodel to treat as the null. It is not necessary to include random effects in this formula (they will be ignored if included – the submodel will be fit with the same random effect structure as the full model regardless of input.)
<code>method</code>	A character variable indicating which method should be used to estimate the distribution of the test statistic under the null.
<code>nboot</code>	Number of bootstrap samples to use if <code>method = "bootstrap"</code> . Ignored if <code>method = "asymptotic"</code> .

**Value**

A list containing

pval	The p-value
F_stat	The calculated F statistic
boot_F	A vector of bootstrapped F statistics if bootstrap has been used. Otherwise NULL.

**Author(s)**

David Clausen

**References**

Willis, A., Bunge, J., and Whitman, T. (2015). Inference for changes in biodiversity. *arXiv preprint*.

**See Also**

[betta](#);  
[betta\\_random](#);  
[betta](#);  
[F\\_test](#)

**Examples**

```
# generate example data
df <- data.frame(chats = c(2000, 3000, 4000, 3000,
2000, 3000, 4000, 3000), ses = c(100, 200, 150, 180,
100, 200, 150, 180),
                Cont_var = c(100, 150, 100, 50,
100, 150, 100, 50),
                Cont_var_2 = c(50,200,25,125,
50,200,25,125))

# fit betta()
example_fit <- betta(formula = chats ~ Cont_var + Cont_var_2, ses = ses, data = df)

# construct L for hypothesis that B_cont_var = B_cont_var_2 = 0
L <- rbind(c(0,1,0),
           c(0,0,1))

F_test_results <- F_test(example_fit, L, nboot = 100)
```

---

toy_metadata	<i>(Data) Data frame of covariate information about toy_otu_table.</i>
--------------	------------------------------------------------------------------------

---

**Description**

(Data) Data frame of covariate information about toy\_otu\_table.

**Usage**

```
toy_metadata
```

**Format**

A data frame with 143 rows and 4 variables:

**Years** Year of sampling

**bloom2** Did the sample correspond to a bloom event?

**Period** What season was sampled?

**Site** Where was the sample taken from? ...

---

toy_otu_table	<i>(Data) A toy OTU table.</i>
---------------	--------------------------------

---

**Description**

Covariate info available in 'toy\_metadata'. A data frame with 448 rows and 143 columns. Rows give the abundance of each taxon; columns give the samples

**Usage**

```
toy_otu_table
```

**Format**

An object of class `data.frame` with 448 rows and 143 columns.

---

toy_taxonomy	<i>(Data) The taxonomy of the OTUs in 'toy_otu_table'.</i>
--------------	------------------------------------------------------------

---

**Description**

(Data) The taxonomy of the OTUs in 'toy\_otu\_table'.

**Usage**

```
toy_taxonomy
```

**Format**

An object of class factor of length 448.

---

true_gini	<i>Calculate the true Gini-Simpson index</i>
-----------	----------------------------------------------

---

**Description**

Calculate the true Gini-Simpson index

**Usage**

```
true_gini(input)
```

**Arguments**

input            A vector of proportions.

**Value**

The Gini-Simpson index of the population given by input.

**Note**

This function is intended for population-level data. If you are dealing with a microbial sample, use DivNet instead.

---

true_hill	<i>Calculate the true Hill numbers</i>
-----------	----------------------------------------

---

**Description**

Calculate the true Hill numbers

**Usage**

```
true_hill(input, q)
```

**Arguments**

input	A vector of proportions.
q	The Hill number of interest. $q = 0$ corresponds to species richness, $q = 2$ corresponds to inverse Simpson, etc.

**Value**

The Hill number of the population given by input.

---

true_inverse_simpson	<i>Calculate the true Inverse Simpson index</i>
----------------------	-------------------------------------------------

---

**Description**

Calculate the true Inverse Simpson index

**Usage**

```
true_inverse_simpson(input)
```

**Arguments**

input	A vector of proportions.
-------	--------------------------

**Value**

The inverse-Simpson index of the population given by input.

**Note**

This function is intended for population-level data. If you are dealing with a microbial sample, use DivNet instead.

---

true_shannon	<i>Calculate the true Shannon index based on proportions</i>
--------------	--------------------------------------------------------------

---

**Description**

Calculate the true Shannon index based on proportions

**Usage**

```
true_shannon(input)
```

**Arguments**

input            A vector of proportions.

**Value**

The Shannon index of the population given by input.

**Note**

This function is intended for population-level data. If you are dealing with a microbial sample, use DivNet instead.

---

true_shannon_e	<i>Calculate the true Shannon's equitability index</i>
----------------	--------------------------------------------------------

---

**Description**

Calculate the true Shannon's equitability index

**Usage**

```
true_shannon_e(input)
```

**Arguments**

input            A vector of proportions.

**Value**

The Shannon E's of the population given by input.

**Note**

This function is intended for population-level data. If you are dealing with a microbial sample, use DivNet instead.



---

true_simpson	<i>Calculate the true Simpson index</i>
--------------	-----------------------------------------

---

**Description**

Calculate the true Simpson index

**Usage**

```
true_simpson(input)
```

**Arguments**

input            A vector of proportions.

**Value**

The Simpson index of the population given by input.

**Note**

This function is intended for population-level data. If you are dealing with a microbial sample, use DivNet instead.

---

wlrm_transformed	<i>The transformed weighted linear regression estimator for species richness estimation</i>
------------------	---------------------------------------------------------------------------------------------

---

**Description**

This function implements the transformed version of the species richness estimation procedure outlined in Rocchetti, Bunge and Bohning (2011).

**Usage**

```
wlrm_transformed(  
  input_data,  
  cutoff = NA,  
  print = NULL,  
  plot = NULL,  
  answers = NULL  
)
```

**Arguments**

input_data	An input type that can be processed by <code>convert()</code> or a phyloseq object
cutoff	Maximum frequency count to use
print	Deprecated; only for backwards compatibility
plot	Deprecated; only for backwards compatibility
answers	Deprecated; only for backwards compatibility

**Value**

An object of class `alpha_estimate`, or `alpha_estimates` for phyloseq objects

**Note**

While robust to many different structures, model is almost always misspecified. The result is usually implausible diversity estimates with artificially low standard errors. Extreme caution is advised.

**Author(s)**

Amy Willis

**References**

Rocchetti, I., Bunge, J. and Bohning, D. (2011). Population size estimation based upon ratios of recapture probabilities. *Annals of Applied Statistics*, **5**.

**See Also**

[breakaway](#); [apples](#); [wlrn\\_untransformed](#)

**Examples**

```
wlrn_transformed(apples)
wlrn_transformed(apples, plot = FALSE, print = FALSE, answers = TRUE)
```

---

wlrn_untransformed	<i>The untransformed weighted linear regression estimator for species richness estimation</i>
--------------------	-----------------------------------------------------------------------------------------------

---

**Description**

This function implements the untransformed version of the species richness estimation procedure outlined in Rocchetti, Bunge and Bohning (2011).

**Usage**

```
wlrn_untransformed(  
  input_data,  
  cutoff = NA,  
  print = NULL,  
  plot = NULL,  
  answers = NULL  
)
```

**Arguments**

input_data	An input type that can be processed by <code>convert()</code> or a phyloseq object
cutoff	Maximum frequency count to use
print	Deprecated; only for backwards compatibility
plot	Deprecated; only for backwards compatibility
answers	Deprecated; only for backwards compatibility

**Value**

An object of class `alpha_estimate`, or `alpha_estimates` for phyloseq objects

**Note**

This estimator is based on the negative binomial model and for that reason generally produces poor fits to microbial data. The result is usually artificially low standard errors. Caution is advised.

**Author(s)**

Amy Willis

**References**

Rocchetti, I., Bunge, J. and Bohning, D. (2011). Population size estimation based upon ratios of recapture probabilities. *Annals of Applied Statistics*, **5**.

**See Also**

[breakaway](#); [apples](#); [wlrn\\_transformed](#)

**Examples**

```
wlrn_untransformed(apples)
```

# Index

## \* datasets

- apples, 5
- hawaii, 23
- pasolli\_et\_al, 33
- soil\_phylo, 42
- toy\_metadata, 45
- toy\_otu\_table, 45
- toy\_taxonomy, 46

## \* diversity

- betta, 5
- betta\_lincom, 8
- betta\_pic, 10
- betta\_random, 11
- breakaway, 13
- breakaway\_nof1, 14
- F\_test, 21
- kemp, 24
- test\_submodel, 43
- wlrm\_transformed, 49
- wlrm\_untransformed, 50

## \* error

- breakaway\_nof1, 14

## \* microbial

- breakaway, 13
- breakaway\_nof1, 14
- kemp, 24

## \* models

- breakaway, 13
- breakaway\_nof1, 14
- kemp, 24
- wlrm\_transformed, 49
- wlrm\_untransformed, 50

## \* nonlinear

- breakaway, 13
- breakaway\_nof1, 14
- kemp, 24

- alpha\_estimate, 3
- alpha\_estimates, 4
- apples, 5, 7, 14, 16, 25, 50, 51

- betta, 5, 9, 11, 12, 21, 44
- betta\_lincom, 8
- betta\_pic, 10
- betta\_random, 11, 21, 44
- breakaway, 7, 13, 14, 16, 25, 50, 51
- breakaway\_nof1, 7, 14, 14, 25
- build\_frequency\_count\_tables, 16

- chao1, 16
- chao1\_bc, 17
- chao\_bunge, 18
- chao\_shen, 19
- check\_format, 19
- convert, 20
- cutoff\_wrap, 20

- F\_test, 21, 44

- get\_F\_stat, 21, 22
- good\_turing, 23

- hawaii, 23

- kemp, 14, 24

- make\_design\_matrix, 25
- make\_frequency\_count\_table, 26

- objective\_bayes\_geometric, 26
- objective\_bayes\_mixedgeo, 28
- objective\_bayes\_negbin, 29
- objective\_bayes\_poisson, 31

- pasolli\_et\_al, 33
- physeq\_wrap, 33
- plot.alpha\_estimates, 34
- poisson\_model, 35
- poisson\_model\_nof1, 35
- proportions\_instead, 36

- rnbinomtable, 36

rztnbinomtable, [37](#)

sample\_inverse\_simpson, [37](#)  
sample\_richness, [38](#)  
sample\_shannon, [38](#)  
sample\_shannon\_e, [39](#)  
sample\_simpson, [39](#)  
sample\_size\_estimate, [40](#)  
sample\_size\_figure, [41](#)  
simulate\_betta, [41](#)  
simulate\_betta\_random, [42](#)  
soil\_phylo, [42](#)

test\_submodel, [43](#)  
toy\_metadata, [45](#)  
toy\_otu\_table, [45](#)  
toy\_taxonomy, [46](#)  
true\_gini, [46](#)  
true\_hill, [47](#)  
true\_inverse\_simpson, [47](#)  
true\_shannon, [48](#)  
true\_shannon\_e, [48](#)  
true\_simpson, [49](#)

wlrm\_transformed, [49](#), [51](#)  
wlrm\_untransformed, [50](#), [50](#)